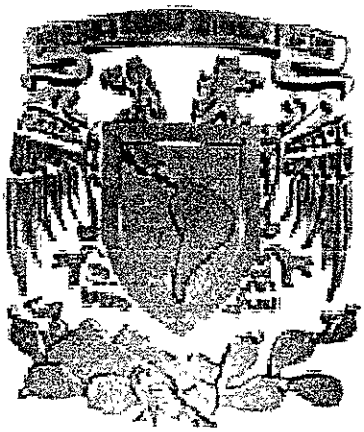
01149



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DIVISION DE ESTUDIOS DE POSGRADO

LA PRECIPITACION PLUVIAL EN LA CIUDAD DE MEXICO COMO SISTEMA DINAMICO CAOTICO Y SU MODELACION Y PRONOSTICO CON REDES NEURONALES ARTIFICIALES

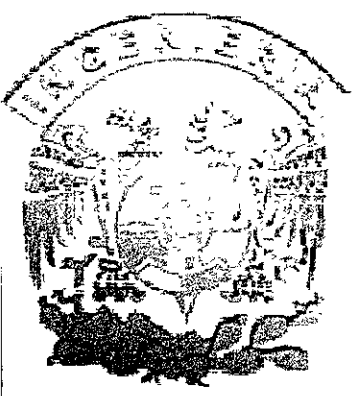
T E S I S

QUE PARA OBTENER EL GRADO DE MAESTRO EN INGENIERIA
(INFORMATICA)

P R E S E N T A

EDGAR PEREZ PEREZ

DIRIGIDA POR EL DR. JAVIER VITELA ESCAMILLA



CIUDAD UNIVERSITARIA 2001.

Agradecimientos

A la Universidad Nacional Autónoma de México (UNAM).
Al Consejo Nacional de Ciencia y Tecnología (CONACYT).
Al Instituto Mexicano del Petróleo (IMP).

Con toda mi gratitud y respeto.

Todos los teoremas son verdaderos, todos los modelos son incorrectos, todos los datos son inexactos..., ¿Qué es lo qué podremos hacer?

"Neural Networks are the second best way of doing just about anything. The best way is to find and tu use the right rules or the optimum algorithm for each particular problem".

John Denker

Indice

Primera Parte. Antecedentes

Por 3 siglos la investigación permaneció inconscientemente enfocada hacia la regularidad...

Hao Bai-lin

1. Introducción

Cuando las *Manifestaciones Caóticas* fueron percibidas en forma consciente en la naturaleza y entendidas como la evolución inherente de una gran variedad de fenómenos cotidianos, el mundo de la *Ciencia Clásica*, al menos como había existido durante los últimos 35 años, pareció marchitarse.

Los científicos, guiados por las ideas y trabajos de la época de *Kepler y Newton*, estuvieron inconscientemente influenciados varios siglos de modo tal que sólo pusieron atención a los *movimientos regulares*, lo cual dió como resultado la incapacidad para percibir los fenómenos irregulares que ocurren en forma continua en la naturaleza. A comienzos de este siglo, sin embargo, *Henri Poincaré* mantuvo una profunda atención hacia la posibilidad de un comportamiento irregular en los sistemas determinísticos. Lamentablemente, debido a la ausencia de algún equipo de cómputo, no pudo avanzar en sus investigaciones. *¡Por tres siglos la investigación permaneció inconscientemente enfocada hacia la regularidad!.*

Sin embargo, a principios de 1970, un grupo multidisciplinario de investigadores emprendió un esfuerzo colectivo con el objetivo de tratar de entender las irregularidades en los fenómenos naturales. Tales irregularidades pueden ser encontradas en la dinámica de los latidos de nuestro corazón, en las variaciones explosivas de la población en cierta especie salvaje, en la turbulencia de un fluido, en el movimiento errático de un meteoro, etc. Los economistas fueron incitados también a estudiar las teorías de los ciclos económicos. Todos estos fenómenos, y una multitud de otros más, fueron observados con curiosidad y analizados cuidadosamente. Ninguno de aquellos científicos se imaginó quizás que precisamente estaban construyendo las bases de los que hoy se conoce como la *Teoría del Caos.*

Gracias a las aportaciones revolucionarias de *Edward Lorenz* en 1963, quien fue el primero de todos los científicos modernos en comprender la esencia del caos y en desarrollar un modelo simplificado de manifestaciones caóticas en el medio ambiente, hoy en día el caos puede percibirse como tal. Verdaderamente, en su modelo ambiental, Lorenz notó la sensibilidad extrema en las respuestas generadas por pequeños cambios en las condiciones iniciales de su modelo y enunció, por primera vez, lo que hoy se conoce como el *efecto mariposa.*

Las ideas del caos se ajustan perfectamente a la mayoría de las investigaciones actuales que se realizan y que no encajan dentro de ninguno de los patrones tradicionales de

pensamiento científico, o a las que, con los métodos clásicos de análisis, proporcionan resultados que no son lo suficientemente adecuados. El estudio riguroso del caos permite descubrir las características universales de respuesta de los sistemas complejos no lineales y nos proporciona elementos muy valiosos para su adecuada interpretación y descripción a través de *modelos matemáticos no lineales*.

En la Meteorología, por ejemplo, en la que hoy en día son ya conocidos la gran mayoría de los principios físicos subyacentes a los fenómenos que repercuten en el estado del tiempo, la construcción de modelos dinámicos de pronóstico basados en sistemas de ecuaciones diferenciales, ha sido y continuará siendo un área primaria de investigación, sin embargo, dado que se sabe ahora también que una cierta *componente caótica* implícitamente estará siempre presente en todas las predicciones, en los grandes centros mundiales de pronóstico se están ya considerando también los efectos del caos en los pronósticos de los modelos.

Uno de los objetivos básicos de la ciencia, tradicional y moderna, es el de hacer predicciones. Dado algún fenómeno o sistema, si se tiene conocimiento de su comportamiento previo y actual, ¿cómo se podrá predecir su comportamiento futuro?. La solución, en la Meteorología tradicional requiere de dos pasos: La construcción de un modelo, basado en las consideraciones teóricas del fenómeno, y el uso de datos observados como condiciones iniciales de entrada. Los modelos atmosféricos actuales de predicción resuelven un conjunto de ecuaciones diferenciales parciales que describen los flujos de los fluidos sobre un globo que permanece en rotación. Los problemas y limitaciones en las predicciones posiblemente se originen desde este nivel, pero adicionalmente, como fue establecido por Thompson [Thompson; 1957], varios problemas significativos adicionales pueden surgir en la siguiente etapa, donde los datos observados son utilizados como condiciones de entrada para el modelo.

La correcta especificación del estado inicial de un modelo atmosférico requiere de las mediciones de varias variables en un volumen tridimensional, sin embargo, normalmente, dadas las características del sistema mundial de observación meteorológica, las observaciones de las variables relevantes son tomadas en sitios ampliamente dispersos, constituyéndose así solamente en un estado inicial discreto del estado del tiempo. Adicionalmente, en un gran número de ocasiones estas observaciones iniciales son, por desgracia, de muy mala calidad. Las ecuaciones diferenciales continuas en el tiempo y en el espacio, simplemente no pueden evolucionar utilizando un conjunto de entrada discreto, [Farmer y Sidorovich;1987]. Debido a esta limitación, inherente a los problemas de fluidos modelados con ecuaciones diferenciales continuas, surge la necesidad de intentar otras soluciones. Es importante observar que se asume la existencia del modelo basado en ecuaciones diferenciales, sin embargo, para una gran variedad de fenómenos no es posible siquiera formular dicho modelo, o incluso aún, si es que el modelo existiera, si los datos de entrada (condiciones iniciales) fueran de excelente calidad, y para evitar las inconsistencias de modelos continuos y datos discretos, se decidiera utilizar la versión discreta del modelo (mediante por ejemplo una aproximación con diferencias finitas, que son ampliamente utilizadas en los modelos meteorológicos), aún así, se tendrían que enfrentar ciertas limitaciones teóricas y prácticas como la estabilidad numérica del algoritmo, las condiciones de frontera, el *caos*, la conservación de energía, de momento, de vorticidad, etc., todo lo cual establece un conjunto de limitaciones verdaderamente serias para el

adecuado desempeño del modelo discretizado, y por tanto, para el correcto pronóstico del tiempo.

Una clase de solución alternativa consiste en construir modelos a partir directamente de los datos observados. Para estos métodos, los datos proporcionados normalmente en forma de series de tiempo, son por lo regular, considerados como la evolución temporal de un proceso dinámico continuo.

En los últimos 15 años los avances en la teoría de los *sistemas dinámicos* han demostrado la existencia de sistemas disipativos cuya evolución asintótica en sus espacios fase no está confinada a ser periódica, casi-periódica, ni mucho menos lineal, sino a ocupar conjuntos que abarcan todo el espacio fase, estos conjuntos son conjuntos *fractales* y se les conoce como *atractores extraños*; los sistemas dinámicos correspondientes, a estos atractores extraños, se conocen como *sistemas dinámicos caóticos* y sus trayectorias, en el espacio fase, nunca se repiten, de tal forma que *su comportamiento es aperiódico pero completamente determinístico.*

El decaimiento de los grados de libertad en el comportamiento asintótico de los sistemas dinámicos es una propiedad determinante para su *fácil* modelación. En este sentido, los modelos no lineales han probado ser mucho más eficientes para la predicción que los tradicionales modelos lineales. De hecho, puesto que un sistema caótico no se presenta a menos que el sistema por sí mismo presente algún tipo de comportamiento no lineal, los modelos no lineales son indispensables para aproximarnos exitosamente a la dinámica caótica. (Aunque los métodos lineales para analizar series de tiempo provenientes de procesos meteorológicos y climatológicos han tenido algo de éxito, especialmente en lo referente a la asociación *causa-efecto* en fenómenos físicos, su poder de predicción es muy limitado).

Es por ello que los modelos no lineales que serán propuestos en este trabajo para la descripción de sistemas dinámicos teóricos y prácticos, serán generados a partir de conjuntos de observaciones univariadas y con la ayuda de las redes neuronales. Para conseguir este propósito, en las secciones siguientes se presentará una metodología que explica la construcción de este tipo de modelos basados exclusivamente en series de tiempo observadas.

Este trabajo está dividido básicamente en 4 secciones: *Antecedentes, Fundamentos Teóricos, Aplicaciones,* y *Conclusiones.* La primera parte presenta, además de la introducción, una descripción detallada del problema resolver, un breve marco histórico que sitúa en el tiempo la problemática que aquí se esta abordando, y finalmente, el objetivo general que se busca resolver con esta pequeña investigación.

La segunda parte, Fundamentos Teóricos, incluye dos capítulos; el capítulo uno tiene como objetivo el proporcionar todas las definiciones básicas, dentro del área de los Sistemas Dinámicos, para una correcta comprensión de todos los resultados presentados en las secciones de las aplicaciones. En el segundo capítulo, y en analogía con el primero, se presentarán una serie de definiciones y de algoritmos relacionados con las Redes Neuronales Artificiales.

El capítulo 3 y el capítulo 4 están en la tercera sección, que es la sección de las Aplicaciones. En el capítulo 3, todos los conceptos de los capítulos 1 y 2 son puestos en práctica sobre un par de sistemas dinámicos clásicos: el oscilador armónico y el sistema de Lorenz. El tipo de modelo no lineal que será utilizado para la descripción del Sistema de Lorenz pertenece a las denominadas *Redes Neuronales,* el algoritmo de entrenamiento, la arquitectura de la red y su vínculo con la reconstrucción de la dimensión del atractor del sistema dinámico (grados de libertad efectivos) serán discutidos a partir de la sección 3.7.
En el capítulo 4 una vez más, todas las mismas herramientas y procedimientos del capítulo 3 son puestas en práctica pero esta vez sobre series de tiempo meteorológicas. El sistema dinámico caótico que se estudiará está determinado por los niveles de precipitación pluvial en la ciudad de México, y el objetivo que se persigue, dado un conjunto histórico de patrones de lluvia, consiste en estimar con tanta certidumbre como sea posible el comportamiento futuro que tendrán los niveles de precipitación.

Finalmente, la cuarta sección está dedicada a las conclusiones.

2. Descripción del problema

En muchas situaciones de la Ciencia y la Tecnología, nos interesa predecir la evolución futura de un sistema a partir de las mediciones pasadas de él. En la física clásica, el procedimiento central consiste en construir un *Modelo Matemático* escribiendo las ecuaciones de movimiento y tratando de integrarlas hacia adelante en el tiempo para predecir el estado futuro.

Matemáticamente podemos decir, que para describir el estado de un sistema, debemos de proponer un punto x en un espacio multidimensional Γ (con una dimensión para cada *grado de libertad* del sistema en cuestión), con lo que la dinámica podrá ser caracterizada como el movimiento de x en Γ.

Sin embargo, este enfoque comienza a mostrar problemas con los sistemas no lineales y de muchos grados de libertad, como con un fluido turbulento, con el estado del tiempo, o con la economía, por citar sólo algunos ejemplos. No es práctico tratar de resolver las ecuaciones de la dinámica del fluido en forma explícita excepto para situaciones especiales bastante simples, y por ello, en general, no podemos seguirle la pista al movimiento en los espacios de dimensión muy grande. Pero no todo esta perdido, Los estudios de la dinámica de sistemas aparentemente caóticos y con muchos grados de libertad, revelan que *la disipación (por ejemplo, la viscosidad) puede reducir el número de grados de libertad efectivamente relevantes, a un número menor, es decir, el movimiento del sistema, el cual en principio ocurre en un espacio Γ de dimensión muy alta, se confina después de algún tiempo a un subespacio Γ_a de menor dimensión llamado un atractor. El atractor Γ_a es frecuentemente un objeto fractal con una dimensión d que no es un número entero [Mandelbrot, 1982].*

Si nosotros podemos de alguna forma identificar las variables que caracterizan al atractor Γ_a, entonces nuestro problema se convertiría en uno de la física simple, con sólo unos pocos grados de libertad. De primera instancia, esto suena bastante difícil o imposible: habiendo tantos grados de libertad en la descripción original del problema, como podemos saber -sin ya antes haber resuelto el problema- cuáles colecciones de variables son las más relevantes; pero en realidad, esto no es un problema del todo. *No es absolutamente trascendente la forma en la que se eligen las nuevas variables siempre y cuando haya un numero grande de variables originales (grados de libertad). En realidad, una elección completamente adecuada para las variables más relevantes estará dada sencillamente por un conjunto de valores previos de la variable que quiere ser pronosticada*, es decir, *¡basta con una serie de tiempo de la variable de interés!.*

La justificación física [Packard, et. al.,1980] para esperar un resultado como el anterior, es que la mayoría de las mediciones que se hagan del sistema contendrán alguna combinación de las *variables relevantes* (puesto que la evolución del sistema se está dando en el atractor), y las mediciones a diferentes tiempos contendrán, en general, diferentes combinaciones. De tal forma que una sucesión de m de tales mediciones deberá contener suficiente información para predecir el movimiento del atractor, siempre que m sea suficientemente grande comparada con la dimensión d del atractor. Packard probó esta idea con experimentos numéricos logrando encontrar buenos resultados.

Hay también un Teorema riguroso, probado por *Takens* [Takens,1981], el cual enuncia que existe una función suave de a lo más $2d+1$ mediciones pasadas (variables) que correctamente predice el valor futuro de la variable en cuestión. La predicción es tan buena como aquella que hubiera resultado de resolver el sistema completo con sus millones de grados de libertad, y aún más, si las variables medidas son conocidas con *infinita precisión*, entonces el resultado es insensible tanto al intervalo de tiempo entre las mediciones pasadas, como a que tan adelante queramos predecir las mediciones futuras. Pero en la practica el ruido y las imprecisiones en los datos limitan la libertad y la confianza con la que dichas cantidades pueden ser elegidas.

De esta forma, en principio sabemos que podemos reducir el problema de predicción para un sistema dinámico complejo cuyo movimiento se encuentra sobre un atractor dimensionalmente menor, a algo parecido con un problema de la física elemental. Lo que el teorema de Takens no proporciona es la forma explícita de la función que lleva a cabo la extrapolación deseada.

Es aquí en donde las *Redes Neuronales* tienen su función, la idea es entrenar una Red Neuronal con algún tipo de arquitectura utilizando el siguiente conjunto de m variables como *los patrones de entrada* (en realidad m es una cota superior bastante buena para el número de grados de libertad significativos; en la práctica los vectores de entrenamiento pueden ser de menor dimensión (pueden ser de dimensión $d+1,d+2,...,m$ donde d es la dimensión del atractor del sistema)).

$$x(t), x(t+h), x(t+2h), ..., x(t+(m-1)h))$$

y los valores (que son conocidos) $x(t+T)$ como *el patrón deseado*, para muchos valores pasados de t.

De esta manera estaremos aproximando la función verdadera desconocida mediante el modelo generado por una red neuronal.

El sistema dinámico caótico que se estudiará en este trabajo (independientemente del sistema de Lorenz), está definido por la evolución temporal y espacial de los niveles de precipitación pluvial en varios puntos (estaciones de monitoreo) de la ciudad de México, aunque el método de análisis que aquí se propone puede extenderse fácilmente hacia cualquier región del país no sólo a nivel microescala sino incluso a escala media (toda la República Mexicana).

Existen algunos elementos que nos hacer suponer que el sistema dinámico definido por la evolución temporal de los patrones de lluvia tiene un *atractor extraño* de baja dimensión, pues de acuerdo con *Fraedrich* [Fraedrich;1986], "recientemente han habido análisis de experimentos hidrodinámicos con el objeto de obtener estimaciones para las dimensionalidades de series de observaciones ambientales, encontrándose evidencias de que un *número relativamente bajo de grados de libertad caracteriza los flujos turbulentos observados...*"

El objetivo consiste en pronosticar la cantidad de precipitación pluvial que se presentará en algún momento futuro del tiempo, para ello el análisis que se desarrollará inicia con la determinación de la dimensión d del atractor de una serie de tiempo de datos de lluvia. Recuérdese que una variable es suficiente para modelar la evolución de un sistema en el cual intervienen muchos grados de libertad, siempre y cuando en dicho sistema dinámico exista un número considerable de variables (grados de libertad) y la variable seleccionada no contenga demasiado *ruido*. En otras palabras, serán pronosticados valores de lluvia utilizando solamente datos históricos de la misma variable lluvia.

La arquitectura de la Red Neuronal que será utilizada corresponde a las denominadas *Redes multicapas de alimentación hacia delante* y el algoritmo de aprendizaje es el de *Retropropagación del error con gradientes conjugados* [Reifman y Vitela;1992]. El número de *neuronas* de entrada para la red $(d+1,d+2,...,m)$ y la definición de los vectores de entrada $(x(t),x(t+h),x(t+2h),...)$ serán determinados en forma muy clara en las secciones siguientes. El periodo de tiempo hacia adelante para hacer los pronósticos será siempre por simplicidad igual a 1, aunque la presentación de los datos no corresponda necesariamente a una periodicidad diaria, sino semanal, quincenal o mensual (para evitar los ceros en los días de no lluvia). Evidentemente y en función de la cantidad de datos que se dispongan, el número de períodos hacia adelante para el pronóstico, puede cambiarse de 1 hacia n, donde $n>1$ y $n \in \aleph$.

Seguramente en este proceso de pronóstico existe un *límite de predictibilidad*, es decir, un valor de máximo de n para el cual es posible realizar una predicción de buena calidad, pero ese es un problema que no será tratado en este trabajo. Otro aspecto de gran importancia,

pero que tampoco será resuelto aquí, consiste en *poner a prueba* la calidad del modelo construido (red neuronal) mediante *pruebas ciegas (blind tests)*, es decir, la red neuronal entrenada puede ser utilizada para predecir valores, evidentemente no conocidos, pero utilizando como vectores de entrada exclusivamente sus propias salidas.

3. Marco histórico

3.1 Redes Neuronales

Conseguir diseñar y construir máquinas capaces de realizar procesos con cierta inteligencia ha sido uno de los principales objetivos y preocupaciones de los científicos a lo largo de la historia. En un principio los esfuerzos estuvieron dirigidos a la obtención de autómatas, en el sentido de máquinas que realizaran, con más o menos éxito, alguna función típica de los seres humanos. Después de varios años de investigación y de desarrollo se ha llegado a disponer ahora de varias herramientas altamente sofisticadas que permiten obtener resultados sorprendentes: la habilidad mecánica de los primeros días ha pasado a convertirse en disponibilidad y capacidades microinformáticas, lo cual ha permitido poder realizar hoy en día algunos procesos similares a los inteligentes y que podemos encuadrar dentro de la denominada *Inteligencia Artificial (IA).*

Sin embargo, a pesar de que hoy se dispone de herramientas y de lenguajes de programación diseñados expresamente para el desarrollo de máquinas inteligentes, existe un problema de fondo que limita enormemente los resultados que se pueden obtener: estas máquinas se implementan sobre ordenadores basados en la filosofía de funcionamiento expuesta por *Von Neumann*, y se apoyan en una descripción *secuencial del proceso de tratamiento de la información.* El elevado nivel y desarrollo de estos ordenadores, por espectacular y complejo que haya llegado a ser, no deja de seguir las líneas anteriormente expuestas: una máquina puramente mecánica que es capaz de realizar tareas secuenciales (de cálculo, de ordenamiento o de control) de forma increíblemente rápida, pero que es incapaz de obtener resultados aceptables cuando se trata de la realización de algún otro tipo de tareas que pudieran ser extremadamente sencillas, incluso para un ser humano de corta edad (reconocimiento de patrones, por ejemplo).

La otra línea de investigación ha tratado de aplicar los principios físicos que se rigen en la naturaleza para obtener máquinas que realicen los trabajos pesados en sustitución de los seres humanos. Así por ejemplo los motores de vapor o explosión emplean un determinado tipo de combustión para obtener la energía que necesitan, al igual que lo hacen los seres vivos. Se puede pensar de igual manera con la capacidad de razonamiento del ser humano: es posible intentar obtener máquinas con capacidades de razonamiento basadas en el mismo principio biológico de funcionamiento (o en algo que tenga cierta similitud con dicho principio).

No se trata de crear máquinas que compitan con los seres humanos, sino que realicen ciertas tareas de rango intelectual con que ayudarle. Los sistemas que se han desarrollado y

los que se lleguen a desarrollar mediante este enfoque no van suponer, como hasta ahora no lo han hecho, la desaparición de las computadoras tal y como hoy las entendemos, por lo menos en aquellas tareas para las que están mejor dotadas incluso que los seres humanos.

De acuerdo con la historia, las primeras explicaciones teóricas sobre el cerebro y el pensamiento fueron dadas por **Platón** (427-347 a.C.) y **Aristóteles** (384-422 a.C.) Las mismas ideas sobre el proceso mental también las mantuvo **Descartes** (1596-1650) y los filósofos empiristas del siglo XVIII.

Aparentemente la *computación neuronal*, tiene más historia de lo que realmente se cree: **Herón** el Alejandrino construyó un autómata hidráulico sobre el año 100 a.C. Además, a través del tiempo se han construido numerosos modelos de animales para demostrar el comportamiento *necesidad-adaptación* sobre diferentes condiciones de vida, como por ejemplo, las numerosas versiones del ratón en el laberinto [Nemes,1969].

Alan Turing, en 1936, fue el primero en estudiar el cerebro como una forma de ver el mundo de la computación; sin embargo, los primeros teóricos que concibieron los fundamentos de la computación neuronal fueron el Neurofisiólogo **Warren McCulloch**, y el matemático **Walter Pitts**, quienes en 1943 lanzaron una teoría acerca de la forma de trabajar de las neuronas [McCulloch,1943]. Ellos modelaron una Red Neuronal simple mediante circuitos eléctricos. Otro importante libro en los inicios de la teoría de redes neuronales fue el escrito en 1949 por **Donald Hebb**, *La organización del Comportamiento*, en el que se establece una conexión entre la Psicología y la Fisiología [Hebb,1949].

En 1957, **Frank Rosenblatt**, comenzó el desarrollo del *Perceptrón*. El Perceptrón es la más antigua red neuronal, y se usa hoy en día de varias formas en el reconocimiento de patrones. Este modelo es capaz de generalizar, es decir, después de haber aprendido una serie de patrones es capaz de reconocer otros similares, aunque no se le hayan presentado anteriormente. Sin embargo, tiene una serie de limitaciones, de entre a cuales posiblemente la más conocida, es su incapacidad para resolver el problema de la función *OR exclusiva (XOR)* y, en general, no es capaz de clasificar clases no separables linealmente [Rosenblatt,1958].

En 1959, **Bernard Widrow** y **Marcial Hoff**, de Stanford, desarrollaron un modelo *ADALINE* (ADAptive LINear Elements). Esta fue la primera red neuronal aplicada a un problema real (filtros adaptativos para eliminar ecos en las líneas telefónicas) y se ha usado comercialmente durante varias décadas [Widrow,1960].

Uno de los mayores investigadores en redes neuronales desde los años 60 hasta nuestros días es **Stephen Grossberg** (Universidad de Boston). A partir de su extenso conocimiento fisiológico ha escrito numerosos libros y desarrollado modelos de redes neuronales. Estudió los mecanismos de la percepción y la memoria. Grossberg realizó en 1967 una red, *Avalancha*, que consistía en elementos discretos con actividad variable en el tiempo y que satisface ecuaciones diferenciales continuas, para resolver actividades tales como reconocimiento continuo del habla y aprendizaje del movimiento de los brazos de un robot [Grossberg,1982].

Rosenblatt en 1962 [Rosenblatt,1962] probo, para el caso más simple de Perceptrón, sin alguna capa intermedia, la convergencia de un algoritmo de aprendizaje, consistente en el cambio iterativo de las conexiones entre las neuronas (pesos w_{ij}), hasta conseguir que algún cálculo fuera realizado. Muchas personas se sintieron enormemente entusiasmadas y pensaron que tales modelos podrían servir de base para la inteligencia artificial.

En 1969 surgieron numerosas críticas que frenaron, hasta 1982, el crecimiento que estaban experimentando las investigaciones sobre redes neuronales. **Marvin Minsky** y **Seymour Papert**, del Instituto Tecnológico de Massachusetts (MIT) publicaron un libro, *Perceptrons* [Minsky,1969], que además de contener un análisis matemático detallado del perceptrón, mostraba que el teorema de Rosenblatt aplicaba sólo a aquellos problemas para los cuales la estructura del perceptrón se ajustaba adecuadamente, es decir, Minsky y Papert mostraron que algunos cálculos bastante elementales no podían ser realizados por el perceptrón de una capa de Rosenblatt, el ejemplo más simple es la función *XOR*.

Rosenblatt había también estudiado estructuras de perceptrones con más de una capa y creía que ellas serían capaces de superar las limitaciones del perceptrón simple. Sin embargo no contaban aún con algún algoritmo de aprendizaje que pudiera determinar los pesos necesarios para realizar algún cálculo dado. Minsky y Papert dudaban que tal algoritmo pudiera ser encontrado y consideraron que la extensión a perceptrones multinivel era completamente estéril. Muchas de las investigaciones dieron un giro hacia la inteligencia artificial, que prometía más por aquel entonces. La mayoría de la comunidad de las Ciencias Computacionales abandonó el paradigma de las redes neuronales por casi 20 años!.

A pesar de libro *Perceptrons*, algunos investigadores continuaron su trabajo. Tal fue el caso de **James Anderson** que desarrolló un modelo lineal, llamado *Asociador Lineal*, que consistía en unos elementos integradores lineales (neuronas) que sumaban sus entradas. Este modelo se basa en el principio de que las conexiones entre neuronas son reforzadas cada vez que están activadas. Anderson diseño una potente extensión del asociador lineal, llamada *Brain-State-in-a Box (BSB)*. [Anderson,1977].

Quizás uno de los desarrollos más significativos en el área de las redes neuronales tenga que ver con la vieja idea de Rosenblatt acerca de un algoritmo de aprendizaje en perceptrones multicapas. A partir de 1972, en forma independiente, muchos investigadores desarrollaron un algoritmo para ajustar las conexiones entre neuronas de capas sucesivas en perceptrones multinivel.

En 1972, **Paul Werbos** defendió como examen doctoral el trabajo titulado *Beyond Regression: New Tools for Prediction and Analysis in the behavioral sciences*, en el cual presentó a la Universidad de Harvard las ideas del algoritmo *Backpropagation*, así como las técnicas para manipular los cálculos involucrados (*regla de la cadena para derivadas ordenadas*).

En 1985, **Rumelhart, Hinton** y **Williams** [Rumelhart,1986a,b], redescubrieron el algoritmo en forma independiente. **Le Cun** [1985] también propuso un algoritmo relacionado.

Aunque este algoritmo no es aún completamente general para resolver cualquier tipo de problema en cualquier tipo de red, sí puede resolver exitosamente una amplia gama de problemas (incluyendo el *XOR*) en comparación con el perceptrón simple.

En Europa y en Japón, las investigaciones también continuaron, **Kunihiko Fukushima** desarrolló el *Neocognitron*, un modelo de red neuronal para el reconocimiento de patrones visuales [Fukushima,1980]. **Teuvo Kohonen**, un ingeniero electromecánico de la Universidad de Helsinki, desarrolló un modelo similar al de Anderson, pero independientemente [Kohonen,1977].

En 1982, coincidieron numerosos eventos que hicieron resurgir el interés por las redes neuronales. **John Hopfield** presentó su trabajo sobre redes neuronales en la Academia Nacional de las Ciencias [Hopfield,1982]. En el trabajo describió con claridad y rigor matemático una red a la que le dio su nombre, que es una variación del asociador lineal, pero, además mostró como tales redes pueden trabajar y que pueden hacer. Además en 1982 se celebró la *U.S. Japan Joint Conference on Cooperative/Competitive Neural Neworks* y **Fujitsu** comenzó el desarrollo en *computadores pensantes* para aplicaciones en robótica.

En 1985, el Instituto Americano de Física comenzó lo que ha sido la reunió anual *Neural Networks for computing*. Esta ha sido la primera de muchas otras. En 1987, el *IEEE* celebró la primera conferencia internacional sobre redes neuronales con mas de 1800 asistentes y 19 nuevos productos mostrados. En el mismo año se formó la *International Neural Network Society* (*INNS*) bajó la iniciativa y dirección de **Grossberg** en U.S.A., Kohonen en Finlandia y **Amari** en Japón: En menos de dos años, la *INNS* tenía mas de 3000 socios. A partir de este momento, el interés por esta área de la ciencia se ha ido incrementado de forma notable, como lo demuestran tanto el numero de congresos y reuniones científicas especializadas a lo largo de todo el mundo, como la aparición de revistas científicas de calidad contrastada dentro del área, así como el interés demostrado por diversos tipos de empresas en utilizar esta tecnología para desarrollar aplicaciones concretas.

En 1988, del espíritu de cooperación en esta nueva tecnología resultó la unión de la IEEE y de la INNS. La ***International Joint Conference on Neural (Networks (IJCNN)*** produjo, en 1989, 430 artículos, 63 de los cuales estuvieron enfocados a una sola aplicación. La IJCNN de enero de 1990, en Washington, incluyó uno hora de concierto de música realizada por redes neuronales. La alternativa europea es la ***International Conference on Artificial Neural Networks (ICANN)***, que fue fundada en septiembre de 1991, y actualmente está organizada por la Sociedad Europea de Redes Neuronales. También merece una referencia aparte la reunión anual ***Neural Information Processing Systems (NIPS)*** celebrada en Denver (Colorado) desde 1987, y que probablemente represente el nivel mas alto de calidad desde el punto de vista científico.

Actualmente son numerosos los trabajos que se realizan y publican cada año, las aplicaciones nuevas que surgen y las empresas que lanzan al mercado productos nuevos, tanto hardware como software (sobre todo para simulación).

En relación con la información que se publica en las revistas especializadas en el área de redes neuronales, cabe destacar entre las más interesantes: *Neural Networks*, revista oficial de la Sociedad Internacional de Redes Neuronales (INNS); *Networks, Computation in Neural Systems*; *IEEE Transaction on Neural Networks*, publicada por *IEEE Neural Networks Council*; *Neural Computation*; e *International Journal of Neural Systems*.

Como ejemplo del resurgir de la investigación sobre redes neuronales, podemos destacar la labor patrocinada por la *Oficina de Tecnología Táctica* de la *Agencia de Proyectos de Investigación Avanzada del Departamento de Defensa de Estados Unidos (DARPA/TTO)* y llevada a cabo en el Instituto Tecnológico de Massachussets (MIT) de octubre de 1987 a febrero de 1988. El resultado de dicho estudio apareció en el libro *Neural Networks Study*, [DARPA,1988], el cual constituye una revisión del estado actual de la tecnología de redes neuronales hasta febrero de 1988, así como sus posibles aplicaciones en el área de defensa y otras áreas, tales como clasificación de patrones, robótica (especialmente control de trayectorias), visión artificial, procesamiento de señales y aplicaciones al habla.

Dentro del entorno europeo, hay que señalar las dos citas anuales de la Sociedad Europea de Redes Neuronales (ENNS) y, como fuente de planificación política de investigación (por medio del financiamiento selectivo de proyectos de investigación), el programa *ESPIRIT*, que durante los últimos años ha financiado una veintena de proyectos que podrían enmarcarse dentro del área de las redes neuronales y sus aplicaciones.

3.2 Caos

Los científicos más notables de la actualidad afirman que la ciencia del siglo 20 será recordada por tres grandes conceptos científico-filosóficos: la relatividad, la mecánica cuántica y el caos. Creen además que la exploración del caos determinará el curso principal de los descubrimientos científicos del siglo 21 y transformará la evolución de la física, la mecánica y también la química; naturalmente, esto también afectará a la ingeniería. De esta forma, los profetas de la nueva ciencia también creen que el caos remplazará los principios de la física de Newton: La relatividad eliminó la ilusión *newtoniana* de tiempo y espacio absolutos; la teoría cuántica eliminó el sueño newtoniano de un proceso controlable de medida y, finalmente, el caos eliminó la fantasía *laplaciana* de una predictibilidad determinística. De esas tres revoluciones científicas, la revolución por el caos, según podemos observar, aplica al universo entero, y sus efectos repercuten, incluso, a escala humana.

Por todos los grandes logros de un gran número de notables científicos, hoy en día debemos verdaderamente de preguntarnos a nosotros mismos, cómo es que este gran edificio de conceptos físicos podría haber continuado con su evolución a través de los años sin proporcionar medios contundentes para contestar algunas de las preguntas más fundamentales acerca de la naturaleza. Cómo empieza la vida y cuál es el misterio de la turbulencia; en un universo gobernado por la entropía e inexorablemente arrastrado hacia un desorden mayor y mayor, cómo es que el orden se establece a él mismo. Más aún, algunos objetos de la vida cotidiana como los fluidos y los sistemas mecánicos no lineales

han llegado a ser tan ordinarios que cualquiera podría llegar a pensar, erróneamente, que ellos están muy bien comprendidos. Hasta hace 30 años, la razón para el desconocimiento y para la ignorancia de tales sistemas fue causada por la ausencia de equipos de cómputo con potencialidades gráficas, pero ahora, en poco tiempo la experimentación y la simulación numérica han demostrado que la ignorancia de la humanidad era muy profunda.

Como la revolución debida al descubrimiento del caos involucra una cascada de nuevas sorpresas, los científicos ahora encuentran bastante natural el regresar al estudio de algunos problemas relacionados directamente con la naturaleza del hombre. Ellos están estudiando tanto las nubes como las galaxias y produciendo artículos que tratan desde la extraña dinámica del reboteo de una pelota sobre una mesa, hasta la misteriosa física cuántica.

El sistema no lineal más simple -y prácticamente todos los sistemas en el mundo real son no lineales- genera problemas extremadamente complicados para la predicción. El orden puede súbitamente degenerar en caos y recíprocamente. En la mayoría de los sistemas podemos descubrir la cohabitación del caos y del orden. Se está buscando una nueva clase de ciencia con la cual se espera atravesar el umbral del conocimiento entre lo que un solo elemento puede hacer -por ejemplo, una molécula de agua o una célula de tejido del corazón- y lo que un conjunto de millones de ellas puede crear en cooperación.

En el pasado, los científicos tradicionalmente deducían de resultados complejos observados, una hipótesis en términos igualmente complejos. Cuando ellos observaban una relación aleatoria entre lo que entraba a un sistema y lo que emergía de él, ellos asumían que deberían introducir aleatoriedad en la teoría desarrollada para reflejar esos efectos de ruido o de *error*. En contraste, el estudio del caos fue motivado por el descubrimiento a inicios de 1960 de que algunas ecuaciones bastante simples podían generar respuestas tan sorprendentes como la inesperada turbulencia de una tormenta. Pequeñas diferencias en las condiciones iniciales de entrada eran rápidamente magnificadas y producían grandes diferencias en la salida. Gracias a Lorenz ahora asignamos este fenómeno a una dependencia extremadamente sensible a las condiciones iniciales. Considerando el ejemplo clásico del estado del tiempo, tales condiciones pueden ser expresadas como parte del efecto mariposa: "*Una mariposa agitando levemente el aire en Beijing podría generar una tormenta el siguiente mes en Nueva York*".

Edward Lorenz fue el primer científico quien a través de sus experimentos numéricos y computacionales, percibió la esencia del caos. Lorenz trabajó en 1961 como meteorólogo en el MIT. Seleccionando un modelo simple de convección Rayleigh-Bernard en una capa de aire activada por la diferencia de temperaturas, propuso representar las respuestas del fenómeno, por un sistema, aparentemente inofensivo, de tres ecuaciones diferenciales ordinarias no lineales. Con este modelo esperaba estudiar el problema de la predicción del tiempo, para lo cual, utilizando una impresora primitiva, graficó la dirección e intensidad del viento. De esta forma hizo el descubrimiento de la época: Las más pequeñas diferencias en las condiciones iniciales, podían producir cursos divergentes en los patrones de viento, los cuales presentaban diferentes comportamientos hasta llegar a una diferencia total. Esta fue la pista para el efecto mariposa y la conclusión de la no predictibilidad del estado del tiempo sobre periodos de tiempo largos. Los descubrimientos de Lorenz fueron

revolucionarios e iniciaron la búsqueda de lo que hoy conocemos como caos. Lorenz también descubrió el correspondiente atractor extraño en el espacio fase.

Mitchell Feigenbaum ingresó al Laboratorio Nacional de Los Alamos en 1974 llevando la convicción de que el conocimiento de los problemas no lineales era prácticamente inexistente. Una de sus primeras investigaciones involucraba el uso de una función logística con una forma cuadrática dependiendo de un solo parámetro. El descubrió que este sistema matemático primitivo producía no solo las respuestas de estado esperadas, sino también, a través de una cascada de bifurcaciones y periodicidades conducía a manifestaciones caóticas mas allá de ciertos valores del parámetro, y que este caos era nuevamente interrumpido por ventanas de respuesta regular. De esta forma Feigenbaum probó la universalidad de sus investigaciones las cuales aplican con una similaridad sorprendente a expresiones matemáticas diferentes y más complejas.

Benoit Mandelbrot, creador del conjunto que lleva su nombre y el cual posiblemente es el objeto más complejo que existe en las matemáticas, inició sus investigaciones en 1979 sobre la generalización de una determinada clase de formas conocidas como Conjuntos de Julia (inventados originalmente por dos distinguidos matemáticos franceses, **Gaston Julia y Pierre Fatou** durante la primera guerra mundial; si ese par de investigadores hubieran trabajado con computadoras y gráficas, no existe duda, habrían sido co-descubridores del caos), Mandelbrot ingeniosamente creó una imagen en un plano complejo que podía servir como catálogo o diccionario de todos los conjuntos de Julia. Científicos como Julia, Fatou, Hubbard, Barnsley y Mandelbrot inventaron reglas novedosas de como construir formas geométricas extravagantes que ahora son conocidas como *fractales* y que son gobernados por el principio de *self-similarily.*

Otro aspecto importante, asociado con la evolución de la teoría del caos, es el del atractor hacia el cual son empujadas las trayectorias en el espacio fase -por ejemplo un punto o un ciclo límite-. Este problema atrajo la atención de dos matemáticos Belgas, **David Ruelle y Floris Takens** quienes, en esos momentos, no estaban enterados de los descubrimientos revolucionarios que Edward Lorenz hizo en 1963 y de la presentación parcial del entonces sin nombre atractor extraño. Ruelle y Takens querían verificar la afirmación de Landau de que, en general, en un fluido la turbulencia es generada por una sucesión infinita de *bifurcaciones Hopf.* Utilizando una argumentación matemática de alto nivel y algunos resultados de Henri Poincaré, ellos demostraron que la afirmación de Landau debería de ser errónea puesto que su esquema no producía ni alargamientos ni pliegues de las trayectorias en el espacio fase y no reflejaba una alta sensibilidad a las condiciones iniciales, las cuales son características esperadas de una transición turbulenta. Más aún, este par de matemáticos construyeron el primer atractor extraño completo y lo denominaron como tal, es decir, crearon el concepto de *atractor extraño.*

En un sentido, *el grado de irregularidad de las formas fractales corresponde a su habilidad para ocupar espacio.* Una línea recta unidimensional, de ninguna forma llena el plano, pero *una curva de Koch* -una clase de copo de nieve idealizado- gobernada por una construcción fractal simple y una longitud infinita pero que encierra un área finita, no tiene una dimensión entera. Su dimensión excede la dimensión uno, de una recta, pero es menor que la dimensión dos, de un área. Mandelbrot determinó esta dimensión fractal como

1.2618. Cualquier atractor cuya dimensión sea fractal, es decir, no entera, se denomina como atractor extraño.

El concepto de dimensión fractal, como es expresado hoy en día por una gran cantidad de definiciones alternativas, abunda ahora en la física y en la teoría de los sistemas no lineales. De esta forma, sabemos hoy que el atractor extraño de Lorenz tiene una dimensión de 2.06. Las dimesiones fractales, por lo tanto, son inherentes a la teoría del caos. Los fractales dominan ahora también en la geofísica en donde tratan de caracterizar la infinita complejidad de la superficie de nuestro planeta. Todo lo anterior y muchos otros descubrimientos han traído la aceptación de la geometría fractal no Euclidiana como una nueva herramienta para resolver muchos problemas.

De esta forma, en términos sencillos, el caos determinístico o simplemente caos, se refiere a la generación de comportamientos impredecibles a partir de una regla simple no lineal. La regla no contiene ruido, aleatoriedad o probabilidades incluidas en ella. En su lugar el comportamiento extremadamente complicado surge simplemente mediante la aplicación repetida de la regla no lineal; es en este sentido que la impredictibilidad emerge con el tiempo.

Existen algunas características que son comunes a todos los sistemas caóticos:

- Los sistemas caóticos determinísticos, son en realidad bastante ordenados e incluso predecibles en escalas cortas de tiempo. El reto consiste en encontrar el orden escondido detrás del caos aparente.

- El comportamiento en el largo plazo de un sistema caótico es difícil o imposible de predecir; incluso las mediciones más exactas de su estado actual se convierten en información inútil para saber cual será el estado futuro del sistema. Es indispensable medir el sistema una vez más para saber cual es su nuevo estado.

- Son altamente sensibles a los cambios en las condiciones iniciales, es decir, empezando en condiciones iniciales muy cercanas, un sistema caótico muy rápidamente se moverá a diferentes estados.

- Un sistema caótico tiene un amplio espectro de frecuencias.

- Un sistema caótico produce una amplificación exponencial de los errores; esto quiere decir que en cualquier sistema dinámico caótico pequeñas magnitudes de ruido externo rápidamente se magnifican y controlan el sistema. Si el ruido está por debajo de la resolución mínima de medición, de tal forma que el observador no puede verlo o controlarlo, entonces el sistema se muestra como impredecible.

- En un sistema caótico coexisten la *Inestabilidad local* contra *estabilidad global*, pues para tener amplificación de los errores pequeños y del ruido, el

comportamiento debe de ser localmente inestable (en períodos cortos de tiempo las condiciones iniciales del sistema muy cercanas se alejarán grandemente unas de otras). Pero para que consistentemente el sistema produzca un comportamiento estable sobre períodos de tiempo largos, el conjunto de comportamientos debe de autocontenerse dentro de él mismo. La coexistencia de esas dos propiedades (inestabilidad local y estabilidad global) conduce a los elegantemente estructurados atractores caóticos.

4. Objetivo

Una vez que la precipitación alcanza la superficie terrestre, incorporándose dentro del ciclo hidrológico del agua, surge el interés por medir su distribución en el tiempo y en el espacio, así como la tasa con la que ésta se acumula en la superficie del planeta, puesto que esta información es de vital importancia para todas la actividades humanas. Las variaciones en la distribución y en la intensidad de las precipitaciones pueden resultar, según se sabe, en sequías o en inundaciones, pero finalmente todo ello se traduce en la disponibilidad del agua como un recurso indispensable para el bienestar y para el desarrollo de los seres humanos.

La modelación numérica de los procesos atmosféricos, oceánicos y geofísicos ha sido siempre una actividad muy demandante en términos computacionales. Un pronóstico típico de 24 horas en un modelo de gran escala (en un *grid* de *50x50x10*, que es equivalente por ejemplo a una región similar a 5 veces la extensión de toda la república Mexicana con resolución de 1 grado de latitud y de 1 de longitud y hasta 15 kilómetros en altura, o a una región equivalente a 2/3 del área de todo el planeta con una resolución de 5 grados de latitud y 5 de longitud y también hasta 15 kilómetros de altura) requiere aproximadamente de 22.5 millones de soluciones. Es claro que solamente los equipos de cómputo más poderosos son capaces de resolver tan sofisticados pero demandantes modelos, y aún en ellos, los tiempos de procesamiento llegan a alcanzar varias horas e incluso días.

Los avances en las capacidades y en el desempeño de las supercomputadoras permiten ahora poder trabajar con problemas más complejos utilizando menos aproximaciones, rejillas más finas, modelos más reales, y todo ello en tiempos de procesamiento más cortos. En estos últimos tiempos, las más recientes arquitecturas altamente paralelas en los equipos de cómputo prometen adicionalmente un incremento en la capacidad de desempeño y una mejora en la exactitud de los modelos. Sin embargo, por desgracia no todo podía ser tan perfecto, pues los modelos actuales de *circulación general*, los *modelos regionales*, o los *modelos anidados* de *alta resolución* (que tienen una resolución máxima de 2 a 20 km en la horizontal), y que son, todos ellos, los que más repetidamente han demostrado su gran capacidad y eficiencia para hacer aceptables pronósticos del tiempo, por ellos mismos no son capaces de determinar la *variabilidad local* de la atmósfera (a resoluciones menores de 1 kilómetro en la horizontal), y no se vislumbra en el futuro inmediato el interés por habilitarles esta capacidad. Por esta razón es indispensable desarrollar un tipo de modelo

alternativo que sea capaz de realizar pronósticos locales (microescala) del estado del tiempo, y muy en particular para los fines de este trabajo, de la precipitación

Hasta ahora, algunas las principales limitaciones con las que se han encontrado los modelos que han probado ser más exactos y más eficientes para la tarea de la predicción del tiempo son las siguientes (nótese que de antemano se descartan la gran mayoría de las técnicas estadísticas clásicas, pues además de carecer las mas de las veces de un sólido carácter científico, generalmente éstas llevan consigo grandes suposiciones acerca del comportamiento del fenómeno en estudio, linealidad, por ejemplo):

- Los Modelos de Circulación General sólo pueden hacer modelación a gran escala y a escala media

- Dado que el emplazamiento, operación y mantenimiento de las estaciones de observación es un proceso caro, el espaciamiento horizontal entre ellas resulta a veces ser demasiado grande como para que los modelos de circulación general puedan detectar los fenómenos meteorológicos que ocurren a escalas menores. Los fenómenos meteorológicos que tienen lugar a escalas de alta resolución (entre ellos las precipitaciones locales) no son *vistos* por estos modelos y por tanto son erróneamente representados.

- El tiempo de procesamiento para la obtención de un pronóstico, es generalmente grande.

- Dado que la atmósfera, estrictamente hablando, es un fluido turbulento, los modelos dinámicos no reflejan su comportamiento real a través de las ecuaciones de la dinámica de los fluidos.

Por lo tanto, el objetivo de este trabajo tiene que ver con el desarrollo de una herramienta alternativa y también complementaria que no tenga las 4 limitaciones anteriores y que, como consecuencia, sí nos permita hacer pronósticos locales (a nivel microescala) de los valores de la precipitación.

Objetivo: Desarrollar un modelo con redes neuronales para pronosticar valores acumulados de precipitación (semanales, quincenales y mensuales). El modelo debe de ser de muy alta resolución, debe de converger en tiempos de operación pequeños y debe de poseer tanta calidad y certeza en sus resultados como calidad y certidumbre tengan los datos que lo generaron.

Segunda Parte. Fundamentos

Bajo el nombre de Caos los métodos de los sistemas dinámicos no lineales han hecho una contribución invaluable a nuestra comprensión hacia un gran número de fenómenos naturales.

Capítulo 1 Sistemas Dinámicos.

En la sección anterior quedó establecido que se tiene por objetivo el hacer pronóstico sobre una serie de datos de precipitación pluvial utilizando redes neuronales, para ello, según se mencionó, es necesario de alguna forma poder identificar la dimensión del atractor de la evolución temporal de los niveles de precipitación, pues ello implícitamente proporciona una cota inferior y una cota superior para el número de neuronas en la capa de entrada de la red neuronal (ver la sección 2 de Antecedentes).

A continuación se presenta una serie de conceptos básicos e indispensables para la adecuada comprensión de los capítulos siguientes, todos ellos relacionados con los Sistemas Dinámicos y con las Redes Neuronales.

1.1 Sistemas Dinámicos

Un sistema dinámico puede ser definido como una *prescripción matemática determinística* de cómo evoluciona el estado de un sistema hacia adelante en el tiempo. Un sistema dinámico consiste de un conjunto de posibles estados y de una regla que determina el estado presente en términos de los estados pasados. Es necesario que la regla sea determinística, lo cual significa que a través de ella sea posible conocer los estados futuros del sistema. Se debe de notar que en la definición de un Sistema Dinámico, se hace hincapié en el aspecto determinístico, es decir, no se permite la aleatoriedad, pues si éste fuera el caso, en lugar de un Sistema Dinámico estaríamos hablando de un *proceso estocástico o aleatorio.*

Tradicionalmente se han distinguido 2 tipos de sistemas dinámicos: si la regla es aplicada en tiempos discretos, el sistema dinámico será discreto. Este tipo de sistemas toman el estado actual como entrada y usando la regla o prescripción matemática actualizan el sistema produciendo un nuevo estado como salida. A estos Sistemas Dinámicos también se les conoce como mapeos. El otro tipo importante de Sistema Dinámico es

esencialmente el límite de los sistemas discretos con tiempos de actualización cada vez más pequeños y pequeños. Las reglas que se utilizan en estos casos son un conjunto de ecuaciones diferenciales y al sistema dinámico se le conoce como continuo.

Un ejemplo de un Sistema Dinámico en el cual la variable tiempo t es continua es un sistema de n ecuaciones diferenciales ordinarias de primer orden y autónomas

$$\begin{aligned} \frac{dx_1}{dt} &= F_1(x_1, x_2, ..., x_N) \\ \frac{dx_2}{dt} &= F_2(x_1, x_2, ..., x_N) \\ &\vdots \\ \frac{dx_N}{dt} &= F_N(x_1, x_2, ..., x_N) \end{aligned} \tag{1.1.1}$$

El cual en forma vectorial se expresa como

$$\frac{dx(t)}{dt} = F(x(t)) \tag{1.1.2}$$

Donde x es un vector *n-dimensional.*

Este es un sistema dinámico porque para cualquier estado inicial $x(0)$ del sistema es posible en principio resolver las ecuaciones diferenciales para obtener el estado futuro $x(t) : t>0$ del sistema. La siguiente figura nos muestra la trayectoria seguida por un sistema hipotético (su evolución en el tiempo) suponiendo que éste se mueve en un espacio tridimensional, es decir $N=3$

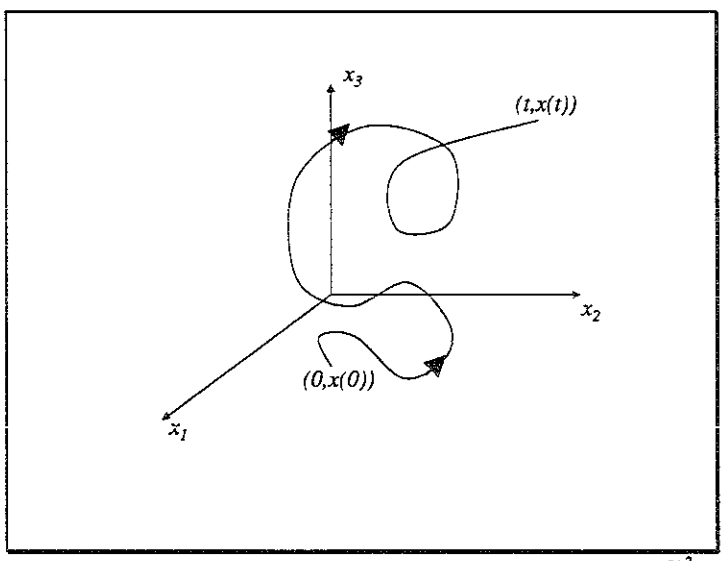


Fig.(1.1.1) Evolución de un Sistema Dinámico continuo en $\Re^3$

El espacio (x_1,x_2,x_3) se le denomina *espacio fase* y la curva dibujada en él por la evolución temporal del sistema se le conoce como una *órbita o una trayectoria* y se denota normalmente como $\{x_0, F(x_0), F^2(x_0), F^3(x_0),...\} = \{F^n(x_0)\}$. Donde $F^n(x_0)$ significa que se ha aplicado n veces el sistema o mapeo, iniciando en el punto x_0. Es común referirse a los sistemas dinámicos continuos en el tiempo con el nombre de *flujos.*

En el caso de los sistemas dinámicos discretos en el tiempo, un ejemplo está dado por los mapeos discretos, los cuales se denotan como una ecuación recurrente de la forma $x_{t+1}=F(x_t)$, donde x_t tiene N componentes $x_t = (x_{t1}, x_{t2}, x_{t3}, ...x_{tN})$.

Dado el estado inicial x_0 podemos obtener el estado en el tiempo $t=1$, como $x_1=F(x_0)$. Habiendo determinado x_1 podemos calcular x_2 haciendo $x_2=F(x_1)$ y así sucesivamente. Por lo tanto para generar una trayectoria del sistema dinámico discreto tendremos que calcular $x_0, x_1, x_2,...$

Es muy razonable hacer la conjetura de que la complejidad en la estructura de las órbitas, puede hacerse mayor para un sistema dinámico de orden elevado. De esta forma una pregunta natural es la siguiente: ¿Qué tan grande debe de ser N (donde N es el numero de ecuaciones o de variables en (1.1.1)) para que en un sistema dinámico un *comportamiento caótico* sea posible?. Para el caso del sistema de ecuaciones diferenciales ordinarias autónomas, la respuesta es que con $N \geq 3$ es suficiente, [Ott;1993].

1.2 Espacio fase

El espacio fase de un sistema dinámico es un espacio matemático con coordenadas ortogonales que representan a cada una de las variables necesarias para la especificación del estado instantáneo del sistema. Por ejemplo, el estado de una partícula en movimiento en una dimensión es especificado por su posición x y su velocidad v, de tal forma que su espacio fase es un plano. Por otro lado, si la partícula se estuviera moviendo en tres dimensiones, entonces su espacio fase sería de seis dimensiones, tres para las posiciones y tres para las velocidades. Un espacio fase puede ser construido de varias formas diferentes, por ejemplo, pueden utilizarse las aceleraciones en lugar de las velocidades.

A las curvas que se dibujan en el espacio fase por la evolución temporal del sistema dinámico se les conoce como órbitas o trayectorias. Una propiedad importante de las trayectorias es que dos trayectorias que se generen con condiciones iniciales similares pasarán muy cerca una de la otra pero nunca se cruzarán. Esta propiedad de no intersección es debida al hecho de que los estados pasado y futuro de un sistema determinístico quedan definidos en forma única por el estado del sistema a un tiempo dado. Una intersección de las trayectorias en el tiempo t introduciría ambigüedad en los

estados pasado y futuro del sistema determinístico y originaría que el sistema quedara indeterminado.

Otra característica importante del espacio fase de algunos sistemas dinámicos es la preservación de áreas o volúmenes; esto significa que todos los puntos que se encuentran en un volumen dado del espacio fase a un tiempo específico se moverán de tal forma que, en un tiempo posterior, los volúmenes ocupados por esos puntos continuarán siendo los mismos.

La propiedad de preservación de áreas, o volúmenes, en los espacios de mayor dimensión, es una característica general de los sistemas conservativos. Esta propiedad induce una clasificación de los sistemas dinámicos en dos categorías: *sistemas dinámicos conservativos* y *sistemas dinámicos disipativos*, dependiendo de si los volúmenes en el espacio fase permanecen constantes o se contraen, respectivamente.

Para determinar si un sistema dinámico es conservativo o disipativo se utiliza el Teorema de la Divergencia del cálculo vectorial

$$\int_S (F \cdot n) dS = \int_R (\nabla \cdot F) dR \qquad (1.2.1)$$

Donde F es cualquier campo vectorial suave y continuo. Si $\nabla \cdot F = 0$ entonces el sistema dinámico será conservativo, y si $\nabla \cdot F < 0$, el sistema dinámico será entonces disipativo.

La justificación matemática del resultado anterior, se comprende mejor si consideramos que el Sistema Dinámico se mueve en un espacio fase tridimensional. Supongamos que las coordenadas del espacio fase son x_1, x_2, x_3, como en la figura (1.2.1)

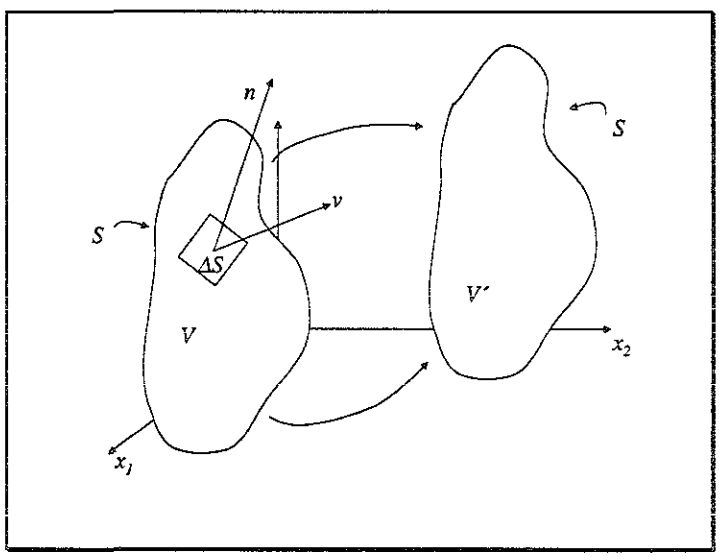


Fig.(1.2.1) Sistema Dinámico conservativo

Las ecuaciones del movimiento del sistema pueden ser expresadas en términos del siguiente sistema de ecuaciones diferenciales autónomo

$$\begin{aligned}\frac{dx_1}{dt} &= F_1(x_1, x_2, x_3)\\ \frac{dx_2}{dt} &= F_2(x_1, x_2, x_3)\\ \frac{dx_3}{dt} &= F_3(x_1, x_2, x_3)\end{aligned} \tag{1.2.2}$$

Ahora considérese una región con volumen R y con superficie S y supóngase que existe un flujo de puntos provenientes de R. Para una región ΔS de S el flujo proveniente de la región R es la componente del vector velocidad v perpendicular a la superficie (o la proyección de v sobre n, tal que n sea un vector normal a la superficie y unitario) multiplicado por ΔS. Puesto que el vector velocidad $v = \left(dx_1/dt, dx_2/dt, dx_3/dt\right)$, está especificado por el conjunto de ecuaciones diferenciales anterior, obtendremos que el flujo que sale de la subregión ΔS está dado por $(v \cdot n)\Delta S = (F \cdot n)\Delta S$. Por lo tanto el flujo neto de toda la superficie es

$$flujo = \int_S (F \cdot n) dS \tag{1.2.3}$$

Ahora bien, recuérdese que el flujo es la cantidad neta de fluido proveniente de R que fluye a través de la superficie S por unidad de tiempo, de tal forma que para el tiempo δt el volumen original V de puntos del espacio fase, habrá cambiado en una cantidad igual a

$$\delta t \times flujo = \delta t \int_S (F.n) dS \tag{1.2.4}$$

Por lo tanto, el área del espacio fase será preservada o no será preservada dependiendo de si la integral de flujo es cero o negativa respectivamente. Generalmente la integral anterior será difícil de evaluar, y por ello de acuerdo al teorema de la divergencia del cálculo vectorial (Teorema de Gauss), tendremos que

$$\int_S (F.n) dS = \int_R (\nabla \cdot F) dR \tag{1.2.5}$$

Este teorema traslada el cálculo de la integral de un campo vectorial F sobre una superficie cerrada S, hacia la integración de la divergencia de F $(\nabla \cdot F)$ en el volumen cerrado R. Adicionalmente, la preservación del volumen del espacio fase debe de ser independiente del volumen R elegido, y por lo tanto es suficiente con examinar $\nabla \cdot F$

solamente. Si esta cantidad es cero, el sistema se denomina conservativo, mientras que si la divergencia es negativa el sistema será disipativo.

1.3 Atractores, Repulsores y Puntos Silla.

Es una característica importante que los Sistemas Dinámicos Disipativos estén caracterizados típicamente por la presencia de conjuntos atrayentes o *atractores* en el espacio fase. Los atractores son subconjuntos acotados hacia los cuales las trayectorias o regiones del Sistema Dinámico (que no tengan volumen cero en el espacio fase) convergen asintóticamente con la evolución del tiempo. Los Sistemas Dinámicos Conservativos no tienen atractores. Más formalmente, sea F un Sistema Dinámico en $\Re^m$ y sea $x_f \in \Re^m$ un punto fijo, es decir, $F(x_f(t))=x_f \ \forall\ t$. Si $\exists\ \varepsilon > 0\ \ni\ \forall x \in V_\varepsilon(x_f)$ sucede que $\lim_{n\to\infty} F^n(x) = x_f$, entonces x_f es un atractor. Si $\exists\ \varepsilon > 0\ \ni\ \forall x \in V_\varepsilon^\circ(x_f)$ sucede que $\lim_{k\to\infty} \left| F^k(x) - x_f \right| > \varepsilon$, entonces x_f es un repulsor. Un punto silla es aquel que tiene al menos una dirección atrayente y al menos una dirección repelente.

1.4 Caos.

Existen en realidad muchas definiciones posibles para caracterizar el *caos*. De hecho no existe un acuerdo general en la comunidad científica en relación a lo que constituye un Sistema Dinámico Caótico. Sin embargo, esto no impide de poder proporcionar una definición que tenga la ventaja de ser fácilmente verificable en varios ejemplos importantes; no obstante se deberá recordar siempre que existen muchas otras formas de capturar la esencia del caos.

Definición (*Conjunto Denso*):

Sea X un conjunto en $\Re^n$ y Y un subconjunto de X. Decimos que Y es denso en X $\Leftrightarrow \forall x \in X,\ \exists\ y \in Y\ \ni \|x - y\| < \varepsilon,\quad \varepsilon > 0$ ■

Definición (*Transitividad*):

Sea F un Sistema Dinámico y sean x,y,z puntos en el espacio fase de F; decimos que F es transitivo $\Leftrightarrow \forall x, y\ \exists\ \ z\ \ni\ si\ |x - z| < \varepsilon \Rightarrow |y - F^k(z)| < \varepsilon$, para alguna k ■.

En otras palabras, un Sistema Dinámico Transitivo tiene la propiedad de que dados cualesquiera dos puntos en su espacio fase, es posible encontrar una órbita que pase

arbitrariamente cerca de ambos. Claramente, un Sistema Dinámico que tiene órbitas densas es transitivo y recíprocamente.

Definición (*Dependencia Sensible a las Condiciones Iniciales*):

Un Sistema Dinámico F, depende sensiblemente de las condiciones iniciales $\Leftrightarrow \exists \beta > 0$ y $k > 0$, $\ni \forall x, y$ con $|x - y| < \varepsilon, \forall \varepsilon > 0, \Rightarrow |F^k(x) - F^k(y)| \geq \beta$ ■.

Donde $F^k(x)$ significa que se aplica k veces el mapeo sobre el punto x. Recuérdese que un mapeo puede ser tanto discreto como continuo.

Definición (*Sistema Dinámico Caótico*):

Un Sistema Dinámico es caótico si y sólo si:

1. Los puntos periódicos de F son densos
2. F es transitivo
3. F es sensiblemente dependiente a las condiciones iniciales.

Una órbita caótica es una órbita que siempre continúa experimentando el comportamiento inestable típico que una órbita exhibe cerca de un repulsor. Una órbita caótica no trata de encontrar un atractor hacia el cual ser atraída (sin embargo, sí existen los atractores caóticos). En cualquier punto de una órbita caótica es posible encontrar otros puntos arbitrariamente cerca que se separarán del punto en cuestión durante las iteraciones posteriores (evolución del sistema), sin embargo el movimiento caótico puede ser atrayente.

Si un movimiento caótico es detectado en el movimiento de un sistema físico, esto debe de ser porque el espacio fase en el cual se está desarrollando el movimiento caótico atrae una porción muy significativa de las condiciones iniciales del sistema; o si en un experimento es posible observar un movimiento caótico, esto querrá decir que se han escogido (frecuentemente en forma aleatoria) unas condiciones iniciales cuyas trayectorias convergen a un atractor caótico.

El movimiento en órbitas caóticas podría ser descrito como estable en el largo plazo (puesto que atrae a un gran conjunto de condiciones iniciales) y como inestable localmente (puesto que es una órbita caótica). Por lo tanto podemos decir, que los atractores caóticos sí existen, o dicho con otras palabras que el movimiento caótico si puede ser atrayente.

En la sección de aplicaciones presentaremos 2 ejemplos que nos mostraran claramente el significado de estas ideas: el primer ejemplo, que es un caso teórico, corresponde al sistema de Lorenz.

$$\begin{aligned} \frac{dx}{dt} &= -\sigma x + \sigma y \\ \frac{dy}{dt} &= -xz + rx - y \\ \frac{dz}{dt} &= xy - bz \end{aligned} \qquad (1.4.1)$$

El modelo de Lorenz, que está descrito por el sistema (1.4.1) es un modelo altamente idealizado de un fluido convectivo en el que el fluido caliente que se encuentra en la parte inferior se mueve hacia arriba y el fluido frío, que se encuentra en la parte superior, va hacia abajo, todo ello en una circulación en el sentido de las manecillas del reloj. El número Prandtl σ, el número de Rayleigh (o de Reynolds) r y b son parámetros del sistema.

Para $\sigma = 10, b = 8/3$ y $r > r^*$ donde $r^* \approx 24.74$ (normalmente se utiliza r=28), Lorenz encontró numéricamente que el sistema se comporta caóticamente y sin embargo si existe un atractor, el llamado *atractor de Lorenz.*

En el segundo ejemplo que corresponde a una situación *más real*, se estudiará como Sistema Dinámico las precipitaciones pluviales registradas durante los últimos 10 años en alguna estación de monitoreo de la ciudad de México. Aunque de antemano es muy probable que no se conozca si existe o no un atractor para este sistema, la intuición (patrones anuales de lluvia) y algunos resultados como el que se mencionará a continuación, nos sugieren que dicho atractor sí existe: "*... hay análisis recientes de experimentos hidrodinámicos para obtener la dimensionalidad de atractores de sistemas físicos a partir de observaciones y existe la evidencia de que un número relativamente pequeño de grados de libertad (variables activas en la evolución del sistema) caracterizan el flujo turbulento observado*" [Fraedrich,1986].

En otras palabras, los patrones anuales de lluvia en la ciudad de México constituyen un Sistema Dinámico que aparentemente tiene un atractor caótico y extraño, para el cual es imposible anticipar su comportamiento localmente en escala temporal y espacial, es decir, es imposible saber de un día a otro qué sucederá con la precipitación o cual será su distribución en escalas pequeñas, e incluso en un mismo día de lluvia es imposible saber con certeza, cuál será su duración, intensidad y cobertura.

Sin embargo, a escalas temporales y espaciales más amplias (en el largo plazo) sí se tiene una idea más o menos precisa del comportamiento de la precipitación, no solamente en la Ciudad de México, sino incluso en toda la República Mexicana (las variaciones recientes en los patrones de lluvia que han tenido lugar no solamente en México, sino a lo largo de todo el mundo, son calificadas por algunos como las consecuencias de un desequilibrio climático general en el planeta, mientras que otros opinan que es la reaparición de comportamientos ya antes ocurridos varios cientos de años atrás. Este segundo enfoque

favorece más el punto de vista de nuestro estudio, pero también nos dejaría ver que las *órbitas del sistema* son de un período demasiado largo)

1.5 Números de Lyapunov

La irregularidad constante de una órbita caótica está cuantificada por los *números de Lyapunov* y por los *exponentes de Lyapunov*. Definiremos los números de Lyapunov como la tasa promedio de divergencia en cada iteración de los puntos cercanos a lo largo de la órbita, y el exponente de Lyapunov como el logaritmo del número de Lyapunov. *El caos se define por un exponente de Lyapunov mayor que cero.*

Definición (*Número de Lyapunov*)**:**

Sea f un mapeo suave del eje real $\Re$, y sea f' su derivada. El número de Lyapunov $L(x_1)$ de la órbita $\{x_1, x_2, x_3, \ldots, x_n, \ldots\}$ está definido como $L(x_1) = \lim_{n\to\infty}\left(|f'(x_1)|+\ldots+|f'(x_n)|\right)^{1/n}$, si es que el límite existe. El exponente de Lyapunov $h(x_1)$ está definido como $h(x_1) = \lim_{n\to\infty}\left(1/n\right)\left(\ln|f'(x_1)|+\ldots+\ln|f'(x_n)|\right)$ si es que el límite existe ■.

Definición (*Orbita asintóticamente Periódica*)**:**

Sea f un mapeo suave. Una órbita $\{x_1, x_2, x_3, \ldots, x_n, \ldots\}$ es llamada asintóticamente periódica si converge a una órbita periódica cuando $n \to \infty$, es decir, si existe una órbita periódica $\{y_1, y_2, \ldots, y_k, y_1, y_2, \ldots, y_k, \ldots\} \ni \lim_{n\to\infty} |x_n - y_n| = 0$ ■.

Definición (*Orbita Caótica*)**:**

Sea f un mapeo del eje real $\Re$, y sea $\{x_1, x_2, x_3, \ldots, x_n, \ldots\}$ una órbita acotada de f. La órbita es caótica si

1. $\{x_1, x_2, x_3, \ldots, x_n, \ldots\}$ no es asintóticamente periódica
2. El exponente de Lyapunov $h(x_1) > 0$ ■.

Definición (*Conjunto Límite hacia Adelante*):

Sea f un mapeo y sea x_0 una condición inicial de f, el conjunto límite hacia adelante de la órbita $\{f^n(x_0)\}$ denotado como $L^+(x_0)$ es el conjunto

$$L^+(x_0) = \{x : \forall N \in \aleph \text{ y } \varepsilon > 0,\ \exists n > N \ni |f^n(x_0) - x| < \varepsilon\}\ \blacksquare.$$

La definición de conjunto límite hacia delante de una órbita es el conjunto de puntos a los cuales la órbita regresa en forma arbitrariamente cerca en una infinidad de ocasiones. Los puntos en una órbita pueden o no pueden estar contenidos en su conjunto límite hacia adelante.

El conjunto límite hacia adelante puede no tener puntos en común con la órbita, como en el caso de un conjunto limite hacia adelante de una órbita que converge hacia un atractor. En este caso el conjunto límite es un punto, el atractor, el cual es aproximado por la órbita en forma tan cercana y tan adelante en el tiempo como se desee. La órbita es atraída hacia el atractor.

Definición (*Atractor Caótico*):

Sea $\{f^n(x_0)\}$ una órbita caótica. Si x_0 está en $L^+(x_0)$, entonces $L^+(x_0)$ es llamado un conjunto caótico. Un *atractor caótico* es un conjunto caótico que también es un atractor $\blacksquare$.

De acuerdo a la definición anterior, un conjunto caótico es un conjunto límite hacia adelante de una órbita caótica en el que la misma órbita caótica está contenida en el conjunto límite hacia adelante.

Las dos definiciones anteriores, nos permiten decir ahora que un atractor es un conjunto límite hacia adelante el cual atrae un conjunto de condiciones iniciales de medida diferente de cero.

Las dos propiedades más importantes de un atractor caótico son:

- Contiene una órbita caótica.
- Atrae un conjunto de valores iniciales que tiene medida diferente de cero,

es decir, el conjunto de valores iniciales con medida diferente de cero es atraído en órbitas caóticas hacia en atractor caótico.

1.6 Isomorfismo, Homeomorfismo, Difhomeomorfismo

Un isomorfismo f es un mapeo biyectivo. Un homeomorfismo f es un mapeo biyectivo, el cual es *bicontinuo*, es decir, f y f^{-1} son continuas. Un difhomeomorfismo es un homeomorfismo en el que f y f^{-1} son derivables ∎.

1.7 Variedades

Una variedad *n-dimensional* es un conjunto que localmente se asemeja al espacio euclidiano $\Re^n$. Por semejanza en este caso estamos suponiendo un punto de vista topológico, es decir, una pieza pequeña de la variedad debe de parecerse a una pieza pequeña de $\Re^n$.

Una variedad unidimensional es localmente una curva. Cada segmento pequeño de una curva puede ser generado alargando y doblando un segmento de recta ($\Re$). Las letras "*D*" y "*O*" son variedades. Las letras "*A*" y "*X*" no lo son puesto que cada una contiene al menos un punto para el cual ninguna vecindad pequeña de él se asemeja a un segmento de recta (las intersecciones).

Formalmente, un subconjunto M de $\Re^n$ se denomina variedad *k-dimensional* (en $\Re^n$) si M es un espacio métrico para el cual cada uno de sus puntos tiene una *vecindad homeomórfica* a $\Re^n$.

1.8 Inmersión e Inscrustamiento.

La evolución de los Sistemas Dinámicos tiene lugar en variedades compactas. Como nosotros estamos interesados en *reproducir* esas variedades a partir de observaciones experimentales y evidentemente se desea encontrar la *mejor reproducción posible*, es necesario investigar el comportamiento local de los mapeos (pues éstos serán los que reproduzcan la variedad experimental a partir de la variedad original) para precisar a que nos referimos cuando hablamos de la mejor reproducción posible.

Para un mapeo f entre 2 variedades compactas $f: M \rightarrow N$, existen 3 casos a considerar: $dim(M)=dim(N)$, $dim(M)<dim(N)$ y $din(M)>dim(N)$.

- **dim(M)=dim(N):** en este caso el mejor comportamiento que un mapeo puede tener, es que éste sea *difhomeomórfico*; si el mapeo no es difhomeomórfico puede ser que la información se pierda.

- **dim(M)>dim(N):** en este caso, necesariamente perdemos información acerca de *M*. La condición más fuerte posible para no perder tanta información es que la derivada del mapeo sea sobreyectiva. Si la derivada de *f* es localmente sobreyectiva en el espacio tangente de *N*, entonces *f* es una submersión local.

- **dim(M)<dim(N):** en este caso el mapeo no puede más ser un difhomeomorfismo, porque *N* tiene más grados de libertad que *M*. Para preservar la estructura de *M* en el espacio más grande, la derivada del mapeo debe de ser invertible y por tanto inyectiva. En este caso se dice que el mapeo es una inmersión.

Para garantizar que la imagen de una inmersión es una variedad, se deben de exigir condiciones adicionales. La figura (1.10.1) muestra un mapeo de un círculo *S* en $\Re^2$ que es una inmersión pero que no tiene una variedad como imagen. El centro de la imagen, en $\Re^2$, no se asemeja a $\Re$ y por tanto no es una variedad.

El problema surge porque el centro de la imagen tiene dos puntos que bajo el mapeo van a dar a él, es decir el mapeo no es inyectivo. El pedir, sin embargo, que la inmersión sea inyectiva no es suficiente. La figura (1.10.2) muestra una inmersión que es inyectiva pero que su imagen no es aún una variedad.

El problema aquí es que los *puntos* $\pm\infty$ son mapeados en una vecindad infinitamente pequeña del origen, evitando una vez más un difhomeomorfismo en $\Re$ (cualquier vecindad del origen, no importa que pequeña sea contendrá las imágenes de $\pm\infty$). Este último problema es evitado pidiendo que el mapeo sea *propio*. Un mapeo es propio si la preimagen de cualquier conjunto compacto es compacta.

Definición (*Incrustamiento*):

Una inmersión inyectiva que es propia es un incrustamiento ∎.

El siguiente teorema establece que este es el mejor comportamiento que *f* puede tener:

Teorema de Incrustamiento

Sean *X* y *Y* dos variedades. Sea *f* un mapeo de *X* a *Y*, $f: X \to Y$, si *f* es un incrustamiento entonces *f* mapea difhomeomórficamente a *X* en una subvariedad de *Y*. ∎

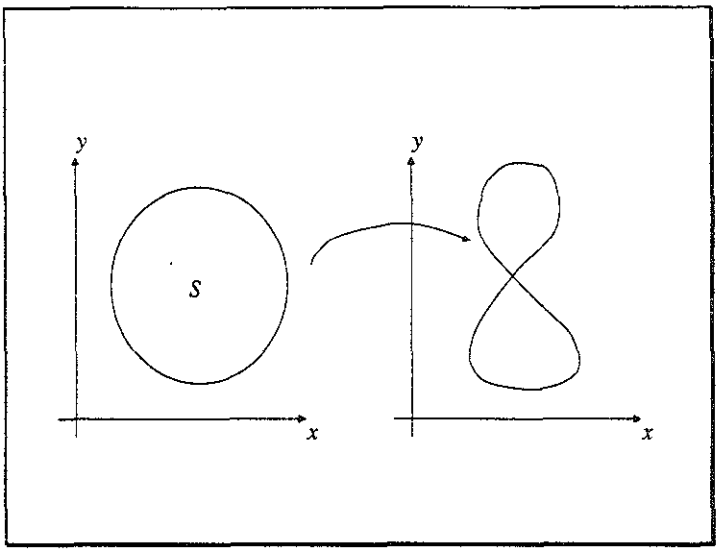


Fig (1.10.1) Mapeo de un círculo S en $\Re^2$ que es una inmersión pero que no tiene una variedad como imagen

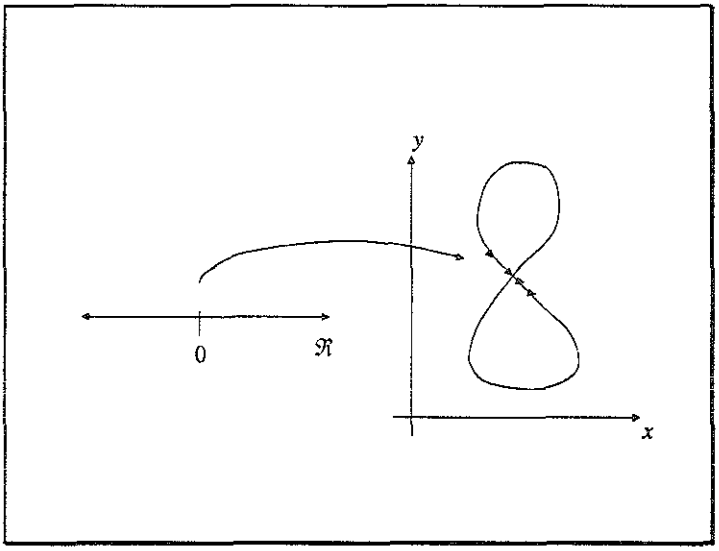


Fig (1.10.2) Una inmersión que es inyectiva pero que su imagen no es aún una variedad

Incrustar variedades en $\Re^n$ es un caso especial que merece mucha atención debido a su relevancia para las observaciones experimentales (series de tiempo). Dada una variedad de dimensión d, qué tan grande debe de ser la dimensión n del espacio real destino?. La respuesta está dada por el siguiente teorema:

Teorema de Incrustamiento de Whitney

Dada una variedad compacta de dimensión d, esta puede ser incrustada en $\Re^{2d+1}$ ∎.

Teorema de Intersección

Sean X y Y dos variedades en $\Re^m$ ∋ X y Y se intersectan en alguna región. Si la dimensión de X es d_1, la dimensión de Y es d_2 y $d_1+d_2<m$, entonces las intersecciones pueden ser removidas a través de perturbaciones arbitrariamente pequeñas de X o de Y ∎

El teorema de Intersección proporciona una muy buena explicación intuitiva de porque el teorema de Whitney debe de ser cierto. Consideremos dos copias idénticas de la misma variedad X, tal que su dimensión es d. Si esas dos variedades fueran colocadas en un espacio de dimensión igual a la dimensión de X, seguramente se intersectarían, pero si las colocamos en un espacio con una dimensión más grande que dos veces la dimensión de X $(2d+1)$, el teorema de intersección nos dice que cualquier intersección remanente no será transversal y podrá ser removida con pequeñas perturbaciones arbitrarias.

1.9 Dimensión y Fractales.

A principios del siglo XX fueron descubiertos una gran diversidad de objetos matemáticos patológicos que necesitaban una adecuada descripción en términos de su dimensión topológica. Las definiciones existentes hasta ese momento no eran lo suficientemente generales para caracterizar lo que hoy conocemos como *fractales*.

Podemos interpretar operacionalmente a un fractal como a un conjunto que tiene un nivel de complicación tal que éste no se simplifica con la magnificación. Tomando como base esta idea podemos considerar a un fractal sobre una rejilla equidistante y contar el número de cajas necesarias para cubrirlo. Es interesante observar como este número cambia a medida que la resolución de la rejilla se hace más fina.

Consideremos una rejilla o partición en el intervalo [0,1] con un incremento de $1/n$, es decir las particiones estarán en $0,\frac{1}{n},\frac{2}{n},...,\frac{(n-1)}{n},1$. Cuál es la dependencia entre el número

de cajas unidimensionales (subintervalos) y el tamaño de la partición?. La respuesta es evidente: hay n cajas de longitud $1/n$. La situación cambia ligeramente si consideramos, por ejemplo el intervalo [0,8]. En este caso necesitaremos $8n$ cajas de longitud $1/n$. La propiedad común para conjuntos unidimensionales, es que el número de cajas de tamaño ε que se requieren para cubrir un intervalo no es más que $c(1/\varepsilon)$, donde c es una constante que depende de la longitud del intervalo.

Esta proporcionalidad es frecuentemente expresada diciendo que el número de cajas de tamaño ε se equilibra con $1/\varepsilon$, queriendo decir que el número de cajas está entre $c_1(1/\varepsilon)$ y $c_2(1/\varepsilon)$, donde c_1 y c_2 son constantes fijas que no dependen de ε.

Ahora bien, el conjunto $\{(x,y):0 \le x,y \le 1\}$ en el plano puede ser cubierto por n^2 cajas de longitud $1/n$ en una rejilla equidistante. Es el exponente 2 lo que diferencía este ejemplo bidimensional del caso unidimensional. Cualquier rectángulo bidimensional en $\Re^2$ puede ser cubierto por $c(1/\varepsilon)^2$ cajas de tamaño ε.

El objetivo aquí consiste en extender esta idea hacia conjuntos más complicados, como los fractales, y en usar esta relación de *equilibrio* entre ε y $1/\varepsilon$ (tamaño y número) para definir la dimensión d de algún objeto en casos en los que no se conozca de antemano la respuesta (como con el atractor, quizás extraño y quizás caótico, de la precipitación pluvial en la ciudad de México (recuérdese que un atractor extraño es aquel que tiene dimensión no entera, mientras que para un atractor caótico lo importante no es su dimensión, que puede o no ser entera, sino el hecho de que atrae órbitas caóticas). En general, si S en un conjunto en $\Re^m$, nos gustaría decir que S es un objeto *d-dimensional* cuando él pueda ser cubierto por $N(\varepsilon) = c(1/\varepsilon)^d$ cajas de longitud ε por lado. Nótese que aquí no se requiere que el exponente d sea entero, y que $N(\varepsilon)$ es el número de cajas de longitud ε.

Definición (*Dimensión Box-Counting):*

Un conjunto acotado S en $\Re^n$ tiene una dimensión d (Box-counting) cuando d se calcula como

$$d = \lim_{\varepsilon \to 0} \frac{\ln N(\varepsilon)}{\ln(1/\varepsilon)}$$

si es que el límite existe. Nótese que se llega a esta definición, partiendo de los razonamientos anteriores, porque c es constante y ε pequeño.

Esta definición nos proporciona una dimensión d que es compatible con la dimensión topológica de conjuntos no fractales, es decir nos proporciona un número entero para los conjuntos "bien portados" y un número no entero en cualquier otro caso.

Es muy importante mencionar además, que otros conjuntos pudieron haber sido utilizados en lugar de cajas en la definición anterior. Es posible, por ejemplo utilizar circunferencias de radio ε.

1.10 Teorema de Takens.

La palabra *genérico* se refiere a un comportamiento que es típico, es decir, a un comportamiento que normalmente se espera observar en un conjunto de objetos, por ello, diremos que una propiedad de funciones suaves es genérica si el conjunto de funciones que poseen dicha propiedad es denso. Esto significa que para cada función suave *f*, independientemente de que cumpla o no alguna propiedad específica, pequeñas y arbitrarias perturbaciones de *f* satisfacerán la propiedad deseada. Un ejemplo de una propiedad genérica de funciones suaves $f:[0,1] \rightarrow \Re$, es que la función tenga a lo más un número finito de ceros.

Por otro lado, existe una relación muy importante, y que a continuación se hará evidente, entre el espacio de estado de un Sistema Dinámico, las observaciones que conforman una *serie de tiempo* del Sistema Dinámico, y lo que será llamado el *espacio de estado reconstruido*. Para empezar supongamos que $\Re^k$ es el espacio de estado de un Sistema Dinámico y que sus trayectorias son atraídas a una variedad *d-dimensional M*.

Supongamos que nos es posible realizar *m* mediciones independientes y simultáneas del sistema en cualquier tiempo dado. Para cada estado, entonces, nuestras *m* mediciones producirán un vector en $\Re^m$. Como es posible repetir este proceso de observación en varios instantes del tiempo, diferentes todos ellos, obtendremos al final varios puntos en $\Re^m$, cada uno representando *m* mediciones simultáneas. Podemos pensar el proceso de medición como una función *F* de $\Re^k$ a $\Re^m$. En cualquier momento, el estado del sistema está representado por un punto de *M* en $\Re^k$, y para él podemos evaluar *F* haciendo las mediciones del vector en $\Re^m$. El siguiente teorema dice que debemos de esperar que *F(M)* represente en forma única todos los estados que existen en la variedad original *M*.

Teorema

Supongamos que *M* es una variedad *d-dimensional* en $\Re^k$. Si $m>2d$ y $F:\Re^k \rightarrow \Re^m$ es una función genérica, entonces *F* es uno a uno en *M*. ∎

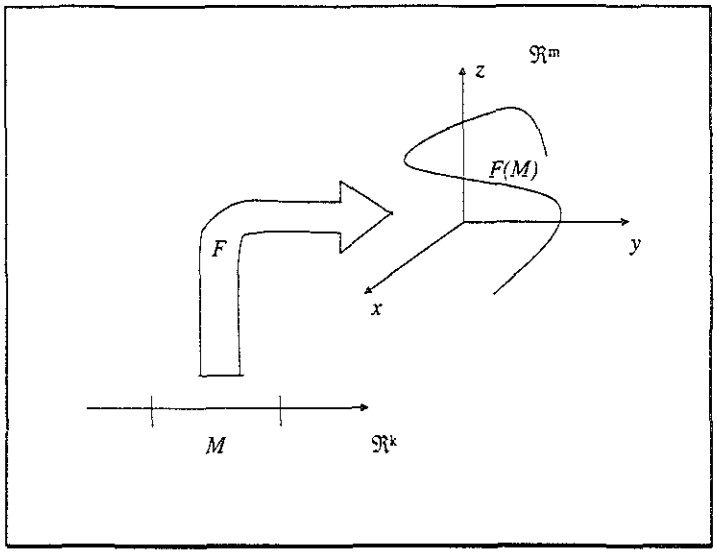


Fig.(1.8.1) Si m>2d, entonces F es uno a uno en M.

Esto significa que si $x, y \in M \ni x \neq y$, entonces $F(x) \neq F(y) \ni F(x)$ y $F(y) \in \Re^m$, es decir, dos estados diferentes de M permanecen diferentes cuando son mapeados en $\Re^m$ (con $m>2d$) o en otras palabras, $F(M)$ no se autointersecta. Nótese que el teorema no es válido para una *dimensión de incrustamiento* menor que $2d+1$.

La razón para exigir que F sea genérica, se justifica de la siguiente manera: aunque la imagen de F, $F(M)$ pudiera tener autointersecciones, otras funciones, las cuales serían perturbaciones extremadamente pequeñas de F, no las tendrán.

El teorema anterior, que en la práctica es sumamente poderoso, es una de las conclusiones del teorema de Incrustamiento de Whitney [Whitney,1936]. El teorema requiere, aunque no se menciona dado que es evidente, que las coordenadas de F sean independientes. Posteriormente fue demostrado [Takens,1981] que es suficiente con elegir F de entre la clase especial de funciones formadas a partir de la reconstrucción de las coordenadas rezagadas de una serie de tiempo univariada. Si a la función de mediciones que da origen a la serie de tiempo univariada, la llamamos $h \ni h: M \subseteq \Re^k \rightarrow \Re$, obtendremos como función con coordenadas reconstruidas la siguiente

$$F(x) = \left(h(x), h(g-T(x)), h(g-2T(x)), \ldots, h(g-(m-1)T(x)\right)$$

Donde g denota el Sistema Dinámico que tiene a M como atractor. g puede ser un mapeo invertible, en cuyo caso g-T denota T pasos del mapeo inverso g^{-1}, o una ecuación diferencial, en cuyo caso g-T denota el estado del sistema T unidades de tiempo atrás. La siguiente figura nos ilustra el Sistema Dinámico g, la función de mediciones escalares h, y la función de coordenadas formadas con rezagos F.

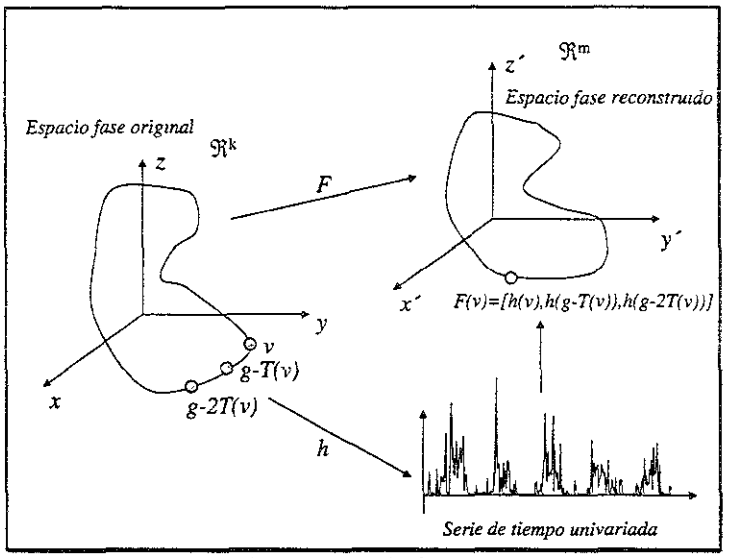


Fig.(1.8.2) Reconstrucción del espacio fase con coordenadas rezagadas

Definición (*Conjunto Invariante):*

Un conjunto invariante S de un flujo F sobre una variedad M es un subconjunto de M definido como $S = \{x \in M : F_t(x) \in S \ \ \forall x \in S \text{ y } \forall t\}$.■

Teorema de Takens (primera versión)

Supongamos que M es un subconjunto de dimensión d de una variedad en $\Re^k$, el cual es invariante bajo el Sistema Dinámico G. Si $m>2d$ y $F:\Re^k \rightarrow \Re^m$ es una función de coordenadas reconstruidas con rezagos a partir de una función genérica de medida h, y un tiempo genérico de rezago T, entonces F es uno a uno en M.■

Este teorema es tan importante que constituye en realidad uno de los ejes centrales de este trabajo; el otro eje importante son las Redes Neuronales.

El teorema de Takens dice que si la dimensión del atractor es el entero d, entonces para funciones genéricas de rezagos, la dimensión de incrustamiento (para *atrapar* al atractor reconstruido) es a lo más $2d+1$, y también dice que la correspondencia entre el estado real del sistema dinámico en $\Re^k$ y un vector con m componentes reconstruidas a partir de los rezagos de una única serie de tiempo univariada del sistema dinámico original es uno a uno (de ahí que la reconstrucción del atractor sea casi perfecta).

El teorema de Takens ha producido una avalancha de investigación puesto que en muchas ramas de la ciencia se ha intentado interpretar las series de tiempo correspondientes a una gran variedad de experimentos, mediante la graficación de coordenadas reconstruidas con rezagos en dimensiones suficientemente grandes para atrapar y desenredar (evitar que existan autointersecciones) el atractor. De hecho las gráficas de coordenadas reconstruidas han sido estudiadas independientemente en mucha literatura de las ciencias físicas [Packard, et al.,1980].

Esta técnica es una de las pocas técnicas disponibles para analizar datos potencialmente caóticos. El hecho de que la reconstrucción sea uno a uno implica afortunadamente que las *Redes Neuronales* pueden ser utilizadas para predecir el comportamiento futuro del sistema, incluso en el caso en el que éste sea caótico.

Los resultados anteriores pueden ser generalizados para atractores A que tengan una dimensión fractal (atractores extraños) en lugar de variedades (dimensión entera).

Teorema de Takens (segunda versión)

Supongamos que A es un subconjunto de $\Re^k$ con dimensión (box-counting) d, el cual es invariante bajo el Sistema Dinámico G. Si $m>2d$ y $F:\Re^k \rightarrow \Re^m$ es una función de coordenadas reconstruidas con rezagos a partir de una función genérica de medida h y a un tiempo genérico de rezago T, entonces F es uno a uno en A.■

En realidad en el trabajo de Takens [Takens, 1981], él propone cuatro teoremas y un corolario a partir de los cuales se desprende mucha de la teoría para la reconstrucción de la evolución del Sistema Dinámico. Uno de esos teoremas (primera versión y segunda versión) es el que más claramente refleja el porque en la práctica, al trabajar con Sistemas Dinámicos que evolucionan en variedades *m-dimensionales* contenidas en $\Re^k$, es suficiente para hacer una representación (reconstrucción) adecuada de ellos, para su posterior modelación y pronóstico, trabajar solamente con una serie de tiempo unidimensional del sistema y con sus rezagos.

De la misma forma en la que la derivada de una función, nos proporciona su mejor aproximación lineal en un punto, la mejor aproximación lineal para una variedad M en un punto x está dada por su espacio tangente; supongamos por ejemplo que tenemos una variedad M de dimensión 2 ubicada en $\Re^3$, en este caso, es sencillo visualizar que el espacio tangente es precisamente un plano. Si tenemos una curva C sobre la variedad M, el vector tangente a C en un punto x es un vector en $\Re^3$ dado por $V_x^C = \frac{dx}{dt}$ (véase el siguiente teorema). El espacio tangente a $\Re^n$ en un punto es el mismo $\Re^n$ nuevamente. En general el espacio tangente tiene la misma dimensión que la variedad.

Teorema de Takens (Versión final)

Sea A una variedad compacta de dimensión d. Para parejas (X,Y) donde X es un campo vectorial suave (al menos de clase C^{2m+1}) y Y una función suave en A, es una propiedad genérica que el mapeo $\phi_{X,Y}: A \to \Re^{2d+1}$, definido por

$$\phi_{X,Y}(x) = \left(y(x), \frac{d}{dt}(y(\varphi_t(x)))\Big|_{t=0}, \ldots, \frac{d^{2d}}{dt^{2d}}(y(\varphi_t(x)))\Big|_{t=0} \right)$$

es un incrustamiento.

Aquí φ_t denota el flujo del campo vectorial X, y Y una *función observable* que da origen a la serie de tiempo univariada.

Corolario

Sea A una variedad compacta de dimensión d. Consideremos arreglos de cuatro componentes, consistentes de un campo vectorial X, una función Y, un punto p y un número real positivo τ. Para funciones genéricas X y Y, y para τ satisfaciendo las condiciones genéricas impuestas por X y p, el conjunto límite hacia adelante de p ($L^+(p)$) es difhomeomórfico con el conjunto de puntos límite de la siguiente sucesión en $\Re^{2d+1}$

$$\left\{ y(\varphi_{k\tau}(p)), y(\varphi_{(k+1)\tau}(p)), \ldots, y(\varphi_{(k+2d)\tau}(p)) \right\}_{k=0}^{\infty}$$

El significado de difhomeomórfico debe de ser claro aquí: significa que existe un incrustamiento suave de A en $\Re^{2d+1}$ mapeando $L^+(p)$ biyectivamente al conjunto de puntos límite de la sucesión anterior.

Lo que el corolario establece simplemente, es que el atractor del Sistema Dinámico original y el atractor de la sucesión de vectores en $\Re^{2d+1}$ formada por una serie de tiempo unidimensional muestreada a intervalos de tiempo τ, son el mismo!.

1.11 Información Mutua

La *Información Mutua* es una medida de la cantidad de información que una variable aleatoria contiene acerca de otra; de alguna forma es una medida de la distancia entre dos

distribuciones de probabilidad o bien, es una reducción en la incertidumbre de una variable aleatoria debido al conocimiento de otra.

Definición (*Información Mutua*):

Sean X y Y dos variables aleatorias definidas sobre el mismo espacio de probabilidad $(\Omega, A, P(.))$. Sea $f_{XY}(x,y)$ la función de densidad conjunta de X y Y, y $f_X(x)$ y $f_Y(y)$ las funciones de densidad marginales de X y de Y respectivamente. Definimos la información mutua entre X y Y como

$$I(X,Y) = \sum_y \sum_x f_{XY}(x,y) \log \frac{f_{XY}(x,y)}{f_X(x) f_Y(y)}$$

La información mutua es una herramienta muy útil para la reconstrucción del espacio fase, pues de acuerdo con el Teorema de Takens, versión final, y al corolario que de él se desprende, el vector en $\Re^{2d+1}$ que es difhomeomórfico con $L^+(p)$, se forma mediante la selección de *rezagos apropiados* $n\tau$: $n=1,2,...,2d+1$ de la serie de tiempo observable, y precisamente la *longitud base* τ, de esos rezagos apropiados está dada por el primer mínimo en la información mutua de la serie de tiempo, es decir, se tendría que calcular $I(x_i, x_{i-\tau})$, para toda x_i en la serie de tiempo.

La información mutua es un estadístico computacionalmente costoso y requiere además grandes cantidades de datos para su cálculo adecuado.

Más recientemente se ha encontrado que la elección de una única τ para generar todo el espacio de incrustamiento (sin importar su dimensión) no garantiza en general que la *redundancia* entre todas las componentes sea efectivamente minimizada; esto ha dado como origen la definición y utilización de *rezagos de incrustamiento no uniformes*, pero este es un aspecto que no será tratado en este trabajo.

Capítulo 2 Redes Neuronales Artificiales.

2.1 Introducción

Actualmente la gran mayoría de las actividades de procesamiento de información son realizadas en computadoras digitales. Esta situación ha generado la amplia creencia de que esta actividad es dependiente exclusivamente de este tipo de computadoras. Sin embargo, si tomamos en cuenta algunos de los desarrollos más recientes en la tecnología de cómputo óptico y recordamos además las actividades que en cómputo analógico fueron desarrolladas antes de 1960, deberá ser suficiente para convencernos de que existen otras alternativas para realizar cálculos y procesar información.

El área de la cibernética reconoce que el procesamiento de la información se origina en el esfuerzo cotidiano que tienen las criaturas vivientes por sobrevivir. Desde este punto de vista podemos comenzar a considerar la posibilidad de existencia de otras técnicas de procesamiento de información que son diferentes de aquellas utilizadas en las computadoras digitales convencionales.

Una dirección en la cual la investigación está siendo vigorosamente influenciada se refiere a la posibilidad de construir dispositivos procesadores de información que emulen las estructuras y los principios operacionales que comúnmente son encontrados en los seres humanos y en otras criaturas vivientes. Esto ha producido una nueva casta de computadoras conocidas como neurocomputadoras, que son diferentes de las computadoras digitales en el sentido de que estas últimas son dispositivos secuenciales diseñados para manipular representaciones simbólicas del mundo externo. Las neurocomputadoras, por otro lado, son estructuras altamente paralelas diseñadas para procesar directamente la información proveniente del mundo externo, sin el paso intermedio de la representación simbólica.

Para entender el interés que existe por la neurocomputación, es útil, además de necesario, el entender también los fundamentos de los conceptos neurobiológicos y su relevancia en el requerimiento de una computación veloz y eficiente.

Las neuronas son células nerviosas y las redes neuronales son redes de esas células. El cerebro (en particular la corteza cerebral que es la capa más externa del cerebro) puede ser pensado como un ejemplo de una red neuronal natural. La motivación para los estudios clásicos en el campo de las redes neuronales artificiales estuvo originada en consideraciones anatómicas y psicológicas.

Los primeros modeladores de las neuronas artificiales estuvieron interesados no solamente en las relaciones estímulo-respuesta de las neuronas biológicas, sino también en la representación de la estructura interna de la neurona. Sin embargo, fueron en realidad las propiedades computacionales de las neuronas artificiales lo que atrajo la atención de los

científicos en varias otras disciplinas. El gran flujo de interés que esta área ha recibido de matemáticos, físicos y personas del área de la computación, se ha ido incrementado gradualmente. El creciente interés en las redes neuronales artificiales está motivado por el deseo de construir una nueva clase de computadoras y de algoritmos más potentes para resolver problemas que han demostrado ser difícilmente manejables con las herramientas tradicionales de cálculo y de procesamiento de la información. Esas tareas son análogas a aquellas realizadas típicamente, en forma natural y velozmente por el cerebro humano, sin embargo, ellas están mucho más allá del alcance de las computadoras convencionales y de los denominados *Sistemas Expertos*. Por ejemplo, un humano puede normalmente reconocer una cara familiar en aproximadamente 200 milisegundos. El ojo humano puede ajustar los niveles de la intensidad de la luz en 7 niveles de magnitud diferentes. Ningún sistema procesador de imágenes hecho por el hombre puede aproximarse siquiera al desempeño de este sistema humano; sorprendentemente este desempeño es obtenido por un sistema cuyos componentes individuales son más voluminosos, más lentos y más inexactos que los actuales componentes electrónicos de vanguardia.

El proceso computacional implicado en el desempeño de una red neuronal artificial es como sigue: una neurona artificial (o elemento procesador) recibe entradas de otras neuronas, de ella misma, o directamente de estímulos externos. Una suma ponderada de estas entradas constituye el argumento para una función de activación (o función de transferencia). Esta función de transferencia, que pretende emular las propiedades de una neurona real, es generalmente no lineal. El valor resultante de la función de transferencia constituye la salida de la neurona artificial. Esta salida es distribuida a través de conexiones ponderadas hacia otras neuronas artificiales. La forma en la cual se definen esas conexiones (conocida como la topología de la red) define el flujo de información en la red y esto a su vez define lo que se conoce como arquitectura de la red. Algunas arquitecturas típicas son por ejemplo: una sola capa, varias capas, alimentación hacia delante, alimentación hacia atrás, recurrencia total y conectividad lateral. Las conexiones ponderadas en esas arquitecturas juegan un papel tan importante que incluso a esas redes se les conoce como *modelos conexionistas de computación.*

El concepto de *memoria* en una computadora convencional es trasladado en el concepto de actualización de los pesos en una red neuronal. No existe una memoria central distinta y separada de la unidad central de procesamiento (CPU). El método utilizado para ajustar los pesos en el proceso de entrenamiento de la red se conoce como la regla o el algoritmo de aprendizaje. Es decir, los sistemas neuronales artificiales no son programados, sino más bien son enseñados o entrenados. El aprendizaje puede ser supervisado o no supervisado. La regla de aprendizaje supervisado que es mayormente utilizada, es el algoritmo de retropropagación del error. Una clase de aprendizaje no supervisado es el método denominado de auto-organización. En resumen, los tres ingredientes necesarios para un sistema computacional basado en las redes neuronales artificiales son: la función de transferencia, la arquitectura y la regla de aprendizaje. Se debe de enfatizar, sin embargo, que los modelos computacionales de este tipo tienen solamente un parecido metafórico con los cerebros reales.

2.2 Fundamentos Biológicos.

De acuerdo a lo que hoy se sabe, la anatomía de una neurona biológica real incluye como sus principales partes las siguientes:

- Una estructura arborecente conocida como las *dendritas*, que es por donde las neuronas reciben o atrapan las señales provenientes de las otras neuronas.
- El cuerpo de la célula (conocido como *soma*) que es donde se localiza el núcleo. Las dendritas están conectadas al cuerpo de la célula.
- El *axon*, que es un tejido largo fibroso y que se extiende desde el cuerpo de la célula hasta eventualmente ramificarse en filamentos y subfilamentos.
- En los extremos de las ramificaciones del axon se encuentran las uniones sinápticas o *sinapsis* hacia las otras neuronas. Las uniones sinápticas juegan el doble papel de terminaciones transmisoras en una célula y terminaciones receptoras en las otras células. Las sinapsis pueden ser encontradas tanto en las dendritas como en el propio cuerpo de la célula. El axon de una neurona típica realiza algunos miles de sinapsis con otras neuronas

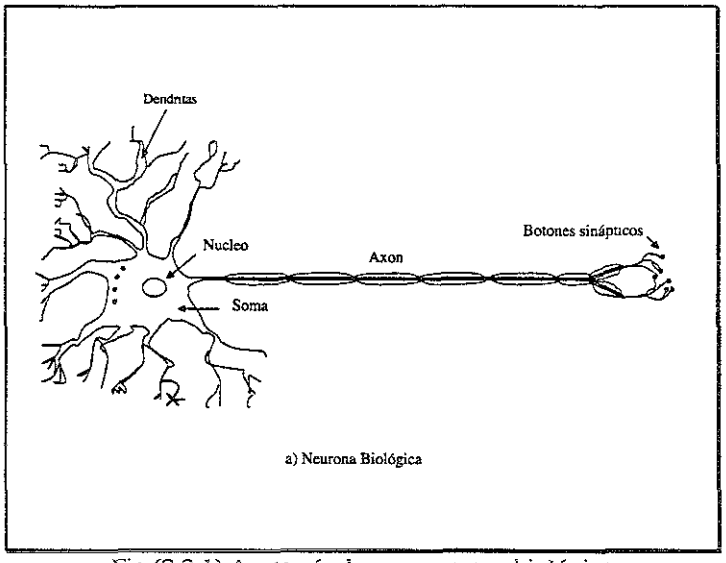


Fig.(2.2.1) Anatomía de una neurona biológica

La transmisión de una señal (típicamente en forma de un tren de pulsos) de una célula a otra a través de la sinapsis es un proceso químico complejo en el cual algunas sustancias específicas que favorecen la transmisión son emitidas del lado de la neurona emisora con el objetivo de incrementar o disminuir el potencial eléctrico dentro del cuerpo de la célula receptora. Si este potencial alcanza un umbral determinado, un pulso de duración e intensidad fija es enviado a través del axon. Es entonces cuando se dice que la neurona ha

disparado. El pulso se ramifica a través de las arborizaciones del axon hasta alcanzar las uniones sinápticas con las otras células. Después de disparar la célula tiene que esperar un intervalo de tiempo conocido como período de refracción antes de que pueda volver a disparar otra vez. Cabe aclarar que una sinapsis, en realidad, no es una conexión física

La figura (2.2.1) que corresponde a un diagrama de una neurona real, muestra lo descrito hasta ahora. La figura (2.2.2) representa una abstracción matemática de una neurona. McCulloch y Pitts [1943] propusieron un modelo simple de neurona como una unidad binaria con umbral. Específicamente, la neurona modelo calculaba una suma ponderada de las entradas provenientes de otras unidades hacia ella y producía un uno o un cero dependiendo de si esta suma era mayor o menor que cierto umbral.

$$x_i(t+1) = \theta(\sum_j w_{ij} x_j(t) - \mu_i) \tag{2.2.1}$$

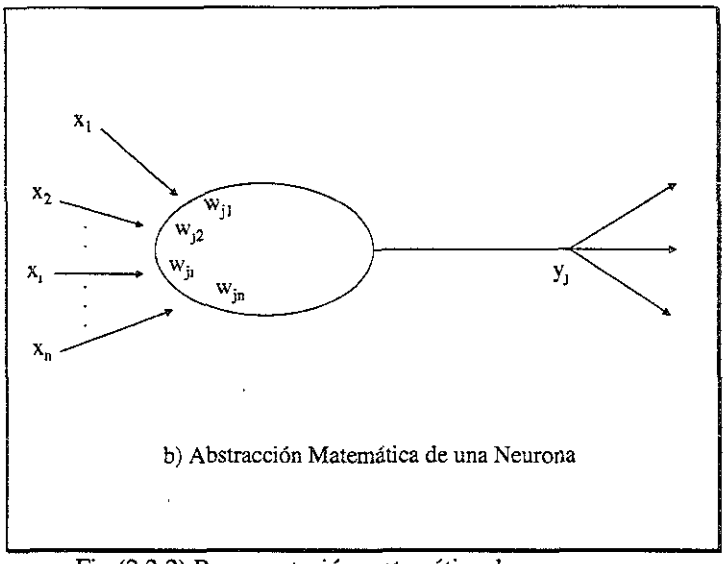
Donde x_i puede ser 1 o 0, y representa el estado de la neurona i disparando o no disparando respectivamente. El tiempo t es tomado en forma discreta, con una unidad de tiempo consumida en cada paso de procesamiento. $\theta(x)$ es la función escalón.

$$\theta(x) = \begin{cases} 1 & si \quad x \geq 0 \\ 0 & en\ otro\ caso \end{cases} \tag{2.2.2}$$

Aquí $x_1, x_2, \ldots, x_n$ se refieren a las entradas recibidas por la neurona i, los pesos w_{ij} representan la intensidad de la sinapsis que une la neurona j con la neurona i, este peso puede ser positivo o negativo correspondiendo a una sinapsis exitatoria o a una inhibitoria respectivamente, o es cero si no existe sinapsis entre i y j. El parámetro μ_i es el valor del umbral para la unidad i; la suma ponderada de entradas debe de igualar o de exceder el valor del umbral para que la neurona dispare.

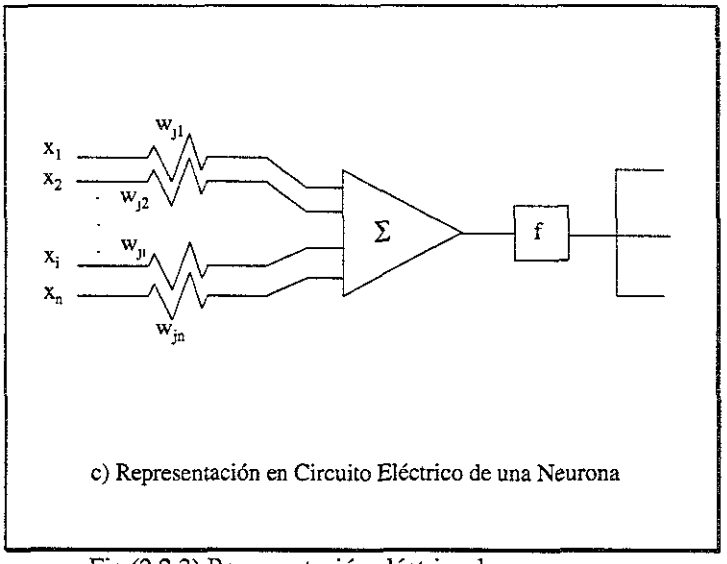
Aunque simple, el modelo de neurona de McCulloch y Pitts es computacionalmente un dispositivo muy poderoso. McCulloch y Pitts mostraron que un arreglo síncrono de tales neuronas es capaz en principio del *cálculo universal*, para un conjunto adecuado de pesos. Esto significa que pueden realizar cualquier tipo de cálculo que una computadora ordinaria, aunque no necesariamente tan rápido o en forma tan conveniente.

La figura (2.2.3) corresponde a la representación en un circuito electrónico a partir del modelo matemático de una red neuronal. Aquí cada x_i es un voltaje y cada w_{ij} está representado por un potenciómetro. La caja triangular, con un signo de suma dentro, es un amplificador operacional configurado como un sumador. La cantidad f es alguna función no lineal apropiada.



b) Abstracción Matemática de una Neurona

Fig.(2.2.2) Representación matemática de una neurona



c) Representación en Circuito Eléctrico de una Neurona

Fig.(2.2.3) Representación eléctrica de una neurona

Una generalización muy simple del modelo McCulloch y Pitts está dado por $x_i = g(\sum_j w_{ij} x_j - \mu_i)$, donde el número x_i se evalúa ahora en forma continua y se denomina como el estado o nivel de activación de la unidad *i*. La función escalón $\theta(x)$ se reemplaza por una función no lineal más general *g(x)* conocida como función de activación o función de transferencia. En lugar de actualizar del tiempo *t* al tiempo *t+1* como se hizo en el modelo anterior, ahora simplemente se da una regla para actualizar x_i siempre que sea necesario. Las unidades son frecuentemente actualizadas en forma asíncrona, es decir, en orden aleatorio para tiempos aleatorios.

Cuando se está trabajando con redes neuronales artificiales es una práctica común hablar de neuronas y de sinapsis, pero más formalmente, dado que las componentes del modelo están lejos de sus contrapartes biológicas, aquí se preferirá utilizar los términos *unidades y pesos* (o conexiones) respectivamente.

Se estima que el cerebro humano tiene entre 10^9 y 10^{12} neuronas, con más de 10^{15} sinapsis entre ellas. Esas células son las unidades básicas procesadoras de información del cerebro. Se estima, de igual forma, que una neurona típica recibe sus entradas de aproximadamente otras 10000 neuronas y envía su señal a quizás 1000 neuronas. La transmisión de esos pulsos de una neurona a otra parece ser un proceso lento (menos de 120 metros/seg.). Cuando una neurona real dispara, su salida es un tren de pulsos cuya frecuencia está relacionada casi en forma logarítmica, a la magnitud de la suma algebraica de entradas recibida. La frecuencia de disparo de una neurona típicamente se encuentra entre 1 y 100 ciclos por segundo, esto implica que sólo unos pocos bits están involucrados en la transmisión de la información de una neurona a otra. A partir de estos parámetros no es posible entender cómo es que el cerebro se desempeña, en las tareas que ya hemos mencionado, a las velocidades que podemos percibir.

Aunque actualmente se tiene una idea muy razonable acerca de la operación de las neuronas reales y un conocimiento acerca de la conectividad entre ellas, ciertamente no sabemos como algunos grupos de neuronas trabajan juntas en forma tan funcional. Lo mejor que para estos tiempos podemos esperar es una imitación adecuada de la estructura y trabajar para que algo de su funcionalidad sea reproducida. Duplicar por completo la estructura del cerebro es algo impráctico y muy costoso; además existen evidencias de que mucho del procesamiento de la información en el cerebro no es solamente un proceso paralelo, sino también de alguna forma, un procesamiento distribuido por función. Algunas evidencias provenientes de la neuropatología, de la neurohistología y de la neurofarmacología indican que existen aproximadamente 1000 regiones localizadas (o módulos) en el cerebro donde las asignaciones de trabajo son realizadas. Por ejemplo, la habilidad para hablar está asociada con el *área de Broca* en el lóbulo frontal izquierdo de la corteza cerebral, y la habilidad para entender el lenguaje natural está asociada con el *área de Wernicke* en la parte posterior izquierda del lóbulo temporal. De esta forma, la capacidad de desempeño de alto nivel que tiene el cerebro es quizás debida al gran número de grupos neuronales que, del alguna manera, se comunican entre ellos; por ejemplo, actualmente se cree que las estructuras que están mayormente involucradas en la formación de la memoria son el cerebelo, el hipocampo, la amígdala y la corteza cerebral. Esta clase de información nos conduce a pensar que con el cerebro humano estamos quizás tratando con una estructura que está regionalmente distribuida por función y que localmente tiene un procesamiento paralelo. Adicionalmente se cree que el "disparo paralelo" de neuronas se lleva a cabo asíncronamente, pues no existe evidencia de un control central para la sincronización. Debido a todas las justificaciones anteriores, se piensa que ahora una red neuronal biológica está tal vez mejor representada por grupos de estructuras de procesamiento paralelo, distribuidas por función y comunicándose entre ellas en alguna forma asíncrona. Imaginemos la potencialidad de un sistema artificial, el cual pudiera trabajar en paralelo como el cerebro pero con la velocidad de los dispositivos electrónicos actuales.

2.3 Aprendizaje.

El aprendizaje es una de las características más importantes de las redes neuronales artificiales. Todo el *conocimiento* en una red neuronal artificial está codificado en los pesos conectores w_{ij}, y el proceso de aprendizaje consiste precisamente en la determinación de esos pesos. Un peso representa la intensidad de asociación entre dos unidades (neuronas). Al nivel de la red, un peso representa qué tan frecuentemente una unidad receptora ha estado activa simultáneamente con la unidad emisora. De esta forma, el cambio en el peso entre dos unidades depende de la frecuencia con que las dos unidades estén teniendo una salida positiva en forma simultánea.

Si una red neuronal es visualizada como un mecanismo de mapeo de un espacio de entrada a un espacio de salida, entonces el problema de aprendizaje es esencialmente equivalente al problema de trabajar con una *memoria asociativa* (en una memoria asociativa el interés consiste en recuperar la información almacenada usando como pistas de entrada conjuntos incompletos o conjuntos corruptos de la misma información) que recupera una salida adecuada cuando se le es presentada alguna entrada y generaliza cuando se le presenta una nueva entrada no conocida. El aprendizaje puede también ser pensado como el problema de *estimar o calcular* el sistema que transforma un conjunto de entradas en un conjunto de salidas a partir de un conjunto dado de ejemplos de entradas-salidas. Un ejemplo clásico para este problema consiste en la teoría de interpolación, en el cual deseamos aproximarnos a una función continua $f(x)$ mediante una función interpoladora $p(w,x)$ que depende de un cierto número de parámetros $w=\{w_1,w_2,...,w_n\}$ y de puntos $x=\{x_1,x_2,...,x_n\}$. La elección de p, típicamente como un polinomio, es equivalente a la elección del conjunto w que proporciona el mejor ajuste. En esto consiste la fase de aprendizaje.

Una de las razones de la gran excitación por el uso de las redes neuronales artificiales es su habilidad para *generalizar* a nuevas situaciones. Después de haber sido entrenada con un número de ejemplos que satisfacen determinada relación, la red puede frecuentemente *deducir* la relación subyacente completa y de esta manera interpolar y extrapolar, a partir de los ejemplos, de una forma muy razonable.

La alta conectividad de las redes se traduce en el hecho de que los posibles errores en algunas unidades serán finalmente intrascendentes. Esto significa que tales sistemas son altamente robustos y que su desempeño se ve afectado mínimamente ante la presencia de ruido o errores. En el cerebro mismo, las células están muriendo todo el tiempo sin afectar su funcionamiento, y probablemente ha sido esta robustez en las redes neuronales biológicas lo que ha sido determinante en la evolución de la inteligencia.

El contraste entre esta clase de procesamiento y el procesamiento convencional de la clase de Von Neumann no podía ser más grande. Aquí tenemos muchos procesadores, cada uno de los cuales está ejecutando un programa muy simple, mientras que en la situación tradicional un solo procesador ejecuta secuencialmente programas complicados. En

contraste con la robustez de una red neuronal, un cálculo secuencial ordinario, en el esquema de Von Neumann, puede fácilmente ser arruinado por un simple error en un bit.

2.4 Antes de la implementación.

Una de las principales causas para el gran entusiasmo hacia las redes neuronales artificiales tiene que ver con el avance que ha habido en la tecnología desde los días del perceptrón. Las primeras implementaciones de redes neuronales artificiales utilizaron amplificadores electrónicos y potenciómetros. La principal desventaja de estos esquemas tiene que ver con la dificultad para ajustar los valores del potenciómetro y con la incapacidad de utilizar miles de amplificadores electrónicos con millones de conexiones. Con la llegada de la tecnología VLSI (integración a muy larga escala), es ahora posible proponer diseños con miles de neuronas y millones de sinapsis.

A pesar de la tecnología VLSI, la mayoría de la investigación y aplicaciones en redes neuronales artificiales es aún realizada con la ayuda de programas simuladores que corren en las computadoras secuenciales o en supercomputadoras. Sin embargo, cada vez más, las implementaciones en hardware están siendo probadas experimentalmente gracias a los avances que han habido en el área del diseño asistido por computadora.

El paralelismo masivo en las redes neuronales es extremadamente atractivo en principio, pero en la práctica existen muchos detalles que deben de ser resueltos antes de que una implementación exitosa pueda ser conseguida para algún problema específico; consideremos por ejemplo lo siguiente para algún problema específico,

- ¿Cuál es la mejor arquitectura?
- ¿Deben o no ser divididas en capas las unidades de la red?
- ¿Cuántas conexiones deben de ser hechas entre las unidades?
- ¿Cuántas unidades de entrada debe de llevar una red, Cuántas intermedias, Cuántas capas?.
- ¿Qué clase de función de activación debe de ser utilizada?
- ¿Cómo debe de realizarse la actualización en las unidades: síncrona o asíncronamente?
- ¿Cuántas unidades se requieren para alguna tarea específica?
- En caso de que una red neuronal artificial pueda aprender, ¿cuántos ejemplos son necesarios para un buen desempeño?
- ¿Cuántos tipos de tareas diferentes puede aprender una red neuronal, qué tan bien y qué tan rápido?
- ¿Qué tan robusta es una red para trabajar con información faltante o incorrecta, o cuando alguna de sus unidades es removida o no trabaja adecuadamente?.

Evidentemente todas estas preguntas están de alguna forma relacionadas y no pueden ser contestadas en forma independiente.

En la sección correspondiente a las aplicaciones serán retomadas varias de las preguntas anteriores y se propondrán un conjunto de respuestas adecuadas en términos de las características específicas de los problemas a resolver, es decir, para algunos ejemplos de sistemas dinámicos clásicos de la literatura y para un conjunto de datos reales de precipitación en la ciudad de México, serán propuestas las arquitecturas correspondientes para las redes neuronales, las capas de entrada, el tipo de conexiones entre las unidades, el número total de unidades, el error de convergencia, la tasa de aprendizaje, etc., que permitirán que una red neuronal artificial sea capaz de aprender y pronosticar adecuadamente algún tipo de comportamiento.

2.5 Red y Algoritmo de Entrenamiento

En la sección anterior fue establecido que los tres componentes fundamentales de un sistema computacional basado en las Redes Neuronales son

- La función de transferencia
- La arquitectura de la red
- La regla o algoritmo de aprendizaje

El interés central en este momento tiene que ver con el punto tres, es decir, tiene que ver con la descripción de alguna regla u algoritmo de aprendizaje para la red neuronal. El algoritmo o los algoritmos que serán descritos en las secciones siguientes han demostrado ser muy eficientes para ciertas arquitecturas de red en la solución de algunos problemas muy específicos, como por ejemplo la modelación de Sistemas Dinámicos. El algoritmo estándar de *retropropagación del error*, que será el primero en mencionarse, nos servirá de base para la definición de su versión con *gradientes conjugados.*

Estos dos algoritmos nos indican la forma en la que los pesos w_{ij}, en cualquier red de alimentación hacia adelante, deben de ser actualizados para aprender un *conjunto de entrenamiento* de *patrones de entradas-salidas*. En este caso solamente la versión con los gradientes conjugados será utilizada en los capítulos correspondientes a las aplicaciones a los Sistemas Dinámicos, pues ha sido demostrado que su velocidad de convergencia y su bondad de ajuste son superiores [Reifman y Vitela;1994] en comparación con la versión estándar.

Estas dos reglas de aprendizaje utilizan alguna variación del algoritmo de gradiente descendente, cada uno con sus particularidades en función de la arquitectura de la red o de algún criterio para acelerar su convergencia.

2.6 Redes de Alimentación hacia adelante.

Se ha mencionado en las secciones anteriores que existen 2 paradigmas generales de aprendizaje: *aprendizaje supervisado y aprendizaje no supervisado.* En el aprendizaje supervisado la red puede comparar sus salidas con las respuestas correctas conocidas y recibir una retroalimentación en función de los errores. A este tipo de aprendizaje también se le conoce como aprendizaje con maestro; el maestro le dice a la red cuáles son las respuestas correctas, o cuando menos (en el aprendizaje reforzado) si las respuestas son correctas o no. En el aprendizaje no supervisado, no existe maestro ni respuestas correctas o incorrectas; la red debe de descubrir por ella misma las categorías interesantes o comunes en los datos de entrada.

Normalmente, al estar trabajando con redes neuronales artificiales, se considera que existe una lista o conjunto de entrenamiento de pares correctos de entradas-salidas. Cuando se aplica una de las entradas de entrenamiento a la red, es posible comparar la salida de la red con la salida correcta y entonces actualizar el valor de las conexiones w_{ij} para minimizar la diferencia. Típicamente esto se hace en forma incremental, realizando mínimos ajustes en respuesta a cada par del conjunto de entrenamiento, de tal forma que los w_{ij}'s convergan a una situación en la cual el conjunto de entrenamiento es conocido y reproducido con gran fidelidad. Es entonces interesante probar con otros patrones de entrada que no pertenezcan al conjunto de entrenamiento, para ver si la red puede en forma exitosa generalizar lo que ha aprendido.

Las redes en capas de alimentación hacia adelante fueron llamadas *Perceptrones* cuando fueron primeramente estudiadas a gran detalle por Rosenblatt y sus colaboradores hace mas de 35 años [Rosenblatt,1962]; la figura (2.6.1) muestra dos ejemplos de perceptrones

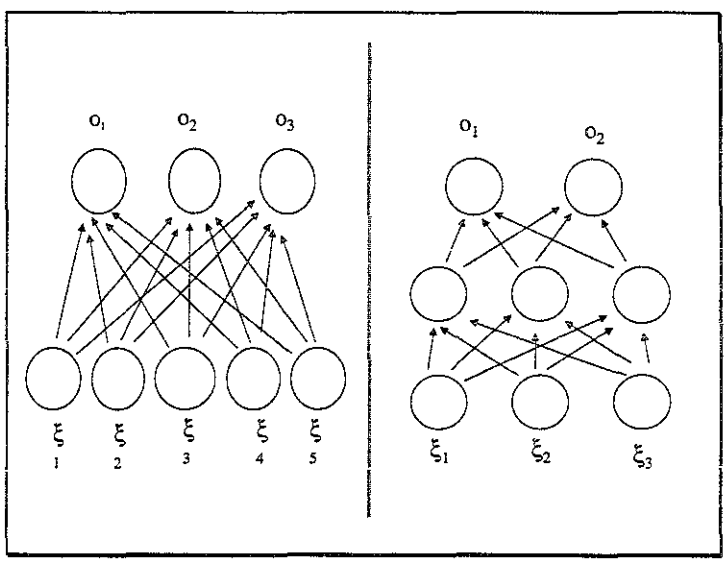


Fig (2.6.1). Dos perceptrones, con 1 y 2 capas respectivamente

Hay un conjunto de unidades de entrada (capa de entrada) cuyo único papel consiste en alimentar los patrones de entrada hacia el resto de la red. Después de la capa de entrada pueden seguir una o más capas intermedias de unidades, seguida por una capa final de salida donde el resultado de los cálculos es leído. En la clase restringida de las redes con alimentación hacia adelante que se están estudiando en este momento, no se permiten las conexiones que van de una unidad a ella misma, o a otras unidades en la capa previa, en la misma capa o a más de una capa adelante. Cada unidad alimenta solamente a las unidades de la siguiente capa. Las unidades de las capas intermedias son frecuentemente llamadas unidades ocultas puesto que ellas no tienen conexión directa con el mundo externo, ni de entrada ni de salida.

Normalmente se utilizan dos convenciones para contar el número de capas en una red neuronal; algunos autores cuentan las unidades de entrada como una capa y otros no. Para las secciones siguientes se utilizará esta última convención, de tal forma que diremos que una red con una capa oculta tiene 2 capas en lugar de 3. Notemos que una red con N capas tiene N capas de conexiones y N-1 capas ocultas. Las redes de alimentación hacia adelante tienen por definición matrices asimétricas de conexiones w_{ij} pues todas las conexiones son unidireccionales.

Considérese ahora una red de alimentación hacia adelante con una sola capa (perceptrón simple), esta red tiene un conjunto de N unidades de entrada y una capa de salida, pero no capas intermedias. La figura (2.6.2) ilustra este tipo de red y da pauta para establecer la notación con la que se trabajará de aquí en adelante.

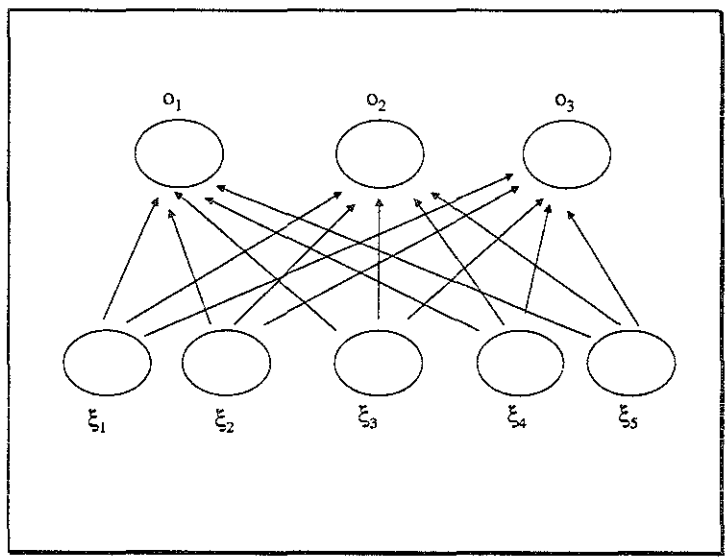


Fig.(2.6.2) Perceptrón Simple.

Las entradas y las salidas son llamadas ξ_k y o_i respectivamente. El valor de o_i está expresado como

$$o_i = g(h_i) = g(\sum_k w_{ik}\xi_k) \tag{2.6.1}$$

Donde $g(\cdot)$ es la función de transferencia que utilizan las unidades. Normalmente $g(\cdot)$ es elegida como una función no lineal, como por ejemplo, una función escalón o una sigmoide continua. También se llegan a utilizar incluso funciones estocásticas.

Debemos de notar que la salida es una función explícita de las entradas. Esta es una caracerística que es válida para todas las redes de alimentación hacia adelante; la entrada es propagada a través de la red y la salida es en forma inmediata. En contraste, algún otro tipo de redes como por ejemplo las redes totalmente recurrentes, siempre necesitan algún tiempo de espera para producir la salida final (por ejemplo, si un patrón entra al tiempo t, la salida se dará al tiempo $t+2$).

Hasta este momento ha sido omitido el uso de los valores de umbral en las unidades (que es la cantidad que debe de ser igualada o excedida para que la neurona dispare), puesto que estos pueden ser siempre tratados como los valores de las conexiones desde una unidad de entrada que permanentemente tiene asignado el valor de -1. Específicamente, es posible definir $\xi_0=-1$ y elegir los valores de las conexiones $w_{i0}=\theta_i$ para obtener

$$o_i = g(\sum_{k=0}^{n} w_{ij}\xi_k) = g(\sum_{k=1}^{n} w_{ij}\xi_k - \theta_i) \tag{2.6.2}$$

donde θ_i son los umbrales. En el caso de que $g(\cdot)$ sea una función continua, θ_i también se conoce como sesgo. La tarea general de asociación puede establecerse ahora de la siguiente manera: preguntarse por un patrón particular de salida ζ_i^μ en respuesta a un patrón de entrada ξ_k^μ. Es decir, el objetivo es que el patrón actual de respuesta o_i^μ sea igual al patrón objetivo conocido ζ_i^μ.

$$o_i^\mu = \zeta_i^\mu \quad : \forall i, \mu \tag{2.6.3}$$

Para el perceptrón simple, la salida actual o_i^μ está dada por (2.6.1). Cuando el patrón de entrada ξ_k es denotado como ξ_k^μ, tendremos que

$$o_i^\mu = g(h_i^\mu) = g(\sum_k w_{ik}\xi_k^\mu) \tag{2.6.4}$$

si p es definido como el número de pares de entradas-salidas en el conjunto de entrenamiento, se tendrá entonces que $\mu = 1,2,...,p$.

Las entradas, las salidas y los valores objetivo pueden ser valores booleanos (por ejemplo ± 1 o 0,1) o variables continuas; para las salidas esto depende, evidentemente, del tipo de función de activación $g(\cdot)$ elegida. Es posible utilizar salidas con valores continuos, y sin embargo tener valores objetivo booleanos, en cuyo caso se debe de esperar solamente que las salidas o_i^μ se encuentren dentro de algún margen de los valores deseados.

Para perceptrones simples en los que teóricamente exista un conjunto de pesos w_{ik} que permitan un determinado cálculo, esos pesos podrán ser encontrados mediante una regla de aprendizaje muy sencilla. La regla de aprendizaje empieza de una primera inicialización (*first guess*) de los valores de los pesos y después mediante mejoras sucesivas actualiza sus valores, hasta que en un número finito de pasos, alcanza una respuesta adecuada.

Existen, sin embargo, algunos cálculos que a pesar de ser muy elementales son conceptualmente muy importantes y que el perceptrón simple no puede resolver. Afortunadamente los perceptrones multicapas pueden resolver muchos problemas que son imposibles con la arquitectura de una sola capa, de hecho, de acuerdo con Cyberkenko [Ciberkenko, 1989] , un perceptrón con una única capa intermedia es suficiente para aproximar cualquier función *suave* y continua.

2.7 Formas cuadráticas

Una forma cuadrática es simplemente una función escalar cuadrática que tiene la siguiente forma:

$$f(x) = \frac{1}{2} x^T A x - b^T x + c \qquad (2.7.1)$$

donde A es una matriz de n x n, x y b son vectores, es decir, matrices de n x 1, y c es un escalar constante. Si A es simétrica y definida positiva, entonces $f(x)$ es minimizada por la solución de $Ax=b$. Además, si A es definida positiva, la superficie definida por $f(x)$ tiene forma de un paraboloide, fig.(2.7.1).

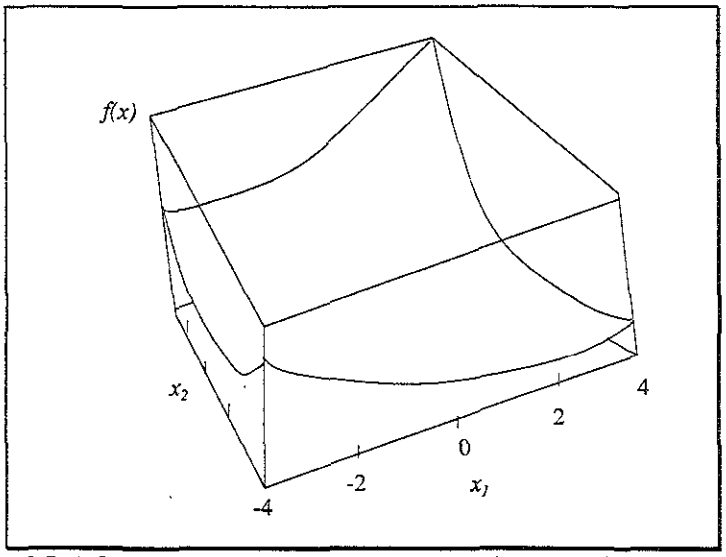


Fig.(2.7.1) Gráfica de una forma cuadrática cuando A es definida positiva

El gradiente de una forma cuadrática se define como

$$f'(x) = \begin{bmatrix} \dfrac{\partial}{\partial x_1} f(x) \\ \dfrac{\partial}{\partial x_2} f(x) \\ \vdots \\ \dfrac{\partial}{\partial x_n} f(x) \end{bmatrix} \tag{2.7.2}$$

El gradiente, como es sabido, es un campo vectorial que para un punto x dado, apunta en la dirección de incremento más grande de $f(x)$. Aplicando (2.7.2) sobre (2.7.1) es posible obtener

$$f'(x) = \frac{1}{2} A^T x + \frac{1}{2} Ax - b \tag{2.7.3}$$

Y si A es simétrica, (2.7.3) se reduce a

$$f'(x) = Ax - b \tag{2.7.4}$$

Por lo tanto, la solución para $Ax=b$ es un punto crítico para $f(x)$, y si A es definida positiva, además de simétrica, entonces esta solución es un mínimo de $f(x)$. Esto quiere decir que $Ax=b$ puede ser resuelto para encontrar el x que minimiza $f(x)$.

El hecho de que $f(x)$ sea un paraboloide es consecuencia directa de que la matriz A sea definida positiva. Si A no es definida positiva, existen otras posibilidades: A puede ser definida negativa. A puede ser singular en cuyo caso no existe una solución única, el conjunto de soluciones es una línea o un hiperplano a lo largo del cual f tiene un valor constante. Si A no es ninguna de las anteriores, entonces x es un punto silla, y las técnicas del *descenso más pronunciado (steepest descent)* y de los *gradientes conjugados* muy probablemente fallarán (estos algoritmos se estudiarán más adelante).

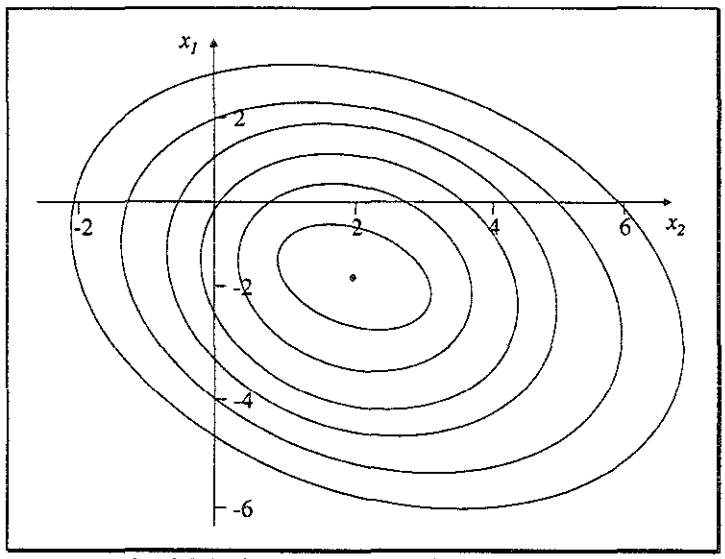


Fig.(2.7.2) Contornos de una forma cuadrática

2.8 Métodos de búsqueda

Los métodos de búsqueda consisten de un conjunto de técnicas para encontrar mínimos de funciones, estos métodos presuponen normalmente el uso de estimaciones del gradiente de la función para indicar la dirección en la cual se encuentra el mínimo de la superficie, por lo tanto también se les conoce como *métodos de descenso*. Como ejemplo se puede considerar el *método de Newton*, el *método del descenso más inclinado (steepest descent)*, y el *método de los gradientes conjugados*. El método de Newton es un método iterativo de búsqueda con gradiente que obliga a que cada una de las componentes del vector de pesos a optimizar sean actualizadas en cada ciclo durante proceso de búsqueda. Los cambios son siempre en la dirección del mínimo de la función objetivo o de costo, siempre que la superficie sea cuadrática.

El método del descenso más inclinado es un método de búsqueda con gradiente que también obliga a que todas las componentes del vector de pesos sean cambiadas a cada paso o iteración del proceso; en este caso, sin embargo, los cambios son en la dirección del gradiente negativo de la superficie de error y por tanto no están necesariamente en la dirección del mínimo de la función, puesto que el gradiente negativo solamente tiende hacia el mínimo cuando su origen (el del gradiente) se encuentra en uno de los ejes principales de la superficie, es decir, cuando el vector gradiente está exactamente sobre uno de los ejes principales de la función. Este algoritmo es de fácil implementación y ha demostrado su valor e importancia en una amplia gama de aplicaciones prácticas.

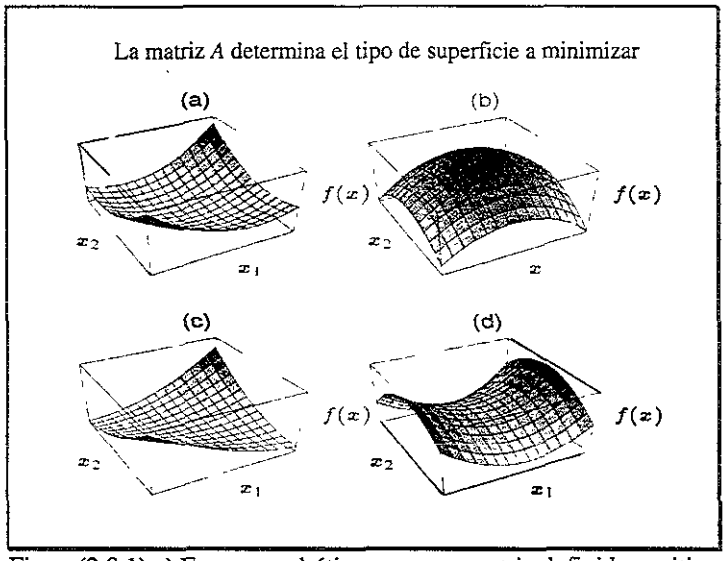
Finalmente, el método de los gradientes conjugados, que también es un método de búsqueda con gradiente, *corrige* muchas de las deficiencias que el método del descenso más inclinado tiene. Este método está muy estrechamente relacionado con el método del steepest descent. Para este método serán presentadas dos versiones, la versión lineal para formas cuadráticas, y la versión no lineal, para encontrar el mínimo de cualquier función continua $f(x)$ a la que se le pueda calcular el gradiente.

2.8.1 Método del descenso más inclinado

El método del descenso más inclinado (*steepest descent*) es un método de búsqueda con gradiente, el cual inicia en un punto arbitrario $x_{(0)}$ y a partir de él se avanza en dirección adecuada hacia el *fondo* de la función. Este algoritmo se ilustra más claramente si se asume que la función a minimizar es una forma cuadrática en la que A es una matriz simétrica y definida positiva, pues bajo tales condiciones $f(x)$ tiene forma de paraboloide, y la solución de $Ax=b$ ($x=A^{-1}b$) es un mínimo global de f, sin embargo debe de tenerse siempre presente que la función a minimizar no necesariamente será una forma cuadrática, pues, según se verá en el capítulo 3, para el entrenamiento de las redes neuronales, se definirá como función a minimizar el error de predicción de la red neuronal (ecuación (2.9.7)), el cual en general, no será una forma cuadrática. Para el caso de una $f(x)$ cuadrática, si A no es definida positiva, existen otras posibilidades: A puede ser definida negativa, A puede ser singular, en cuyo caso no existe una solución única y el conjunto de soluciones es una línea o un hiperplano a lo largo del cual f tendrá un valor constante. Si A no es ninguna de las anteriores, entonces x es un punto silla, y las técnicas del *descenso más pronunciado (steepest descent)* y de los *gradientes conjugados* muy probablemente fallarán. La figura (2.8.1) ilustra más claramente este comportamiento.

En este algoritmo, después de haber iniciado en un punto $x_{(0)}$, se *calculan* una serie puntos $x_{(1)}, x_{(2)}, x_{(3)},...$, hasta que se está satisfecho con la cercanía de la aproximación calculada en relación con el mínimo verdadero de la función. Cada que se da un nuevo paso (para avanzar del punto $x_{(i)}$ al punto $x_{(i+1)}$, se elige la dirección en la cual f decrece más rápidamente, que es la dirección opuesta a $f'(x)$. De acuerdo a la ecuación (2.7.4), esta dirección es $-f'(x_{(i)}) = b - Ax_{(i)}$.

En el método del descenso más inclinado, sin embargo, la dirección del gradiente negativo de la superficie de error no está necesariamente en la dirección del mínimo de la función, puesto que el gradiente negativo solamente tiende hacia el mínimo cuando su origen (el del gradiente) se encuentra en uno de los ejes principales de la superficie (los ejes principales de la superficie pueden ser identificados localizando el eje mayor y el eje menor de las elipses que se forman con las curvas de nivel de la superficie. Ver figura (2.8.3)), es decir, cuando el vector gradiente está exactamente sobre uno de los ejes principales de la función.



Figura(2.8.1) a) Forma cuadrática para una matriz definida positiva b) Para una matriz definida negativa; c) Para una matriz singular y indefinida positiva, la linea que atraviesa el fondo del *valle* es el conjunto de soluciones; d) Para una matriz indefinida: la solución es un punto silla

Las siguientes definiciones son muy importantes para este método y para el de gradientes conjugados: el *error* $e_{(i)} = x_{(i)}-x$ es un vector que indica que tan lejos se está de la solución óptima: el *residual* $r_{(i)} = b - Ax_{(i)}$ indica que tan lejos se está del valor correcto de b. Es sencillo observar que $r_{(i)}= -Ae_{(i)}$, y por lo tanto los residuales pueden pensarse como los errores transformados por A dentro del mismo espacio de b. Adicionalmente, y más importante todavía es el hecho de que $r_{(i)}= - f'(x_{(i)})$, y por tanto puede pensarse a los residuales como la dirección del descenso más pronunciado.

Supóngase que para un problema dado, el algoritmo inicia en un punto $x_{(0)}=[-2,-2]^T$. El siguiente paso en la dirección del descenso más pronunciado, *caerá* en algún lugar del vector con origen en *(-2,-2)*, figura (2.8.2 a). En otras palabras, se eligirá un punto

$$x(1) = x(0) + \alpha\, r(0) \qquad (2.8.1)$$

donde *r(0)* es la dirección de descenso más pronunciado con respecto al punto *x(0)*, y α es un número real constante. La pregunta es: *qué tan grande es el paso que se debe dar?.*

Una búsqueda en línea es un procedimiento que permite determinar el α que minimiza una función f a lo largo de una dirección dada. La figura (2.8.2 b) ilustra este procedimiento: estamos restringidos a elegir un punto en la intersección del plano vertical y del paraboloide. La figura (2.8.2 c) es la parábola definida por la intersección de esas dos superficies, y lo que se desea es determinar el valor α que minimiza esa función. Cuál es el valor de α en el mínimo de la parábola?.

α minimiza f cuando la derivada direccional $\frac{d}{d\alpha} f(x_{(1)})$ es igual a cero. Por la regla de la cadena $\frac{d}{d\alpha} f(x_{(1)}) = f'(x_{(1)})^T \frac{d}{d\alpha} x(1) = f'(x_{(1)})^T r_{(0)}$. Igualando esta expresión a cero, encontramos que α debe de ser elegido de tal forma que $r_{(0)}$ y $f'(x_{(1)})$ sean ortogonales. Fig. (2.8.2 d).

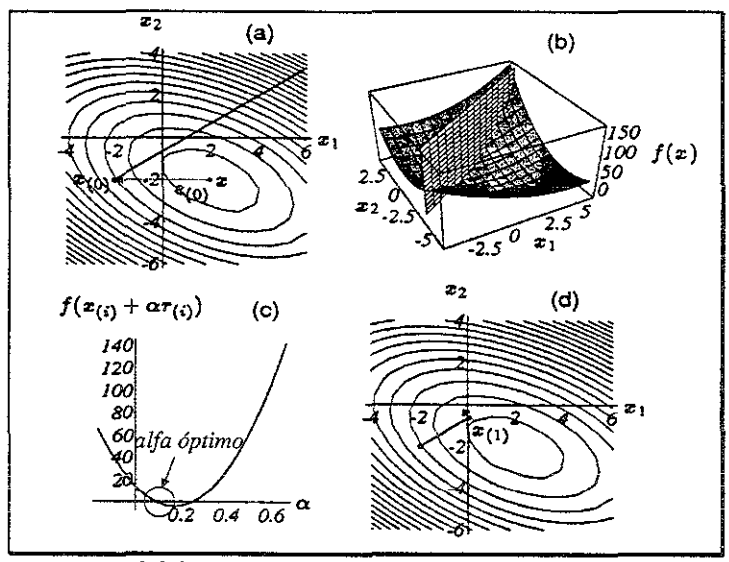


Fig.(2.8.2) El método del descenso más pronunciado

Para determinar α, nótese que $f'(x_{(1)}) = -r_{(1)}$, y por lo tanto:

$$\begin{aligned} r_{(1)}^T r_{(0)} &= 0 \\ (b - Ax_{(1)})^T r_{(0)} &= 0 \\ (b - A(x_{(0)} + \alpha r_{(0)}))^T r_{(0)} &= 0 \\ (b - Ax_{(0)})^T r_{(0)} - \alpha (Ar_{(0)})^T r_{(0)} &= 0 \\ (b - Ax_{(0)})^T r_{(0)} &= \alpha (Ar_{(0)})^T r_{(0)} \\ r_{(0)}^T r_{(0)} &= \alpha r_{(0)}^T (Ar_{(0)}) \\ \alpha &= \frac{r_{(0)}^T r_{(0)}}{r_{(0)}^T (Ar_{(0)})} \end{aligned} \qquad (2.8.2)$$

Por lo tanto, escribiéndolo todo junto, el método *steepest descent* está dado por

$$r_{(i)} = b - Ax_{(i)}, \qquad (2.8.3)$$

$$\alpha_{(i)} = \frac{r_{(i)}^T r_{(i)}}{r_{(i)}^T A r_{(i)}}, \qquad (2.8.4)$$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} r_{(i)}. \qquad (2.8.5)$$

Y típicamente su comportamiento geométrico se ilustra en la figura (2.8.3)

Este algoritmo, en la forma (2.8.3), (2.8.4), (2.8.5,) requiere dos multiplicaciones matriz-vector por iteración. En general, el costo computacional de los algoritmos iterativos está dominado por los productos matriz-vector; en este caso, afortunadamente, uno de ellos puede ser eliminado. Multiplicando $-A$ por la izquierda en ambos lados de la ecuación (2.8.5), y sumando b, se obtendrá

$$r_{(i+1)} = r_{(i)} - \alpha_{(i)} A r_{(i)} \qquad (2.8.6)$$

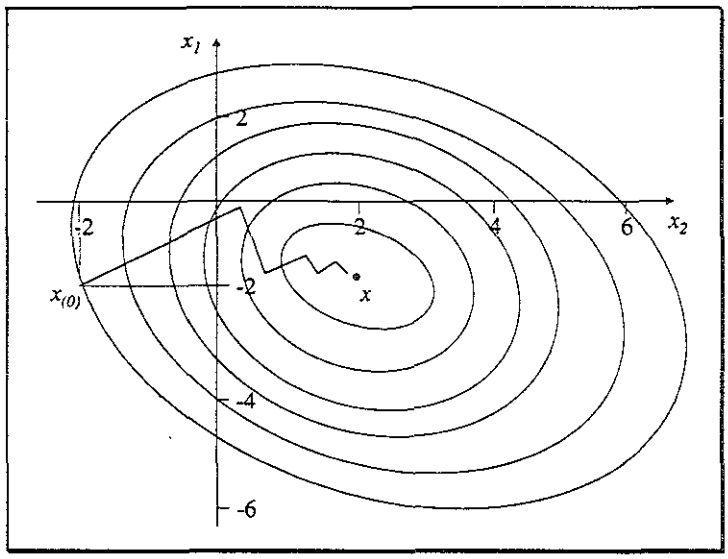


Fig.(2.8.3) El método de steepest descent iniciando en (-2,-2) y convergiendo a (2,-2)

Aunque la ecuación (2.8.3) es necesaria para calcular $r_{(0)}$, la ecuación (2.8.6) puede ser utilizada en cada iteración subsecuente. El producto $Ar_{(i)}$, que ocurre en las ecuaciones (2.8.4) y (2.8.6), necesita ser calculado solamente una vez. La desventaja de utilizar esta expresión recurrente es que la sucesión generada por la ecuación (2.8.6) es generada sin ninguna retroalimentación del valor $x_{(i)}$, de tal forma que una acumulación de errores de punto flotante podría causar que $x_{(i)}$ convergiera solamente a un punto cercano al x deseado.

Este efecto puede ser evitado si periódicamente se utiliza la ecuación (2.8.3) para recalcular el residual exacto.

2.8.2 Método de los gradientes conjugados

2.8.2.1 Direcciones Conjugadas.

El método steepest descent frecuentemente repite las mismas direcciones de búsqueda a lo largo del proceso de optimización, fig.(2.8.3). Evidentemente sería mejor que cada vez que se tomara alguna dirección, el tamaño de α fuera el *correcto* desde la primera vez, de tal forma que cualquier dirección previamente utilizada jamás volviera a utilizarse. Considérese la siguiente idea: escójanse un conjunto de *direcciones de búsqueda ortogonales* $d_{(0)}, d_{(1)}, d_{(2)}, \ldots d_{(n-1)}$. En cada dirección de búsqueda, se tomará exactamente un paso, y este paso será precisamente de la longitud adecuada para alinearse con la solución óptima x. Después de n pasos se habrá llegado al mínimo.

La siguiente figura ilustra la idea anterior utilizando las direcciones de los ejes cartesianos como las direcciones de búsqueda. El primer paso (horizontal) nos conduce a la coordenada adecuada x_1; El segundo paso (vertical) dará en la solución óptima. Nótese que $e_{(1)}$ es ortogonal a $d_{(0)}$. En general, para cada nueva iteración se selecciona un punto como el siguiente

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)} \tag{2.8.7}$$

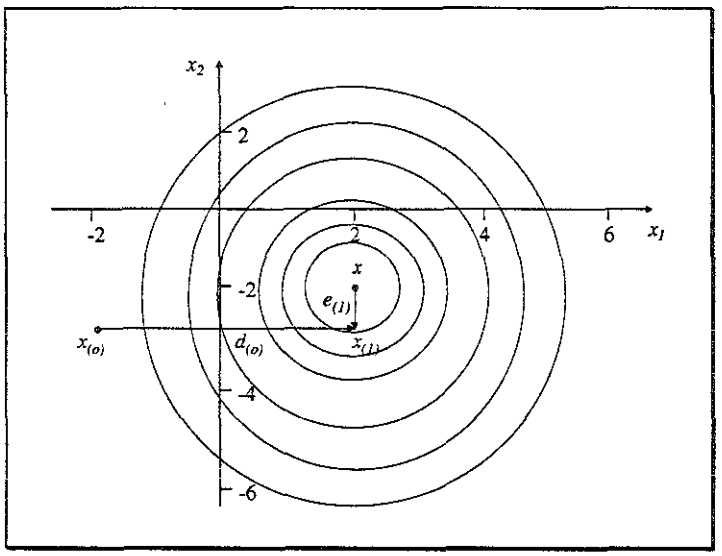


Fig.(2.8.4) Método de las direcciones ortogonales. Este método trabaja solamente si la respuesta correcta se conoce de antemano

Para encontrar el valor de $\alpha_{(i)}$ se debe de considerar el hecho de que $e_{(i+1)}$ debe de ser ortogonal a $d_{(i)}$, para que de esta forma no se vuelva a utilizar la dirección $d_{(i)}$ nuevamente. Usando esta condición se tiene que

$$\begin{aligned} d_{(i)}^T e_{(i+1)} &= 0 \\ d_{(i)}^T (e_{(i)} + \alpha_{(i)} d_{(i)}) &= 0 \\ \alpha_{(i)} &= -\frac{d_{(i)}^T e_{(i)}}{d_{(i)}^T d_{(i)}} \end{aligned} \tag{2.8.8}$$

Desafortunadamente, con todas las ideas expuestas hasta ahora no se ha resuelto nada, puesto que, de acuerdo a la ecuación anterior, no es posible calcular el valor de $\alpha_{(i)}$ sin el conocimiento previo de $e_{(i)}$, pero si se conociera $e_{(i)}$ el problema estaría ya resuelto!.

La solución consiste en hacer los vectores de búsqueda *A-ortogonales* en vez de solamente ortogonales. Dos vectores son *A-ortogonales,* o *conjugados con respecto a la matriz A*, si

$$d_{(i)}Ad_{(j)} = 0$$

Por lo tanto, el nuevo requerimiento es que $e_{(i+1)}$ sea *A-ortogonal* a $d_{(i)}$. Esta condición de ortogonalidad se traduce nuevamente en encontrar el punto mínimo α en la dirección del vector de búsqueda $d_{(i)}$, al igual que se hizo en el caso del steepest descent. Para entender lo anterior, y utilizando (2.8.7), obsérvese lo siguiente

$$\begin{aligned} \frac{d}{d\alpha} f(x_{(i+1)}) &= 0 \\ f'(x_{(i+1)})^T \frac{d}{d\alpha} x_{(i+1)} &= 0 \\ -r_{(i+1)}^T d_{(i)} &= 0 \\ d_{(i)}^T A e_{(i+1)} &= 0 \end{aligned}$$

Por lo que el valor para $\alpha_{(i)}$ cuando las direcciones de búsqueda son *A*-ortogonales, está dado por

$$\begin{aligned} \alpha_{(i)} &= -\frac{d_{(i)}^T A e_{(i)}}{d_{(i)}^T A d_{(i)}} \\ \alpha_{(i)} &= \frac{d_{(i)}^T r_{(i)}}{d_{(i)}^T A d_{(i)}} \end{aligned} \qquad (2.8.9)$$

Lo cual, contrariamente con (2.8.8), sí es posible calcular. Obsérvese como si los vectores de búsqueda fueran los residuales, la expresión anterior sería la misma que se utilizó en el método del steepest descent.

Para comprobar que con este procedimiento realmente se calcula el valor de x en n iteraciones, es posible expresar el término de error como una combinación lineal de las direcciones de búsqueda, con lo cual se obtendrá

$$e_{(0)} = \sum_{j=0}^{n-1} \delta_j d_{(j)} \qquad (2.8.10)$$

Los valores de δ_j pueden ser calculados observando lo siguiente. Puesto que las direcciones de búsqueda son *A-ortogonales*, es posible en la ecuación anterior eliminar todos los valores de δ_j excepto uno, de la siguiente forma:

$$
\begin{aligned}
& d_{(k)}^T A e_{(0)} = \sum_j \delta_{(j)} d_{(k)} A d_{(j)} \\
& d_{(k)}^T A e_{(0)} = \delta_{(k)} d_{(k)} A d_{(k)} && \text{(por A - ortogonalidad de los vectores d)} \\
& \delta_{(k)} = \frac{d_{(k)}^T A e_{(0)}}{d_{(k)} A d_{(k)}} && (2.8.11) \\
& \quad = \frac{d_{(k)}^T A (e_{(0)} + \sum_{i=0}^{k-1} \alpha_{(i)} d_{(i)}}{d_{(k)}^T A d_{(k)}} && \text{(por A - ortogonalidad de los vectores d)} \\
& \quad = \frac{d_{(k)}^T A e_{(k)}}{d_{(k)}^T A d_{(k)}} && \text{(por ecuación (2.8.7))}
\end{aligned}
$$

Usando (2.8.9) y (2.8.11), es posible observar que $\alpha_i = -\delta_{(i)}$. Lo anterior nos proporciona un nuevo enfoque para describir el término de error

$$
\begin{aligned}
e_{(i)} &= e_{(0)} + \sum_{j=0}^{i-1} \alpha_{(j)} d_{(j)} \\
&= \sum_{j=0}^{n-1} \delta_{(j)} d_{(j)} - \sum_{j=0}^{i-1} \delta_{(j)} d_{(j)} \qquad (2.8.12) \\
&= \sum_{j=i}^{n-1} \delta_{(j)} d_{(j)}
\end{aligned}
$$

En forma nada sorprendente la ecuación anterior, muestra como el proceso de ir *construyendo* el vector x componente a componente, puede también ser entendido como un proceso de ir *disminuyendo* el término de error componente a componente. Despues de n iteraciones, cada componente habrá sido disminuida y se tendrá que $e_{(n)}=0$.

Todo lo que falta por hacer ahora es encontrar un conjunto de direcciones de búsqueda $\{d_{(i)}\}$ que sean A-ortogonales, para ello se seguirá el proceso de *Gram-Schmidt*. Es importante mencionar que aunque se está trabajando con la *versión lineal de los gradientes conjugados*, este proceso de ortogonalización aplicará también para el caso en el que la función de error no sea cuadrática y no exista por lo tanto una matriz A que sea simétrica y definida positiva (ecuación (2.7.3)).

Ahora bien, como se mencionó anteriormente, para la gran mayoría de los problemas reales que estaremos interesados en resolver, la función de error, o error de predicción, ecuación (2.9.7), nunca será cuadrática, y por lo tanto ni siquiera llegará a existir A, y por ello se deberá de utilizar la *versión no lineal de los gradientes conjugados,* con una forma alternativa para la minimización, local o global, de la función de error.

2.8.2.2 Conjugación Gram-Schmidt

Supóngase que se tiene un conjunto de n vectores linealmente independientes $u_0,u_1,u_2, ..., u_{n-1}$. Para construir $d_{(i)}$ tómese u_i y réstensele todas las componentes que no sean A-ortogonales a los d vectores previos, fig.(2.8.5). En otras palabras, sea $d_{(0)}=u_0$, y para $i>0$ calcúlese

$$d_{(i)} = u_i + \sum_{k=0}^{i-1} \beta_{ik} d_{(k)} \tag{2.8.13}$$

donde el valor de las β_{ik}, para $i>k$ estará dado por

$$\begin{aligned} d_{(i)}^T A d_{(j)} &= u_i^T A d_j + \sum_{k=0}^{i-1} \beta_{ik} d_{(k)}^T A d_{(j)} \\ 0 &= u_i^T A d_{(j)} + \beta_{ij} d_{(j)}^T A d_{(j)}, \quad i > j \\ \beta_{ij} &= -\frac{u_i^T A d_{(j)}}{d_{(j)}^T A d_{(j)}} \end{aligned} \tag{2.8.14}$$

La dificultad que surge con el uso de la conjugación Gram-Schmidt en el método de las direcciones conjugadas es que todos los vectores de búsqueda viejos deben de ser mantenidos en memoria para construir cada uno de los nuevos, y eso se traduce en una complejidad de orden n^3 para el algoritmo. Como consecuencia el método de las direcciones conjugadas se utilizó muy poco hasta el descubrimiento del método de los gradientes conjugados, el cual es *un* método de direcciones conjugadas, pero sin las desventajas anteriores.

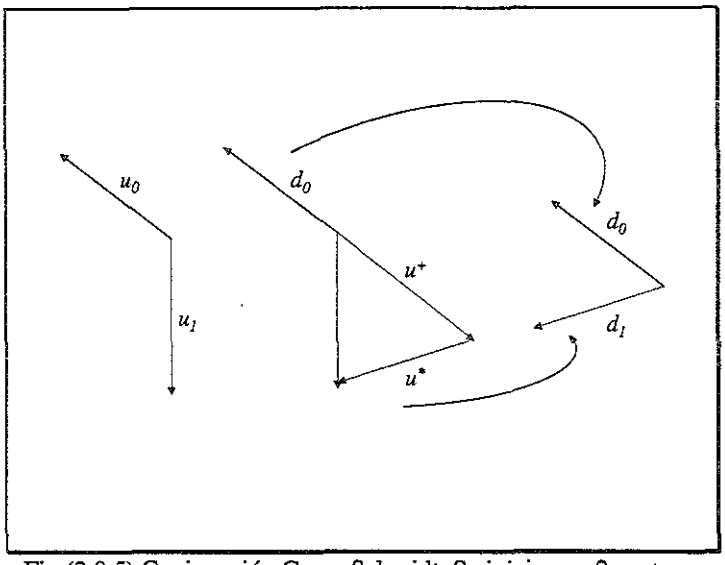


Fig.(2.8.5) Conjugación Gram-Schmidt. Se inicia con 2 vectores linealmente ind. u_0 y u_1. $d_{(0)}=u_0$. El vector u_1 está formado por 2 componentes: u^* que es A-ortogonal a $d_{(0)}$ y u^+ que le es paralelo. Después de la conjugación sólo la parte A-ortogonal permanece, y $d_{(1)}=u^*$.

2.8.2.3 Propiedades de los residuales.

A continuación se mencionarán algunas propiedades necesarias para el algoritmo de los gradientes conjugados:

- Al igual que con el método del Steepest Descent, el número de productos Matriz-vector por iteración puede ser reducido a uno utilizando un proceso recurrente para calcular el valor del residual

$$\begin{aligned} r_{(i+1)} &= -Ae_{(i+1)} \\ &= -A(e_{(i)} + \acute{a}_{(i)}d_{(i)}) \\ &= r_{(i)} - \acute{a}_{(i)}Ad_{(i)} \end{aligned} \qquad (2.8.15)$$

- La relación de recurrencia anterior *sugiere* que, al mismo tiempo que el término de error está siendo disminuido componente a componente (en aquellas componentes que son paralelas a las direcciones de búsqueda $d_{(i)}$), el residual está siendo también disminuido componente a componente (en aquellas componentes que son paralelas a un

conjunto diferente al de las direcciones de búsqueda $Ad_{(i)}$). Esta *sugerencia* queda confirmada si la ecuación (2.8.12) es multiplicada por la izquierda por $-A$

$$r_{(j)} = -\sum_{k=j}^{n-1} \delta_{(k)} A d_{(k)}$$

- El producto interno de $d_{(i)}$ con la expresión anterior es

$$d_{(i)}^T r_{(j)} = 0, \quad i < j \qquad \textit{(por A - ortogonalidad de los d - vectores)} \qquad (2.8.16)$$

Alternativamente, la relación anterior pudo haber sido obtenida observando lo siguiente: recuérdese que una vez que se ha dado un paso en alguna dirección de búsqueda, nunca se volverá a utilizar esa dirección otra vez; el término de error será por siempre *A-ortogonal* a todas las direcciones de búsqueda utilizadas, y puesto que $r_{(i)} = -Ae_{(i)}$, el residual será por siempre ortogonal a todas las direcciones de búsqueda anteriores

- Tomando el producto interno de la ecuación (2.8.13) y $r_{(j)}$ se obtendrá

$$d_{(i)}^T r_{(j)} = u_i^T r_{(j)} + \sum_{k=0}^{i-1} \beta_{ik} d_{(k)}^T r_{(j)} \qquad (2.8.17)$$

$$0 = u_i^T r_{(j)}, \qquad i < j \quad \textit{(por (2.8.14))} \qquad (2.8.18)$$

Cada residual es ortogonal a todos los vectores u previos. Esto no debe de ser sorprendente puesto que los vectores de búsqueda son construidos precisamente a partir de los vectores u; por lo tanto, los vectores de búsqueda $d_{(0)},\ldots,d_{(i)}$ generan el mismo subespacio que $u_0,\ldots,u_i$, y $r_{(j)}$ (para toda $j>i$) es ortogonal a este subespacio.

- Finalmente, otra identidad que será utilizada posteriormente y que se deriva de (2.8.17) está dada por

$$d_{(i)}^T r_{(i)} = u_i^T r_{(i)} \qquad (2.8.19)$$

2.8.2.4 Método de los gradientes conjugados (versión lineal).

El método de los gradientes conjugados es simplemente un método de direcciones conjugadas, pero en este caso las direcciones de búsqueda son construidas por conjugación de los residuales, es decir haciendo $u_i = r_{(i)}$.

Partiendo del hecho de que los residuales tuvieron mucho que ver en la determinación del α óptimo en el método steepest descent, parecía bastante natural el intentar utilizarlos nuevamente en la construcción de las direcciones de búsqueda, pero esta vez, para el algoritmo de los gradientes conjugados.

De esta forma, la ecuación (2.8.18) se transforma en

$$r_{(i)}^T r_{(j)} = 0, \quad i \neq j \tag{2.8.20}$$

Puesto que cada residual es ortogonal al subespacio generado por los vectores de búsqueda previos, todos los residuales deben de ser mutuamente ortogonales. Esta relación permite poder simplificar los terminos β_{ij}, para ello obsérvese como el numerador en la ultima parte de la ecuación (2.8.14) es posible expresarlo de la siguiente forma:

$$r_{(i)}^T r_{(j+1)} = r_{(i)} r_{(j)} - \alpha_{(j)} r_{(i)}^T A d_{(j)}, \qquad \text{por (2.8.18)}$$

$$\alpha_{(j)} r_{(i)}^T A d_{(j)} = r_{(i)} r_{(j)} - r_{(i)}^T r_{(j+1)}$$

$$r_{(i)}^T A d_{(j)} = \begin{cases} \dfrac{1}{\alpha(i)} r_{(i)}^T r_{(i)}, & \text{i = j,} \\ -\dfrac{1}{\alpha(i-1)} r_{(i)}^T r_{(i)}, & \text{i = j+1, por (2.8.23)} \\ 0 & \text{en otro caso.} \end{cases}$$

$$\therefore \beta_{ij} = \begin{cases} \dfrac{1}{\alpha(i-1)} \dfrac{r_{(i)}^T r_{(i)}}{d_{(i-1)}^T A d_{(i-1)}} & \text{i = j+1, por (2.8.17)} \\ 0 & \text{i > j+1} \end{cases} \tag{2.8.21}$$

Obsérvese como la mayoría de los términos β_{ij} han desaparecido!, es decir, no es más necesario guardar los vectores de búsqueda anteriores (viejos) para garantizar la *A-ortogonalidad* de los nuevos vectores de búsqueda. Esta agradable propiedad es la que hace del algoritmo de gradientes conjugados un algoritmo importante en si, pues se reduce tanto la complejidad espacial como la complejidad de tiempo por iteración pasando de $O(n^2)$ a $O(m)$, donde m representa el número de componentes diferentes de cero de A.

Si adicionalmente, sólo por notación, se expresa $\beta_{(i)} = \beta_{i,i-1}$, la ecuación anterior se reduce a:

$$\beta_{(i)} = \frac{r_{(i)}^T r_{(i)}}{d_{(i-1)}^T r_{(i-1)}} \qquad \text{por (2.8.9)}$$

$$= \frac{r_{(i)}^T r_{(i)}}{r_{(i-1)}^T r_{(i-1)}} \qquad \text{por (2.8.19)}$$

Por lo que finalmente, el método de los gradientes conjugados (versión lineal) puede expresarse como:

$$\begin{aligned} d_{(0)} &= r_{(0)} = b - Ax_{(0)} \\ \alpha_{(i)} &= \frac{r_{(i)}^T r_{(i)}}{d_{(i)}^T A d_{(i)}} \\ x_{(i+1)} &= x_{(i)} + \alpha_{(i)} d_{(i)} \\ r_{(i+1)} &= r_{(i)} - \alpha_{(i)} A d_{(i)} \\ \beta_{(i+1)} &= \frac{r_{(i+1)}^T r_{(i+1)}}{r_{(i)}^T r_{(i)}} \\ d_{(i+1)} &= r_{(i+1)} + \beta_{(i+1)} d_{(i)} \end{aligned} \qquad (2.8.22)$$

2.8.2.5 Método de los gradientes conjugados (versión no lineal).

Para la gran mayoría de los problemas reales que estaremos interesados en resolver, la función de error, o error de predicción, ecuación (2.9.7), nunca será cuadrática, y por lo tanto ni siquiera llegará a existir una matriz A simétrica o definida positiva con la cual trabajar, y por ello, se deberá de utilizar la *versión no lineal de los gradientes conjugados,* con una forma alternativa para la definición de las direcciones ortogonales, esto quiere

decir que el algoritmo de los gradientes conjugados puede ser utilizado no solamente para encontrar el punto mínimo de una forma cuadrática, sino también para minimizar cualquier función continua $f(x)$ a la cual sea posible calcularle el gradiente.

De entre las posibles aplicaciones para este algoritmo pueden mencionarse una gran variedad de problemas de optimización, tales como control óptimo, regresiones no lineales, y por supuesto, entrenamiento de redes neuronales.

Para derivar el algoritmo no lineal de los gradientes conjugados, básicamente se deben de hacer 3 cambios a la versión lineal que se estudió anteriormente: la fórmula recursiva para el residual no puede más ser utilizada, se vuelve más complicado calcular el tamaño del paso α, y existen diferentes alternativas para la elección de β.

En esta versión no lineal, el residual es siempre igualado con el negativo del gradiente, $r_{(i)} = -f'(x_{(i)})$. Las direcciones de búsqueda son calculadas aplicando el proceso de ortogonalización de Gram-Schmidt sobre los residuales, al igual que se hizo en la versión lineal. El proceso de las búsquedas lineales, a lo largo de las direcciones ortogonales, para la determinación del $\alpha_{(i)}$ óptimo, resulta ser mucho más complicado que en la versión lineal, y una gran variedad de procedimientos pueden ser utilizados. En la sección 2.11, donde se enunciará la variante de esta versión no lineal de los gradientes conjugados para el entrenamiento de las redes neuronales, se utilizará, para la determinación de la tasa de aprendizaje $\alpha_{(i)}$, una combinación de la *regla de la sección dorada* con una interpolación cúbica. Al igual que en el caso lineal, el valor de la tasa de aprendizaje que minimiza $f(x_{(i)} + \alpha_{(i)} d_{(i)})$ es encontrado cuando se garantiza que el gradiente es ortogonal a la dirección de búsqueda, y para ello es posible utilizar cualquier algoritmo con el que sea posible encontrar los ceros de la expresión $[f'(x_{(i)} + \alpha_{(i)} d_{(i)})]^T d_{(i)}$

Para la versión lineal de los gradientes conjugados existen diferentes expresiones equivalentes para el valor de β. Para la versión no lineal, esas expresiones dejan de ser equivalentes. Aún en estos días se sigue investigando cuál es la mejor opción para su valor. Dos de las alternativas mayormente utilizadas están dadas por la fórmula Fletcher-Reeves, que fue la que se utilizó en la versión lineal, y la fórmula Polak-Ribière

$$\beta^{FR}_{(i+1)} = \frac{r^T_{(i+1)} r_{(i+1)}}{r^T_{(i)} r_{(i)}}, \qquad \beta^{PR}_{(i+1)} = \frac{r^T_{(i+1)} (r_{(i+1)} - r_{(i)})}{r^T_{(i)} r_{(i)}}.$$

El método Fletcher-Reeves converge si el punto de inició está suficientemente cerca del mínimo deseado, mientras que el método Polak-Ribière, para algunos *casos raros*, itera en forma infinita sin converger jamás. Sin embargo, frecuentemente la fórmula Polak-Ribière converge mucho más rápido.

Afortunadamente, la convergencia del método Polak-Ribière puede ser garantizada escogiendo $\beta = max\{\beta^{PR}, 0\}$.

Por lo que finalmente, el método de los gradientes conjugados (versión no lineal) puede expresarse como:

$$d_{(0)} = r_{(0)} = -f'(x_{(0)}),$$

Encontrar el $\alpha_{(i)}$ que minimiza $f(x_{(i)} + \alpha_{(i)} d_{(i)})$,

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)}$$
$$r_{(i+1)} = -f'(x_{(i+1)}),$$
$$\beta_{(i+1)} = \frac{r_{(i+1)}^T r_{(i+1)}}{r_{(i)}^T r_{(i)}} \quad \text{ó} \quad \beta_{(i+1)} = max\left\{\frac{r_{(i+1)}^T (r_{(i+1)} - r_{(i)})}{r_{(i)}^T r_{(i)}}, 0\right\},$$
$$d_{(i+1)} = r_{(i+1)} + \beta_{(i+1)} d_{(i)}.$$

2.9 Algoritmo de Retropropagación estándar. (Backpropagation con steepest descent)

Las redes neuronales de alimentación hacia adelante con capas intermedias u ocultas no tienen las limitaciones que los perceptrones simples en cuanto a la incapacidad de poder realizar ciertos cálculos (la representación de la función *XOR* por ejemplo). Aunque el gran poder de las redes multicapas fue descubierto hace mucho tiempo, fue hasta fechas más recientes cuando se descubrió como conseguir que ellas aprendieran una tarea particular, utilizando, por ejemplo, el algoritmo de retropropagación del error u otros métodos. La ausencia por mucho tiempo de una regla de aprendizaje, aunada a la demostración de Minsky y Papert [Minsky,1969] de que solamente las funciones linealmente separables podían ser representadas por los perceptrones simples, condujo a la pérdida de interés en las redes multicapas hasta fechas más recientes.

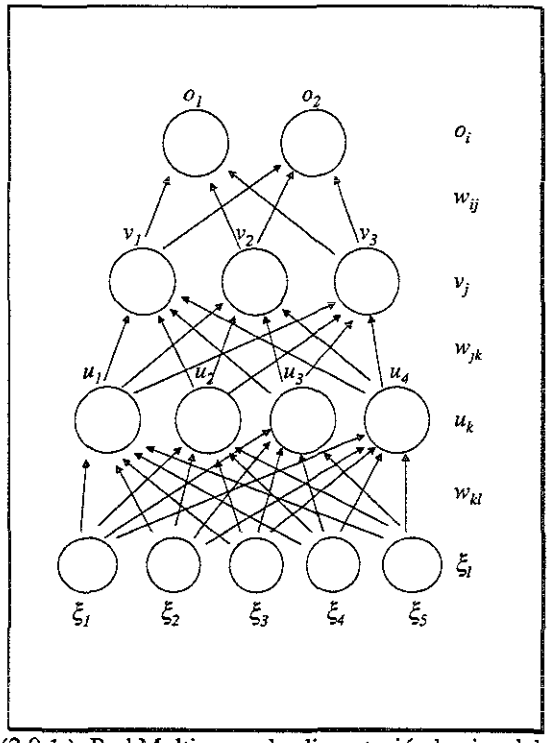
El algoritmo de retropropagación es central para mucho del trabajo actual en el entrenamiento de las redes neuronales. Este algoritmo fue inventado varias veces en forma independiente, por Werbos [Werbos,1974], Parker [Parker,1985] y Rumelhart et.al. [Rumelhart,1986]. El algoritmo nos proporciona un método para actualizar los pesos w_{pq} en cualquier red de alimentación hacia adelante para aprender un conjunto de entrenamiento de pares de entradas-salidas $\{\xi_k^\mu, \zeta_i^\mu\}$.

Consideremos la siguiente red de alimentación hacia adelante (fig (2.9.1)). Las unidades de salida son denotadas como o_i, las unidades intermedias de la segunda capa por v_j, las

unidades intermedias de la primera capa por u_k y las unidades de entrada por ξ_l. Existen conexiones w_{kl} de las unidades de entrada a las unidades intermedias en la primera capa, w_{jk} de las unidades intermedias en la primera capa a las unidades intermedias en la segunda capa, y w_{ij} de las unidades intermedias de la segunda capa a las unidades de salida. Notemos que el índice i se refiere a una unidad de salida, j a una unidad oculta en la segunda capa, k a una unidad oculta en la primera capa y l a una unidad de entrada.

Las entradas corresponden siempre a valores particulares de alguna aplicación (en este trabajo corresponderán a datos meteorológicos de lluvia). Los diferentes patrones de entrada serán denotados con el superíndice μ, de tal forma que la entrada l sera denotada como ξ_l^μ, cuando el patrón μ esté siendo presentado. Los ξ_l^μ's pueden ser binarios (0,1,o ± 1) o continuos.

Utilizaremos N para denotar el número de unidades de entrada y p para el número de patrones de entrada ($\mu = 1,2,...,p$).



Fig(2.9.1). Red Multicapas de alimentación hacia adelante

De esta forma, dado el patrón μ, la unidad oculta k en la primera capa intermedia recibe la siguiente entrada neta

$$\sum_l w_{kl}\xi_l \qquad (2.9.1)$$

y produce como salida

$$g(\sum_l w_{kl}\xi_l) = u_k \tag{2.9.2}$$

Entonces, la unidad v_j de la segunda capa intermedia recibe como entrada neta

$$\sum_k w_{jk}u_k = \sum_k w_{jk}\, g(\sum_l w_{kl}\xi_l) \tag{2.9.3}$$

y produce como salida

$$g(\sum_k w_{jk}\, g(\sum_l w_{kl}\xi_l)) = v_j \tag{2.9.4}$$

con lo que la unidad de salida o_i recibe

$$\sum_j w_{ij}v_j = \sum_j w_{ij}\, g(\sum_k w_{jk}\, g(\sum_l w_{kl}\xi_l)) \tag{2.9.5}$$

y produce como su salida final

$$o_i = g(\sum_j w_{ij}\, g(\sum_k w_{jk}\, g(\sum_l w_{kl}\xi_l))) \tag{2.9.6}$$

En general, los umbrales θ_i deben también de ser tomados en cuenta, pero dado que ellos pueden ser tratados como pesos con unidades de entrada constantes (añadiendo una unidad de entrada adicional con valor fijo de –1 y conectada a todas las unidades de la red), podemos restringir el análisis solamente al ajuste de los pesos y asumir que el error de predicción es una función de los pesos solamente.

De esta forma, nuestra medición del error obtenido o función objetivo estará dada por

$$E[w] = \frac{1}{2}\sum_{\mu i}(\zeta_i^{\mu} - o_i^{\mu})^2 \tag{2.9.7}$$

o bien

$$E[w] = \frac{1}{2}\sum_{\mu i}(\zeta_i^{\mu} - g(\sum_j w_{ij}\, g(\sum_k w_{jk}\, g(\sum_l w_{kl}\xi_l^{\mu}))))^2 \tag{2.9.8}$$

Esta es una función continua y diferenciable de los pesos, por lo que podemos utilizar la técnica del gradiente descendente para encontrar los pesos adecuados.

Por lo que, denotando por Δw_{ij} a $-\eta\frac{\partial E}{\partial w_{ij}}$, donde η es un escalar conocido como tasa de aprendizaje, obtendremos que para las conexiones que van de la segunda capa intermedia a la capa de salida, la técnica del gradiente descendente proporciona

$$\Delta w_{ij} = -\eta\frac{\partial E}{\partial w_{ij}} = \eta\sum_{\mu}(\zeta_i^{\mu} - o_i^{\mu})g'(\sum_j w_{ij}v_j^{\mu})v_j^{\mu} \tag{2.9.9}$$

$$= \eta\sum_{\mu}\delta_i^{\mu}v_j^{\mu} \tag{2.9.10}$$

donde

$$\delta_i^{\mu} = g'(\sum_j w_{ij}v_j^{\mu})(\zeta_i^{\mu} - o_i^{\mu}) \tag{2.9.11}$$

Ahora bien, para las conexiones que van de la primera capa intermedia hacia la segunda capa intermedia obtendremos

$$\Delta w_{jk} = -\eta\frac{\partial E}{\partial w_{jk}} = \eta\sum_{\mu i}(\zeta_i^{\mu} - o_i^{\mu})g'(\sum_j w_{ij}v_j^{\mu})w_{ij}g'(\sum_k w_{jk}u_k^{\mu})u_k^{\mu} \tag{2.9.12}$$

$$= \eta\sum_{\mu i}\delta_i^{\mu}w_{ij}g'(\sum_k w_{jk}u_k^{\mu})u_k^{\mu} \tag{2.9.13}$$

$$= \eta\sum_{\mu}\delta_j^{\mu}u_k^{\mu} \tag{2.9.14}$$

donde

$$\delta_j^{\mu} = g'(\sum_k w_{jk}u_k^{\mu})\sum_i w_{ij}\delta_i^{\mu} \tag{2.9.15}$$

Finalmente, para las conexiones que van de la capa de entrada hacia la primera capa intermedia tendremos

$$\Delta w_{kl} = -\eta\frac{\partial E}{\partial w_{kl}} = \eta\sum_{\mu}(\zeta_i^{\mu} - o_i^{\mu})g'(\sum_j w_{ij}v_j^{\mu})\sum_j w_{ij}g'(\sum_k w_{jk}u_k^{\mu})w_{jk}g'(\sum_l w_{kl}\xi_l^{\mu})\xi_l^{\mu} \tag{2.9.16}$$

$$= \eta\sum_{\mu}\sum_j g'(\sum_k w_{jk}u_k^{\mu})\sum_i(w_{ij}\delta_i^{\mu})w_{jk}g'(\sum_l w_{kl}\xi_l^{\mu})\xi_l^{\mu} \tag{2.9.17}$$

$$= \eta \sum_{\mu} \sum_{j} (\delta_j^{\mu} w_{jk}) g'(\sum_{l} w_{kl} \xi_l^{\mu}) \xi_l^{\mu} \tag{2.9.18}$$

$$= \eta \sum_{\mu} \delta_k^{\mu} \xi_l^{\mu} \tag{2.9.19}$$

donde

$$\delta_k^{\mu} = g'(\sum_{l} w_{kl} \xi_l) \sum_{j} w_{jk} \delta_j^{\mu} \tag{2.9.20}$$

Notemos que (2.9.10), (2.9.14), (2.9.19) tienen la misma forma pero con una definición diferente para las $\delta's$. En general, con un número arbitrario de capas, el algoritmo de retropropagación siempre tiene la forma

$$\Delta w_{pq} = \eta \sum_{patrones} \delta_{salida} v_{entrada} \tag{2.9.21}$$

donde salida y entrada se refieren a los dos extremos p y q de la conexión involucrada y v representa el nivel de activación apropiado que se forma con la información proveniente de alguna capa intermedia o directamente del conjunto de entradas.

El significado de δ depende de la capa involucrada; para la última capa de conexiones, ésta está dada por (2.9.11), mientras que para todas las otras capas esta representada por una ecuación como (2.9.15) o (2.9.20).

La ecuación (2.9.15) nos permite determinar el valor de δ para una unidad escondida v_j en términos de las δ 's correspondientes a las unidades o_i que la mencionada unidad escondida v_j alimenta. Los coeficientes son precisamente los pesos usuales que *van hacia adelante* $w_{ij}'s$, pero en este caso ellos están *propagando errores (δ 's) hacia atrás* en lugar de *señales hacia adelante*; de aquí el nombre del algoritmo: *retropropagación del error* o simplemente retropropagación. Por lo tanto es posible utilizar la misma red, o mejor dicho una versión bidireccional de ella, para calcular tanto los valores de las salidas o_i como las δ 's. La siguiente figura (fig. (2.9.2)) ilustra esta idea en una red de 3 capas.

Aunque las reglas de actualización (2.9.10),(2.9.14) y (2.9.19) quedaron expresadas como sumas sobre todos los patrones μ, para una gran variedad de problemas ellas son utilizadas en forma incremental (*on line*): un patrón μ es presentado como entrada y entonces todos los pesos son actualizados antes de que el siguiente patrón sea considerado. Claramente lo anterior hace que la función objetivo (2.9.7) sea más sencilla en cada iteración y permite que mediante iteraciones sucesivas se obtenga una adaptación al gradiente local. Si los patrones son elegidos en forma aleatoria, esto hace que la búsqueda del mínimo en el espacio de pesos sea también estocástica, permitiendo una exploración más amplia de la superficie objetivo o de costo. La alternativa en modo *batch*, es decir tomando (2.9.10),(2.9.14) y (2.9.19) literalmente, y sólo actualizar después de que todos los patrones han sido presentados, requiere de espacio de almacenamiento adicional para cada conexión.

La efectividad relativa de las dos alternativas está en función del problema a resolver, pero aparentemente la alternativa *on-line* es superior en varios casos, especialmente si los conjuntos de entrenamiento son muy regulares o redundantes [Hertz;1991]. Sin embargo, es posible encontrar también algunas otras situaciones en las que claramente la versión batch se comporta más eficientemente, [Reifman y Vitela; 1993].

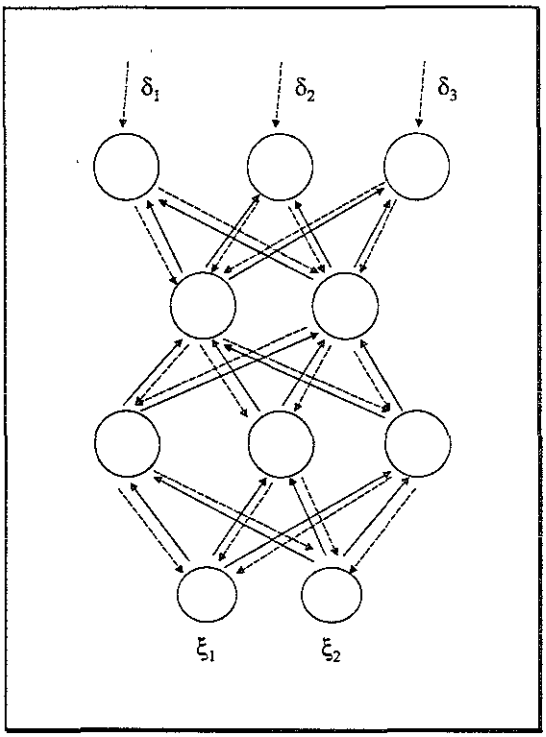


Fig.(2.9.2). Retropropagación en una red de 3 capas. Las líneas sólidas muestran la propagación hacia adelante de las señales y las líneas punteadas muestran la retropropagación de los errores.

Es muy común utilizar como funciones de activación o transferencia $g(\cdot)$ una función sigmoidal. Evidentemente la función debe de ser diferenciable, y normalmente lo que se desea es que la función se sature en ambos extremos. De esta forma es posible utilizar alguna de las siguientes funciones

$$g_1(h) = \frac{1}{1+e^{-2\beta h}} \qquad \text{con rango (0,1)}$$

$$g_2(h) = \tanh(\beta h) \qquad \text{con rango (-1,1)}$$

Normalmente el parámetro β es elegido como 1 o ½ para la primera función. Las derivadas de esas funciones son muy atractivas puesto que quedan expresadas en términos de ellas mismas

$$g_1'(h) = 2\beta g_1(1-g_1)$$

$$g_2'(h) = \beta(1-g_2^2)$$

A partir de lo anterior es posible poder expresar (2.9.11) como $\delta_i^\mu = g(\sum_j w_{ij} v_j^\mu)(1-g(\sum_j w_{ij} v_j^\mu))(\zeta_i^\mu - o_i^\mu)$ para $\beta = 1/2$.

El hecho de que las derivadas de la función objetivo (2.9.7) puedan ser calculadas por retropropagación de los errores es sumamente atractivo, pero adicionalmente origina dos consecuencias muy importantes:

- La regla de actualización que se obtiene, es decir ecuación (2.9.21), es *local,* esto quiere decir que para calcular el cambio en el peso para alguna conexión, solamente necesitamos un par de cantidades disponibles (después de la retropropagación de las δ 's) una de cada lado de la conexión involucrada. Esto hace que el algoritmo de retropropagación sea muy adecuado para implementarlo en forma paralela (correrlo en computadoras que tengan múltiples procesadores).

- La complejidad computacional del algoritmo es menor de lo que se hubiera esperado. Si tenemos n conexiones en total, el cálculo de la función objetivo (2.9.7) sería de complejidad n, y por tanto el cálculo en forma directa de las n derivadas necesarias sería del orden de n^2. En contraste el algoritmo de retropropagación nos permite conocer todas las derivadas en sólo n operaciones.

Se mencionó anteriormente que muchas veces en la práctica, para simplificar el problema de entrenar la red, se trabaja con la opción *on-line* o incremental en la que un patrón μ es presentado como entrada y entonces todos los pesos son actualizados antes de que el siguiente patrón sea considerado. A continuación se presenta esta versión del algoritmo en forma completa y lista para ser implementada en una red con M capas, $m=1,2,\ldots,M$, en las que v_i^m representa la salida de la i-ésima unidad en la capa m. Para ello se definirá $v_i^0 = \xi_i$, es decir, la i-ésima entrada y w_{ij}^m como la conexión entre v_j^{m-1} y v_i^m.

1. Inicializar todos los pesos de la red con valores pequeños (por ejemplo [-1,1])
2. Escoger un patron de entrada ξ_k^μ y aplicárselo a la capa de entrada, es decir, $\xi_k^\mu = v_k^0 \quad \forall\ k$.

3. Propagar la señal hacia adelante en la red neuronal construida, haciendo $v_i^m = g(h_i^m) = g(\sum_j w_{ij}^m v_j^{m-1})$ $\forall i$ y $\forall m$, hasta que todas las v_i^m hayan sido calculadas.
4. Calcular las $\delta's$ para la capa de salida $\delta_i^M = g'(h_i^M)(\zeta_i^\mu - v_i^M)$ donde ζ_i^μ son los valores deseados y v_i^m son las salidas obtenidas para el patrón μ considerado.
5. Calcular las $\delta's$ para el resto de las capas, propagando los errores hacia atrás

$$\delta_i^{m-1} = g'(h_i^{m-1})\sum_j w_{ji}^m \delta_j^m : m = M, M-1,...,3,2.$$

hasta que las $\delta's$ hayan sido calculadas para todas las unidades.
6. Definir $\Delta w_{ij}^m = \eta \delta_i^m v_j^{m-1}$ para actualizar todas las conexiones de acuerdo con

$$w_{ij}^{nuevo} = w_{ij}^{viejo} + \Delta w_{ij}$$

7. Regresar a 2 y repetir para el siguiente patrón.

La versión *on-line* del algoritmo de retropropagación no sólo se traduce en una función objetivo más sencilla a minimizar, sino que además facilita la comprensión del proceso de optimización, sin embargo, a pesar de esas ventajas, desafortunadamente esta *versión sencilla* del algoritmo muy pocas veces es la mejor, pues como se verá en las secciones siguientes, el algoritmo estándar de retropropagación, tanto en su forma on-line como en su modalidad batch, presenta varias limitaciones teóricas, y por ello algunas otras variantes del algoritmo de retropropagación resultan mejores en términos de la rapidez de convergencia y de la exactitud de los resultados que producen.

2.10 Limitaciones del algoritmo de retropropagación estándar

Un gran porcentaje de toda la investigación, teórica y aplicada, que se desarrolla aún hoy en día para el aprendizaje supervisado de las redes neuronales, está basado en el método de optimización del steepest descent, en el que el gradiente negativo de la función de error, con respecto a todos los pesos, es calculado durante cada iteración y un paso es tomado en esa dirección. Si el paso es *adecuadamente* pequeño, la reducción de error estará garantizada. De está forma se obtendrá una sucesión de vectores de pesos w_n, que convergerá al mínimo local de la función de error

$$w_{n+1} = w_n - \eta \frac{\partial E}{\partial w_{ij}}$$

Dado que se está interesado en converger hacia el mínimo local en el tiempo más corto posible (éste no será siempre el caso, pues para la estabilidad numérica de ciertos algoritmos, se requiere que la resolución temporal sea *adecuadamente pequeña*), no existe en realidad una buena justificación para mantenerse restringido al uso del algoritmo steepest descent, pues existen al menos un par de buenas justificaciones en favor de otros métodos, que resultan ser más exactos y más eficaces en el proceso de convergencia.

La gran mayoría de las limitaciones del algoritmo de retropropagación tradicional, son heredadas directamente del algoritmo de steepest descent, pues este algoritmo no es siempre la mejor forma de decidir en que dirección avanzar en busca del mínimo, puesto que en algunas ocasiones puede conducir directamente a un comportamiento oscilatorio (movimiento en forma de zig-zag, de un lado hacia el otro en el *valle* de alguna función).

De entre los principales problemas que es posible encontrarle en la práctica al método del steepest descent es posible resaltar los siguientes:

- El método del steepest descent es un método de búsqueda con gradiente, sin embargo, la dirección del gradiente negativo de la superficie de error no está necesariamente en la dirección del mínimo de la función, puesto que el gradiente negativo solamente tiende hacia el mínimo cuando su origen (el del gradiente) se encuentra en uno de los ejes principales de la superficie, es decir, cuando el vector gradiente está exactamente sobre uno de los ejes principales de la función.

- En las ecuaciones (2.9.10),(2.9.14) y (2.9.19), la tasa de aprendizaje η es un parámetro libre y por lo tanto debe de ser elegido cuidadosamente para cada problema: si es elegido demasiado pequeño, el progreso del algoritmo será también demasiado lento, pero si es elegido demasiado grande, podría suceder que el algoritmo presentara problemas de oscilaciones.

- Incluso en la situación óptima en la que iteración tras iteración se da un paso adecuado, es decir, un paso en la dirección de descenso más pronunciado, la cual conduciría al mínimo local de la función, puede ser probado que el algoritmo del steeepest descent se comportará particularmente lento para ciertos tipos de funciones, de tal forma que de una iteración a otra, el error previo de convergencia disminuirá solamente un 0.000000001%!. En general, la convergencia en el método steepest descent es lenta, pero llega a ser más lenta conforme el algoritmo se encuentra más cerca de la solución óptima.

- Si la función de error es una forma cuádrática, un cociente grande entre el máximo eigenvalor de A (ecuación (2.7.1)) con el mínimo eigenvalor de A, dará como resultado un movimiento en forma de zig-zag durante el proceso de búsqueda

- Otro problema del algoritmo de retropropagación estándar es la ocurrencia del fenómeno conocido como *saturación prematura* de los nodos de la red [Rezgui; 1990], [Chen;1990], [Reifman y Vitela; 1994]. Para ciertos valores iniciales de los pesos (seleccionados aleatoriamente en la primera iteración del algoritmo), los valores de salida dados por (2.9.6), pueden prematuramente saturarse con los valores 0 ó 1. Cuando este fenómeno ocurre, los valores en la ecuación (2.9.11) llegan a ser muy pequeños, y cada nueva iteración del algoritmo no producirá cambios significativos en los valores de los pesos, dando como consecuencia un progreso casi nulo en la dirección del mínimo de la función. Normalmente este comportamiento persiste durante un número grande de iteraciones hasta que los valores de salida (2.9.6) se restablecen de su condición de saturamiento, y los pesos comienzan nuevamente a moverse en dirección del mínimo de la función de error. El efecto negativo de la saturación prematura, es por supuesto, un incremento en el tiempo necesario para lograr la convergencia hacia el mínimo de la función.

Todas estas limitaciones repercuten directamente en el entrenamiento de una red neuronal, pues incrementan notablemente el tiempo que se requiere para la tarea de aprendizaje, al obligar a ejecutar un número mayor de iteraciones y al obligar también a realizar un número mayor de corridas prueba-error para determinar los parámetros óptimos (tasa de aprendizaje, por ejemplo) del algoritmo.

2.11 Algoritmo de Retropropagación con la versión no lineal de los Gradientes Conjugados.

Un mejor aprendizaje para las redes neuronales puede ser conseguido aplicando *el método de optimización no lineal de los gradientes conjugados*. Contrariamente al algoritmo de retropropagación estándar, el cual minimiza el error de predicción moviéndose en dirección del gradiente negativo, y para el cual no está garantizada su convergencia en un número finito de iteraciones [Wismer; 1978], el método de los gradientes conjugados se mueve a lo largo de las direcciones conjugadas, y por lo tanto garantiza la convergencia hacia el mínimo de una función cuadrática, asumiendo que no hay errores por redondeo, en un número finito de pasos [Fletcher; 1964]. Para funciones que no son cuadráticas, como en el caso de la función de error, el proceso es iterativo y el algoritmo conducirá hacia el mínimo de cualquier valle en el que se inicie.

Además de tener mejores propiedades de convergencia y de ser menos sensible al conjunto inicial de pesos, el método de los gradientes conjugados proporciona una forma sistemática para actualizar los parámetros óptimos del algoritmo (tasa de aprendizaje, por ejemplo) en

cada iteración. Conforme el algoritmo se aproxima más hacia el mínimo de la función, su exactitud se hace más precisa y esto lo obliga a converger más rápidamente.

La siguiente sucesión de pasos muestra el algoritmo, basado en la versión no lineal de los gradientes conjugados, para el entrenamiento de una red neuronal multicapas. El algoritmo es iterativo, y en cada iteración requiere del conocimiento del gradiente de la función de error a minimizar ($\nabla E(w)$), el cual es obtenido exactamente de la misma forma que en la versión estándar del backpropagation, para el cálculo de las direcciones conjugadas $d_{(i)}$

1. Seleccionar aleatoriamente dentro del intervalo [-1,1], los pesos iniciales para la red neuronal, e igualar a uno el contador del algoritmo, es decir hacer $c=1$.

2. Presentar los p patrones de entrada-salida a la red ($\mu=1,2,...,p$), y calcular las diferencias $\zeta_i^\mu - o_i^\mu$ ($\mu=1,2,...,p;\ i=1,2,...,n$). , y detenerse si el criterio de convergencia, $|\zeta_i^\mu - o_i^\mu| < \varepsilon$, es cumplido. En cualquier otro caso, continuar hacia el paso 3.

3. Calcular el gradiente de la función de error, $g_c = \nabla E(w_c)$, en la misma forma que se hizo para el algoritmo de retropropagación estándar, es decir

$$\frac{\partial E}{\partial w_{ij}} = -\sum_{\mu} \delta_i^\mu v_j^\mu$$

si es que se tiene una red como la de la figura (2.9.1) y el peso w_{ij} va de la segunda capa intermedia hacia la capa de salida, y así sucesivamente. El peso w_{ij}, es una componente del vector de pesos w_c en la iteración c

4. Calcular las direcciones conjugadas, $d_{(i)}$ de la siguiente forma

$$d_{(c)} = \begin{cases} -g_c + \dfrac{g_c \cdot g_c}{g_{c-1} \cdot g_{c-1}} d_{(c-1)}; \text{ para } c > 1 \\ -g_1 \qquad\qquad\qquad\qquad ; \text{ para } c = 1 \end{cases}$$

donde el vector $d_{(i)}$ es una componente del vector $d_{(c)}$ en la iteración c.

5. Obtener el valor de η_k que minimiza $E(w_k + \eta \vec{i}_k)$ con respecto a η a través de una búsqueda lineal.

6. Actualizar los pesos $w_{k+1} = w_k + \Delta w_k$ donde el cambio en los pesos Δw_k está dado por $\Delta w_k = \eta_k \vec{i}_k$.

7. Regresar al paso 2 y actualizar el contador c de la siguiente forma: si $c<n$, hacer $c=c+1$; pero si $c=n$, entonces hacer $c=1$.

El algoritmo anterior es similar a la versión estándar ya antes vista. Los primeros 3 pasos anteriores son esencialmente los mismos en ambos algoritmos. Los pasos 4,6 y 7 anteriores, corresponden al cálculo de las direcciones conjugadas $d_{(k)}$, a la actualización de los pesos, y la actualización del contador respectivamente. El paso 5 de este algoritmo, que proporciona una regla sencilla para la actualización de la tasa de aprendizaje, no tiene equivalente en la versión estándar.

Con el algoritmo anterior concluye este capítulo, que ha correspondido a los conceptos teóricos necesarios para comprender los resultados que se presentarán en las secciones siguientes. El capítulo 3 corresponderá a las aplicaciones sobre sistemas dinámicos teóricos, y en el capítulo 4 se analizará un sistema dinámico real.

Tercera Parte. Aplicaciones.

Capítulo 3 Aplicación a algunos Sistemas Dinámicos Clásicos.

Este capítulo y el siguiente están destinados a la aplicación de todos los conceptos y los resultados que se han ido estableciendo a lo largo de las secciones anteriores. Los *Sistemas Dinámicos Clásicos* que se tomarán como base en este capítulo para dicho propósito son: un mapeo logístico, que mediante la variación de su parámetro dará lugar a una serie de tiempo periódica y a una serie de tiempo caótica, un movimiento armónico oscilatorio y, finalmente, el Sistema de Lorenz [Lorenz, 1963]. El sistema de Lorenz corresponde a un modelo simplificado de circulación convectiva de la atmósfera por calentamiento en la superficie y fue propuesto por Edward Lorenz en 1963 como un ejemplo de un sistema altamente sensible a pequeños cambios en las condiciones iniciales, lo cual impide, de acuerdo a las conclusiones enunciadas por Lorenz, poder hacer pronósticos confiables a muy largo plazo de la evolución atmosférica.

En este capítulo serán definidos también una serie de nuevos conceptos, como por ejemplo: las *Gráficas de Recurrencia*, la *Integral de Correlación* y la *Región de Escalamiento*, que serán de gran ayuda para la obtención del objetivo final: *desarrollar un modelo de pronóstico de precipitación pluvial con redes neuronales.*

Posteriormente en el capítulo siguiente, una vez que todos los conceptos y todas las herramientas hayan sido implementadas y validadas sobre el sistema de Lorenz, todo este mismo proceso de análisis será nuevamente puesto en práctica, pero esta vez sobre el conjunto de datos de precipitación pluvial en la Ciudad de México.

3.1 Reconstrucción del Espacio Fase

La dinámica del estado del tiempo y de los sistemas climáticos puede ser simulada por ecuaciones diferenciales parciales que describan los procesos físicos subyacentes en la atmósfera. Esas ecuaciones originales pueden ser transformadas a un conjunto de n ecuaciones diferenciales ordinarias dependientes del tiempo; el conjunto resultante de ecuaciones diferenciales ordinarias define la evolución temporal de las n variables $x_j(t)$ del sistema dinámico:

$$x'_j = f_j(x_1, x_2, \ldots, x_n): j = 1, \ldots, n. \qquad (3.1.1)$$

De esta forma el *espacio fase,* que contiene la evolución en el tiempo del sistema dinámico subyacente, es generado por las n variables $x_j(t)$, $j=1,...,n$ del proceso. Las curvas de evolución en el tiempo del sistema dinámico están formadas por las trayectorias que en este espacio fase n-dimensional se generan por la variación temporal de todas las $x_j(t)$, estas curvas de evolución exhiben diferentes patrones de comportamiento llamados atractores. Bajo ciertas circunstancias y para determinados sistemas físicos, en algunos atractores pueden aparecer movimientos irregulares o caóticos dando origen a los denominados *atractores extraños.*

El primer ejemplo de un atractor extraño relacionado con procesos atmosféricos fue presentado por Edward N. Lorenz en 1963. La dinámica de ese sistema, aunque determinística, revela un comportamiento irregular o caótico, el cual está caracterizado por una gran dependencia en las condiciones iniciales (característica típica de los sistemas atmosféricos).

Los atractores forman subespacios del sistema dinámico original, y usualmente tienen dimensiones d más pequeñas que las del espacio fase, es decir $d<n$. Los atractores con comportamiento irregular, no necesariamente están destinados a tener dimensiones enteras, por el contrario, las dimensiones fractales o no enteras parecen ser bastante comunes para muchos sistemas turbulentos que a pesar de ser determinísticos muestran una gran sensibilidad a las condiciones iniciales.

El atractor de un sistema dinámico está representado por los límites asintóticos de las trayectorias que en el espacio fase son descritas por aquellas variables independientes que definen la dinámica del proceso. En la mayoría de las aplicaciones prácticas, cuando se desea conocer algún tipo de información o las propiedades topológicas de un sistema, tal como la dimensión de su atractor, generalmente se está limitado a disponer solamente de un conjunto de observaciones del proceso, estructuradas en forma de *serie de tiempo univariada* de alguna variable o variables de estado (en algunos casos se pueden tener varias series de tiempo para un mismo proceso, cada una representando la evolución temporal de alguna variable de interés). Para obtener la información deseada cuando el proceso de estudio es un flujo turbulento o caótico, Packard et al. [Packard;1980] sugieren una *reconstrucción del espacio fase* y del comportamiento del atractor en él. Para conseguirlo hay que transformar la dinámica del proceso (serie de tiempo univariada) en una nueva dinámica, multivariada y generalmente formada mediante la incorporación sucesiva de nuevas variables independientes a un conjunto original hasta que no más información es *ganada* añadiendo una nueva coordenada (variable) independiente al espacio fase que se está reconstruyendo (piénsese, por ejemplo, en el proceso de construcción de una base para un espacio vectorial). La serie de tiempo univariada original (observable) representará una de tales coordenadas, y sus $(m-1)$ derivadas complementarán las variables independientes que se requieren. Las curvas de evolución del sistema dinámico podrán ser construidas en este espacio fase *m-dimensional reconstruido* generado a partir de la serie de tiempo univariada y de sus derivadas sucesivas.

Consideremos, por ejemplo, los datos generados por la siguiente ecuación diferencial de segundo orden que describe el movimiento de un oscilador armónico:

$$\frac{d^2x}{dt^2} = -bx \tag{3.1.2}$$

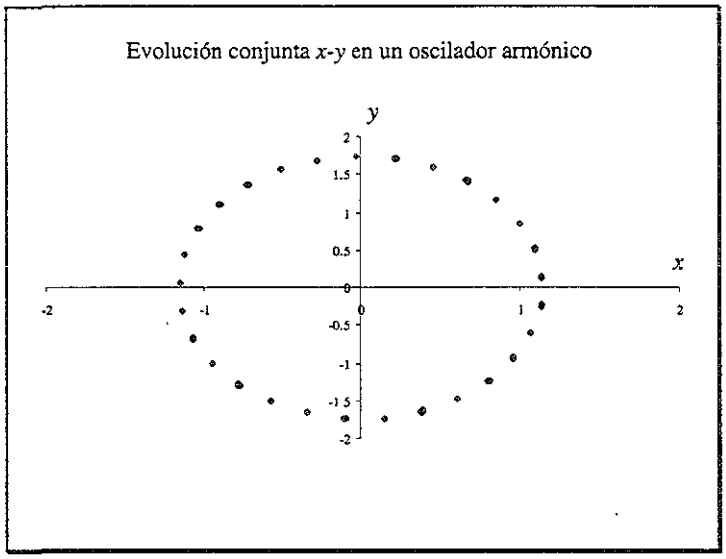


Fig.(3.1.1) Plano fase original (x,y) del sistema dinámico.

Claramente, esta ecuación puede ser reescrita en términos de dos ecuaciones diferenciales de primer orden

$$\frac{dx}{dt} = y$$
$$\frac{dy}{dt} = -bx \tag{31.3}$$

Las variables x y y forman el ***plano fase*** y las ecuaciones anteriores describen el *flujo* de la dinámica en este plano.

Supongamos que obtenemos una serie de tiempo $x(t)$ de la ecuación de segundo grado original Fig (3.1.2). ¿Cómo podemos reconstruir el plano fase y los flujos en él a partir exclusivamente de los datos $x(t)$ medidos?. En cualquier instante, la posición del sistema en el espacio fase está dada por las coordenadas (x,y), mientras que la serie de tiempo, por ella misma, nos proporciona, solamente, los valores de x a cada instante.

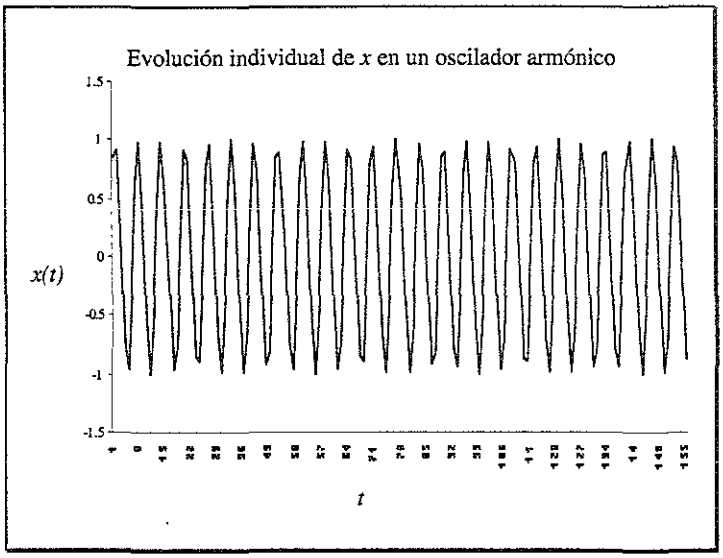


Fig.(3.1.2) La variable x medida de la ecuación (3.1.3)

Podemos medir $y(t)$ a partir de $x(t)$ observando que en la ecuación (3.1.3) $y = \frac{dx}{dt}$. Si graficamos $\frac{dx}{dt}$ contra x, obtendremos la trayectoria del sistema en el plano fase. *¡Con esto describiremos el flujo del sistema basados, exclusivamente, en los datos medidos!.*

Ahora bien, dada una serie de tiempo $x(t)$, ¿cómo podemos calcular su derivada, es decir, $\frac{dx}{dt}$?. Los datos de los fenómenos que frecuentemente estaremos interesados en estudiar son colectados periódicamente (en forma diaria, semanalmente, mensualmente, etc.) a través de *observación en campo*, o a través de dispositivos electrónicos vinculados a equipos de cómputo, de tal forma que las mediciones $x(t)$ en realidad consisten de una secuencia de observaciones realizadas a intervalos discretos de tiempo $x_0, x_1, x_2, \ldots, x_n, \ldots$

Utilizando una de las definiciones posibles de derivada de una función x en un punto t observamos lo siguiente:

$$\frac{dx(t)}{dt} = \lim_{h \to 0} \frac{x(t+h) - x(t)}{h} \tag{3.1.4}$$

Claramente podríamos extender la definición anterior para el cálculo de las *derivadas de la serie de tiempo* observada, de acuerdo a la siguiente expresión:

$$\frac{dx_t}{dt} = \frac{x_{t+h} - x_t}{h} \tag{3.1.5}$$

Para las mediciones discretas en el tiempo, de que constan las series de tiempo, h puede tomar sólo los valores 0,1,2,3,..., y no puede tomar valores fraccionarios, sin embargo, en términos prácticos, el valor más pequeño que h puede tomar es uno, aunque en algunas ocasiones será apropiado seleccionar un valor mayor.

De esta forma, observamos que la reconstrucción del espacio fase se reduce a graficar $\frac{x_{t+h} - x_t}{h}$ contra x_t. Notemos que solamente están involucradas dos cantidades x_{t+h} y x_t. Ellas contienen *toda* la información del sistema y es relativamente sencillo graficarlas. En la figura (3.1.3) se muestra la reconstrucción del espacio fase con $h=8$

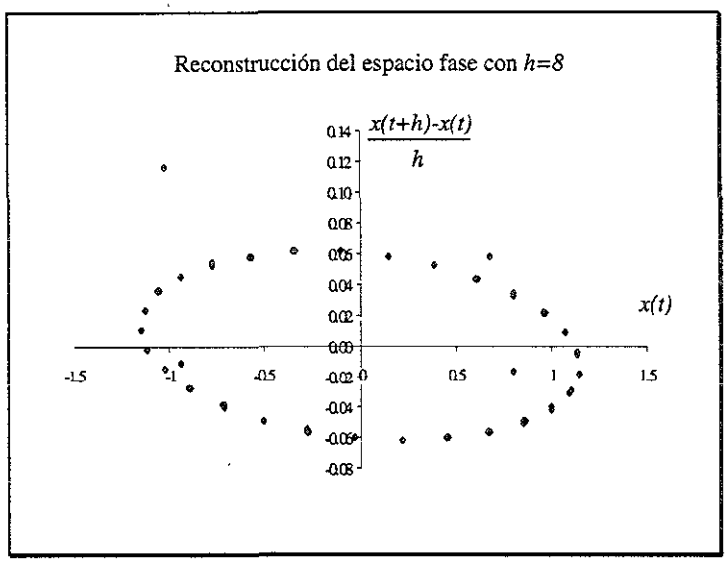


Fig.(3.1.3) Reconstrucción del espacio fase a partir de una serie de tiempo observada $x(t)$

La ecuación (3.1.3) es simplemente una situación especial puesto que en este caso $\frac{dx}{dt}$ nos proporciona en forma directa el valor de y, pero en general, la dinámica en el espacio fase está determinada por un par de ecuaciones diferenciales del siguiente estilo

$$\frac{dx}{dt} = f(x,y), \qquad \frac{dy}{dt} = g(x,y) \tag{3.1.6}$$

Si tomamos como punto de partida este esquema más general, y si en algún problema práctico solamente podemos medir u observar una serie de tiempo $x(t)$, ¿cómo podemos calcular el valor de $y(t)$?. En la gran mayoría de los problemas reales, frecuentemente estamos imposibilitados para obtener los valores de y o de z o de cualquier otra variable que, aunque sepamos que es importante

para la dinámica del proceso en estudio, las limitaciones en cuanto a tiempo, a equipo, a personal capacitado o a recursos económicos, no nos permiten medir u observar todas las variables a la vez sino solamente a algún subconjunto de ellas, sin embargo, ***esto no representa un gran problema pues para muchas aplicaciones, incluyendo los fines que perseguimos con este trabajo, ¡no nos hacen falta!***. **Para reconstruir en el espacio fase toda la información relevante acerca de la dinámica del fenómeno en estudio, nos es suficiente con una sola serie de tiempo.** Notemos que si contamos con una medición $x(t)$ y calculamos su derivada con respecto al tiempo, es decir $\frac{dx(t)}{dt}$, tendremos tanto una medición directa de x como un valor calculado de $f(x,y)$.

Algo de información acerca de y esta contenida en el valor de $f(x,y)$ (obtenido solamente a partir de $x(t)$) , y frecuentemente esta información es suficiente para obtener una muy buena idea acerca del comportamiento del sistema dinámico. La sucesión de figuras siguiente (figs.(3.1.4),(3.1.5),(3.1.6)) nos proporciona un ejemplo que nos ilustra en forma bastante clara como la reconstrucción del espacio fase (x_t,x_{t+h}) se compara con el espacio fase original (x,y) a pesar de que nunca se hicieron mediciones directas de la variable dinámica y. En la práctica el espacio fase puede ser reconstruido tomando como coordenadas $((x_{t+h}-x_t)/h,\ x_t)$, o directamente (x_{t+h},x_t). La calidad de la reconstrucción dependerá básicamente de la elección de h.

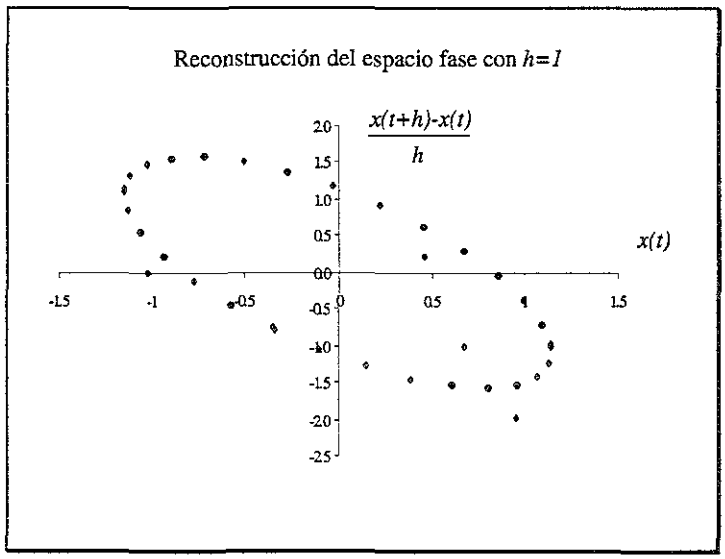


Fig.(3.1.4) Reconstrucción del espacio fase a partir de una serie de tiempo observada $x(t)$

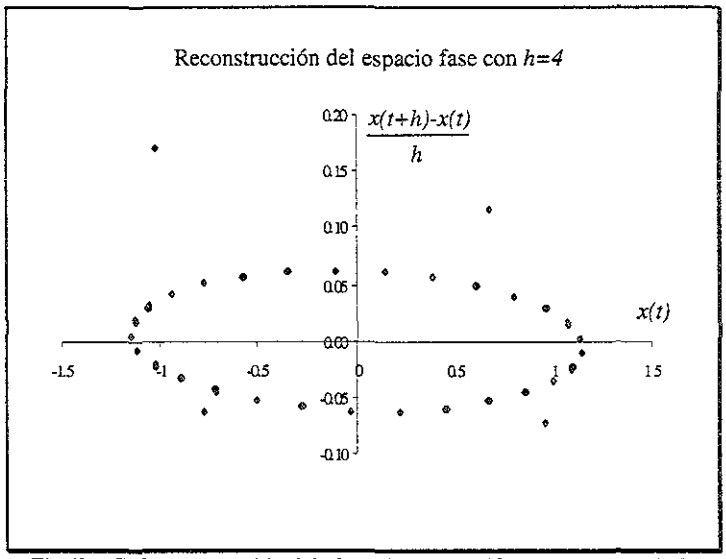


Fig.(3.1.5) Reconstrucción del plano fase para diferentes valores de *h*. Nótese como la calidad de la reconstrucción depende del valor de *h* seleccionado

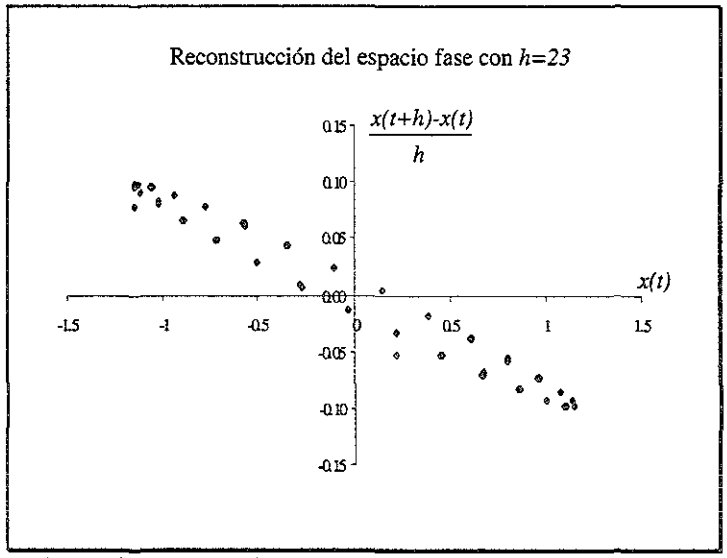


Fig.(3.1.6) Reconstrucción del plano fase para diferentes valores de *h*. Nótese como la calidad de la reconstrucción depende del valor de *h* seleccionado

3.2 Incrustando una Serie de Tiempo

El conjunto de *n* ecuaciones diferenciales ordinarias

$$\frac{dx_j(t)}{dt} = f_j(x_1,...,x_n): \qquad j = 1,...,n \tag{3.2.1}$$

modela la dinámica del estado del tiempo y de los sistemas climáticos en un espacio fase *n-dimensional.* El espacio fase es generado por *n* coordenadas x_j , $j=1,...,n$, las cuales son definidas por las *n* variables independientes del sistema dinámico.

De esta forma, la evolución del sistema en el tiempo está dada por un vector $\xi(t) = [x_1(t),...,x_n(t)]$, cuyas componentes definen la posición de la trayectoria en el espacio fase.

Ahora bien, notemos que el sistema de ecuaciones (3.2.1) puede ser reducido a una *única ecuación diferencial no lineal de orden n* para alguna de las *n* variables del sistema dinámico, por ejemplo para $x_j(t)=x(t)$, mediante diferenciación de la restantes *n-1* variables, con lo cual obtendríamos una ecuación como la siguiente

$$\frac{d^n x(t)}{dt^n} = f(x(t), x'(t), x''(t),.., x^{n-1}(t)) \tag{3.2.2}$$

Es importante enfatizar en el significado de lo anteriormente expuesto: hemos visto que un sistema de *n* ecuaciones diferenciales describe el estado del tiempo en un espacio *fase n-dimensional*, sin embargo dicho sistema inicial de *n* ecuaciones diferenciales puede ser reducido a una sola ecuación diferencial de orden *n* para alguna de las *n* variables del sistema dinámico y sus respectivas *n-1* derivadas, lo cual quiere decir que *una sola ecuación de grado n (para alguna de las variables del sistema y de sus derivadas) es capaz de describir la dinámica del estado del tiempo!.*

Notemos además que en esta única ecuación diferencial nueva, equivalente al conjunto original de *n* ecuaciones, las *n* variables que ahora se utilizan y que constituyen el espacio fase reconstruido, en lugar de ser $x_1(t), x_2(t),..., x_n(t)$, pasan a ser $x_i(t), x_i'(t),..., x_i^{(n-1)}(t)$, es decir, un vector con componentes $\boldsymbol{x}_i(t) = (x_i(t), x_i'(t), x_i''(t),..., x_i^{(n-1)}(t))$ es el que ahora define la posición de las trayectorias (del sistema dinámico) y su evolución en el tiempo. Incluir variables (derivadas) adicionales, por ejemplo $x_i^{(n)}(t)$, al vector *n*-dimensional anterior es superfluo puesto que ello no proporcionará más información independiente.

Ahora, si consideramos trayectorias dentro del espacio fase *n-dimensional* original $x_1, x_2,..., x_n$ (por ejemplo un atractor o cualquier otro objeto geométrico de dimensión $d<<n$) **éste podrá también**

ser descrito en el nuevo espacio fase $x_i, \dot{x}_i, \ddot{x}_i, \ldots$, generado por *la i-esima* variable y sus derivadas.

Es importante aclarar que la dimensión del nuevo espacio fase (donde se reconstruye la dinámica del atractor o del objeto geométrico) puede ser más pequeña que la del espacio fase original, sin embargo tanto el atractor u objeto matemático original como el reconstruido tendrán la misma dimensión. Esto se conoce como *incrustación* y está justificado sólidamente por el *Teorema de incrustación de Takens* [Takens;1981] que es aplicable para la gran mayoría de los sistemas dinámicos.

El teorema establece que objetos geométricos *d-dimensionales* descritos por el sistema dinámico (3.2.1) y evolucionando en el espacio fase con coordenadas $x_j(t)$: $j = 1,2,\ldots,n$ pueden ser incrustados dentro de un *espacio fase (m=2d+1)-dimensional* (por ejemplo el generado por alguna variable y sus sucesivas derivadas). Este teorema, que requiere una dimensión de incrustamiento de *m=2d+1* coloca el análisis de la dimensionalidad de atractores o de objetos geométricos, dentro de un *lado seguro*, es decir, para determinar las dimensiones de atractores de sistemas físicos es suficiente con garantizar dimensiones *m* de incrustamiento que sean suficientemente más grandes que las de los atractores (la que tendrían si pudiéramos verlos en su espacio fase real o teórico), las cuales son generalmente de menor dimensión con respecto al espacio fase original, es decir $d < m \leq n$. En la práctica cualquier valor de $m \geq d$ es frecuentemente adecuado para la reconstrucción, sin embargo la única garantía viene cuando $m \geq 2d + 1$.

De esta forma, para determinar la dimensión de atractores a partir de una única variable de estado es suficiente incrustarlos en un espacio *m-dimensional* generado por la serie de tiempo y sus *(m-1)* derivadas $\boldsymbol{x}_i(t) = (x_i(t), \dot{x}_i(t), \ddot{x}_i(t), \ldots, x_i^{(n-1)}(t))$, es decir, **no es necesario conocer el espacio fase original, ni su dimensión, ni el conjunto de variables de estado independientes siempre y cuando *m* sea elegida lo *suficientemente grande.***

Por otro lado, las coordenadas del espacio fase reconstruido deben de ser por lo menos linealmente independientes pues sólo ello garantizará una adecuada reconstrucción de la dinámica en ese espacio (recordemos, por ejemplo, que en el espacio euclidiano los vectores canónicos que lo generan son linealmente independientes, además de ortogonales). Para datos observados la independencia funcional que se requiere puede ser conseguida escogiendo la cantidad *h*, mencionada en la sección anterior, de tal forma que las observaciones $x(t), x(t+h), x(t+2h), \ldots$, sean independientes entre ellas. Nótese que lo ideal es pedir *independencia funcional* y no solamente independencia lineal

Como una primera aproximación pudiera pensarse en el coeficiente de correlación lineal como una vía para determinar el valor de *h*, sin embargo debemos de recordar que el valor de este estadístico solamente nos proporciona el grado de asociación lineal que existe entre dos variables y lo que nosotros en realidad deseamos es conocer cualquier tipo de dependencia funcional que pudiera existir entre los datos de nuestra serie de tiempo para de esta forma poder elegir aquel valor de *h* en donde dicha dependencia funcional sea mínima o nula y nos garantice, por lo tanto, una total independencia en las observaciones, de tal forma que la calidad en la reconstrucción del espacio fase sea *óptima*. (Nótese como en la sucesión de figuras (3.1.4), (3.1.5), (3.1.6) se utilizan diferentes valores para *h* y se obtienen diferentes calidades en el espacio fase reconstruido).

La información mutua es una medida de la cantidad de información (no sólo información lineal) que una variable aleatoria (dato observado) contiene acerca de otra (dato observado), en este sentido nos proporciona una medida de la distancia entre dos distribuciones de probabilidad, y por ello es una mejor herramienta para la elección de *h*. De hecho, de acuerdo con Fraser [Fraser;1990] una elección *óptima* para *h* está dada por el primer mínimo en la información mutua. Compárese la figura (3.2.2) con la sucesión de figuras (3.1.4),(3.1.5),(3.1.6).

Observese que sucede si en la sucesión de figuras antes mencionada en lugar de escoger, valores arbitrarios para *h* seleccionamos el primer mínimo proporcionado por la información mutua.

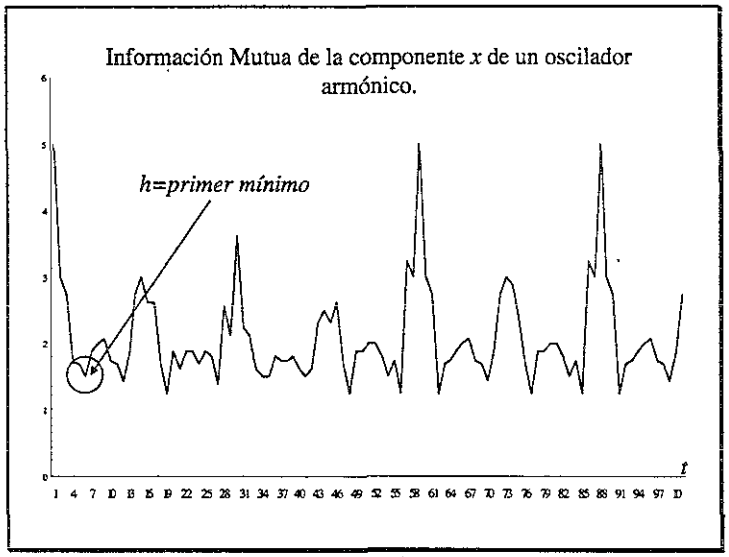


Fig.(3.2.1) Determinación del *h* óptimo con Información Mutua

En general, cuando se desea reconstruir la geometría de un *sistema caótico que es continuo en el tiempo*, y que está evolucionando en un espacio de dimensión *n*, a partir únicamente de una sucesión discreta de observaciones, la serie de tiempo unidimensional debe de ser *incrustada* en un espacio *m-dimensional,* graficando los siguientes arreglos de *m* coordenadas

$$\boldsymbol{x}_t=(x_t,x_{t+h},x_{t+2h},\ldots,x_{t+(m-1)h})$$

Donde los $\boldsymbol{x}_t$ denotan los vectores incrustados y los x_t denotan una única medición al tiempo t o en alguno de sus múltiplos.

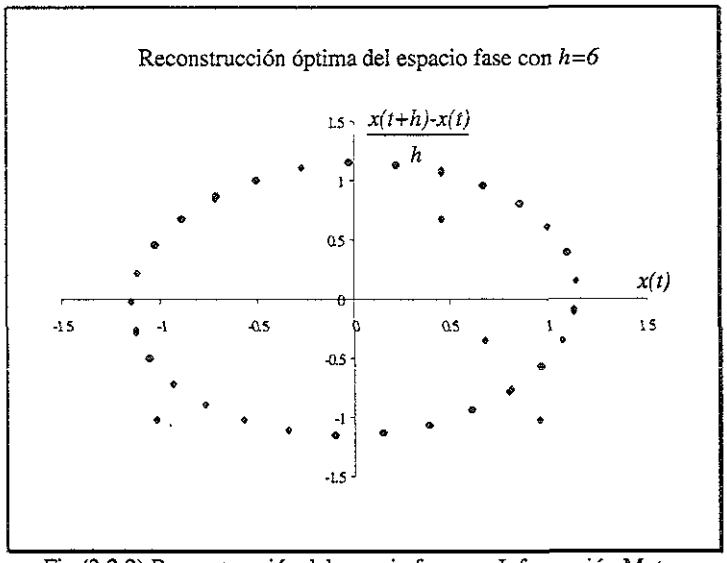


Fig.(3.2.2) Reconstrucción del espacio fase con Información Mutua

Esta técnica de representar una serie de tiempo medida u observada como una sucesión de puntos en un espacio de *m* dimensiones se conoce como un *incrustamiento de rezagos en el tiempo*. El *Teorema de incrustamiento de Takens* establece que la dinámica reconstruida del sistema dinámico es geométricamente similar a la dinámica original tanto para sistemas dinámicos continuos como para sistemas dinámicos discretos. La sucesión de vectores creados al incrustar una serie de tiempo se conoce como la *trayectoria reconstruida, m* se conoce como la *dimensión de incrustamiento*, y *h* como el *rezago de incrustamiento.*

3.3 Gráficas de Recurrencia

Las variables de estado de datos observados están dadas por $x(t_i) = (x(t_i), x(t_i + h), ..., x(t_i + (m-1)h))$. Cada $x(t_i)$ es un punto en un espacio de incrustamiento *m-dimensional* y la serie de tiempo incrustada puede ser considerada como una sucesión de puntos. Cada punto representa el estado del sistema en el tiempo t_i.

Podemos calcular la distancia entre dos puntos en los tiempos i y j de la siguiente forma

$$\delta_{ij} = \left| x(t_i) - x(t_j) \right| \tag{3.3.1}$$

Si la serie de tiempo es periódica con periodo T, entonces $\delta_{ij} = 0$ cuando $|i-j| = nT$ para $n=0,1,2,3,...$

Supongamos que seleccionamos alguna distancia r y nos preguntamos ahora cuando $|x(t_i) - x(t_j)| < r$; una forma de resolverlo consiste en graficar en el plano i-j el punto (i,j) si $|x(t_i) - x(t_j)| < r$. Este tipo de gráficas llamadas de recurrencia reflejan como la trayectoria reconstruida es recurrente o se repite a ella misma.

Para una señal periódica de periodo T la gráfica de recurrencia está constituida por una serie de franjas a 45° que están separadas por una distancia T en las direcciones horizontal y vertical, para cada valor pequeño de r. Figura (3.3.1) y (3.3.2).

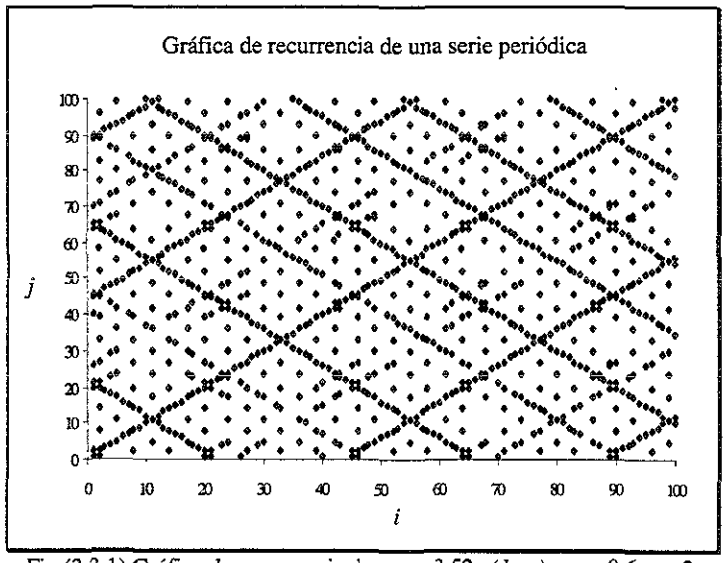


Fig.(3.3.1) Gráfica de recurrencia de $x_{n+1}=3.52x_n(1-x_n)$: $x_0 = 0.6$, $m=2$, $r=0.001$, $h=1$

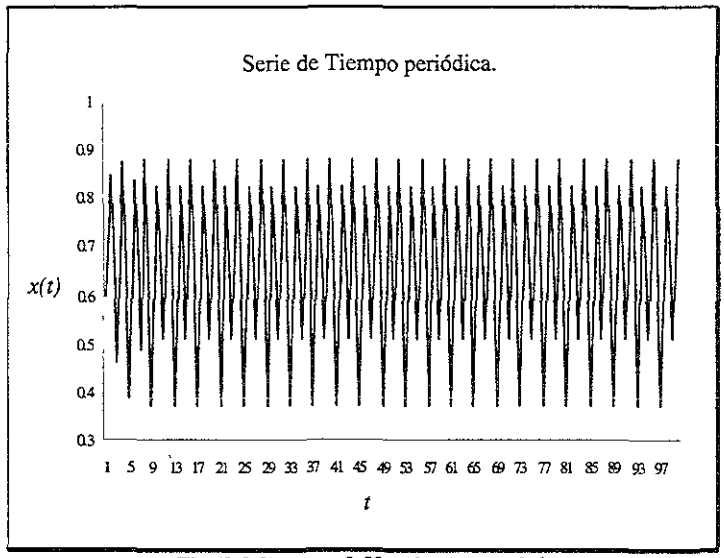


Fig.(3.3.2) $x_{n+1}=3.52x_n(1-x_n)$: $x_0 = 0.6$

Para una serie de tiempo caótica, este tipo de gráficas representan un patrón de comportamiento más complicado. Pueden aparecer eventualmente pequeños brotes de regularidad (franjas a 45°) pero también es frecuente observar patrones que no reflejan algún tipo de comportamiento periódico. Figura (3.3.3).

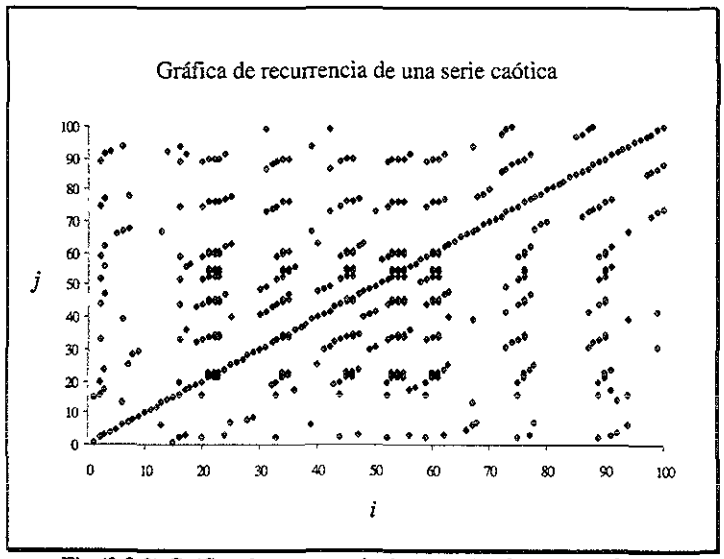


Fig.(3.3.3) Gráfica de recurrencia de $x_{n+1}=4x_n(1-x_n)$: $x_0 = 0.456$

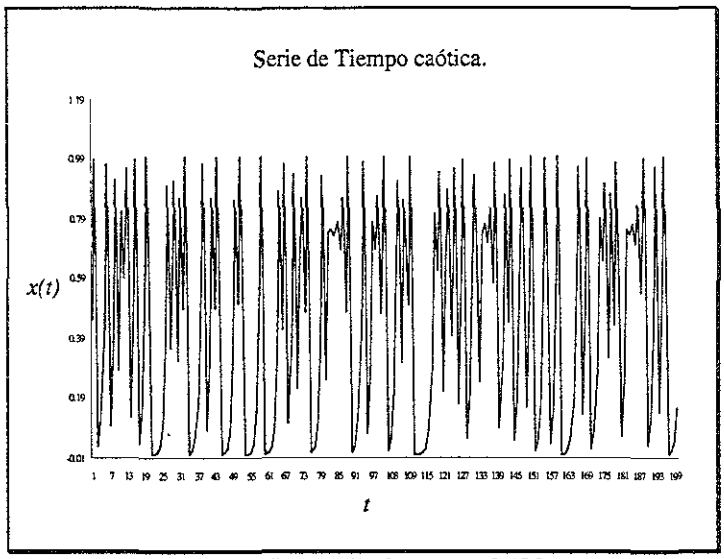


Fig.(3.3.4) $x_{n+1}=4x_n(1-x_n)$: $x_0 = 0.456$

Para una serie de números aleatorios, no es evidente algún tipo de patrón de regularidad. Sin embargo, todas las gráficas de recurrencia existirá al menos una franja de puntos en la diagonal principal correspondiente a $i=j$.

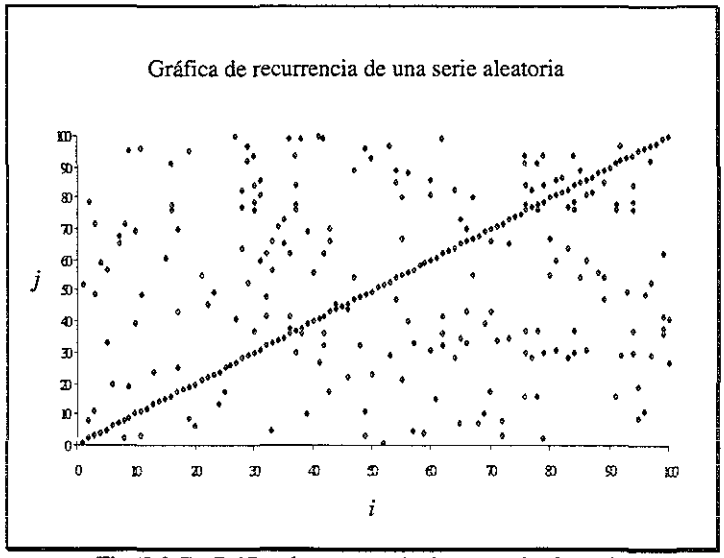


Fig.(3.3.5). Gráfica de recurrencia de una serie aleatoria

El número de puntos en una gráfica de recurrencia nos indica cuántas veces la trayectoria se mantuvo dentro de una determinada distancia r.

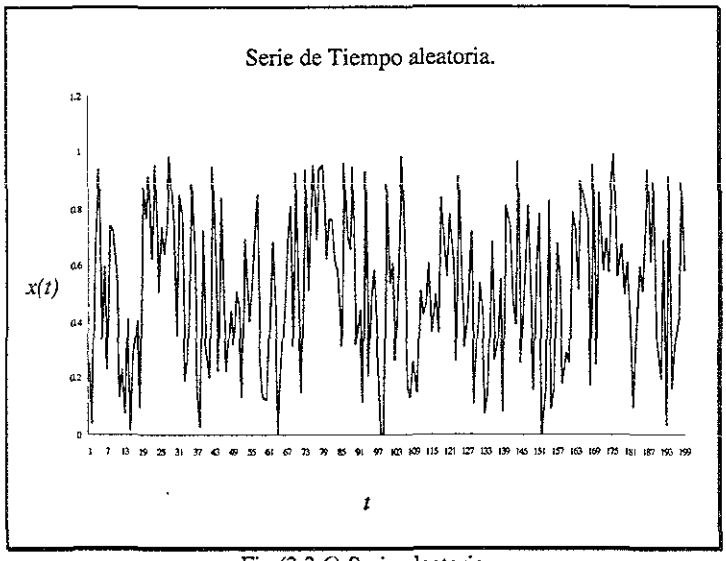


Fig.(3.3.6) Serie aleatoria

3.4 Integral de Correlación

Una herramienta que está íntimamente relacionada con las gráficas de recurrencia y que nos proporciona un conjunto de elementos adicionales para el análisis de una serie de tiempo es la *integral de correlación C(r)* de los elementos de la serie de tiempo, la cual se define como el cociente entre el número de pares de puntos $(x(t_i),x(t_j))$ tal que su distancia δ_{ij} es menor que r, y el total de pares de puntos que es posible formar (a saber n^2). Es muy importante notar que los puntos $x(t_i)$ están en algún espacio *m-dimensional* ($m = 1,2,3,\ldots$).

$$C(r) = \frac{1}{n^2}\sum_{i,j=1}^{n} H(r - |x(t_i) - x(t_j)|) \tag{3.4.1}$$

Donde H esta dada por $H(x) = \begin{cases} 0 : x \le 0 \\ 1 : x > 0 \end{cases}$ (3.4.2)

$C(r)$ describe cómo se incrementa el número de parejas cercanas conforme se incrementa la distancia r. La integral de correlación es una de las cantidades más importantes en el análisis de series de tiempo caóticas. Lo que es importante no es el valor de $C(r)$ para algún valor particular de r, sino los cambios que se dan en ella mediante el cambio gradual de r. Cuando r aumenta, una mayor cantidad de puntos aparecen en la gráfica de recurrencia y por lo tanto $C(r)$ aumenta. La sucesión de figuras (3.4.1)-(3.4.3) nos muestra la integral de correlación para una serie periódica, para una serie caótica y para una serie aleatoria (ruido blanco).

Para un sistema perfectamente periódico, incrementar r en poco no cambiara en mucho el numero de puntos $C(r)$, para un sistema caótico, el mismo incremento de r provocará que más puntos se integren a $C(r)$, pero indiscutiblemente el incremento más notable para $C(r)$ se dará cuando el conjunto de datos constituya un ruido blanco.

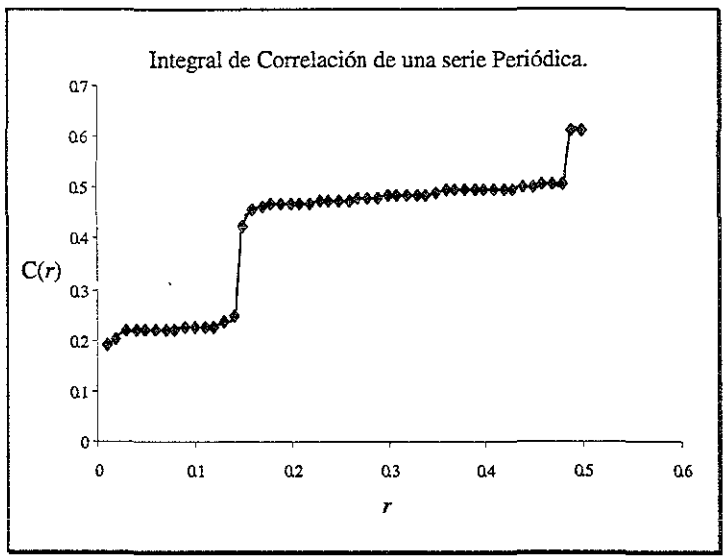


Fig.(3.4.1) Integral de correlación de $x_{n+1}=3.52x_n(1-x_n)$: $x_0 = 0.6$

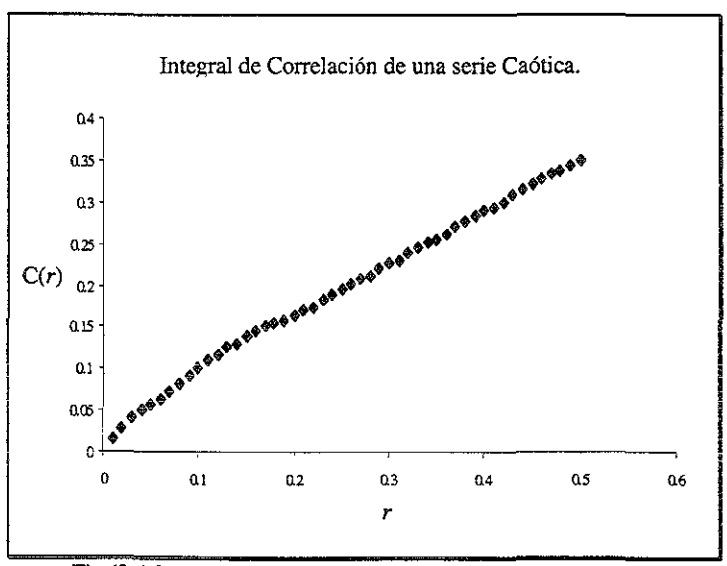


Fig.(3.4.2) Integral de correlación de $x_{n+1}=4x_n(1-x_n)$: $x_0 = 0.6$

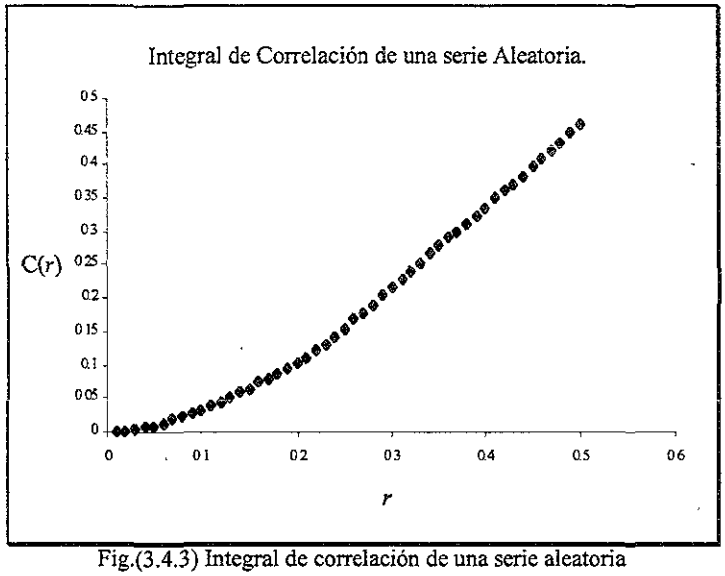


Fig.(3.4.3) Integral de correlación de una serie aleatoria

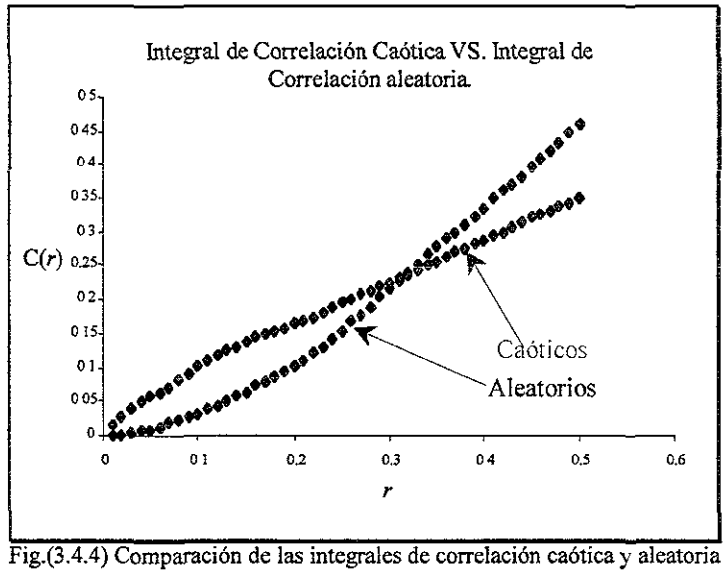


Fig.(3.4.4) Comparación de las integrales de correlación caótica y aleatoria

Notamos que la gráfica de $C(r)$ es plana para el sistema periódico; que tiene una pendiente moderada para los datos del sistema caótico y que tiene una pendiente más pronunciada para el sistema aleatorio.

Existe una relación muy estrecha entre la integral de correlación $C(r)$ y el concepto de dimensión de un objeto geométrico. Supóngase por ejemplo que se tiene un conjunto de puntos dispersos en forma más o menos uniforme en una curva unidimensional. Si se fija uno de los puntos como de referencia y se cuenta cuántos de los otros puntos se encuentran a lo más a una distancia r del de referencia, se encontrará que a medida de que r aumenta el número de puntos dentro de la distancia r también aumentará y en forma directamente proporcional a la longitud r. Ahora, si los puntos

estuvieran dispersos en forma más o menos uniforme pero en una superficie bidimensional, al elegir a uno de ellos como de referencia, se observaría que la cantidad de puntos, dentro de la distancia r estaría relacionada con el área del círculo de radio r, es decir πr^2, pues a una mayor área corresponderá un número mayor de puntos; similarmente si los puntos estuvieran dispersos en un volumen tridimensional, el numero de puntos con distancia menor a r con respecto al de referencia estaría relacionado con el volumen de una esfera de radio r, es decir $4/3\pi r^3$. En general, para puntos dispersos dentro de un objeto *d-dimensional* la cantidad de puntos que se encuentran a una distancia menor que r del de referencia es proporcional a r^d.

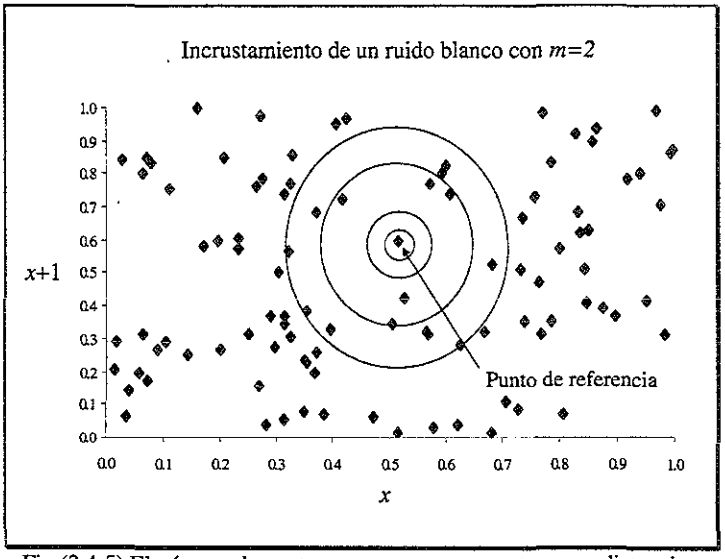


Fig.(3.4.5) El número de puntos cercanos con respecto a una distancia r se incrementa proporcionalmente con el área de un círculo de radio r.

3.5 Dimensión de correlación, Dimensión del atractor y Región de escalamiento.

En general, para calcular la integral de correlación de un conjunto de puntos, se deberá de utilizar a cada uno de los puntos como de referencia y se deberá de contar cuántos de los otros puntos se encuentran dentro de una distancia r, esto sugiere que la integral de correlación de un conjunto de puntos dispersos dentro de un volumen *d-dimensional* deberá de ser proporcional a r^d, es decir,

$$C(r) = Ar^d \tag{3.5.1}$$

O bien $C(r) \approx r^d$, donde A es una constante de proporcionalidad.

Si tomamos logaritmo de ambos lados obtendremos

$$\log C(r) = d \times \log(r) + \text{constante} \qquad (3.5.2)$$

Por lo tanto, para determinar el valor de *d*, conocido como la *dimensión de correlación*, simplemente se deberá de graficar *logC(r)* contra log(*r*) y se deberá de calcular la pendiente de las curvas resultantes para varios valores de *r*. La ecuación (3.5.1) también se conoce como *regla de escalamiento*.

La dimensión de correlación puede ser utilizada para buscar atractores en series de tiempo. Originalmente esta idea fue propuesta por Grassberger & Procaccia en 1983 [Grassberger;1983] y está basada en el hecho de que los atractores de los sistemas caóticos son con mucha frecuencia bastante similares y pueden ser descritos por una dimensión fractal. Si una serie de tiempo proviene de un sistema dinámico que está en un atractor, entonces la trayectorias reconstruidas mediante la incrustación de la serie de tiempo tendrán las mismas propiedades topológicas que el atractor original (siempre y cuando la dimensión de incrustamiento *(m)* sea lo suficientemente grande y la elección de los rezagos en el tiempo *(h)* sea adecuada). *En particular la trayectoria reconstruida tendrá la misma dimensión que la original.*

Dada una serie de tiempo arbitraria, no es posible en general saber de antemano el valor *m* de la dimensión de incrustamiento que se debe de utilizar para determinar, empleando la integral de correlación, el valor *d* de la dimensión de su atractor. En realidad, si pudieramos de primera instancia conocer para cualquier serie de tiempo el valor *m* (que es una cota superior bastante razonable para el número de variables suficientes para modelar la dinámica del atractor) tendríamos más de la mitad del problema resuelto, pues de acuerdo al teorema de Takens (si el atractor en el espacio fase original tiene dimensión *d*, entonces una dimensión de incrustamiento de $m=2d+1$ será adecuada para reconstruirlo) podriamos proponer $d=(m-1)/2$ como una primera aproximación a los grados de libertad relevantes después de que se estabiliza en sistema (y también como el número de nodos en la capa de entrada para la *Red Neuronal*) y a partir de ahí podríamos mejorar el valor de dichos grados de libertad por *prueba y error* o siguiendo la metodología de Grassberger y Procaccia pero ya directamente en un espacio *m-dimensional*, sin tener que *explorar* las *m-1* dimensiones restantes.

Es importante enfatizar que en la práctica, dado un sistema dinámico evolucionando en el tiempo, la gran mayoría de las veces solamente dispondremos, de un conjunto discreto de observaciones (serie de tiempo) que incluso pueden no ser las más representativas del fenómeno. Normalmente en casi todas las aplicaciones de este tipo ni siquiera es posible llegar a identificar en su totalidad a las *n* variables que están involucradas en el evolución del sistema y por lo tanto es aún mucho más difícil poder pensar en identificar la dimensión del atractor, si es que éste existe, o en poder pensar en identificar la dimensión total en la que está evolucionando el sistema dinámico completo.

Puesto que el objetivo del análisis de Grassberger y Procaccia (integral de correlación) es el de encontrar la dimensión *d* del atractor (y según como hemos visto no conocemos *m*), la solución a este problema consiste en calcular *d*, a partir de la integral de correlación *C(r)* empleando diferentes valores para *m*, desde $m=1$ hasta conseguir una *saturación* en el valor de *d*. Figs. (3.5.1),(3.5.2),(3.6.5).

La explicación para este proceso se entiende perfectamente si consideramos una serie de tiempo consistente de ruido blanco y nos preguntamos por la dimensión *d* de su atractor, si es que éste existe. Podemos observar que cuando los datos son incrustados con $m=2$, se obtendrá un conjunto

de puntos dispersos en el plano que tenderán a abarcar todo el plano (o así lo harían si tuviéramos una cantidad suficiente de datos), podemos pensar entonces que su dimensión es 2 (es decir $d=2$ cuando $m=2$). Similarmente, si incrustamos los datos con $m=3$ obtendríamos un conjunto de puntos en el espacio que tendería a llenarlo si tuviéramos un número suficiente de ellos, es decir $d=3$ cuando $m=3$. En el *caso ideal*, para un auténtico ruido blanco $d=m$, donde por ideal deberemos de entender que contamos con una cantidad *infinita* de datos. Ahora bien, evidentemente en la práctica nunca es posible disponer de un número infinito de datos, sin embargo, es posible mostrar que para conjuntos *pequeños* de datos, la relación $d \approx m$ se preserva y podemos pensar, aunque no es una regla infalible, que 10^m datos son necesarios para mostrar que $d \approx m$ para cualquier valor de m, [Kaplan;1995]. Mientras que, por otro lado, para una serie de tiempo proveniente de un sistema dinámico con un atractor, el crecimiento más alla de toda cota de d terminará en algún momento para algún valor finito y *pequeño* de m. Precisamente ese valor al que converge d será la dimensión del atractor.

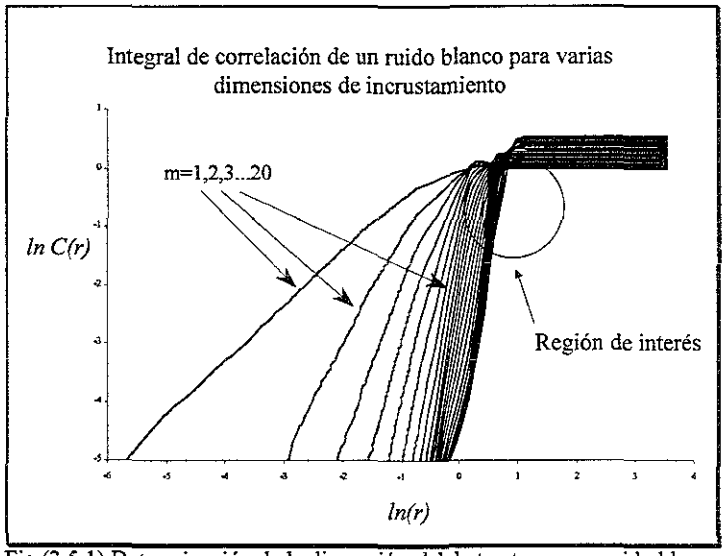


Fig.(3.5.1) Determinación de la dimensión d del atractor en un ruido blanco

Es posible ilustrar los comentarios anteriores con el siguiente ejemplo. Considérense 2 series de tiempo: un ruido blanco (fig. (3.3.6)) y la serie de tiempo de la componente x en el ejemplo del oscilador armónico de la sección 3.1 (fig. (3.1.2)). Obsérvese que sucede en la determinación de d, en las gráficas $ln(r)$ contra $ln(C(r))$, figuras (3.5.1) y (3.5.2).

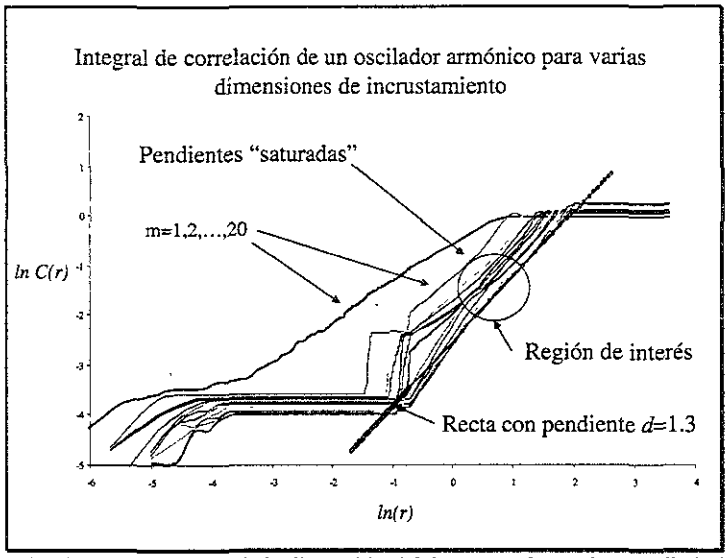


Fig.(3.5.2) Determinación de la dimensión *d* del atractor de un sistema dinámico por saturamiento de las pendientes.

Nótese la diferencia en las dos gráficas al calcular *ln C(r)* para valores de las abscisas mayores que cero y menores, para este caso, que dos (obsérvese lo mismo en la gráfica correspondiente al sistema de Lorenz en la siguiente sección Fig.(3.6.5)).

Para la determinación de *d*, existen ciertos métodos *empíricos* como el siguiente [Hao Bai-lin,1984], la regla de escalamiento, ecuación (3.5.1), será válida sobre aquella región intermedia, *región de escalamiento*, donde *ln r* es más pequeño que el tamaño del *objeto buscado* (atractor) y más grande que la distancia más pequeña entre los puntos del conjunto. En este caso, tales regiones intermedias, magnificadas, son mostradas en las figuras (3.5.3) y (3.5.4).

Obsérvese como para un ruido blanco la pendiente de las curvas no se *satura*, mientras que para la componente *x* del oscilador armónico, el valor de *d* es encontrado en donde la pendiente de las curvas se hace constante dentro de la región de escalamiento.

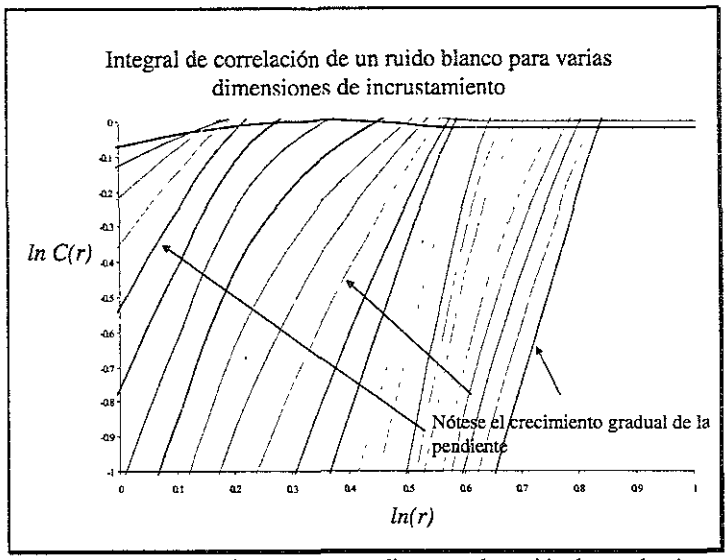


Fig. (3.5.3) Comportamiento de las pendientes en la región de escalamiento para un ruido blanco.

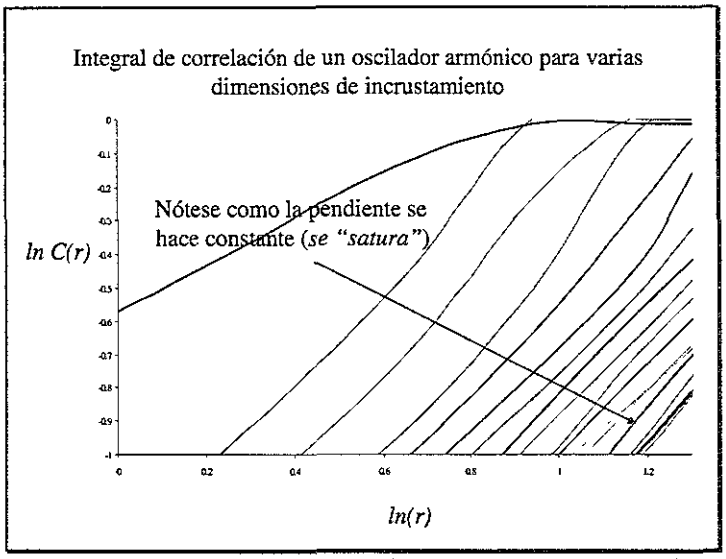


Fig. (3.5.4) Comportamiento de las pendientes en la región de escalamiento para la componente *x* de un oscilador armónico.

Lo anteriormente establecido puede ser resumido en la siguiente gráfica.

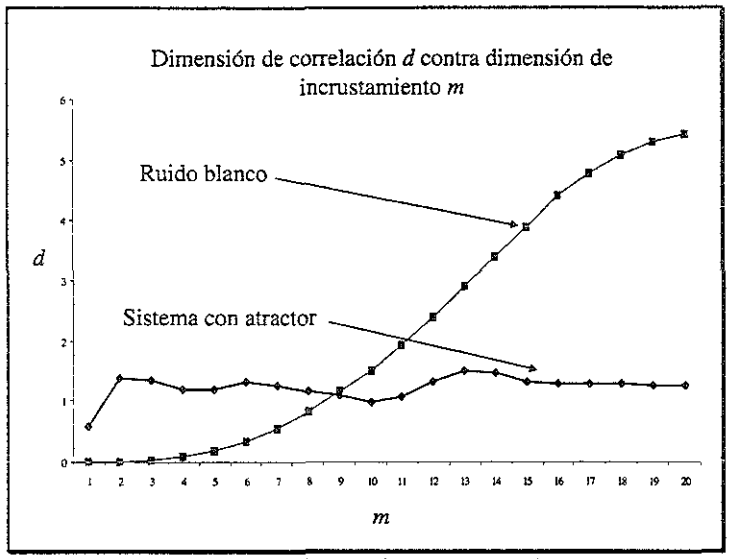


Fig.(3.5.5) En el ruido blanco las pendientes crecen más alla de toda cota ($d \approx m$ para 10^m datos), mientras que en los Sistemas con atractor la pendiente tiende a hacerce constante y a indicarnos el valor de d

Es importante observar que las gráficas no son *perfectas*, es decir, no reflejan con una precisión del 100% lo que la teoría indica; existen muchas razones del porque sucede esto, entre ellas, por ejemplo, es posible mencionar que solamente fueron utilizados alrededor de 200 datos para el análisis mostrado en las gráficas anteriores, mientras que la teoría dice que para observar claramente como $d \approx m$ en un ruido blanco se requieren aproximadamente 10^m observaciones; esto querría decir que en el ejemplo anterior, en el que se exploró el comportamiento de los datos hasta con una $m=20$, se necesitarían alrededor de 10^{20} datos!. No obstante, a pesar de que solamente fue utilizada poca información, los resultados son bastante consistentes con lo que debería de esperarse: en el ruido blanco la pendiente crecerá más alla de toda cota y para el sistema dinámico con atractor, la pendiente se volverá constante después de algunas iteraciones.

La principal limitación experimental que surge con la aplicación de este procedimiento es la *necesidad de satisfacer el límite cuando* $n \to \infty$. El número de puntos necesarios para satisfacer este límite es una función fuertemente dependiete de la dimensión, es decir, el número de puntos necesarios para medir la dimensión de un objeto *d-dimensional* puede crecer tan rápido como 10^d o como 100^d. La prueba práctica para determinar el tamaño adecuado del conjunto de datos consiste en incrementar paulatinamente el número de puntos hasta que las estimaciones de la dimensión (pendientes) dejen de cambiar. Desafortunadamente, al ganar precisión en la estimación de d mediante el incremento del número de puntos, se pierde velocidad en los cálculos, pues la dimensión de correlación, ecuacion (3.4.1), es un algoritmo de orden n^2 (el número de operaciones necesarias es cuadrático en el número de puntos). Los algoritmos de orden n^2 son famosos en las ciencias computacionales pues dejan de ser funcionales cuando n es demasiado grande, ya que el tiempo de procesamiento se vuelve verdaderamente enorme. El lector interesado puede comprobar fácilmente que las afirmaciones anteriores no son exageradas, pues por ejemplo para el sistema de Lorenz que se explica en la siguiente sección, se necesitaron algunos días para obtener la gráfica (3.6.5)

Finalmente, debe de tenerse siempre presente que la precisión finita de los cálculos y la falta de información abundante pondrán irremediablemente una cota superior a la calidad con la que el

espacio fase puede ser reconstruido, de tal forma que las trayectorias que observacionalmente pudieran ser *identicas*, matemáticamente pudieran ser diferentes,

En las secciones siguientes se harán aún más explícitos los resultados anteriores, trabajando directamente sobre el Sistema de Lorenz.

3.6 Sistema de Lorenz.

El sistema de Lorenz, propuesto por Edward Lorenz en 1963, [Lorenz;1963], corresponde a un modelo simplificado de circulación convectiva Rayleigh-Bernard en una capa de aire activada por la diferencia de temperaturas. Con este modelo Lorenz esperaba estudiar el problema de la predicción del tiempo, para lo cual, utilizando una impresora primitiva, graficó la dirección e intensidad del viento para diferentes corridas del modelo. De esta forma hizo el descubrimiento de la época: las más pequeñas diferencias en las condiciones iniciales, producían cursos completamente divergentes en los patrones de viento, los cuales presentaban diferentes comportamientos hasta llegar a una diferencia total. Esta fue la pista para el *efecto mariposa* y para la conclusión de la no predictibilidad del estado del tiempo sobre periodos de tiempo largos. Los descubrimientos de Lorenz fueron revolucionarios e iniciaron la búsqueda de lo que hoy conocemos como caos.

Lorenz observó que en ciertos casos, las trayectorias de su sistema nunca acababan en un punto fijo ni en un ciclo límite estable, y sin embargo nunca divergían a infinito. Lo que Lorenz descubrió no había sido nunca antes visto en la comunidad matemática, y no se le prestó atención durante muchos años. En la actualidad, el atractor generado por las ecuaciones de Lorenz *es el atractor extraño más conocido.*

El Modelo de Lorenz es un sistema de tres ecuaciones diferenciales no lineales acopladas, para cada una de las tres variables x, y, z que describen una circulación convectiva en sentido de las manecillas del reloj. Estas ecuaciones son:

$$\begin{aligned} \frac{dx}{dt} &= -\sigma x + \sigma y \\ \frac{dy}{dt} &= -xz + rx - y \\ \frac{dz}{dt} &= xy - bz \end{aligned} \tag{3.6.1}$$

Donde el número Prandtl σ, el número de Rayleigh (o de Reynolds) r y b son los parametros del sistema. Para $\sigma = 10, b = 8/3$ y $r > r^*$ donde $r^* \approx 24.74$ (normalmente se utiliza r =28), Lorenz

encontró numéricamente que el sistema se comporta caóticamente y sin embargo si existe un atractor, el atractor de Lorenz.

Se puede demostrar que, por ejemplo, para los valores de los parámetros $\sigma = 10, b = 8/3$ *y* $r = 14$, el comportamiento del sistema es estable, de forma que con en el tiempo tiende a uno de dos puntos fijos, normalmente llamados *C+ y C-*. Es posible comprobar, variando las condiciones iniciales del sistema, pero no el valor que se ha fijado en los parámetros, que en unos casos el sistema tiende a uno de los dos puntos fijos y en otros casos tiende al otro.

La posición exacta de los puntos fijos *C+* y *C-* varía a medida que los parámetros de control cambian (siempre que los cambios no sean muy grandes pues de lo contrario C+ y C- dejarían de ser puntos fijos). El sistema tiene otro punto fijo en el origen de coordenadas del espacio fase (*x=0, y=0, z=0*). Este punto no cambia de posición al variar los parámetros, pero es inestable. Para verlo, se deben de situar las condiciones iniciales en *(0,0,0)* y observar como el sistema no se mueve de ese punto. Sin embargo, si se eligen como condiciones iniciales, por ejemplo *(0.2,0,0)*, el sistema tardará en moverse, pero acabará escapando del origen y convergiendo a *C+* o *C-*. Debido a que todas las trayectorias que empiezan cerca del origen acaban escapando de él, este punto recibe el nombre de repulsor. Por el motivo contrario, C+ y C- reciben el nombre de atractores.

Como se ha mencionado anteriormente, *C+* y *C-* se mueven a medida que los parámetros cambian. El parámetro *r* se utiliza habitualmente para controlar este hecho. Es posible intentar por ejemplo disminuir el valor de *r* de una unidad en una unidad, y observar que los puntos fijos se acercan más y más entre sí hasta que finalmente, cuando *r* se hace más pequeño que 1, los dos puntos fijos estables se encuentran en el origen y permanecen allí. Es decir, por debajo de *r=1* el origen es un punto fijo estable, el único. De hecho, aquí se ha seguido el procedimiento en sentido contrario al que se hace habitualmente. Si el comportamiento del sistema se estudia para *r* crecientes desde 0, se observa como el punto estable en el origen se bifurca en dos puntos fijos estables y uno inestable. Este fenómeno se conoce como *bifurcación de horca*, y tiene lugar para *r=1*.

Si *r* aumenta por encima de 24.74 ($\sigma = 10, b = 8/3$), ninguno de los puntos fijos es ya un atractor de la trayectoria. Sin embargo, puede demostrarse que la trayectoria está acotada, es decir, queda *encerrada* en una región de su espacio fase, y sin embargo todos los puntos fijos existentes en esta zona acotada del espacio son repulsores!. La trayectoria no tiene ningún sitio hacia donde tender, sino que va de un lado a otro siendo repelida por los tres puntos fijos, y sin embargo acotada en el espacio. Lo único que la trayectoria puede hacer es doblarse sobre sí misma de manera muy compleja, como si se corta a sí misma una infinidad de veces, pero esto es sólo una consecuencia de las limitaciones gráficas en 2 o en 3 dimensiones (Figs. (3.6.1) y (3.6.2)). De hecho se puede demostrar, [Kathleen;1997], que una trayectoria en el espacio fase no se cortará a sí misma, por muy larga que ésta sea!.

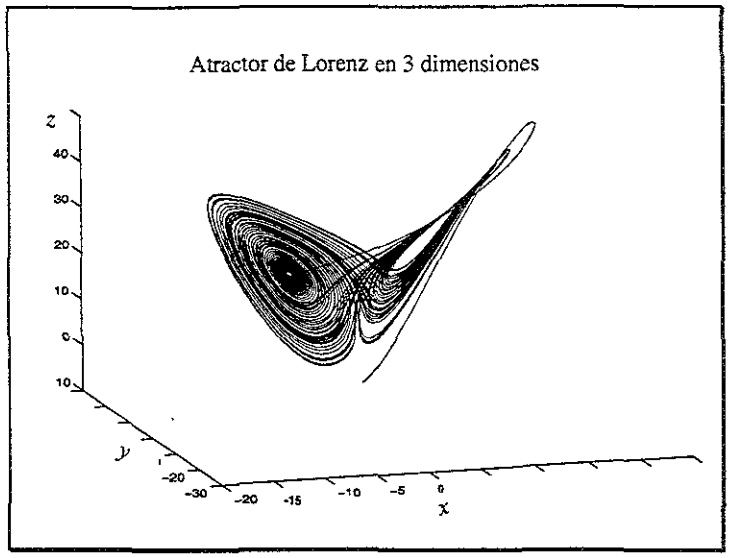


Fig.(3.6.1) Atractor de lorenz en 3 dimensiones. d=2.06 $\sigma = 10, b = 8/3, r = 28$; 4669 puntos. (x_0, y_0, z_0)=(1,3,-2)

El conjunto de puntos en los que la trayectoria se encuentra recibe el nombre de atractor. Dadas las características tan especialmente complejas de este objeto recibe el nombre de *atractor extraño*. Toda trayectoria sigue un camino en este atractor, pero el camino concreto que sigue es impredecible, y muy inestable. Hoy se sabe que la dimensión del atractor de Lorenz es 2.06, lo cual es intuitivamente claro pues éste no está en un plano, pero tampoco ocupa todo el espacio.

En la gráfica (3.6.3) se muestra la gran sensibilidad del sistema de Lorenz a pequeños cambios en las condiciones iniciales; para ello fueron generadas dos series de tiempo de la variable x mediante la integración numérica del sistema (3.4.1) usando el método de *Runge-Kutta* de 4 orden con condiciones iniciales (1,1,1) para un caso, y (1,1.01,1) para el otro.

Al inicio, dado que las condiciones iniciales son prácticamente las mismas, el comportamiento temporal es verdaderamente indistinguible, pero paulatinamente, con el tiempo, los comportamientos comienzan a separarse, para finalmente a futuro mostrar patrones completamente divergentes.

De la misma forma que en la sección anterior fueron analizadas las Integrales de Correlación para un oscilador armónico y para una serie de tiempo consistente de un ruido blanco, y en base a ellas se discutió la existencia de un atractor y de su posible dimensión, ahora ha llegado el momento de implementar esos mismos cálculos y procedimientos sobre el sistema de Lorenz.

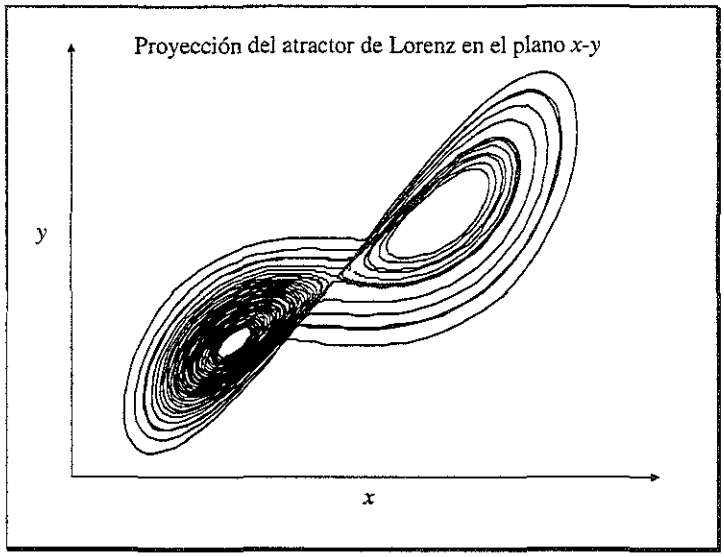


Fig.(3.6.2) Atractor de lorenz en 2 dimensiones. $\sigma = 10, b = 8/3, r = 28$; 4669 puntos. (x_0, y_0, z_0)=(1,3,-2)

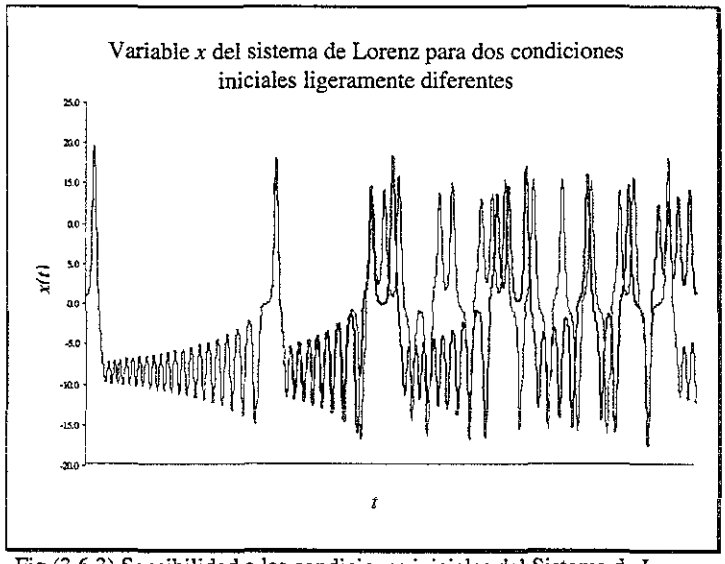


Fig.(3.6.3) Sensibilidad a las condiciones iniciales del Sistema de Lorenz

Lo que se necesita, en primer lugar, es una serie de tiempo univariada y proveniente del sistema (3.6.1), para ello, al igual que en el caso del oscilador armónico, se debe de integrar numéricamente el sistema de ecuaciones diferenciales que describen la dinámica del proceso. En este caso el sistema de Lorenz. En la figura (3.6.4) se muestra solamente la variable x de la integración numérica graficada contra el tiempo (un poco más de 16000 datos).

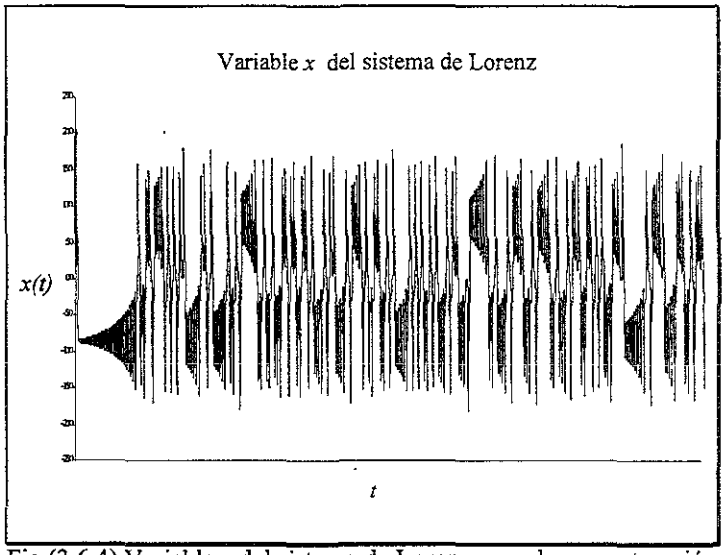


Fig.(3.6.4) Variable x del sistema de Lorenz, para la reconstrucción del atractor

Para calcular la integral de correlación, así como todos los resultados que le preceden, fueron utilizados solamente 10000 puntos, pues como se explicó anteriormente, el algoritmo para calcular $C(r)$ es muy demandante en términos computacionales. $C(r)$ fue calculada en espacios de diferente dimension, desde $m=1$, hasta $m=8$. La figura (3.6.5) muestra la integral de correlación.

Obsérvese nuevamente el comportamiento que tienen las pendientes de las curvas en la región de escalamiento, que en este caso corresponde aproximadamente al intervalo (0,2). Es sencillo distinguir en la figura que las pendientes tienden a hacerse constantes, y de acuerdo con los cálculos que fueron realizados, el número al que tienden es aproximadamente d=2.06.

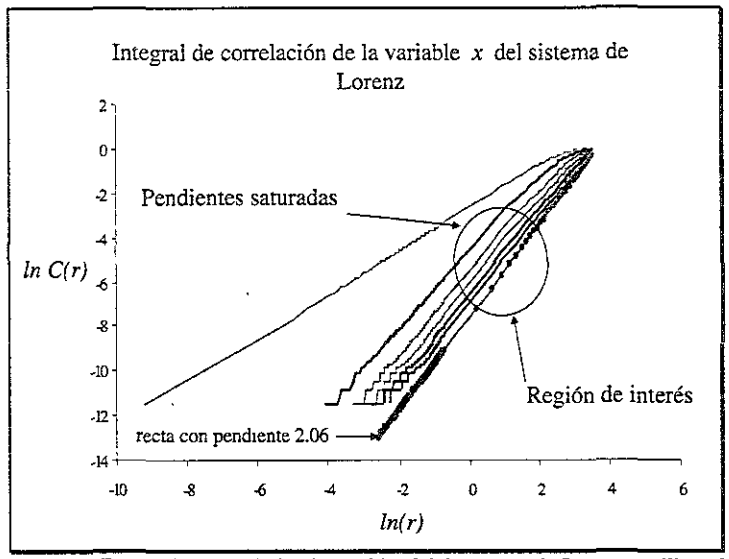


Fig.(3.6.5) Determinación de la dimensión *d* del atractor de Lorenz, utilizando sólo la variable *x*

Cuál es la relación entre el valor $d = 2.06$ encontrado por la saturación de las pendientes y la dimensión teórica, verdadera y conocida del atractor de Lorenz?. La respuesta no podía ser mas satisfactoria: *son iguales!* . Es decir con una sola medición del sistema, la variable x (aunque teóricamente pudo haber sido cualquier otra, es decir y o z), fue posible reconstruir exactamente la misma dimensión de su atractor. Y no solamente eso, también es posible reconstruir el atractor mismo en un espacio de dimensión $2(2.06)+1 \approx 5$, (5 para fines prácticos).

Dado que no es posible observar gráficamente el atractor reconstruido en su *espacio seguro de incrustamiento* que según hemos visto es de dimensión 5, y dado que en realidad, ya era sabido que con una dimensión igual a 3 es suficiente (pues el atractor real tiene dimensión 2.06, y los grados de libertad son x,y,z), es posible entonces observar el atractor reconstruido en un espacio de dimensión 3 . Fig(3.6.6)

Con estos resultados y de acuerdo a lo que fue establecido en las secciones precedentes, es posible ahora construir una red neuronal para hacer pronóstico. La información valiosa que ha sido obtenida mediante todo este proceso, consiste afortunadamente de una cota superior para el número de unidades en la capa de entrada de la red neuronal ($m=2d+1 \approx 5$). Sin embargo, también es posible intentar arquitecturas de red con capas de entrada que tengan desde d unidades (o el entero inmediato siguiente), o $d+1$, o $d+2$, etc., hasta llegar a m. La ventaja de usar menos de m unidades, d por ejemplo, tiene que ver con la rapidez en los procesos de cálculo, pero desafortunadamente muchas veces los resultados no son tan buenos como los que se obtienen con las m unidades.

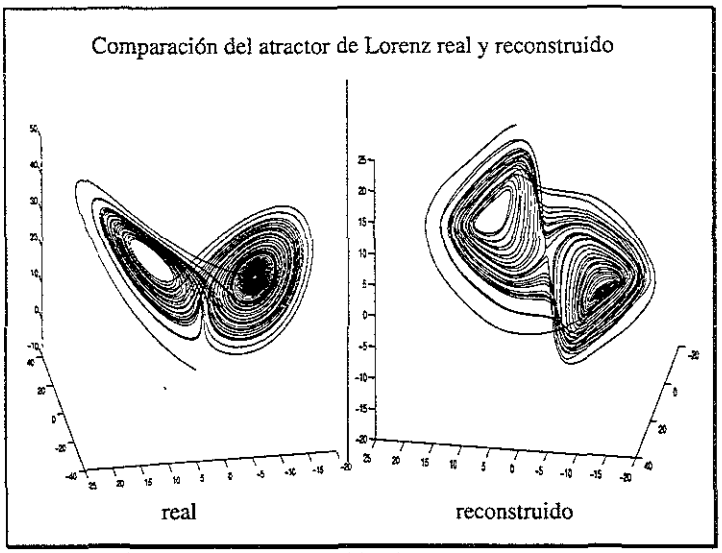


Fig.(3.6.6) Reconstrucción del atractor de Lorenz

3.7 Pronóstico del Sistema de Lorenz.

Se ha establecido anteriormente que la determinación de la dimensión *d* del atractor es muy importante puesto que ella indica el número de grados de libertad o variables *realmente significativas* en la evolución temporal de un Sistema Dinámico. Para el caso particular del sistema de Lorenz, se encontró que una cota superior para ese número de grados de libertad significativos y explicativos de su evolución temporal en el espacio fase es $m \approx 5$.

3.7.1 Definición de la arquitectura de la red neuronal.

De acuerdo a la sección 2 de la primera parte, lo que resta por hacer ahora, es construir una red neuronal multicapas y con alimentación hacia adelante que como capa de entrada tenga un número de unidades igual a la dimensión del espacio seguro de incrustamiento, 5 a saber, y en la capa de salida sólo una unidad, si es que se desea solamente hacer predicción sobre una de las variables del sistema (en este caso la variable x). *Pero sorprendentemente el alcance del método que se está proponiendo permite obtener una estimación en el espacio fase reconstruido de la dinámica completa del atractor, es decir de todas sus variables!* ((x,y,z) en este caso, o sólo de 2 de ellas: xy,xz, etc.). Esto quiere decir que es posible obtener una estimación muy acertada de dónde estará ubicado el siguiente punto del atractor dentro del espacio fase. Para ello tanto la arquitectura de la red como los conjuntos de entrenamiento serán diferentes, pues mientras que para el primer caso se requiere solamente una unidad de salida (pronóstico de la variable x al tiempo $t+1$), para el segundo caso se necesitarían 3 o 2 unidades, en función de lo que se desee. Nótese también que

solamente se pronosticará un paso adelante en el tiempo, aunque en teoría es posible predecir tantos como se necesiten (en la práctica, esta capacidad para predecir en el tiempo, estará limitada por la calidad y la cantidad de los datos).

Se han desarrollado varios criterios para determinar el número óptimo de unidades y de capas intermedias en una red neuronal multicapas, pero por desgracia dichos criterios no tienen mucha utilidad práctica, pues mientras que en algunos casos el número óptimo de unidades en la capa intermedia se conoce hasta después de haber finalizado la etapa de entrenamiento y de disponer de los errores que la red comete en la etapa de generalización! [Utans;1991], en otros más se requiere del conocimiento de la función f!! [Barron;1994]. Nótese que todo lo que se ha escrito hasta este momento es porque en el 99% de los problemas reales no se conoce el comportamiento de F_t. (F_t es la función teórica que describe el comportamiento verdadero de la serie de tiempo, que es una de las componentes del Sistema Dinámico completo).

Afortunadamente no es indispensable conocer de antemano el número exácto de unidades intermedias en la red neuronal, pues es posible iniciar el entrenamiento con *muchas* unidades e ir removiendo paulatinamente, durante o después de la fase de entrenamiento, las que sean *superfluas*. Para el sistema de Lorenz se utilizarán únicamente 5 unidades intermedias.

Por otro lado, teóricamente es sabido que una red neuronal de alimentación hacia adelante requiere solamente de una capa intermedia para poder modelar cualquier función continua [Cybenko;1989], sin embargo en la práctica se ha visto que 2 capas intermedias pueden ser más eficientes en la obtención de mejores resultados [Hartman;1991], pero por desgracia, las más de las veces el costo que hay que pagar por utilizar dos capas es demasiasdo elevado, pues la etapa de entrenamiento se vuelve mucho más difícil. En realidad, las redes neuronales con muchas capas intermedias no se justifican a menos que la superficie de desición sea muy complicada. Para el sistema de Lorenz se utilizará solamente una capa intermedia.

De esta forma, se tiene finalmente que la arquitectura de la red neuronal multicapas y con alimentación hacia adelante que modelará el sistema de Lorenz es la siguiente: 2 capas, 5 unidades de entrada, 5 unidades intermedias y 1 unidad de salida (ó 2 ó 3 dependiendo de lo que se desee).

El algoritmo de entrenamiento que será utilizado es el de retropropagación del error con gradientes conjugados. Para la etapa de entrenamiento serán utilizados los primeros 2000 datos del conjunto de 10000 con el que se trabajo en la sección anterior, y el modelo que se obtenga será probado sobre los 8000 datos restantes.

Es muy importante hacer en este momento un breve repaso de los procedimientos que hasta este momento se han seguido, así como de los resultados obtenidos, y que son de esencial importancia para el entrenamiento de la red neuronal: se desea saber para el sistema de Lorenz , sistema (3.6.1) cuál será su estado en el tiempo $t+1$, para ello las ecuaciones (3.6.1) son integradas numéricamente y se obtienen 10000 puntos, es decir, para 10000 pasos en el tiempo se conoce ahora el estado *real* del sistema en cada una de sus variables (x,y,z). De el conjunto de las 10000 observaciones (x,y,z), se selecciona un subconjunto (x,y,z) consistente de los primeros 2000 datos, que se utilizará como conjunto de entrenamiento, y el número restante de observaciones (10000-2000), será el conjunto de prueba. Tanto para el conjunto de entrenamiento como para el conjunto de prueba, solamente se utilizará una de las variables del sistema (la variable x). Dado que se conoce el estado actual del sistema (dentro del conjunto de entrenamiento) en t=2000, y el pasado en t=1999, t=1998, ..., t=1, se desea saber ahora cual será el estado futuro de la variable x, es decir su estado en el tiempo

t=2001, t=2002,t=2003,...t=10000. Nótese que para contestar esta pregunta directamente a través del sistema (3.6.1), tendría que ser encontrada una función F, solución del sistema, y ésta tendría que ser evaluada en t=2001, t=2002,..., t=10000. No hay necesidad de decir que hasta este momento no ha sido encontrada dicha función.

Otra solución, completamente evidente y que quizás al lector se le pudiera ocurrir, consistiría en continuar la solución numérica n iteraciones más (n=10001,10002,...), es decir, en lugar de detener el ciclo de cálculos en el tiempo t, dejarlo correr n ciclos más y detenerlo hasta el tiempo $t+n$. Cuál es el problema con esta solución?, pues para el sistema de Lorenz, o para cualquier otro sistema en el que se conozca un conjunto de ecuaciones como el (3.6.1), no habría ningún problema, al contrario, sería quizás más rápido y pudiera incluso ser más exacto. Pero qué es lo que pasa con la gran mayoría de los problemas reales?, se dispone también de un sistema de ecuaciones como el (3.6.1)?; evidentemente que no!, y es precisamente esa una de las grandes dificultades a las que el científico se tiene que enfrentar para obtener alguna respuesta. Para una gran variedad de problemas reales, simples o complejos, lamentablemente NO existe algún tipo de modelo que permita explicarlos, y peor aún, varios de los modelos que actualmente ya existen y se utilizan en varias disciplinas científicas, probado está que tienen errores y algunos de ellos son muy severos. Y es por todo ello que, como una alternativa, se ha preferido todo este largo proceso de análisis, y que algunos autores lo denominan como *modelación empírica.*

Por lo tanto, con los 10000 datos de la variable x (que en la vida real pudiera pensarse como 10000 observaciones de datos de lluvia), se desea saber qué sucederá con a ella en el tiempo 10001, así como también qué sucederá con todo el sistema *dentro del atractor* (es decir, con relación al ejemplo de la lluvia, cuánto lloverá en el día 10001 y cuál será el *estado general* del tiempo!! (*Aquí en realidad, aunque la teoría dice que sí el posible pensar de esta forma, en términos prácticos solamente podría ser válido si es que existiera un conjunto de variables que en forma explícita y precisa pudieran explicar el comportamiento de la lluvia.*), o más correctamente cuál será el estado de las *variables básicas* que definen la evolución del tiempo, si es que éstas se conocieran, si es que la precipitación pudiera expresarse explícitamente como función de ellas, y si es que se lograra encontrar la dimensión del atractor de la precipitación entendida como un sistema dinámico), por ello, mediante el uso de la integral de correlación $C(r)$ fue estimada la dimensión d del atractor, utilizando únicamente la variable x, y se encontró que el valor estimado d=2.06 coincidió exactamente con el valor real, por ello, de acuerdo al teorema de Takens, un espacio seguro de incrustamiento para el atractor reconstruido, es aquel que como dimensión tiene $2d$+1 grados de libertad, o en este caso una dimensión igual a $2(2.06)+1 \approx 5$. Esto quiere decir que a lo más son 5 las variables que se requieren para describir el sistema de Lorenz, aunque en realidad se sabía que con 3 era suficiente.

3.7.2 Definición de las variables de entrada para la red neuronal.

De esta forma, la pregunta natural que surge es la siguiente: Cuáles 5 variables son las que se deben de utilizar para la capa de entrada de la red neuronal, si hasta este momento únicamente disponemos de 2000 observaciones de una sola variable, la variable x del conjunto de entrenamiento? (recuérdese que se asume que y y z no se conocen. Normalmente en la gran mayoría de los sistemas reales sucede que se dispone tan sólo de alguna o algunas pocas variables en forma de series de tiempo y a partir de ellas o de ellas se desea hacer inferencia sobre todo el sistema).

De acuerdo a la sección 3.2, la respuesta está dada por el teorema de Takens, pues para formar el espacio de 5 dimensiones que se requiere dado que solamente se dispone de la variable x, es necesario encontrar el rezago de incrustamiento h, que permitirá que las nuevas variables x-h, x-$2h$, x-$3h$, x-$4h$, en conjunción con x, tengan ninguna dependencia funcional entre ellas, o en la práctica muy *poca* (justo como sucede con las componentes de cualquier espacio Euclidiano $x_1,x_2,\ldots,x_n$, en donde todas las x_i's son linealmente independientes además de ortogonales). También fue establecido en la sección 3.2 que dicho rezago óptimo de incrustamiento h está dado por el primer mínimo en la información mutua de la varialbe x. Para los 10000 datos de la variable x se encontró que dicho valor óptimo h es igual a 12, por lo que las 5 variables que se están buscando están dadas por: $x(i)$, $x(i\text{-}12)$, $x(i\text{-}24)$, $x(i\text{-}36)$, $x(i\text{-}48)$: $i = 49,\ldots,10000$.

3.7.3 Entrenamiento de la red neuronal.

De esos (10000-49) vectores de entrada en $\Re^5$ se tomarán los primeros 2000 como conjunto de entrenamiento y los restantes serán el conjunto de prueba, es decir, se entrenará la red con parejas de *entradas-salidas* como las siguientes:

$$(x(1), x(13), x(25), x(37), x(49)) \rightarrow x(50)$$
$$(x(2), x(14), x(20), x(38), x(50)) \rightarrow x(51)$$
$$\vdots$$
$$(x(2000), x(2012), x(2024), x(2036), x(2048)) \rightarrow x(2049)$$

y se pobará que tan capaz fue la red de *decifrar* la dependencia funcional entre las entradas y las salidas, con los siguientes vectores de prueba:

$$(x(2001), x(2013), x(2025), x(2037), x(2049)) \rightarrow ?$$
$$(x(2002), x(2014), x(2026), x(2038), x(2050)) \rightarrow ?$$
$$\vdots$$
$$(x(9952), x(9964), x(9976), x(9988), x(10000)) \rightarrow ?$$

3.7.4 Resultados

Establecido lo anterior queda ya completo el esquema básico para poder echar a andar la red neuronal. La figura (3.7.1) muestra el desempeño de la red en la reproducción del conjunto de entrenamiento (t=1,...,2000), y la figura (3.7.2) muestra el desempeño de la red neuronal en la predicción de la variable x al tiempo $t+1$: t=2000, 20001,..., 10000. Compárese esta última gráfica con la figura (3.3.20).

Finalmente, de acuerdo a la teoría, también es posible mediante un conjunto de entrenamiento como el siguiente, hacer directamente una predicción razonable sobre el comportamiento al tiempo $t+1$ del atractor. Fig (3.7.3)

$$(x(1), x(13), x(25), x(37), x(49)) \rightarrow (x(50),x(62),x(74))$$
$$(x(2), x(14), x(20), x(38), x(50)) \rightarrow (x(51),x(63),x(75))$$

.

.

.

$$(x(2000), x(2012), x(2024), x(2036), x(2048)) \rightarrow (x(2049),x(2061),x(2073))$$

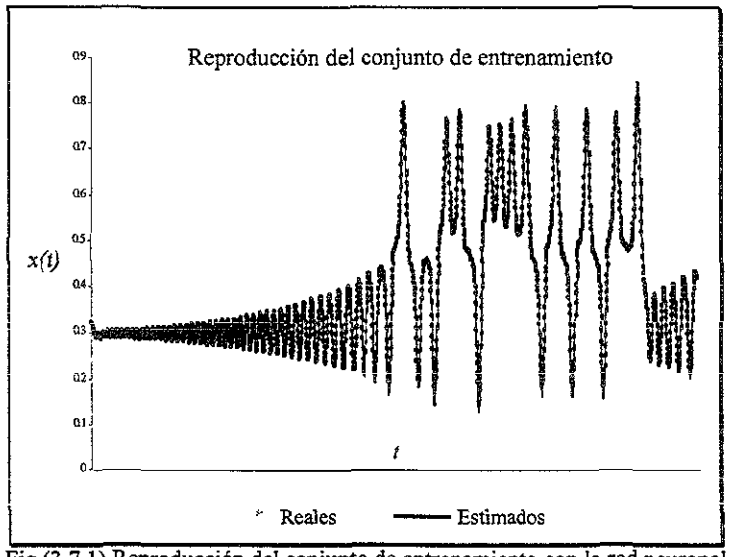


Fig.(3.7.1) Reproducción del conjunto de entrenamiento con la red neuronal

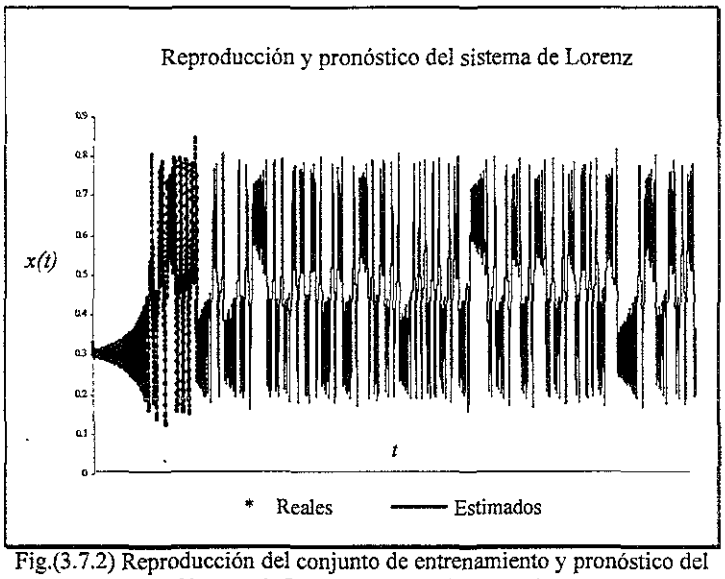


Fig.(3.7.2) Reproducción del conjunto de entrenamiento y pronóstico del Sistema de Lorenz con la red neuronal.

Es decir, para el sistema de Lorenz, teóricamente existe una función $\hat{f}$ que hace posible que $\hat{f}(x, x-h, x-2h) = F(x, y, z)$, o para mayor seguridad

$$\hat{f}(x, x-h, x-2h, x-3h, x-4h) = F(x, y, z) \tag{3.7.1}$$

donde F es la verdadera función que describe la dinámica de (x,y,z) para el sistema de Lorenz en sus espacio fase. De igual forma, teóricamente tambien existe una función $g_i(\,)$, tal que cada una de las n variables del sistema puede ser expresada al tiempo t con una ecuación del siguiente estilo

$$x_i^{(t)} = g_i(x_i^{(t)}, x_i^{(t)} - h, x_i^{(t)} - 2h, \ldots, x_i^{(t)} - mh) : \; i = 1, \ldots n; t = 1, 2, 3, \ldots \tag{3.7.2}$$

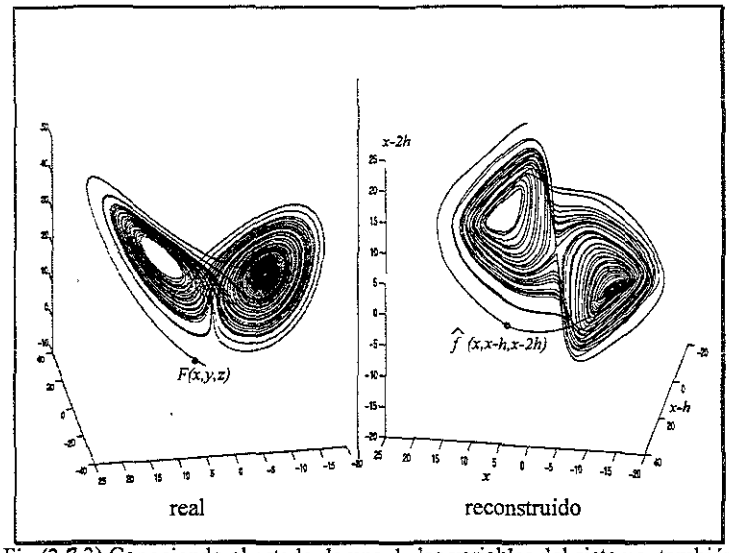


Fig.(3.7.3) Conociendo el estado de una de las variables del sistema, también es posible conocer el comportamiento que tendrán las otras.

En el siguiente capítulo, todo el mismo proceso de análisis que se siguió con el sistema de Lorenz, será ahora puesto en práctica con series de tiempo de mediciones acumuladas de lluvia en la ciudad de México.

Capítulo 4 Aplicación a series de tiempo de precipitación pluvial.

"El comandante general está perfectamente consciente de que los pronósticos del tiempo no son buenos, sin embargo los necesita para fines de planeación..."

Kenneth Arrow.

4.1 Introducción

La precipitación pluvial en todas sus formas no es sino una manifestación más en la que el agua se hace presente en el sistema conjunto tierra-atmósfera. La precipitación es un elemento muy importante y no solamente es esencial para la sobrevivencia de todos los seres vivos sobre el planeta, sino que además su presencia es fundamental en el funcionamiento interno de la capa atmosférica.

La precipitación puede ser dividida en 2 categorías generales: aquella que está asociada con corrientes de aire que ascienden lentamente y que repercuten en escalas del orden de los 1000 kilómetros (lloviznas), y aquellas que están asociadas con masas de aire locales muy inestables y que originan una gran variabilidad en la convergencia del aire dando origen a precipitaciones que repercuten en escalas del orden de decenas de kilómetros (chubascos). El primer tipo de lluvia es ligera pero persistente, cubre grandes extensiones de tierra y puede tener una duración de varios días, esta precipitación es benéfica para la agricultura puesto que su baja intensidad rara vez excede las tasas de infiltración en el suelo. Los chubascos y las tormentas, que están asociados con el segundo tipo de precipitación, son de gran intensidad, corta duración, cubren pequeñas extensiones de superficie y pueden ser benéficas o dañinas para la agricultura dependiendo de algunos factores tales como: la cantidad y la intensidad de la lluvia, la época del año, la precipitación previa, el tipo de cultivo, las características del suelo, etc.

Cada precipitación implica posibles beneficios y posibles costos para la gente. Dentro de los beneficios es posible mencionar: incremento en el crecimiento de las plantas y fijación de nitrógeno, disminución de los incendios forestales, incremento en las aguas superficiales disponibles para fines ganaderos, agrícolas, domésticos, hidroeléctricos, etc. Algunos de los costos más evidentes están relacionados con la pérdida de bienes materiales y de seres vivos, con inundaciones (o sequías si la precipitación es poca), con la pérdida o daño a cultivos, con las grandes inversiones económicas que se hacen para remediar sus daños, con la interrupción de la actividad económica, etc.

Los costos más obvios y mayormente publicitados están relacionados con fatalidades, con daños y con perdidas de propiedades; éstos normalmente aparecen en los encabezados de

los periódicos y en las noticias estelares de los programas de televisión, y sirven para justificar la *desaparición* de enormes sumas de recursos públicos cada año. Recuérdense por ejemplo, las más recientes y recurrentes sequías e inundaciones, que durante los últimos 10 años, han estado presentes en varios estados de nuestro país, así como también los recurrentes e inútiles programas *extraordinarios* de emergencia y de apoyo para *mitigar* las consecuencias de las lluvias o las sequías. Los gobiernos en turno y los funcionarios correspondientes siempre implementando programas correctivos y de emergencia en lugar de desarrollar programas preventivos y de investigación de las causas de tales fenómenos.

Es menos probable que los efectos benéficos de las precipitaciones despierten tanto la atención pública, más aún en un México en el que ya a últimas fechas, hasta el maíz y el frijol se importan de otros países.

En años recientes las frecuentes y severas sequías han llegado a ser el mayor desastre climático a lo largo del mundo, afectando en forma igualmente adversa una gran variedad de regiones. Indudablemente las sequías son uno de los principales fenómenos naturales adversos al hombre. En forma contraria a otros desastres naturales, las sequías pueden ocurrir en cualquier momento; sus efectos negativos duran, normalmente, por más de una temporada, pudiendo extenderse por varios años, o llegar, incluso, a décadas. Las sequías más severas, que son aquellas que persisten de una temporada a otra, tienen repercusiones económicas que van mas allá de un simple decremento en la producción agrícola, pues son causantes también de miseria, migraciones, inestabilidades sociales y muerte, tal y como sucede en la gran mayoría de los *minifundios* Mexicanos de temporal.

Las inundaciones, por otro lado y al igual que las sequías, son uno de los desastres naturales más destructivos y con mayor frecuencia de aparición en todo el mundo. Su presencia origina perdidas de vidas, destrucción de propiedades y del medio ambiente, así como interrupción de la actividad económica.

Entender y eventualmente predecir las variaciones en el estado del tiempo es una tarea particularmente desafiante debido a las interacciones tan complejas y a todas las retroalimentaciones que existen entre todas las diferentes componentes del sistema atmosférico. La modelación numérica de los procesos atmosféricos, oceánicos y geofísicos ha sido siempre una actividad muy demandante en todos los sentidos, a pesar de ello, sin embargo, varios métodos han sido desarrollados para pronosticar el estado del tiempo, y para pronosticar en particular el valor de la precipitación. Estos métodos están basados principalmente en técnicas estadísticas, en modelos de balance de energía, o en modelos de circulación general.

Las primeras supercomputadoras que fueron utilizadas para fines distintos a la defensa militar estuvieron dedicadas a la predicción numérica del tiempo y al procesamiento de datos geofísicos. Los grandes avances que ha habido en las capacidades y en el desempeño de las supercomputadoras ha permitido a la comunidad científica poder trabajar ahora con problemas más complejos utilizando menos aproximaciones, rejillas mas finas y modelos más reales.

En estos últimos tiempos, las más recientes arquitecturas altamente paralelas en los equipos de computo, prometen adicionalmente un incremento en la capacidad de desempeño y una mejora en la exactitud de los modelos y de los pronósticos.

4.2 Modelación de la Atmósfera. Algunos comentarios.

El clima es lo que se espera, el tiempo es lo que se tiene

La predicción numérica del tiempo, mediante el pronóstico de las variables meteorológicas que son soluciones computacionales de las ecuaciones matemáticas que *describen* algunos procesos físicos en la atmósfera, ha sido una actividad de rutina durante los últimos 30 años en varios centros internacionales de pronóstico de primer nivel. Los modelos numéricos han sido aplicados operacionalmente a la predicción de sistemas atmosféricos de gran escala, es decir, aquellos sistemas que horizontalmente llegan a cubrir distancias de varios miles de kilómetros y persisten como perturbaciones identificables del flujo atmosférico de 1 a 4 días. Motivados por el éxito de esos modelos, los meteorólogos recientemente han comenzado la experimentación con modelos numéricos de fenómenos atmosféricos que ocurren a escalas espaciales y temporales más pequeñas. Es precisamente en esta escala media y microescala en donde se presentan muchos fenómenos severos del tiempo, tales como huracanes, tornados, tormentas de nieve, lluvias de gran intensidad, contaminación severa del aire, etc.

Para modelar adecuadamente cualquier fenómeno atmosférico, tradicionalmente el primer paso ha consistido en la identificación y en el entendimiento de los procesos físicos importantes que afectan sus características durante su tiempo de vida. Posteriormente, esos procesos físicos importantes deben de ser representados por ecuaciones matemáticas que puedan ser resueltas con equipos de cómputo. Normalmente este segundo paso siempre requiere de algunas simplificaciones que tienen que ver con la idealización de ciertos procesos físicos y con la capacidad y rapidez del equipo empleado para realizar los cálculos. La representación matemática de los procesos físicos produce un conjunto de ecuaciones de predicción para la temperatura, la presión, los vientos y la humedad. Esas ecuaciones son resueltas en muchos puntos sobre un arreglo horizontal (*grid*) y a varios niveles de la atmósfera.

Un *grid* tridimensional de grandes dimensiones, podría consistir, por ejemplo, de una matriz horizontal de *50x50* aplicada a 10 niveles de la atmósfera, produciendo un total de 25000 puntos o incógnitas para el modelo. Si las 9 variables mencionadas anteriormente (temperatura, 3 fases del agua, presión, densidad y 3 componentes de la velocidad del viento) son definidas en cada punto del *grid*, entonces se deberá determinar 225000 variables para poder representar el estado de la atmósfera en un momento dado.

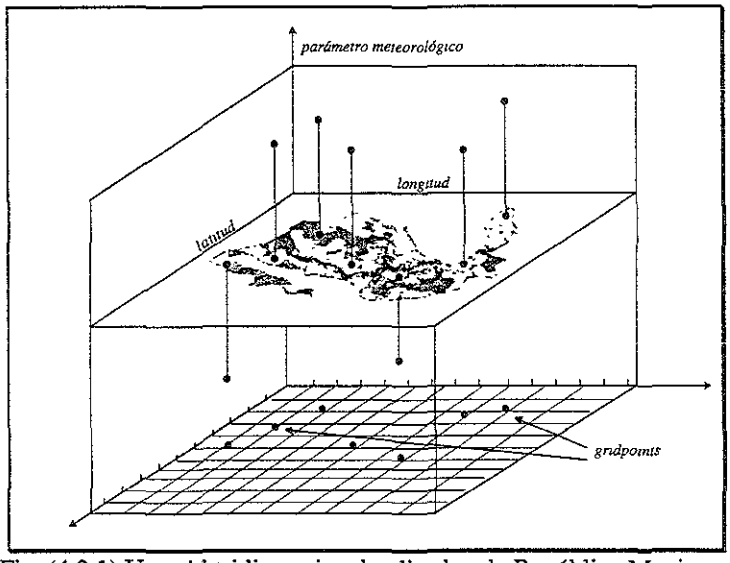


Fig. (4.2.1) Un grid tridimensional aplicado a la República Mexicana

Obtener una configuración exacta inicial de las condiciones de la atmósfera en el arreglo tridimensional es un aspecto muy importante del problema de modelación numérica global pues constituye el punto de arranque para la evolución y adecuado pronóstico del sistema. Esta fase de inicialización, que es el principal obstáculo para la modelación a pequeña escala, consiste de observaciones y de datos calculados. Obtener una observación es un proceso caro y por ello nunca hay tantas observaciones como puntos en el *grid* tridimensional. Normalmente el espaciamiento horizontal promedio que existe entre las estaciones de observación da como consecuencia que algunos sistemas importantes, pero con dimensiones horizontales menores a la resolución de la red de observación, no sean detectados y por tanto sean erróneamente representados por los modelos.

El cuarto paso en el problema de la modelación numérica tradicional de la atmósfera, consiste propiamente en la generación del pronóstico mediante la solución de las ecuaciones matemáticas generadas en el segundo paso. Dada la estructura inicial de la atmósfera (paso anterior), las ecuaciones de predicción pueden ser utilizadas para obtener el estado de la atmósfera en el futuro inmediato. Cuando este proceso es repetido una y otra vez, es posible obtener con un día o más de anticipación, un bosquejo de lo que será la estructura tridimensional de la atmósfera y del estado del tiempo.

Un pronóstico típico de 24 horas en un modelo de gran escala requiere aproximadamente 100 soluciones sucesivas para cada ecuación en cada punto, esto querría decir, por ejemplo en la estructura del *grid* mencionado anteriormente, que se necesitarían aproximadamente 22.5 millones de soluciones para conocer el estado del tiempo para el día siguiente!. Es claro que solamente los equipos de cómputo más poderosos

(supercomputadoras con decenas de procesadores) son capaces de resolver tan sofisticados pero demandantes modelos, y aún en ellos, el tiempo de procesamiento llega a alcanzar varias horas e incluso días.
Varios métodos han sido desarrollados para pronosticar las precipitaciones. Estos métodos están basados principalmente en técnicas estadísticas que utilizan datos históricos [Barry & Perry;1981]; [Jauregui;1995]. Algunos modelos estadísticos utilizan correlaciones rezagadas entre las variables meteorológicas regionales en superficie a gran escala, de tal forma que las relaciones más significativas son trasladadas a ecuaciones de regresión múltiple para predecir los valores mensuales, o con otra periodicidad, de la precipitación [Klein & Bloom;1987]. Algunos de los problemas más frecuentes a los que se enfrentan estos modelos son: dificultad para *adivinar* cuales son las *conexiones* más importantes, bajas correlaciones entre las variables y falta de información.

En forma alternativa también se utilizan los modelos numéricos (son códigos computacionales basados en expresiones discretizadas de las soluciones de las ecuaciones que gobiernan el movimiento de un fluido, incluyendo parametrizaciones de procesos físicos (convección, turbulencia, etc.) a escalas más pequeñas) como una forma alternativa de predecir el estado del tiempo, entre ellos es posible mencionar por ejemplo, los modelos de balance de energía, tal como el *modelo termodinámido de Adem* que ha sido utilizado para predecir la temperatura en superficie y las precipitaciónes [Adem;1982]. Sin embargo, la dificultad con este tipo de modelos surge en el momento en el que las ecuaciones diferenciales parciales (*continuas*) que describen el comportamiento (*idealizado*) de los fenómenos que fueron considerados como los más importantes por el modelador, son traducidas a su versión discreta, pues normalmente el esquema numérico de solución (diferencias finitas, por ejemplo) presentará, en mayor o en menor medida, algún tipo de *inestabilidad numérica*, inherente al propio esquema numérico, o a errores de punto flotante. Este tipo de inestabilidades numéricas han recibido una gran atención en la literatura especializada de modelación atmosférica, pues sus atroces efectos son capaces de *destruir completamente* la dinámica de un fluido en tan solo unas cuantas iteraciones.

El desarrollo de los Modelos de Circulación General (MCG) constituye un paso fundamental en la moderna predicción numérica del tiempo. Los Modelos de Circulación General han llegado a ser una herramienta muy importante en el estudio de las circulaciones a gran escala y del clima global, siendo capaces hoy en día de reproducir el ciclo anual o de simular eventos anómalos tales como aquellos asociados con *El niño* y con *La Oscilación del Sur* [Philander;1990], sin embargo, y esto es muy importante, esos modelos por ellos mismos no son capaces de determinar la variabilidad regional y local (microescala) de la atmósfera debido, entre otras cosas, a su poca resolución espacial y al hecho de que la atmósfera sea un flujo turbulento y que, por lo tanto, no exista hasta el momento, un conjunto de ecuaciones primitivas capaces de modelar su comportamiento preciso. Existen también los denominados modelos anidados de área limitada que cubren alguna zona de interés a escala regional, con resoluciones espaciales típicas de 25 a 50 kms. en la horizontal y de 100 a 500 mts. en la vertical; y los modelos de alta resolución, con una resolución espacial de 2 a 20 kms. en la horizontal y de 10 a 100 mts. en la vertical, para realizar pronósticos locales y de corto plazo; pero también en estos casos

estos tipos de modelos no son capaces de detectar, y por lo tanto hacen una interpretación errónea, de los sistemas meteorológicos que ocurren a escalas menores.
Durante las últimas 3 décadas, las simulaciones numéricas del estado del tiempo a través de modelos numéricos han llegado a ser la herramienta más poderosa para el pronóstico de las condiciones de la atmósfera. El pronóstico numérico ha alcanzado una etapa de desarrollo tal, que los modelos globales permiten ahora obtener pronósticos incluso a escalas planetarias. Estos modelos son normalmente corridos por centros de pronóstico internacionales, tienen una resolución espacial típica de aproximadamente 100 a 200 kilómetros en la horizontal y de 500 a 1000 metros en la vertical, y a través de ellos es posible realizar pronósticos con un rango de hasta 10 días. Los resultados de estos modelos (matrices de puntos o campos escalares) son enviados a subcentros o centros de menor tamaño en donde se corren modelos a una escala más limitada. En estos subcentros, los resultados de los modelos globales son utilizados como condiciones iniciales o como condiciones de frontera para correr modelos anidados de área limitada que cubren alguna zona de interés a escala regional, por ejemplo Europa. Los modelos anidados tienen una resolución típica de 25 a 50 kilómetros en la horizontal y de 100 a 500 metros en la vertical y con ellos es posible realizar pronósticos con un rango de hasta 3 días.

Nuevamente la salida del modelo de área limitada puede ser utilizada como condición inicial o de frontera para correr modelos de alta resolución (resolución espacial de 2 a 20 kilómetros en al horizontal y de 10 a 100 metros en la vertical) para realizar pronósticos locales y a corto plazo.

En el futuro, los modelos de alta resolución podrían llegar a ser más y más operacionales debido, entre otras cosas, a la creciente disponibilidad de equipos de cómputo de alto desempeño y a la mejora en las redes y en los equipos de observación de la atmósfera.

4.3 Limitaciones de los modelos actuales de predicción del estado del tiempo

Todos los teoremas son verdaderos,
Todos los modelos son incorrectos,
Todos los datos son inexactos,
¿Qué es lo qué podremos hacer?

El flujo de un fluido está gobernado por ecuaciones no lineales que tienen una antigüedad de aproximadamente 160 años. El aire de la atmósfera y el agua de los océanos son fluidos, sin embargo, la aplicación de aquellas ecuaciones a la dinámica de la atmósfera empezó aproximadamente hace 110 años, y no ha sido sino hasta hace relativamente poco tiempo, que se ha tenido éxito en relacionar las soluciones de las ecuaciones a fenómenos atmosféricos reales; esto último se ha debido principalmente a las dificultades que son inherentes a las aplicaciones físicas: los datos están muy dispersos, el sistema de

coordenadas básico es no inercial (esta en rotación), y el flujo es turbulento en muchas escalas (el sistema es caótico). Como consecuencia de lo anterior las Ciencias Atmosféricas se han mantenido esencialmente como una ciencia empírica el mayor tiempo de su existencia.

Existen algunas soluciones analíticas para las ecuaciones que describen los flujos geofísicos que son afectados por un sistema de referencia en rotación, y también existen, evidentemente, para los flujos en un sistema de referencia que no está en rotación. Si adicionalmente el flujo presenta ondas y vórtices, es posible también encontrarle algunas soluciones analíticas. La gran mayoría de los modelos *convencionales* de pronóstico del tiempo están construidos con las ecuaciones anteriores.

Los flujos de la atmósfera contemplados a gran escala evolucionan, aparentemente, en forma uniforme y determinística (de tal forma que es posible modelarlos utilizando alguna de las ecuaciones mencionadas anteriormente), sin embargo, desde una perspectiva más fina, es posible descubrir que los flujos geofísicos están impresionantemente llenos de ondas y de vórtices.

La solución matemática para un flujo que contiene ondas y vórtices aleatorios, denominado *flujo turbulento*, no ha sido encontrada todavía [Brown, 1990]. La atmósfera es un flujo turbulento y por lo tanto esto quiere decir, que no existe un conjunto de ecuaciones matemáticas que permitan representar su comportamiento preciso fielmente (macro y micro). Es por ello, en conjunción con otras limitaciones ya antes mencionadas (falta de datos, uso de ecuaciones incorrectas, inestabilidad numérica del algoritmo), que la gran mayoría de los actuales modelos numéricos del tiempo no pueden hacer pronóstico a una escala espacial de muy alta resolución.

Por lo tanto, si el campo de interés es el estado del tiempo y el clima a gran escala, será suficiente trabajar solamente con los modelos numéricos de circulación general actualmente disponibles. Sin embargo, en la predicción del estado del tiempo a nivel regional, se ha encontrado que los detalles de los efectos de la escala pequeña eventualmente llegan a ser importantes. En los conceptos de la teoría del caos, las perturbaciones más pequeñas imaginables pueden cambiar la naturaleza de la solución a gran escala. Las capacidades computacionales están creciendo rápidamente, pero los modelos numéricos para dominios de gran escala, que tengan además tamaños de *grid* suficientemente pequeños para incluir la turbulencia de la microescala, no están contemplados en el futuro cercano, pues podrían quizás, más que contribuir a la generación de un pronóstico exacto, producir patrones de pronóstico totalmente divergentes debido a pequeñas variaciones en la etapa de inicialización del modelo de una corrida a otra. Aquí cabe aclarar que en el *Centro Europeo de Pronóstico*, al percatarse de esta situación, y para evitar los efectos del caos (un sistema caótico es altamente sensible a pequeñas variaciones en las condiciones iniciales) se corren múltiples simulaciones de sus modelos variando las condiciones iniciales en una vecindad adecuada, y tomando como pronóstico final la *trayectoria promedio* o *banda más representativa* de todas las corridas. A este proceso de pronóstico se le conoce como

ensembles, y es una aplicación directa para remediar los efectos del caos. Figuras (4.3.1) y (4.3.2). A este tipo de gráficas también se les conoce como *gráficas spaghetti.*

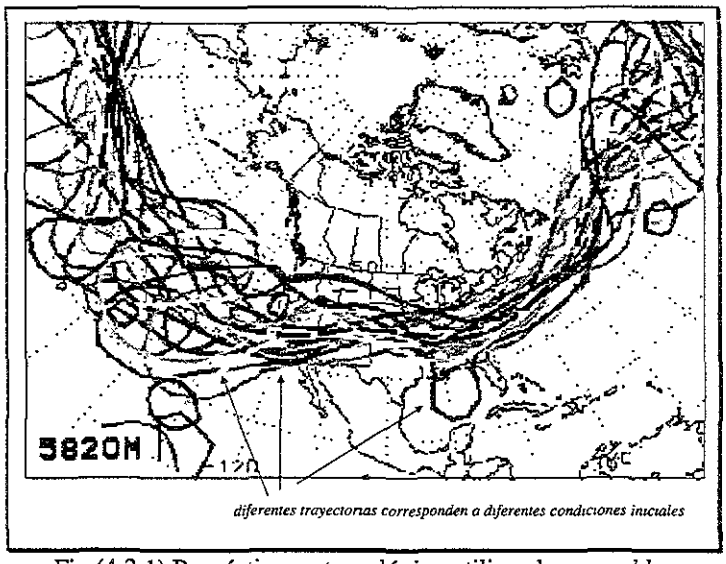


Fig (4.3.1) Pronóstico meteorológico utilizando *ensembles*

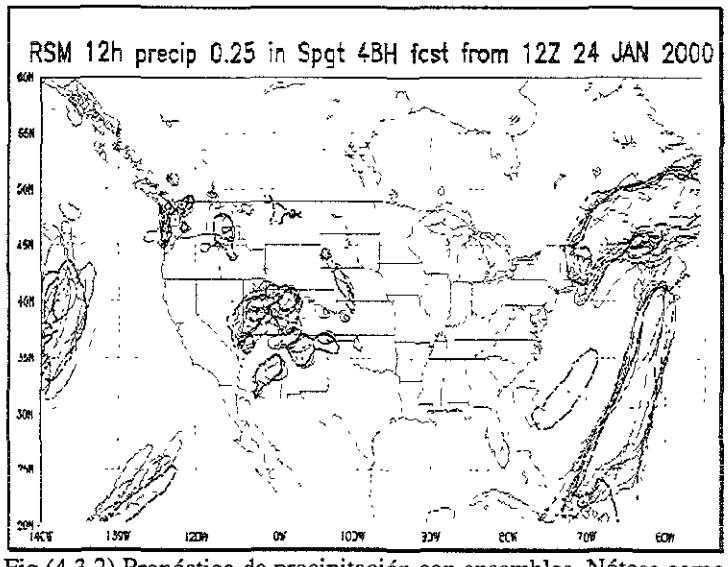


Fig.(4.3.2) Pronóstico de precipitación con ensembles. Nótese como pequeñas variaciones en las condiciones iniciales del modelo (el mismo modelo), pueden producir patrones de pronóstico totalmente diferentes.

Entonces surge la pregunta, qué herramienta o qué modelo, puede ser utilizado para hacer predicción local del estado del tiempo?. Una posible respuesta, aunque no es la única y no es absolutamente completa, está dada por los modelos que se obtienen directamente a

partir de los datos (no se basa en las ecuaciones que describen la dinámica de la atmósfera, ni tampoco se reduce al viejo esquema estadístico de análisis de las series de tiempo o de regresiones lineales), es decir, la *modelación empírica.*

4.4 Modelación de la lluvia con una red neuronal

No podía ser más evidente la necesidad de una herramienta alternativa, en este caso un *Modelo Conexionista (Red Neuronal)*, que se basa exclusivamente en los datos disponibles y en las *relaciones ocultas* que existen entre ellos, y que es capaz de hacer predicciones del tiempo a nivel local (microescala) con tanta validez como validez tengan las estaciones de observación generadoras de los datos que lo alimentan, es decir, se estaría hablando de una resolución espacial de 1mm a 1000 mts. en la horizontal y en superficie [WMO; 1993].

Los modelos generados con redes neuronales no son tan *triviales* como los tradicionales modelos estadísticos basados en correlaciones y en regresiones lineales (como si el comportamiento de las variables atmosféricas estuviera regido por simples relaciones lineales), o en la engorrosa y muchas veces poco precisa metodología de Box y Jenkins. Los modelos generados con redes neuronales tampoco hacen uso de las ecuaciones tradicionales de la dinámica de fluidos que al ser *simplificaciones y aproximaciones* de la realidad, a veces muy severas, ya llevan consigo un considerable margen de error, el cual tiende a incrementarse en la medida en la que el esquema numérico de solución no sea capaz de evolucionar establemente (*estabilidad numérica del algoritmo*).

Las redes neuronales hacen uso de los datos y de *las relaciones ocultas* que existen entre ellos, y como ahora ya se sabe, normalmente un conjunto *unidimensional* de datos, proveniente de un sistema dinámico *n-dimensional*, es suficiente para modelar eficazmente el comportamiento futuro del fenómeno. Los datos no son abstracciones, aproximaciones o suposiciones, los datos son la realidad del fenómeno, son su manifestación ante nuestros sentidos. Es cierto, para determinadas aplicaciones, la calidad de los datos deja mucho que desear pero, tanto en el esquema estadístico como en el esquema dinámico, también se hace uso de ellos.

Al hacer uso solamente de la realidad del fenómeno a través de los datos, las redes neuronales no proporcionan resultados válidos únicamente bajo tal o cual suposición o abstracción de la realidad. Los modelos neuronales generan resultados tan reales y tan confiables como reales y confiables hayan sido los datos que lo alimentaron.

Adicionalmente, las redes neuronales, cuando menos las que aquí se presentan, no necesitan de varios días de cálculos computacionales para generar, a lo más, un pronóstico con una resolución muy baja (de varios kilómetros en la horizontal). Por el contrario, a una red neuronal entrenada con un conjunto de datos procesados con la metodología que aquí se presenta, le bastan solamente algunos minutos para hacer un

pronóstico de precipitación acumulada (1 semana, 15 días, o un mes) con una muy alta resolución.

¿Qué más se podría necesitar?, desde luego la precisión y la viabilidad operativa en el pronóstico, pero eso es algo que en tanto el modelo y los conceptos que lo sustentan no sean implementados en un marco de pronóstico real, no podrá ser optimizado, por ahora solamente se garantiza, y estamos seguros de ello, una precisión teórica.

Para determinar el modelo de pronóstico a partir de una red neuronal, es indispensable, en primer lugar, tener definidos un conjunto de *comportamientos entradas-salidas* (patrones), que desempeñen el papel de conjunto de entrenamiento. Como es sabido, este conjunto de entrenamiento deberá de ser presentado a la red neuronal un número suficientemente grande de veces hasta que ésta logre *descifrarlo y reproducirlo* dentro de ciertos márgenes de error. Una vez que la red ha aprendido el conjunto de entrenamiento, tiene lugar la etapa de generalización o *pronóstico* sobre un conjunto de datos desconocidos por la red.

Evidentemente, para poder trabajar con la red neuronal es indispensable haber determinado previamente su arquitectura , es decir, el número de capas y el número de unidades en cada una de ellas. Como se ha explicado anteriormente, el algoritmo de entrenamiento que será utilizado es el de retropropagación del error con la versión no lineal de los gradientes conjugados.

Para la determinación de la arquitectura de la red deberá recordarse el resultado establecido por Cybenko: *una red neuronal con una única capa intermedia es suficiente para modelar cualquier función continua*, es decir, la red neuronal que será utilizada para la modelación y pronóstico de los niveles de lluvia en alguna estación de monitoreo de la ciudad de México, tendrá solamente dos capas: la capa intermedia y la capa de salida. Recuérdese que la capa de entrada no se considera como una capa formal.

El número de unidades en cada capa de la red es un aspecto central en la definición de una buena arquitectura. Recuérdese la discusión en las secciones (3.6) y (3.7). Para esta aplicación nuevamente será utilizada una única unidad en la capa de salida, aunque como se discutió en la sección (3.7), también es posible utilizar 2 o 3, o en general n unidades de salida, tantas como periodos hacia adelante se esté interesado en predecir.

El número de unidades intermedias será definido en función del número de unidades de entrada, y como es sabido, este número de unidades de entrada en forma óptima está determinado por la integral de correlación de los datos de lluvia pues ésta nos indica la dimensión d del atractor del sistema dinámico (evolución temporal de los niveles de lluvia en este caso), por lo que de acuerdo al teorema de Takens, dicho número óptimo de unidades de entrada será a lo más de $2d+1$ unidades.

Así pues, para poner en marcha todo el proceso de pronóstico de precipitación pluvial, se necesita como primer paso, el conjunto de datos que describen este fenómeno.

4.5 Los datos para el pronóstico

Alimentemos un modelo con basura,
Y no obtendremos otra cosa que basura

La red de estaciones manuales (convencionales) de observación de fenómenos atmosféricos de superficie en México, está integrada por aproximadamente 80 nodos. Existen también estaciones automáticas de superficie y estaciones en las que se realizan observaciones de altura (se lanzan globos sonda para monitorear el estado de la atmósfera en las capas superiores). Figura (4.5.1) . Para esta aplicación se utilizarán los datos de lluvia diaria correspondientes a 10 años de observaciones (recuérdese que un aspecto muy importante de esta metodología de análisis tiene que ver con la cantidad de datos; sección (3.5)), del periodo comprendido del 1 de enero de 1990 al 31 de diciembre del año 1999. La estación de observación es una estación meteorológica automática de la *red GASIR* y está ubicada en la Ciudad Universitaria de la Ciudad de México, aunque este mismo proceso de análisis es igualmente válido para cualquier otra estación de observación. Los datos de lluvia diaria fueron proporcionados por la Gerencia de Aguas Superficiales del Servicio Meteorológico Nacional. La figura (4.5.2) muestra la serie de tiempo de los datos diarios.

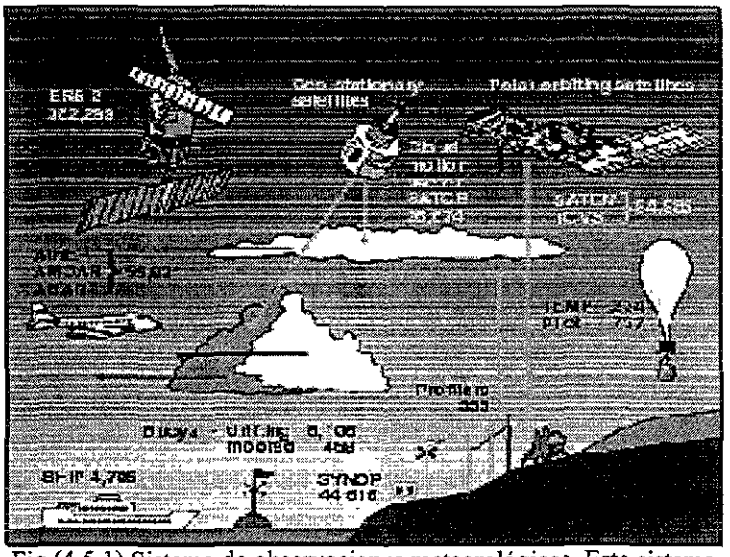


Fig.(4.5.1) Sistema de observaciones meteorológicas. Este sistema proporciona datos de superficie y de altura en tierra y mar.

Es sencillo observar que una serie de tiempo de datos de lluvia diaria contendrá demasiados *huecos* correspondientes a los días de no lluvia; esto no sería en realidad un problema, pues podría asignárseles el valor cero, de no ser porque la serie de tiempo también contiene días con lluvia cero, es decir, en la misma serie de tiempo hay días que

no tienen asignado algún valor pues aparentemente no llovió, pero también hay días que tienen asignado el valor cero!. Cuál es la diferencia entre un día de no lluvia y un día con lluvia cero?. Aparentemente, y en términos prácticos, ninguna, por ello para este análisis a ambos se les asigno el valor cero. Sin embargo, los problemas no terminan aquí, pues como es sabido, por lo regular las observaciones meteorológicas, y muy en particular las observaciones automáticas de lluvia, contienen un margen de error que a veces resulta ser demasiado grande.

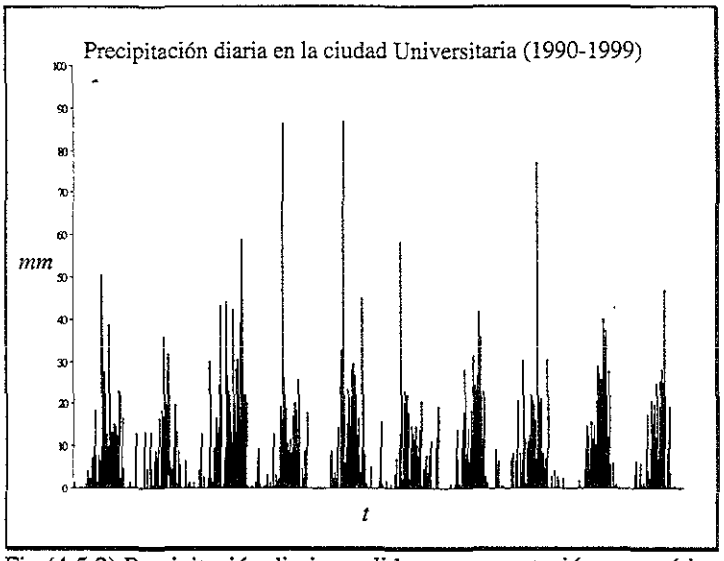


Fig.(4.5.2) Precipitación diaria medida con una estación automática de observación en superficie

Para tratar de evitar cualquier tipo de inconsistencia con los datos de lluvia en su presentación diaria, se optó por *suavizar* la información, y para ello, se decidió trabajar con acumulados semanales, quincenales y mensuales, es decir, cada dato de la nueva serie de tiempo corresponderá ahora a la cantidad de lluvia que se presentó durante una semana completa, o 15 días o un mes, en la estación de observación. Los acumulados semanales, quincenales y mensuales son mostrados en la sucesión de gráficas (4.5.3)-(4.5.5).

Para las secciones siguientes, incluyendo la modelación y pronóstico con la red neuronal, se trabajará solamente con los datos acumulados quincenales. Esta decisión fue tomada básicamente por la mala calidad de los datos. Resulta casi impensable el intentar desarrollar un *modelo sencillo* con datos diarios, no solamente por la complejidad de la función (fig.(4.5.2)), sino porque es sabido además que en su presentación diaria los datos de lluvia suelen contener niveles de ruido muy considerables.

Para los acumulados semanales se intentaron varias arquitecturas de red, y se obtuvieron de hecho algunos modelos que reproducían bastante bien el conjunto de entrenamiento,

pero desafortunadamente en la etapa de generalización, en la que se utilizan datos no conocidos por la red, ésta tendía a cometer errores demasiado severos. Este problema podría deberse básicamente a dos causas:

1. La red más que aprender del conjunto de entrenamiento y *entender* el comportamiento subyacente a ellos (lo cual le permitiría generalizar sobre ese tipo de características), simplemente está *memorizando* toda la información que se le presenta, pues invariablemente, para los acumulados semanales, el tiempo de convergencia resultó ser demasiado grande.
2. El ruido en los datos no es un *ruido sistemático,* es decir, esas pequeñas o no tan pequeñas variaciones que existen con respecto a los valores verdaderos de lluvia, han ido cambiando a lo largo de los 11 años de que se dispone información, pues normalmente tanto los equipos de observación como las personas encargadas de procesar y almacenar los datos han ido cambiando.
   En otras palabras, lo que la red neuronal aprendió en el conjunto de entrenamiento, es un conjunto de datos que tiene un nive de error diferente al nivel de error contenido en el conjunto de prueba, y a decir verdad, bajo estas condiciones una predicción exitosa se vuelve prácticamente imposible.

Es por ello que para la obtención del modelo de pronóstico, se decidió trabajar con los acumulados quincenales, pues a pesar de que los datos *fuente* (diarios) siguen conteniendo los mismos niveles de ruido, al acumularlos sobre periodos de tiempo más amplios, los niveles de error tienden a hacerse más homogéneos, dando como resultado una función a aproximar (fig.(4.5.4)) más sencilla. Desde luego que los pronósticos que se obtengan con este modelo deberán de entenderse también como la lluvia acumulada para los próximos quince días.

Lo anterior no significa, sin embargo, que una red neuronal no sea capaz de pronosticar lluvia diaria o lluvia semanal, lo que sí quiere decir, es que si los datos son *erróneos*, por más procesamiento que se haga de la información, no habrá red neuronal o algoritmo alguno capaz de modelarlos exitosamente. *Alimentemos un modelo con basura y no obtendremos otra cosa que basura*

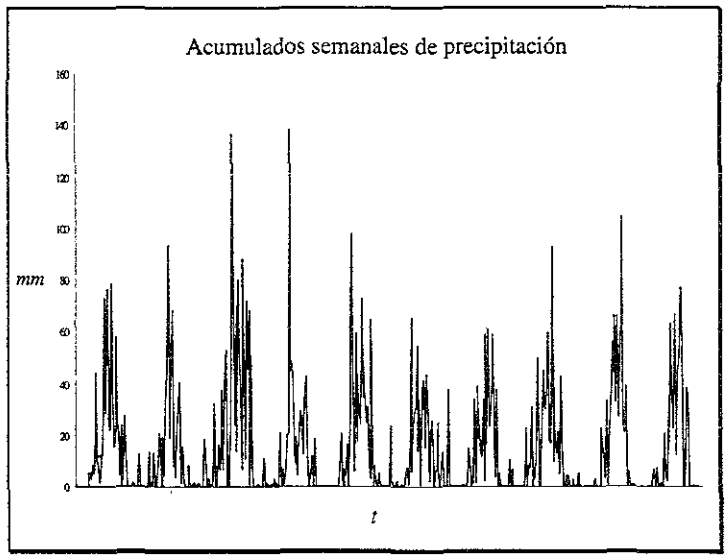


Fig.(4.5.3) Precipitación acumulada semanalmente

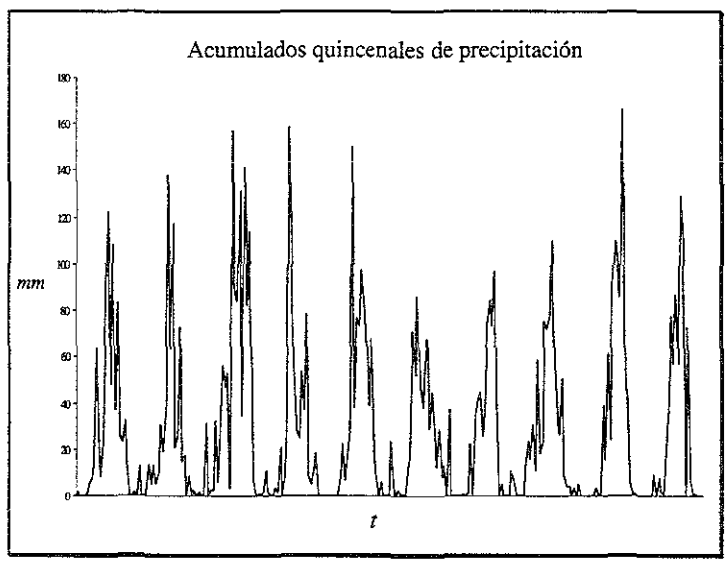


Fig.(4.5.4) Precipitación acumulada quincenalmente

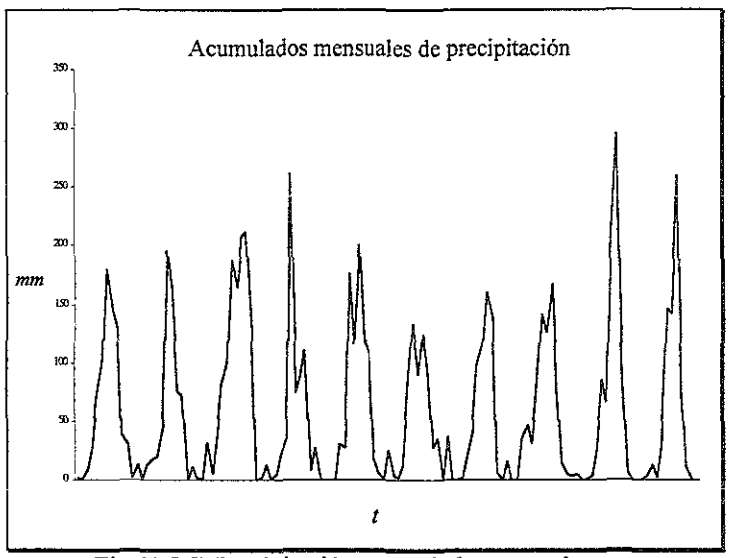


Fig.(4.5.5) Precipitación acumulada mensualmente

4.6 Integral de correlación de los datos de lluvia.

Como es sabido, la integral de correlación de una serie de tiempo está estrechamente relacionada con la dimensión del atractor del sistema dinámico (sección (3.4)).

Para los datos de lluvia acumulada quincenalmente, la integral de correlación se muestra en la figura (4.6.1). De acuerdo con la discusión presentada en la sección (3.5), es posible proponer en este caso, como la dimensión d del atractor, el número $d = 4.8$; esto querría decir, de acuerdo al teorema de Takens, que la región segura de incrustamiento tendrá una dimensión igual a $2d+1$, es decir, $m \approx 10$ ó $m \approx 11$. Por lo tanto, la capa de entrada para la red neuronal tendrá a lo más $m = 11$ unidades.

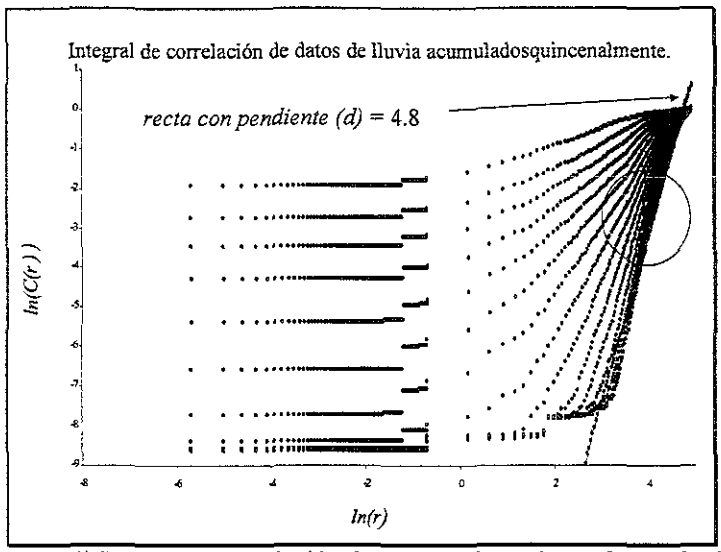


Fig.(4.6.1) Integral de correlación de acumulados quincenales de lluvia. La gráfica sugiere que el valor de d es 4.8

4.7 Arquitectura de la red neuronal para el pronóstico de lluvia.

En la sección anterior se determinó que una cota superior para el número de grados de libertad significativos en la evolución temporal del sistema dinámico, está dada por el valor de m=11 . Esto quiere decir que la capa de entrada de la red neuronal contendrá a lo más 11 unidades, aunque como se explicó en la sección (3.5) también es posible intentar arquitecturas con d unidades, o d+1, d+2, ... etc. (es decir, arquitecturas con 5,6,7,... unidades). En particular, para los acumulados quincenales, se decidió utilizar una arquitectura con solamente 10 unidades de entrada.

Para la capa intermedia de la red neuronal, fueron utilizadas 30 unidades, esta cantidad se determinó en base a numerosas corridas (prueba y error), pues como se mencionó anteriormente, no existe hasta el momento algún criterio que sugiera *a priori* el número óptimo de unidades ocultas. De esta manera, se decidió finalmente que una arquitectura adecuada para la red estaría dada por: 2 capas (de alimentación hacia adelante), con 30 unidades en la capa intermedia y una única unidad de salida. Como patrones de entrada se utilizaron vectores de dimensión m=10.

Si se revisa nuevamente la sección (3.7), se encontrará que para poder realizar el proceso de pronóstico en forma exitosa, es indispensable aún, la determinación del rezago de incrustamiento h, el cual, según se ha dicho, está dado por el primer mínimo en la información mutua de la serie de tiempo de datos de lluvia. Figura (4.7.1). (En realidad el

valor de h fue determinado desde la sección 4.6, pues su valor es indispensable para el calculo de la integral de correlación).

De acuerdo a la gráfica (4.7.1), es posible observar que el valor óptimo para h es 2 (2 períodos de 15 días en este caso). Este valor de h en conjunción con el valor de m elegido, permiten finalmente poder definir en forma precisa los patrones de *entrada-salida* para el entrenamiento de la red neuronal.

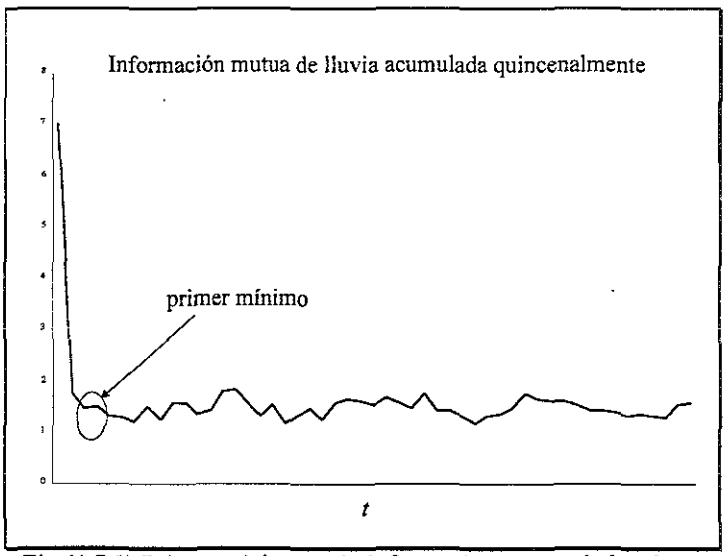


Fig.(4.7.1) Primer mínimo en la información mutua de los datos acumulados de lluvia. La gráfica indica que el primier mínimo se alcanza en h=2

4.8 Pronóstico de la lluvia acumulada.

El caos no nos impide hacer predicciones, simplemente nos restringe el horizonte de tiempo efectivo sobre el cual dichas predicciones pueden ser hechas.

El conjunto de entrenamiento para la red neuronal consiste de aproximadamente 240 datos (figura (4.5.4)), con los cuales es posible formar los siguientes vectores de entrada y salida

Entradas *salidas*

$$(x_i, x_{i+2}, x_{i+4}, x_{i+6}, \ldots, x_{i+18}) \longrightarrow (x_{i+19}) : \quad i=1,2,3,\ldots 240$$

Es importante mencionar que los datos del conjunto de entrenamiento deben de ser *normalizados*, es decir, deben de ser transformados linealmente del rango de valores original (observado) al intervalo (0,1), que corresponde al rango de la función de transferencia (sigmoide) utilizada en cada una de las unidades de la capa intermedia y de la capa de salida de la red neuronal (sección (2.9.2)).

El conjunto de prueba consiste de aproximadamente 17 datos (todas las quincenas del 1 de enero del año 2000 al 15 de septiembre del mismo año). Una vez que la red neuronal ha sido entrenada, es posible que ella haga una reproducción del conjunto de entrenamiento, figura (4.8.1), y es posible también que sea capaz de generalizar y proporcionar un *pronóstico* (sobre el conjunto de prueba), para el tiempo $t+n$: donde t=259 y n=1,2,3,...,17. Figura (4.8.3).

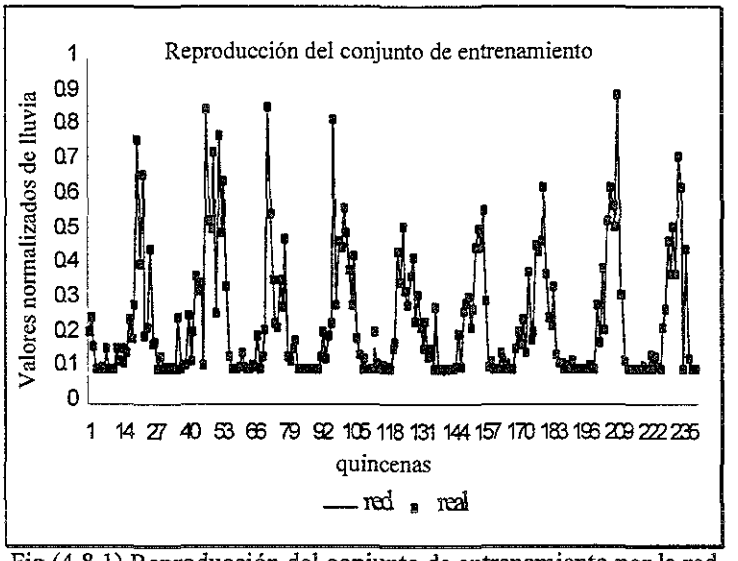


Fig.(4.8.1) Reproducción del conjunto de entrenamiento por la red neuronal. 240 datos quincenales correspondientes a las quinenas de 1990 a 1999.

Podemos observar, figura (4.8.2), el comportamiento de la función error, ecuación (2.9.7). Estas *curvas de aprendizaje* son obtenidas durante el proceso de entrenamiento de la red

neuronal y nos indican el error total que la red está cometiendo para reproducir todo el conjunto de prueba (dado que estamos trabajando con la versión batch). Nótese como dicha función error paulatinamente converge hacia el valor ε =0.05. Ver paso 2 de la sección (2.11).

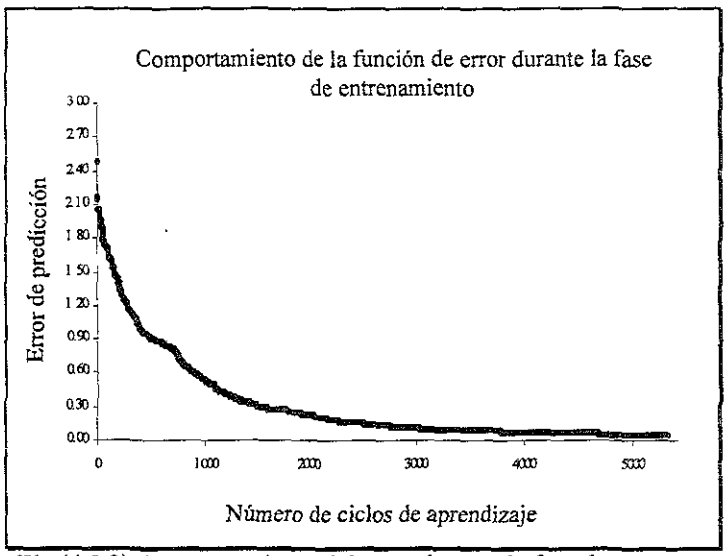


Fig.(4.8.2) Comportamiento del error durante la fase de entrena – miento de la red neuronal. Los errores convergen a ε = 0.05.

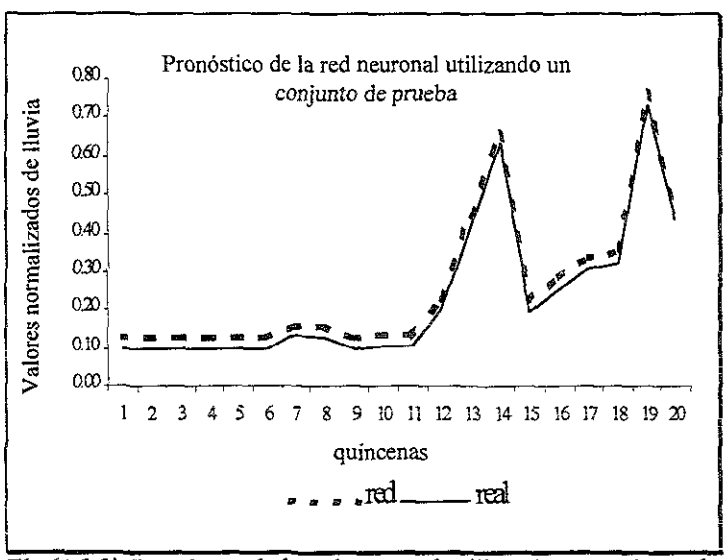


Fig.(4.8.3) *Pronóstico* de la red neuronal utilizando un conjunto de Prueba (pronósticos de las primeras 20 quincenas del año 2000).

Lo que podemos observar en las figuras (4.8.1)-(4.8.3) es que después del proceso de entrenamiento, la red neuronal logra descifrar el conjunto de prueba y logra también aprender a generalizar, de acuerdo a las características del conjunto de entrenamiento. Esto en general no es algo sencillo de conseguir, pues el obtener buenos resultados en la etapa de entrenamiento no garantiza buenos resultados en la etapa de prueba (*calibración del modelo*). Anteriormente se mencionó que con datos acumulados semanalmente, después de la etapa de aprendizaje también se obtuvieron excelentes reproducciones del conjunto de entrenamiento, pero desafortunadamente, durante las etapas de prueba y de pronóstico con la red neuronal entrenada, los resultados que se obtuvieron no fueron nada satisfactorios. Sin embargo en este caso, con los datos quincenales, el desempeño de la red en la etapa de prueba es muy bueno y ello nos proporciona un margen considerable de confianza para los futuros pronósticos que pudieran hacerse con el modelo desarrollado.

No obstante, es importante mencionar que la red entrenada solamente será capaz de predecir acumulados quincenales de lluvia en la estación meteorológica de donde provinieron los datos para la etapa de entrenamiento, y asumiendo que *los patrones dinámicos* subyacente en ellos no cambian a lo largo del tiempo.

Un verdadero pronóstico con la red entrenada puede ser hecho a partir de la segunda quincena del mes de septiembre del año 2000. Aqui es importante resaltar el gran valor de las *pruebas ciegas*, pues además de que contribuyen a saber que tan *robusto* es el modeo obtenido, permiten también obtener n periodos de pronóstico (alimentando el modelo con sus propias salidas)

Finalmente, otra forma de obtener n periodos como pronostico se consigue con un conjunto diferente de entrenamiento, y por supuesto con una arquitectura diferente para la red neuronal. Los patrones de salida del conjunto de entrenamiento y la capa de salida de la red deberán de tener n componentes.

Cuarta Parte. Conclusión

Durante los últimos 40 años la modelación de la atmósfera y de los fluidos geofísicos, así como la predicción del tiempo han evolucionado en forma vertiginosa. Esta evolución en los modelos y en las técnicas de modelación se ha desarrollado en forma paralela con el perfeccionamiento gradual de los equipos de cómputo y de los dispositivos meteorológicos para la observación de las condiciones del tiempo. Sin embargo, a pesar de disponer hoy en día de todo este gran cúmulo de mejores equipos y procedimientos de predicción (sin mencionar toda la experiencia que a lo largo de estos últimos años se ha ganado), este conjunto de nuevas y poderosas herramientas de pronóstico es insuficiente no sólo para producir una predicción exitosa del estado del tiempo a un nivel de microescala, sino que incluso, los pronósticos a escala global del comportamiento atmosférico siempre llevan consigo un considerable margen de error, el cual tiende a magnificarse a medida que el periodo pronosticado se hace mayor.

El problema no son únicamente los modelos, así como tampoco lo son únicamente los datos, el verdadero problema son tanto los modelos como los datos. Ambos, tanto conjunto de ecuaciones o *estructuras internas* (cuando el modelo se construye a partir de los datos) en el caso de los modelos, como los conjutos de información o series de tiempo (para inicializar o construir del modelo), en el caso de los datos, están invariablemente contaminados por errores de observación y por errores propios de la etapa de modelado .

En otras palabras, lo que hasta hoy la comuniad científica ha sido capaz de producir, es un conjunto de *modelos imperfectos* que por si fuera poco se alimentan de datos también imperfectos. El resultado por supuesto tenía que ser, y no hay que dedicar demasiado tiempo a la reflexión, un *pronóstico imperfecto* del estado del tiempo. Quizás más correctamente tendríamos que decir un *pronóstico del tiempo que es doblemente imperfecto.* En este caso estamos hablando de pronósticos del tiempo, pero por desgracia la misma situación se reproduce hacia todas las disciplinas científicas donde se utilizan modelos y se utilizan datos para interactuar con ellos, es decir, estamos hablando prácticamente de toda la ciencia.

Qué es más imperfecto un modelo (*dinámico*) que se construye en base a las ecuaciones que describen el comportamiento de los flujos atmosfericos (primera fuente de error), que como se mencionó anteriormente son fluidos turbulentos (caóticos), y que después dicho modelo continuo para poder ser resuelto tiene que ser expresado en su versión discreta (segunda fuente de error) a través de alguna técnica numérica de solución (diferencias finitas por ejemplo), y que finalmente, para obtener algún resultado, dicha versión discreta del modelo debe de ser alimentada con datos cargados de una gran incertidumbre (tercera fuente de error), o es más imperfecto un modelo (*empírico*) que se construye, como aquí se ha explicado, a partir exlcusivamente de los datos observados (primera fuente de error), y que se *calibra* posteriormente con datos también observados (segunda fuente de error), y que finalmente, dicho modelo validado debe de ser alimentado con datos cargados también de una gran incertidumbre (tercera fuente de error)...?. No lo sabemos, no sabemos cual de las dos alternativas de modelado sea *más imperfecta*, y no ha sido el objetivo de este trabajo el enarbolar o el menospreciar las capacidades de cualquiera de ellas. Nadie se atrevería a negar los excelentes logros y las grandes capacidades de los modelos dinámicos actuales, los cuales son capaces de simular con asombrosa precisión los ciclos anuales y

la evolución global de la atmósfera, pero tampoco estamos de acuerdo en descartar algunas otras técnicas de modelación que, visto está, se desempeñan mucho mejor en donde los primeros son absolutamente incapaces de producir algún resultados sensato.

Más correctamente tendríamos que decir, para el caso que aquí nos ocupa, que cada técnica tiene sus ventajas y cada técnica tiene sus limitaciones, es por ello que de acuerdo a lo que mencionamos desde las primeras secciones de este trabajo, creemos que la mejor alternativa consiste en complementar las capacidades de uno, con las bondades del otro.

Es en este marco de cooperación y de trabajo conjunto, de validación y de complemento de los resultados para la creación de un producto mayor y de mejor calidad, en el que pretendemos que, ya para estas fechas, esta común y sencilla investigación sea colocada. Los resultados, como se explicó en la seccion 4.8, son muy satisfactorios y complementan perfectamente, en general, las predicciones globales o a grandes escalas de los modelos dinámicos, y en particular se constituye como una herramienta hasta hoy inexistente en los centros nacionales de pronóstico del tiempo.

Bibliografía

Thompson, P.D, 1957: Uncertainty of initialstate as a factor in the predictability of large scale atmospheric flow patterns. Tellus,9, 275-295.

Farmer, J.D., and J.J. Sidorowich, 1987: Predicting chaotic time series. Phys Rev. Lett.,59, 845-848

Takens, F,1981: Detecting strange attractors in turbulence. Dynamical systems and Turbulence, Lecture Notes in Math. 898. Springer, 366-381.

Fraedrich,K., 1986: Estimating the dimensions of weather and climate attactors. J. Atmos. Sci.,432, 419-432.

Reifman, J., and Vitela, J.,1994: Accelerating learning of neural networks with conjugate gradients for nuclear power plant applications. Nuclear Technology, 106, 225-241.

Vitela, J., and Reifman, J., 1997: Premature saturation in Backpropagtion Networks: Mechanism and necesary conditions. Neural Networks, 10, 721-735.

Werbos, P., 1974: Beyond regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph.D. thesis, Harvard University, Cambridge, MA.

Rumelhart, D.E., 1986: Learning representations by backpropagations errors. Nature, 323, 533-536.

Cybenko, G. 1989: Approximation by superpositions of a sigmoidal function. Mathematics of control, signals, and Systems, Vol. 2 No.4, 303-314.

Barron, A.R. 1994: Aproximation and estimation bounds for Artifitial Neural Networks. Machine Learning 14, 115.

Utans, J. 1991: Selecting Neuran Networks Architecure via the prediction risk. Proc. First Intl. Conf.on AI appl. On wall street, IEEE press, Los Alamitos CA.

Hartman, E. 1991: Predicting the future: Advantages of semilocal units. Neural Comput. 3, 566.

Rezgui, A. 1990: The effect of the slope of the activation function on the backpropagation algorithm. Proc. Int. Joint conf. Neural Networks, Washington, D.C., January 15-19, 1990, Vol. 1, p. 707.

Fletcher, R. 1964: Function Minimization by Conjugate Gradients. Comput. J., 7, 149.

Lorenz, E.N.,1963: Deterministic nonperiodic flow. J. Atmos. Sci., 20, 130-141.

Grassberger, P. And I. Procaccia, 1983: Characterization of strange attractors. Phys. Rev. Lett., 50, 346-349.

Kaplan Daniel, 1995. Understanding Nonlinear Dynamics. Springer-Verlang. New York, Inc.

Ott Edward, 1993. Chaos in Dynamical Systems. Cambridge University Press.

Argyris John, 1994. An exploration of chaos. North Holland.

Hao Bai-Lin, 1984. Chaos. World Scientific Publishing.

Hertz, J. 1991. Introduction to the theory of Neural Computation. Adisson Wesley publishing company.

Cover, T. 1991. Elements of Information Theory. Wiley Series in Telecomunications.

Devaney Robert L., 1989. An introduction to Chaotic Dynamical Systems, 2nd edition. Addison-Wesley Publishing Co.

Kathleen T. Alligood, et. al, 1997. Chaos, and Introduction to Dynamical Systems. Springer.

Hao Bai-Lin, 1987. Directions in chaos, Vol. 2. World Scientific Series on Directions.

Brown Robert A., 1990. Fluid Mechanics of the Atmosphere. Academic Press.

Sumner, G. 1988. Precipitation process and Analysis. EE.UU. Jonh Wiley and Sons LTD.

Garcia F. (editor), 1997. Numerical simulations in the Environmental and Earth sciences. London. Cambridge Univ. Press.

Edwin Kessler, 1986. Thunderstorm Morphology and Dynamics. U
niversity of Oklahoma Press.

Edwin Kessler, 1983. Thunderstorm in Human affairs, 2nd edition. University of Oklahoma Press.

World Meteorological Organization (WMO), 1993. Technical Note 158.

World Meteorological Organization (WMO), 1993. Compendio de apuntes para la formación de personal meteorológico. Volumen Meteorología general, Volumen Climatología.

World Meteorological Organization (WMO), 1995. Climate System Monitoring.