



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
ACATLÁN**

**PRONÓSTICO DE DONACIONES DE CÉLULAS
MADRE POR CORDÓN UMBILICAL DEL CENTRO
NACIONAL DE TRANSFUSIÓN SANGUÍNEA**

TESIS

**QUE PARA OBTENER EL TÍTULO DE:
LICENCIADA EN MATEMÁTICAS APLICADAS Y
COMPUTACIÓN**

**PRESENTA:
RUBÍ ÁNGELES PÉREZ**

**ASESORA: DRA. MARÍA DEL CARMEN GONZÁLEZ
VIDEGARAY**

OCTUBRE 2016

SANTA CRUZ ACATLÁN, NAUCALPAN, ESTADO DE MÉXICO



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Quiero agradecer a todas las personas que me apoyaron en el transcurso de la carrera, que contribuyeron a mi formación como persona y profesionista.

A mis padres, por su apoyo incondicional, siempre dándome una palabra de aliento, por impulsarme a superarme, por cuidarme y sobre todo por darme la vida. Mejores padres no pude haber tenido.

A mi asesora María del Carmen, que estuvo apoyándome en este proceso, por compartirme sus conocimientos. Gracias por su paciencia para la realización de este trabajo.

A mis sinodales, en especial a los profesores Gabriel Delgado y Jaime Vergara por su ayuda y disponibilidad.

A la Facultad de Estudios Superiores Acatlán por darme la oportunidad de formar parte de una institución con un alto grado académico.

A mis amigos: Oscar, Dulce, Karina, Selene, María, Josué, Jonatan por su ayuda y amistad, lo que permitió que la etapa de la universidad fuera más agradable.

A todos los profesores de la carrera por contribuir a mi formación profesional, por compartir su conocimiento, experiencia y sus buenos consejos.

*“ Los sueños no se hacen realidad
sin muchos fracasos y esperanzas”*

-Walt Disney

Índice general

Introducción	7
Objetivo	9
Justificación	9
1 Pronóstico de las células madre usando Box-Jenkins	10
1.1 Metodología Box-Jenkins	10
1.1.1 Verificar que la serie sea estacionaria	11
1.1.2 Identificación	11
1.1.2.1 Interpretación de la ACF	12
1.1.2.2 Interpretación de la PACF	14
1.1.2.3 Periodograma	15
1.1.2.4 Periodograma integrado	16
1.1.3 Estimación	16
1.1.4 Diagnóstico	17
1.1.5 Pronósticos	18
1.2 Modelos ARIMA	18
1.2.1 Concepto de estacionaridad	18
1.2.2 Modelos estacionarios	19
1.2.2.1 Modelos autorregresivos	19
1.2.2.2 Modelos de medias móviles	20
1.2.2.3 Modelos mezclados	20
1.2.3 Modelos no estacionarios	21
1.2.4 Modelos con variación estacional	22
1.2.4.1 Modelos puramente estacionales	22
1.2.4.2 Modelo SAR(P)	22
1.2.4.3 Modelos SMA(Q)	22
1.2.4.4 Modelos SARIMA(P,D,Q)	23
1.2.5 Modelos multiplicativos	23
1.3 Caso de aplicación: predicción de número de donaciones de células madre por cordón umbilical	24
1.3.1 Células madre de cordón umbilical	24
1.3.2 Proceso para obtener las células madre del cordón umbilical	25

1.3.3	Beneficios de las células madre obtenidas de la sangre del cordón umbilical	25
1.3.4	Donaciones de células madre por cordón umbilical en el mundo	26
1.3.5	Donaciones de células madre por cordón umbilical en México .	28
1.4	Modelado por la metodología Box-Jenkins	29
1.4.1	Identificación de un modelo	32
1.4.2	Estimación de los parámetros	34
1.4.3	Diagnóstico del modelo	35
1.4.3.1	Significación estadística de los parámetros	35
1.4.3.2	Condiciones de estacionaridad e invertibilidad	35
1.4.3.3	Análisis de los residuales del modelo	35
1.4.3.4	Datos faltantes y atípicos	37
1.4.3.5	Comparación de modelos	38
1.4.3.6	Elección del modelo	39
2	Pronóstico con otro modelo matemático	41
2.1	Qué son las redes neuronales artificiales	41
2.1.1	Elementos básicos de una RNA	42
2.1.2	Funciones de activación	43
2.2	Arquitectura de una RNA	47
2.2.1	Clasificación según el número de capas	48
2.2.2	Clasificación según tipo de conexiones	48
2.2.3	Clasificación según tipo de información	49
2.3	Aprendizaje y recuerdo	49
2.3.1	Fase de aprendizaje o entrenamiento	50
2.3.1.1	Algoritmos de aprendizaje	50
2.3.1.2	Resumen de algoritmos más utilizados	53
2.3.1.3	Terminología	53
2.3.1.4	Algoritmo de entrenamiento Backpropagation o retropropagación	55
2.3.2	Evaluación del aprendizaje	56
2.4	Modelos de RNA aplicados para predicción	57
2.4.1	Red neuronal perceptrón multicapa	57
2.4.1.1	Características del perceptrón multicapa	57
2.4.1.2	Componentes del perceptrón multicapa	57
2.4.2	Red neuronal adaline/madaline	58
2.4.3	Red neuronal de función de base radial (RBF)	60
2.5	Metodología para predicción con RNA	60
2.6	Modelado con RNA	63
3	Comparación de modelos	67
3.1	Análisis comparativo	67
3.1.1	Ajuste y predicción	67
3.1.2	Gráficas de ajuste y predicción de los modelos	67

	6
3.2 Discusión del análisis	68
3.3 Pronósticos del modelo seleccionado	69
Conclusiones	71
Anexos	73
Bibliografía	87

Introducción

Una de las maravillas que existen en el universo es el ser humano, ya que puede pensar, crear y crecer. Todo esto se puede llevar a cabo gracias a la estructura de su cuerpo, que lo conforman órganos, los cuales a su vez están formados por células. Nuestro cuerpo está formado aproximadamente de cien billones de diminutas células: óseas, sanguíneas, entre otras. Sin embargo, existen células que sobresalen por tener la capacidad de convertirse en cualquier otro tipo de célula, de regenerarse a sí mismas y regenerar órganos, estas reciben el nombre de células madre.

Las células madre son células biológicas que se encuentran en la mayoría de los organismos multicelulares y que tienen la capacidad de dividirse o auto replicarse a sí mismas para producir más células madre (a través de la mitosis) por periodos indefinidos durante toda la vida de un individuo y que, bajo las condiciones apropiadas o señales correctas del microambiente, se pueden diferenciar para dar origen a diversos tipos de células especializadas de distintos linajes con características y funciones especializadas como miocitos, neuronas o hepatocitos. Existen diferentes tipos de células madre. Dentro de los mamíferos hay dos grandes tipos de células madre: las células madre embrionarias o fetales que se encuentran en los órganos del feto y células madre adultas que se encuentran dentro de un organismo desarrollado como el de los niños, así como de adultos.

Son muy valiosas como herramienta de investigación, para ser utilizadas habitualmente en tratamientos médicos contra algunas enfermedades, pero en un futuro podrían ser útiles para tratar una amplia variedad de enfermedades. Los pacientes que reciben alguno de estos tipos de trasplante tienen una mejor atención, esto ha ayudado a que los médicos puedan determinar eficazmente qué pacientes tienen mayor posibilidad de obtener mejores resultados tras haber recibido algún de trasplante de células madre.

El Centro Nacional de Transfusión sanguínea (CNTS) proporcionó los datos de transfusión de sangre de cordón umbilical desde el comienzo de su campaña en el año 2003 hasta el 2014. Esta investigación utilizará dichos datos para realizar un pronóstico y conocer el futuro próximo de las donaciones de sangre de cordón umbilical.

Para pronosticar el número de donaciones de células madre por cordón umbilical que se tendrán en el CNTS en los próximos dos años, se utilizará la metodología Box-Jenkins para realizar el pronóstico. La cual consiste en aplicar filtros a la serie de tiempo hasta obtener residuales cuyo comportamiento sea semejante al ruido blanco, para que no afecten considerablemente al resultado final. La metodología de Box-Jenkins se realiza de forma iterativa tomando siempre en cuenta dos principios fundamentales, el primero es el principio de parsimonia que nos dice que no debemos agregar al modelo elementos que no estén justificados, el

segundo principio es el de mejoramiento iterativo que nos permitirá ir mejorando el modelo de forma sucesiva.

También se obtendrá un modelo por medio de Redes Neuronales Artificiales (RNA). La cual consiste en dos fases, la fase de entrenamiento y la fase de prueba. La fase de entrenamiento consiste en usar un conjunto de datos o patrones de entrenamiento para determinar los pesos (valores pequeños entre 0 y 1) que definen el modelo de la RNA, esto se hace de forma iterativa, de acuerdo con los valores de entrenamiento. La fase de prueba consiste en utilizar un segundo grupo de datos o patrones diferentes a los del entrenamiento, que permite controlar el proceso de aprendizaje.

El modelo obtenido de una RNA se comparará con el modelo que se utilizará de la metodología Box-Jenkins, para saber cuál de los dos modelos es el que nos da un mejor pronóstico.

El trabajo se encuentra dividido de la siguiente manera:

- Capítulo 1. Pronóstico de las células madre usando Box-Jenkins
Se explica de manera breve la metodología Box-Jenkins y los modelos ARIMA. Así como también se exponen las especificaciones de caso de aplicación y encontrar un modelo adecuado para pronosticar, haciendo uso de los software Statgraphics [16] y SPSS [17].
- Capítulo 2. Pronóstico con otro modelo matemático
Se explica de manera breve las principales características de una RNA y su funcionamiento, y particularmente, algunos modelos usados para series de tiempo. También se describe el proceso que se usa en la aplicación de una RNA, para obtener un modelo para pronosticar las donaciones de células madre del cordón umbilical del CNTS, utilizando el software estadístico R Studio [18].

- Capítulo 3. Comparación de modelos
Se analizan los modelos que se seleccionan de los capítulos 1 y 2, para comparar y seleccionar el mejor modelo que pronostique. Se realiza un análisis comparativo de ajuste y de los pronósticos de los modelos ARIMA y de la RNA.
- Conclusiones
Finalmente se muestran las conclusiones obtenidas en la investigación.

Objetivo

El resultado de esta investigación permitirá conocer el futuro de las próximas donaciones de células madre por cordón umbilical que se tendrán en el CNTS y observar si habrá un crecimiento de donaciones.

Pronosticar el número de donaciones de células madre del cordón umbilical que habrá en los próximos dos años en el CNTS usando la metodología Box-Jenkins y obtener un modelo que nos permita dar un mejor pronóstico de las próximas donaciones de células madre por cordón umbilical.

Justificación

Los avances en la medicina son muy notables, esto se puede ver en las investigaciones de las células madre que se han realizado en los últimos años para combatir enfermedades. Por su gran utilidad hay países que realizan algún tipo de trasplante de células madre, los países que cuenta con el más alto número de donantes de algún tipo de trasplante de célula madre es Estados Unidos y España. México también lleva a cabo este tipo de trasplante, cuenta con el Centro Nacional de Transfusión sanguínea (CNTS), Instituto Mexicano del Seguro Social (IMSS) entre otras instituciones.

En un futuro, este método podría reemplazar a otros para curar o sanar una amplia variedad de enfermedades. Para llevar a cabo esto existen instituciones que se encargan de recibir algún tipo de donación de células madre para poder someter a los pacientes a un tratamiento y mejorar su calidad de vida.

El interés científico de la presente investigación, se basa en obtener un modelo matemático que nos permita realizar un pronóstico para conocer el futuro próximo de donaciones de cordón umbilical que se pueden tener en los próximos años en el CNTS, para conocer si existe o no, un crecimiento de las donaciones de cordón umbilical.

Capítulo 1 Pronóstico de las células madre usando Box-Jenkins

1.1 Metodología Box-Jenkins

Para explicar la metodología del procedimiento de un pronóstico con Box-Jenkins haré uso de lo que emplea la Dra. González Videgaray[13], que consiste en extraer los datos y separarlos de la parte no predecible.

La serie de tiempo pasa por filtros, hasta obtener residuales no predecibles, el comportamiento de dichos residuales tienen poca influencia en el resultado final y se deben comportar como ruido blanco. La metodología hace uso de tres filtros lineales: el autorregresivo, de integración y de medias móviles. También ofrece recursos para evaluar y mejorar los diversos modelos tentativos, de una manera sucesiva que permiten determinar lo que falta o sobra en un modelo.

El proceso iterativo de Box-Jenkins para construir modelos lineales de series de tiempo se basa en dos grandes principios:

- *El principio de parsimonia.* Consiste en elegir siempre el modelo más sencillo que sea suficientemente representativo de los datos.
- *El principio del mejoramiento iterativo.* Consiste en partir de un modelo sencillo y factible, al cual se harán mejoramientos sucesivos, hasta llegar a un modelo satisfactorio.

La figura 1.1 representa el funcionamiento del método de Box-Jenkins.

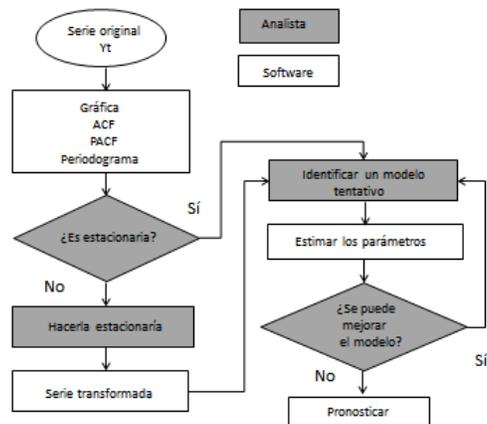


Figura 1.1. Metodología Box-Jenkins [13].

De acuerdo con los dos principios antes mencionados, el objetivo será identificar un modelo tentativo para las observaciones. Posteriormente se construirán otros modelos, con el fin de escoger el más apropiado.

1.1.1 Verificar que la serie sea estacionaria

Como primer paso se debe verificar que la serie sea estacionaria y tratar de eliminar la variación estacional, en caso de que exista. Para ello se sugiere lo siguiente:

- Revisar la posible existencia de heteroscedasticidad, en caso de que exista, estabilizar la varianza a través de una transformación apropiada.
- Observa si la serie presenta algún tipo de tendencia. En caso de que exista, hacer las diferencias ordinarias que se requieran.
- Determinar si la serie contiene alguna variación estacional, si esto ocurre, debe probarse la aplicación de diferencias estacionales.

1.1.2 Identificación

Después de hacer la serie estacionaria, se debe identificar un modelo tentativo. Para ello, es necesario contar con una serie de elementos básicos, que permitan la identificación de un modelo:

- La función autocorrelación (ACF)
- La función autocorrelación parcial (PACF)
- El periodograma
- El periodograma integrado

Para efectuar la identificación de un modelo, es necesario realizar lo siguiente:

1. Observar la gráfica de la ACF muestral, si es decreciente infinita o se trunca; en este último caso, en qué valor ocurre esto.
2. Observar la gráfica de la PACF muestral, si es decreciente infinita o se trunca, en este último caso, en qué valor ocurre esto.
3. Analizar tanto el periodograma como el periodograma integrado, junto con ACF y PACF, para determinar si aún existe variación estacional, después de haber aplicado alguna diferencia estacional. Por medio de los valores numéricos del periodograma se puede constatar el orden de la variación estacional. En dado caso de que algún valor resalte, se debe analizar si se ha descubierto algo o tal vez se trate de algún error en el procedimiento.
4. Con los resultados de los paso anteriores, se podrán construir varios modelos tentativos.

En los pasos 1 y 2 se cuentan con pruebas de hipótesis para cada autocorrelación y autocorrelación parcial.

1.1.2.1 Interpretación de la ACF

Para determinar si se debe considerar que la ACF es decreciente infinita o se trunca en un cierto valor de k , se utiliza en primera instancia una inspección visual de la gráfica (correlograma), auxiliada por un intervalo de confianza que permitirá probar la siguiente hipótesis para cada una de las autocorrelaciones teóricas ρ_k :

$$H_0 : \rho_k = 0$$

$$H_1 : \rho_k \neq 0$$

Si la hipótesis nula no se rechaza, se considera que la autocorrelación correspondiente al intervalo k es estadísticamente insignificante, esto quiere decir, que no hay autocorrelación entre Y_t y Y_{t-k} . El estadístico de prueba se obtiene de la siguiente forma:

$$t_{rk} = \frac{r_k - 0}{s_{rk}} \tag{1.1}$$

donde r_k es el coeficiente de autocorrelación simple para un retraso de k periodos.

La varianza del estimador r_k está dada por:

$$Var(r_k) \approx \frac{1}{N}[1+2(\rho_1^2+\rho_2^2+\dots+\rho_q^2)]\forall k > q \quad (1.2)$$

El estimador de la varianza será:

$$Var(r_k) = s_{r_k}^2 \approx \frac{1}{N}[1+2(r_1^2+r_2^2+\dots+r_q^2)]\forall k > q \quad (1.3)$$

El estadístico que se usa de prueba sigue aproximadamente una distribución de probabilidad t de Student, con tantos grados de libertad como datos disponibles, para un nivel de confianza de $(1-\alpha)100\%$.

La metodología Box-Jenkins requiere de contar por lo menos con ochenta datos, para que sea posible detectar tendencia y variación estacional, así como hacer diferencias de cualquier tipo sin reducir demasiado el tamaño de muestra, puede considerarse que el valor de la t de Student, para $\alpha = 0.05$ es 1.96, aproximadamente dos.

La regla de decisión es la siguiente:

- No rechazar H_0 si $|t_{r_k}| < 2$, es decir, considerar que ρ_k no es significativo.
- Rechazar H_0 , si se considerar que ρ_k es significativo.

El intervalo de confianza se construye como:

$$P[\rho_k - 2s_{r_k} < r_k < \rho_k + 2s_{r_k}] = 0.95 \quad (1.4)$$

Los límites inferior y superior del intervalo de confianza se pueden agregar a la gráfica ACF, para identificar cuáles son los valores de las r_k que quedan fuera del intervalo $[-2s_{r_k}, 2s_{r_k}]$ y que por tanto, son significativas.

Otra manera útil resultaría probar con un conjunto de K autocorrelaciones y determinar si son o no, todas insignificantes, la prueba de hipótesis es la siguiente:

$$H_0 : \rho_1 = \rho_2 = \dots = \rho_k = 0$$

$$H_1 : \text{Al menos una } \rho_k \neq 0, \forall k \leq K$$

El estadístico de prueba que se utiliza es el estadístico Box-Pierce:

$$Q(K) = N(N+2) \sum_{k=1}^K \frac{r_k^2}{N-k} \quad (1.5)$$

Que se distribuye aproximadamente como una X^2 (ji-cuadrada) con $K-p-q-1$ grados de libertad, con un nivel de confianza del $(1-\alpha)100\%$, donde p es el orden de la parte autorregresiva y q el orden de la parte de medias móviles. Además se resta uno por la estimación de la constante.

La regla de decisión se formula como:

- No rechazar H_0 si $Q(K) < X_{N,1-\alpha}^2$, es decir, las K primeras autocorrelaciones son estadísticamente insignificantes, con lo cual puede considerarse que los datos analizados son ruido blanco.
- Rechazar H_0 , en el caso de que se considere que al menos una de las autocorrelaciones es significativa.

1.1.2.2 Interpretación de la PACF

Para considerar que la PACF es decreciente infinita o si se trunca en cierto valor de k , se utiliza un intervalo de confianza que permitirá probar la siguiente hipótesis para cada una de las autocorrelaciones ρ_{kk} :

$$H_0 : \rho_{kk} = 0$$

$$H_1 : \rho_{kk} \neq 0$$

Si la hipótesis nula no se rechaza, se considera que la autocorrelación parcial correspondiente al intervalo k es estadísticamente insignificante, esto quiere decir, que la variable autorregresiva Y_{t-k} no se considera para el modelo $AR(k)$. De lo contrario, la autocorrelación correspondiente es significativa.

El estadístico de prueba es el siguiente:

$$t_{r_{kk}} = \frac{r_{kk} - 0}{s_{r_{kk}}} \quad (1.6)$$

donde r_{kk} es el coeficiente de autocorrelación parcial para un retraso de k periodos.

La varianza del estimador r_{kk} está dada por:

$$Var(r_{kk}) \approx \frac{1}{N} \forall k > p \quad (1.7)$$

Por lo tanto, el estimador de la varianza será el mismo valor:

$$Var(r_{kk}) = s_{r_{kk}}^2 \approx \frac{1}{N} \forall k > p \quad (1.8)$$

La regla de decisión se establece de la siguiente manera:

- No rechazar H_0 si $|t_{r_{kk}}| < 2$, es decir, considerar que ρ_{kk} no es significativo.
- Rechazar H_0 , es decir, considerar que ρ_{kk} es significativo.

Para identificar qué valores sobresalen de la PACF, se construye el siguiente intervalo:

$$P[\rho_{kk} - 2s_{r_{kk}} < r_{kk} < \rho_{kk} + 2s_{r_{kk}}] = 0.95 \quad (1.9)$$

Este intervalo se puede agregar a la gráfica PACF, para identificar qué valores de las r_{kk} quedan fuera de $[-2s_{r_{kk}}, 2s_{r_{kk}}]$, y que por tanto, significativas con un 95% de confianza.

1.1.2.3 Periodograma

Cuando existen componentes periódicos en una serie de tiempo, en particular cuando son amplitudes muy bajas, en la gráfica difícilmente se podrá visualizar esta situación. Para hacerla más clara, Schuster (1898) propuso utilizar la gráfica del periodograma para para buscar un componente periódico en una serie, esta se calcula con la siguiente ecuación:

$$I(\omega_k) = \begin{cases} N\alpha_0^2 & k = 0, \\ \frac{N}{2}(\alpha_k^2 + b_k^2) & k = 1, 2, \dots, [\frac{N-1}{2}], \\ N\alpha_{N/2}^2 & k = \frac{N}{2} \text{ cuando } N \text{ es par.} \end{cases} \quad (1.10)$$

donde a_k y b_k son los coeficientes de Fourier y N el número de observaciones de la serie de tiempo.

El periodograma se define como la suma de cuadrados de la serie de tiempo, es decir, $\sum_{t=1}^N Y_t^2$. Cuando tomamos los cuadrados de las amplitudes y los multiplicamos por el valor de N , los valores grandes se harán aún más grandes, mientras los pequeños irán disminuyendo. De esta manera se incrementará el contraste entre unos y otros.

1.1.2.4 Periodograma integrado

El periodograma integrado muestra la suma normalizada de los valores del periodograma que se dividen entre la suma de los valores de las ordenadas de todas las frecuencias de Fourier $f_k = k/N$ y se representa a través de una prueba de bondad de ajuste no paramétrica de Kolmogorov-Smirnov. La suma normalizada se calcula con la siguiente ecuación:

$$IP(\omega_k) = \sum_{j=0}^k \frac{I(\omega_j)}{\sum_{t=1}^N Y_t^2} \quad (1.11)$$

donde $I(\omega_j)$ es la función del periodograma y Y_t^2 es la serie de tiempo al cuadrado.

1.1.3 Estimación

Después de identificar algunos modelos tentativos, se habrá de estimarse los parámetros, es decir, el término constante, los coeficientes autorregresivos y de medias móviles. En general, para la estimación de parámetros puede utilizarse algunos de los siguientes métodos: *momentos*, *máxima verosimilitud* o *mínimos cuadrados*.

Para realizar la estimación de parámetros, se requiere del apoyo de un software y efectuar las siguientes acciones:

1. Como el método de estimación será de tipo numérico e involucra los datos nombrados prehistóricos, es necesario asignar valores a estas variables. A esto se llama *inicialización de la serie de tiempo*.

2. Por otro lado, el modelo generará ecuaciones donde se quiere minimizar el error, sin embargo, presenta una forma no lineal. Con el objetivo de linealizarlo, será necesario contar con valores iniciales de los parámetros que deben estimarse.

1.1.4 Diagnóstico

Una vez que se ha elegido a un modelo tentativo y se han estimado sus parámetros, habrá que diagnosticar el modelo. Para el diagnóstico se requieren las siguientes herramientas básicas:

- Los residuales, es decir, las diferencias entre cada dato y el modelo.
- Los pronósticos de los últimos datos, que pueden retirarse temporalmente del modelo.

Al diagnosticar un modelo se recomienda considerar los siguientes pasos:

1.- Verificar nuevamente que los datos cumplan con las condiciones de estacionaridad.

2.- Verificar si el modelo está sobrestimado, esto quiere decir, si sobran variables que no aportan un mejor ajuste; en ese caso deben retirarse. Esto se hace mediante pruebas de hipótesis, con la ayuda de herramientas estadísticas, como son: la desviación estándar estimada del estimador del parámetro $S_{\hat{\beta}_j}$, el estadístico t del estimador del parámetro t_{β_j} y el valor-P o P - *value* del estimador del parámetro.

Todos ellos se utilizan para probar la siguiente hipótesis para cada parámetro:

$$H_0 = \beta_j = 0$$

$$H_1 = \beta_j \neq 0$$

Donde β_j es el j -ésimo parámetro del modelo. Para probar esta hipótesis, se construye el estadístico:

$$t_{\hat{\beta}_j} = \frac{\hat{\beta}_j - 0}{S_{\hat{\beta}_j}} \quad (1.12)$$

La regla de decisión es:

- No rechazar H_0 si $|t_{\hat{\beta}_j}| < 2$. En este caso el parámetro β_j es estadísticamente insignificante y debe retirarse del modelo.
- En caso de no cumplirse H_0 , el parámetro debe dejarse en el modelo.

La regla de decisión se puede construir utilizando el $P - value$, de la siguiente forma:

- No rechazar H_0 si $P - value > \alpha$. En este caso el parámetro β_j es estadísticamente insignificante y debe retirarse del modelo.
 - En caso de no cumplirse H_0 , el parámetro debe dejarse en el modelo.
- 3.- Verificar si el modelo está subestimado; en dado caso habrá que agregar parámetros. Esto se hará a través de evaluar el comportamiento de los residuales que deberán aproximarse a ruido blanco.
- 4.- Revisar la posible ocurrencia de datos atípicos o faltantes, para darles el tratamiento adecuado.
- 5.- Por último, se toman todos los posibles modelos y se comparan de manera global, por medio de un cuadro comparativo.

1.1.5 Pronósticos

Después de encontrar un modelo apropiado, se pronosticarán los valores de la serie de tiempo. Por medio de los valores obtenidos (valores puntuales y por intervalo), corresponde hacer la interpretación, recomendaciones, discusión y sugerencias.

1.2 Modelos ARIMA

En esta sección se explica de forma breve los modelos ARIMA y sus características con base en lo que exponen la Dra. González Videgaray [13] y González Casimiro [24].

1.2.1 Concepto de estacionaridad

- *En sentido fuerte o estricto*

Una serie de tiempo es estacionaria en sentido fuerte o estricto si su función de distribución conjunta de probabilidad permanece idéntica en el tiempo. Esto es si se cumple lo siguiente:

$$f(Y_t, Y_{t-1}, \dots, Y_{t-k}) = f(Y_{t+\tau}, Y_{t-1+\tau}, \dots, Y_{t-k+\tau}) \forall \tau \in \mathbb{R} \quad (1.13)$$

- *En sentido débil o amplio*

Una serie de tiempo es estacionaria en sentido débil o amplio si se satisfacen tres condiciones:

1.-Su media es constante en el tiempo. Es decir:

$$E(Y_t) = E(Y_{t+\tau}) = \mu, \forall \tau \in \mathbb{R} \quad (1.14)$$

2.-La varianza es constante en el tiempo.

$$Var(Y_t) = Var(Y_{t+\tau}) = \gamma_0, \forall \tau \in \mathbb{R} \quad (1.15)$$

3.-La autocovarianza es independiente del tiempo.

$$Autocorr(Y_t, Y_{t-k}) = Autocorr(Y_{t+\tau}, Y_{t+\tau-k}) = \frac{\gamma_k}{\gamma_0} = \rho_k, \forall \tau \in \mathbb{R} \quad (1.16)$$

1.2.2 Modelos estacionarios

1.2.2.1 Modelo autorregresivo

Un modelo es autorregresivo si la variable Y_t de un periodo t es explicada por las observaciones de ella misma correspondientes a periodos anteriores, añadiéndose, un término de error. Este modelo se denota por $\mathbf{AR}(p)$ donde p indica el orden del modelo, es decir, el número de observaciones retrasadas de la serie temporal analizada que intervienen en la ecuación.

La ecuación general del modelo $\mathbf{AR}(p)$ es:

$$Y_t = \delta_0 + \phi_{\delta 1} Y_{t-1} + \dots + \phi_p Y_{t-p} + e_t \quad (1.17)$$

El modelo $\mathbf{AR}(p)$ presenta las siguientes propiedades:

- Tendrá condiciones de estacionaridad para sus parámetros.
- La ACF es decreciente infinita, con variantes que dependen de los signos y magnitudes de los coeficientes.
- La PACF se trunca en $K = p$.

1.2.2.2 Modelos de medias móviles

Un modelo denominado medias móviles es aquel que explica el valor de una determinada variable en un período t en función de un término independiente y una sucesión de errores correspondientes a períodos precedentes, ponderados convenientemente. El modelo se denota por $\mathbf{MA}(q)$ puesto que contiene q variables históricas autorregresiva. En el modelo se acostumbra formular con signos negativos en los coeficientes, con el objetivo de facilitar el uso de notación abreviada. Si el coeficiente aparece en el modelo con un valor negativo, esto implica que su valor es positivo y viceversa.

La ecuación general del modelo $\mathbf{MA}(q)$ es:

$$Y_t = \delta_0 + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q} \quad (1.18)$$

Donde e_t son ruido blanco gaussiano, es decir, variables aleatorias independientes que siguen una distribución Normal con media cero, varianza constante σ^2 y son no correlacionadas.

El modelo $\mathbf{MA}(p)$ presenta las siguientes propiedades:

- Carece de condiciones de estacionaridad para sus parámetros, es decir, siempre es estacionario.
- Tiene condiciones de invertibilidad para sus parámetros.
- La ACF se trunca en $k = q$.
- La PACF es decreciente infinita, con variantes que dependen de los signos y magnitudes de los coeficientes.

1.2.2.3 Modelos mezclados

Algunas veces es necesario construir un modelo que incluya términos tanto de autocorrelación como de medias móviles. Los modelos $\mathbf{ARMA}(p, q)$ contienen $p+q$ parámetros, este tipo de modelos tendrán tanto condiciones de estacionaridad como condiciones de invertibilidad.

La ecuación general del modelo $\mathbf{ARMA}(p, q)$ es la suma de los modelos $\mathbf{AR}(p)$ y $\mathbf{MA}(q)$:

$$Y_t = \delta_0 + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + e_t - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q} \quad (1.19)$$

El modelo $ARMA(p, q)$ que es construido a partir de la mezcla de los modelos $AR(p)$ y $MA(q)$, se puede suponer que tendrá condiciones de estacionaridad siempre que $\sum_i^p \phi_i < 1$ y de invertibilidad siempre que $\sum_i^q \theta_i < 1$. Así como también, su ACF será decreciente infinita, que termina en $k = q$ con un efecto procedente del modelo $MA(q)$ y su PACF también será decreciente infinita, con un efecto procedente del modelo $AR(p)$ que desaparece después de $k = p$.

1.2.3 Modelos no estacionarios

La mayoría de las series de tiempo en casos reales, no cumplen con las condiciones de estacionaridad en sentido amplio, ya que presentan varianza creciente, tendencia, variación estacional, puede que presente algunas de estas características o todas.

En estos casos es necesario modificar temporalmente los datos, a través de operaciones matemáticas sencillas, de manera que se ajusten a las condiciones de estacionaridad. De esta manera, será posible ajustar los datos a algún modelo que sea estacionario. Con el modelo obtenido se generan los pronósticos, posteriormente, se regresan los pronósticos a las unidades de los datos originales, aplicando funciones inversas.

Para ajustar un modelo no estacionario a un modelos estacionario, es necesario:

1.- *Estabilizar la varianza.* Los datos tienen que ser homoscedásticos, gráficamente los datos tienen que estar dentro de un intervalo, alrededor de la media, sin importar si esta es constante o no. Para lograr estabilizar la varianza es necesario aplicar una o más transformaciones como raíz cuadrada, logaritmo o recíproco.

2.- *Eliminación de la tendencia.* Una vez que ya se estabilizó la varianza, se observa si la serie de tiempo presenta algún tipo de tendencia. Si es el caso, es necesario eliminar la tendencia a través de diferencias ordinarias o finitas. Después de que los datos tengan media constante, se observa si existe variación estacional.

3.- *Tratamiento de la variación estacional.* La variación estacional se puede visualizar en algunos casos desde la gráfica de los datos originales o después de estabilizar la varianza. En algunos datos se puede presentar después de eliminar la tendencia. Para tratar la variación estacional, existen dos alternativas:

a) Por medio de diferencias estacionales.

b) Representar la variación estacional a través de modelos multiplicativos que incluyan el componente estacional.

1.2.4 Modelos con variación estacional

En muchas series de tiempo que presentan variación estacional, se puede lograr eliminar a través de diferencias estacionales o bien, también es posible modelarla, incluyendo variables apropiadas en los modelos, para ello existen dos tipos de modelos.

1.2.4.1 Modelos puramente estacionales

Los modelos estacionales con variación estacional avanzan o retroceden en periodos de longitud s , a estos modelos se les llama modelos exclusivamente estacionales (seasonal models). Se denotan de las siguientes formas: $SARMA(P, D, Q)$ o $ARMA(P, D, Q)_s$

donde:

- P es el orden de la parte autorregresiva estacional SAR .
- D es el orden de las diferencias estacionales de longitud s .
- Q es el orden de la parte de medias móviles estacionales SMA .

A continuación se describe de forma breve estos modelos.

1.2.4.2 Modelo $SAR(P)$

El modelo general autorregresivo estacional es de la siguiente forma:

$$Z_t = \delta_0 + \phi_1 Z_{t-s} + \phi_2 Z_{t-2s} + \dots + \phi_P Z_{t-Ps} + e_t \quad (1.20)$$

Este modelo tiene condiciones de estacionaridad pero no de invertibilidad.

Por otro lado, la ACF de estos modelos es decreciente infinita con valores significativos en múltiplos de s y la PACF se trunca en $k = P$.

1.2.4.3 Modelos $SMA(Q)$

El modelo general de medias móviles estacional es de la siguiente forma:

$$Z = \delta_0 + e_t - \Theta_1 e_{t-s} - \Theta_2 e_{t-2s} - \dots - \Theta_Q e_{t-Qs} \quad (1.21)$$

Este modelo tiene condiciones de invertibilidad pero no de estacionaridad.

Por otro lado, la ACF se trunca en $k = Q$ con valores significativos en múltiplos de s y la PACF es decreciente infinita.

1.2.4.4 Modelos SARIMA(P, D, Q)

El modelo general estacional es de la siguiente forma:

$$Z_t = \delta_0 + \phi_1 Z_{t-s} + \dots + \phi_P Z_{t-Ps} + e_t - \theta_1 e_{t-s} - \dots - \theta_Q e_{t-Qs} \quad (1.22)$$

Sus parámetros tienen condiciones de estacionaridad como de invertibilidad.

El comportamiento de las ACF y PACF son decrecientes infinitas en múltiplos de s .

1.2.5 Modelos multiplicativos

Todos los modelos antes mencionados, estacionales y no estacionales, se pueden combinar en uno general de forma multiplicativa, a este modelo se conoce como modelo multiplicativo y se denota de la siguiente forma:

$$\text{ARIMA}(p, d, q) \times (P, D, Q)_s$$

o bien, de la siguiente forma:

$$\text{ARIMA}(p, d, q) (P, D, Q)_s$$

donde:

- El valor p es el orden de la parte AR.
- El valor d es el orden de las diferencias ordinarias o no estacionales.
- El valor q es el orden de la parte MA.
- El valor P es el orden de la parte SMA.
- El valor D es el orden de las diferencias estacionales.
- El valor de Q es el orden de la parte SMA.

La ACF y PACF de este modelo depende por la interacción entre la parte regular y estacional.

1.3 Caso de aplicación: predicción de número de donaciones de células madre por cordón umbilical

Nuestro cuerpo contiene diferentes tipos de células que son indispensables para nuestra salud. De ellas depende nuestro cuerpo para que esté en funcionamiento, para hacer que nuestro corazón lata, para el funcionamiento de nuestro cerebro, para reemplazar las células de nuestra piel a medida que se renueva, entre muchas otras funciones. Sin embargo, quienes se encargan de formar todos estos tipos de células, son las células madre. Las células madre son las proveedoras de nuevas células. Cuando las células madre se dividen pueden hacer más de sí mismas o más de otros tipos de células. Por ejemplo, las células madre de la piel pueden formar más células madre de piel o logran formar diferentes células que tienen trabajos específicos como producir el pigmento de melanina.

Las células madre se pueden obtener de dos fuentes, de los embriones en sus primeras fases de desarrollo (células madre de origen embrionario) o del organismo adulto (células madre adultas), como las células que se presentan en el cordón umbilical o se pueden encontrar en el organismo, distribuidos en órganos y tejidos, como la piel, el cerebro, músculo, intestino, corazón, córnea, retina, médula ósea, grasa, vasos, etc [5].

En esta investigación nos centraremos solo en las células de cordón umbilical.

1.3.1 Células madre de cordón umbilical

El cordón umbilical conecta a un bebé en el vientre de su madre. Se extiende desde una abertura en el estómago del bebé a la placenta en el útero. El cordón mide unos 50 cm de largo.

Para qué sirve el cordón umbilical

En la placenta, el oxígeno y los alimentos del torrente sanguíneo de la madre pasan al torrente sanguíneo del bebé a lo largo del cordón umbilical.

La sangre circula a través del cordón por unos vasos sanguíneos que la transportan aportando oxígeno y nutrientes imprescindibles para el bebé. Dos arterias llevan los productos de deshecho como el dióxido de carbono desde el bebé a la placenta. Estos vasos sanguíneos están encerrados y protegidos por una sustancia pegajosa llamada gelatina de Wharton, que a su vez está cubierta por una membrana llamada amnios.

Hacia el final del embarazo, la placenta pasa anticuerpos al bebé a través del cordón umbilical, dándole inmunidad a las infecciones durante unos tres meses después del nacimiento. Sin embargo, sólo transmite los anticuerpos que ya tiene [4].

El cordón umbilical contiene células madre. Estas son células que pueden convertirse en tipos específicos de células en su cuerpo. Las células madre pueden utilizarse para tratar algunas enfermedades, tales como ciertas formas de cáncer y anemia.

A veces la sangre del cordón se desecha junto con el cordón umbilical y la placenta después del nacimiento del bebé. Pero algunas personas desean guardar o donar la sangre del cordón (en un banco de sangre de cordón umbilical) para que pueda utilizarse más adelante para tratar enfermedades [6].

1.3.2 Proceso para obtener las células madre del cordón umbilical

Para los tratamientos, los médicos suelen obtener sangre del cordón umbilical de las placentas de donantes voluntarias después de dar a luz.

En el CNTS, la recolección de la sangre de cordón umbilical la realiza el ginecólogo responsable del parto o cesárea.

El CNTS entrega a la donante un equipo de recolección (“kit”), el cual contiene todo lo necesario para la obtención de la sangre. Después del nacimiento por parto o cesárea, y generalmente antes de la salida de la placenta, el médico debe limpiar con un antiséptico el sitio del cordón umbilical donde se efectuará la punción. La punción se efectúa con una aguja que va unida a un tubo que conduce a la bolsa colectora. El médico debe cerciorarse que se extraiga el máximo volumen posible (mínimo 80 mililitros). Finalizada la extracción, se guarda un fragmento de cordón umbilical en uno de los tubos contenidos en el equipo de recolección y se toma una muestra de sangre materna para estudios posteriores.

Es indispensable asegurarse que todos los documentos contenidos en el equipo de recolección se llenen adecuadamente y que se colecten las firmas de las personas que se solicitan y, una vez llena la bolsa con la sangre de cordón umbilical, las muestras sanguíneas y la documentación anexa, sean colocadas dentro de la bolsa de cierre hermético. Se conserva a temperatura ambiente (entre +12^o C y +24^o C) y habitualmente es un familiar quien lleva el equipo de recolección al CNTS, dentro de las primeras 24 horas después del nacimiento [7].

1.3.3 Beneficios de las células madre obtenidas de la sangre del cordón umbilical

Las células madre pueden curar y tratar enfermedades. Sin embargo, debido a los diferentes tipos de células madre que existen, cada una ofrece distintos beneficios, esto es porque tienen diferentes aplicaciones. Tales como los siguientes [4]:

- Las células madre del cordón umbilical tienen la propiedad de convertirse en diferentes tipos de células especializadas.
- La manera de obtención de las células madre del cordón umbilical es muy fácil, indolora y el riesgo es nulo. No hay peligro para la madre ni mucho menos para el bebé.
- Estas células son prácticamente vírgenes, es decir, no contienen virus, elementos cancerígenos ni ningún otro daño. Esto es debido a que han estado protegidas en el útero durante toda su vida.
- Su esperanza de vida es superior a los 15 años gracias a la congelación. Los expertos aseguran que pueden llegar a vivir muchos más años sin problemas.
- Las células madre del cordón umbilical son 100% compatibles con el bebé y mucho más compatibles con sus familiares que las células madre de médula ósea.
- Hay una mayor concentración de células madre en el cordón umbilical que en la misma porción de médula ósea.
- Como el donante es el mismo que el receptor o un familiar, no se invierte tiempo en la búsqueda de un donante. La disponibilidad de las células es inmediata.
- Durante su obtención no debe destruirse ningún embrión por lo que no existe controversia.

Actualmente, las células madre del cordón umbilical se pueden utilizar para tratamientos para combatir cerca de 80 enfermedades potencialmente mortales. Entre las enfermedades tratables se incluyen una amplia gama de cánceres, problemas relacionados con la médula ósea, trastornos de la sangre o hemoglobinopatías, trastornos metabólicos, inmunodeficiencias, tratamientos emergentes entre otras enfermedades.

1.3.4 Donaciones de células madre por cordón umbilical en el mundo

En 1988 se reportó en Francia el primer trasplante exitoso de células madre obtenidas de cordón umbilical. El receptor fue un pequeño de cinco años que padecía una rara forma de anemia, y las células fueron obtenidas del cordón umbilical de su hermana. A partir de entonces, esta terapéutica se convirtió en una realidad salvadora de vidas en todo el mundo y comenzaron a proliferar los bancos de cordón umbilical. Las investigaciones con este tipo de células han atraído reflectores porque prometen tratar enfermedades contra las que hoy no se puede hacer mucho [9].

CAPÍTULO 1 PRONÓSTICO DE LAS CÉLULAS MADRE USANDO BOX-JENKINS 27

Actualmente España ocupa el segundo lugar en el mundo en donaciones de cordón umbilical, con el 10% de todas las unidades almacenadas en el mundo, que han permitido trasplantar células madre sanguíneas a 2,500 personas, de las que tres cuartas partes son de fuera de España, según la Organización Nacional de Trasplantes (ONT). Solo Estados Unidos los supera con una población seis veces mayor que en España [10].

En la figura 1.2 se muestra la gráfica de unidades disponibles de sangre de cordón umbilical por banco en España en los últimos 10 años [11].

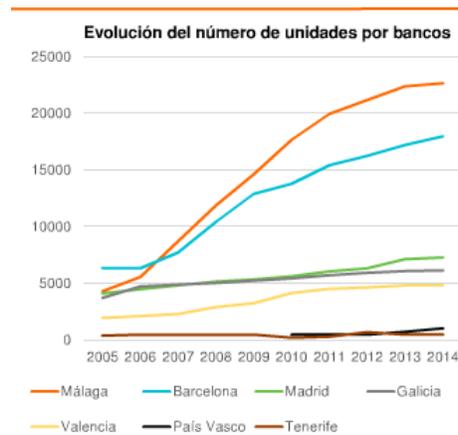


Figura 1.2. Unidades de sangre de cordón umbilical en España [11].

En la figura 1.3 se muestra la gráfica de los trasplantes de células madre que se han hecho en Estados Unidos; las células madre de cordón umbilical son ya la fuente preferida, según el programa de donantes de médula ósea—Estados Unidos de América [12]. En la gráfica se puede ver el crecimiento que ha tenido esta fuente a partir del 2001 hasta el 2010.

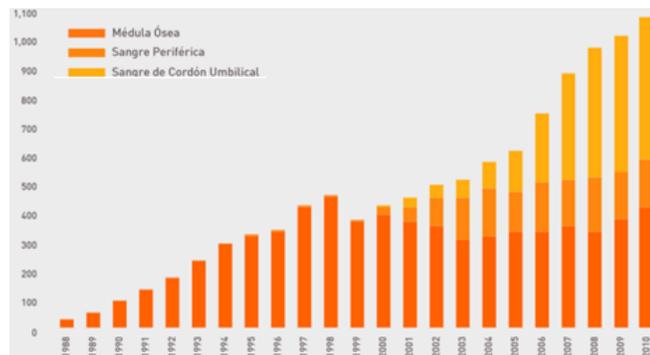


Figura 1.3. Trasplante de células madre en Estados Unidos [12].

1.3.5 Donaciones de células madre por cordón umbilical en México

A principios de este siglo, llegaron a México los primeros bancos privados para donaciones de células madre de cordón umbilical, y en el 2007 se reportó que existían por lo menos diez empresas, convirtiendo a México en el cuarto país con mayor número de estos bancos en el mundo, según datos de la Organización Médica Colegial de España [9]. El problema es que el costo es muy elevado y prohibitivo para la mayoría de las personas, aunque la cura de muchas enfermedades a futuro se vislumbran muy alentadoras.

En México solo existen tres bancos públicos de sangre de cordón umbilical: el (CNTS) de la Secretaría de Salud, el del hospital la Raza, que pertenece al IMSS, y el hospital Universitario de la Universidad Autónoma de Nuevo León.

Se estima que hay más de una decena de bancos privados que operan en el país, aunque solo tres cuentan con permiso de la Comisión Federal para la Protección contra Riesgos Sanitarios (Cofepris): BCU, Lamport Genética y Desarrollo e Innovaciones, y Desarrollo en Biotecnología Celular.

Quienes desean preservar los cordones umbilicales de sus hijos pagan alrededor de 800 dólares para los primeros trámites y 100 dólares al año por preservarlo.

Para donaciones, México necesita por lo menos 10,000 unidades de sangre de cordón umbilical, sin embargo, entre 2003 y 2012 el CNTS de la Secretaría de Salud solo recibió 5,000 de ellas.

No todos los cordones pueden ser utilizados con éxito. De acuerdo con un estudio de la American Society for Blood and Marrow Transplantation las probabilidades de éxito es de 1 entre 2,500.

El reto para la donación es que en algunas ocasiones las muestras que se deben extraer de los cordones no alcanzan los 90 mililitros de sangre (la cantidad mínima necesaria) o su calidad es deficiente. Eva Calderón Garcidueñas, especialista en hematología de la Facultad de Química de la UNAM, explica que la Academia Estadounidense de Pediatría (AAP, por sus siglas en inglés) calcula que las posibilidades de que un niño necesite sus propias células madre de la sangre del cordón en el futuro son de una en 200,000 [8].

En la gráfica 1.4 se muestran las donaciones de células madre del cordón umbilical, así como el número de células rechazadas y aceptadas del año 2003 al 2014 [7].

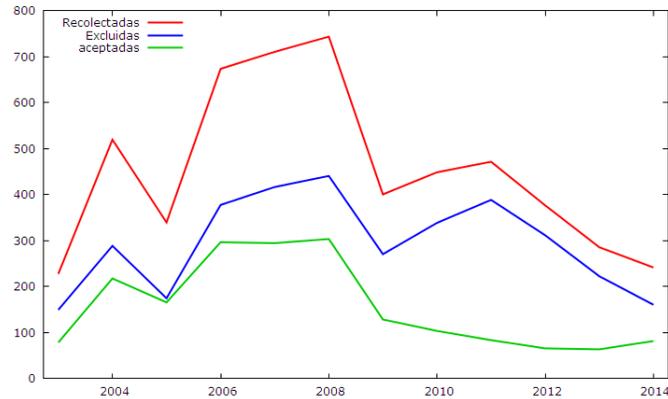


Figura 1.4. Células madre del cordón umbilical en el CNTS [7].

1.4 Modelado por la metodología Box-Jenkins

El CNTS cuenta con los datos anuales del número de donaciones de células madre del cordón umbilical desde el año 2003 al año 2014. Sin embargo, el número de datos son insuficientes para realizar un pronóstico usando la metodología Box-Jenkins, además el CNTS no lleva el registro mensual de donaciones de células madre por cordón umbilical. Se vio la necesidad de repartir los datos anuales en sus correspondientes meses.

Para obtener los datos mensuales, se sacó el promedio anual, se obtuvo la desviación estándar de las donaciones anuales, después de obtener la desviación estándar, se dividió entre 12 la desviación anual, para obtener la desviación mensual, después se sacó el promedio de cada año. Posteriormente, se estableció un intervalo por cada año de la siguiente manera: de acuerdo con el promedio de cada año se restó la desviación mensual para el límite inferior, mientras que para el límite superior se sumó al promedio la desviación mensual. Finalmente se generaron 12 números aleatorios por año, de acuerdo con los intervalos que se establecieron en cada año, de tal forma que sumaran el total de donaciones que se obtuvieron por año.

La siguiente gráfica nos ilustra los datos de la serie de tiempo de donaciones de células madre por cordón umbilical. Como se puede observar en la figura 1.5, la serie parece ser homoscedástica, en otras palabras, la varianza parece ser constante, la tendencia era creciente hasta finales del año 2008 y en los años posteriores la serie tiende a ser decreciente, además, pareciera que no existe variación estacional, pero esto pudiera surgir a través del análisis y presenta fluctuación aleatoria.

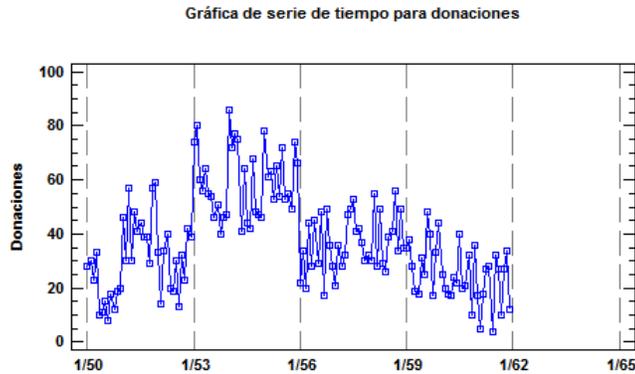


Figura 1.5. Serie de tiempo original.

La figura 1.6 se puede contemplar el periodograma integrado de la serie de donaciones de células madre por cordón umbilical. Podemos notar que el periodograma integrado muestra un escalón al inicio de la gráfica, en otras palabras, presenta frecuencias cercanas a cero, esto quiere decir, que la serie tiene tendencia.

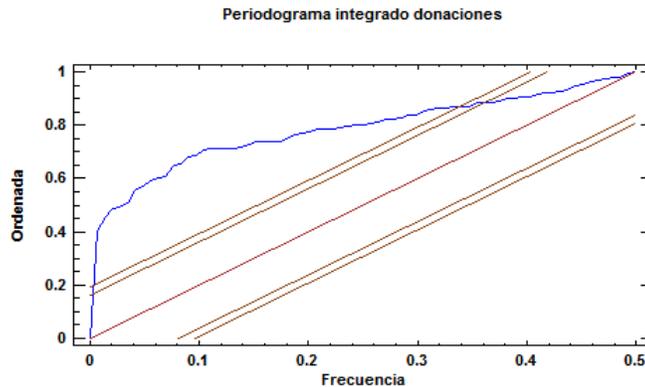


Figura 1.6. Periodograma integrado de las donaciones del CNTS.

La función de autocorrelación (ACF) que se muestra en la figura 1.7, se puede observar que la función decrece infinitamente, pero se puede ver que es un poco lento, esto es porque la serie presenta tendencia. Por otro lado, la función de autocorrelación parcial que se muestra en la figura 1.8, se puede notar que sobresale en $k = 1$ y $k = 2$.

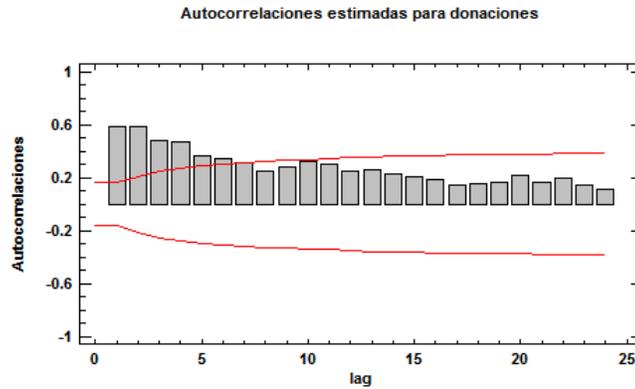


Figura 1.7. Gráfica original de la ACF.

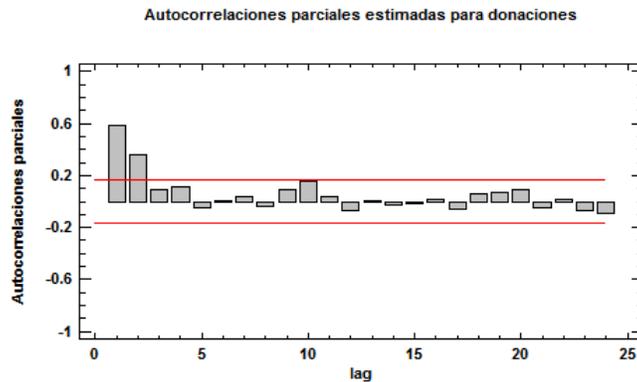


Figura 1.8. Gráfica original de la PACF.

De lo anterior, se destaca lo siguiente:

- La varianza parece ser constante, dicho de otra forma, la serie es homoscedástica.
- La serie presenta tendencia, esto es sobresaliente en el peridograma integrado, se puede ver un escalón al inicio de la gráfica (figura 1.6).
- Pareciera que no presenta variación estacional, sin embargo, puede presentarse a lo largo del análisis.

De esto se concluye, que los datos no son estacionarios, por lo tanto, se tendrán que hacer modificaciones. Para llegar a un modelo tentativo, es necesario conseguir que la serie sea estacionaria, para ello, se tendrá que eliminar la tendencia mediante diferencias ordinarias.

1.4.1 Identificación de un modelo

Después de observar detenidamente cada una de las gráficas anteriores e identificar los componentes de la serie de tiempo, se procede a realizar una o dos diferencias ordinarias para eliminar la tendencia. La figura 1.9 se muestra la serie después de aplicar una diferencia ordinaria. Podemos ver que la media y la varianza de la serie de tiempo son constantes, y no presenta variación estacional, esto último lo corroboraremos a continuación con la función de autocorrelación (ACF) y la función de autocorrelación parcial (PACF).

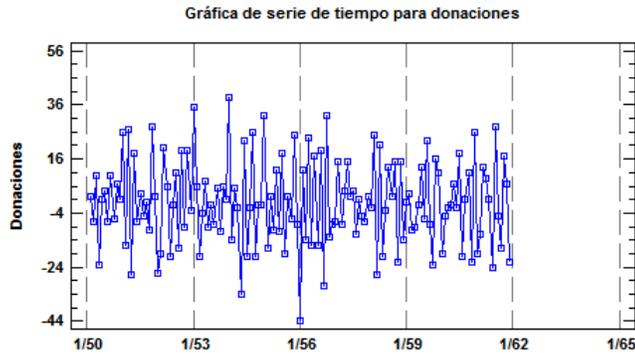


Figura 1.9. Gráfica serie de tiempo con una diferencia ordinaria.

La ACF (figura 1.10) muestra un valor en $k = 1$ y la PACF (figura 1.11) muestran un valor en $k = 1$ y ligeramente sobrepasa valores en $k = 2$, $k = 3$ y $k = 9$, el resto de los valores se van disminuyendo y se mantienen dentro de los intervalos de confianza. Los correlogramas de la serie aplicando una sola diferencia parecen ser limpios.

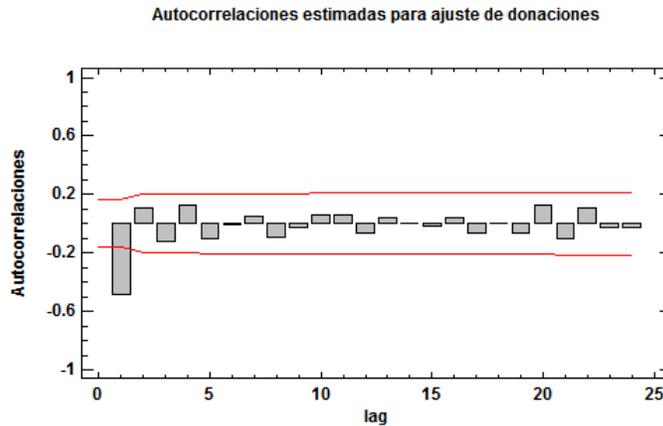


Figura 1.10.ACF con una diferencia ordinaria.

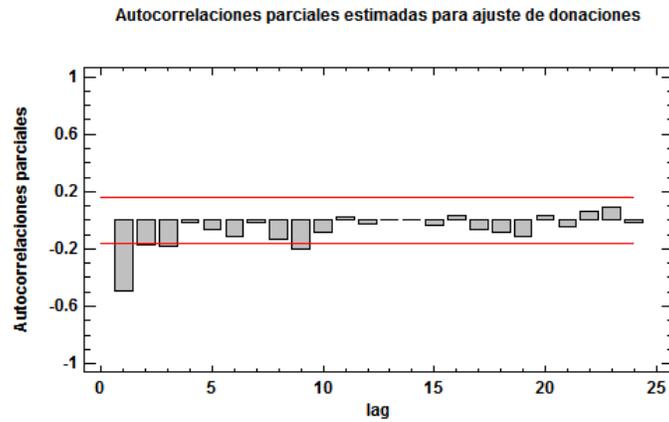


Figura 1.11. PACF con una diferencia ordinaria.

Aplicando una segunda diferencia, podemos ver que los correlogramas son menos limpios que los de una diferencia. En la ACF que se muestra en la figura 1.12, el valor en $k = 1$ es más cercano a 1 y resalta un valor en $k = 2$, por otra parte, la PACF que se muestra en la figura 1.13, decrece infinitamente. Con base en las gráficas, se prefieren los correlogramas con una sola diferencia ordinaria.

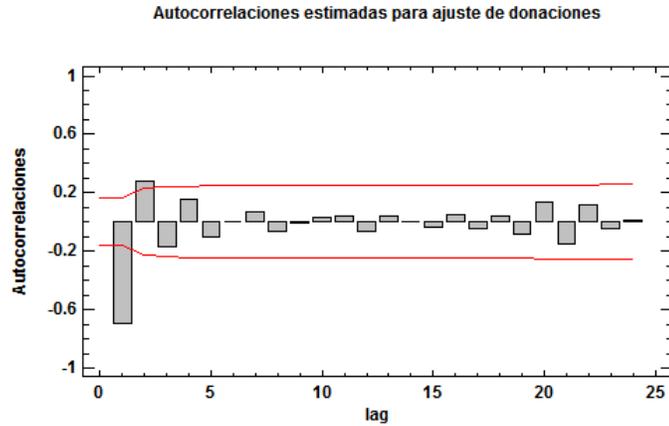


Figura 1.12. ACF con dos diferencias ordinarias.

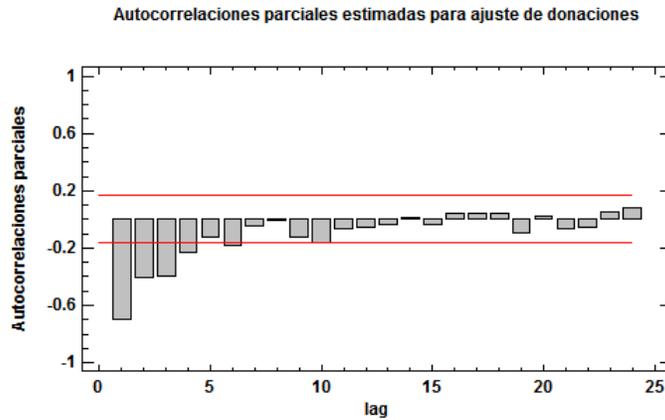


Figura 1.13. PACF con dos diferencias ordinarias.

Recordando que si la ACF mostrara una estructura decreciente y en la PACF tuviera p números de coeficientes significativos se trataría de un AR. En el caso de MA sería justo a la inversa. Sin embargo, no es el caso, ya que la ACF (figura 1.10) se trunca en $k = 1$ y la PACF (figura 1.11) se trunca en $k = 1$ y ligeramente en los valores $k = 2$, $k = 3$ y $k = 9$, por lo tanto, se sugiere partir como primer modelo tentativo a un $ARIMA(1, 1, 1)$.

1.4.2 Estimación de los parámetros

Una vez de tener uno o varios modelos iniciales, la siguiente etapa de la metodología Box-Jenkins es la de estimación de los parámetros los parámetros ϕ (correspondiente al AR) y θ (correspondiente al MA) del modelo a partir del conjunto de 143 observaciones (144-1, observación que se pierde tras diferenciar).

Los resultados de la estimación de los parámetros del modelo $ARIMA(1, 1, 1)$ se muestran en la figura 1.14.

Resumen del modelo ARIMA				
Parámetro	Estimado	Error Est.	t	P-value
AR(1)	-0.0217646	0.133736	-0.162743	0.870955
MA(1)	0.620379	0.106931	5.80168	0.00000
Mean	-0.03603559	0.441645	-0.0815947	0.935086
Constante	-0.0368202			

Figura 1.14. Estimación del modelo propuesto.

1.4.3 Diagnóstico del modelo

1.4.3.1 Significación estadística de los parámetros

Cada uno de los parámetros autorregresivo o media móvil incluidos en el modelo debe ser estadísticamente significativo. Se observa en la figura 1.14, el estadístico t del parámetro $AR(1)$ y de la constante, no es mayor en valor absoluto a dos y su P -value no es menor a 0.05, esto indica que no son significativos, por lo cual, serán eliminados el parámetro $AR(1)$ y la constante del modelo, en el caso del parámetro $MA(1)$ es significativo, por lo tanto se propone un $ARIMA(0, 1, 1)$.

Los resultados de la estimación de los parámetros del modelo $ARIMA(0, 1, 1)$ se muestran en la figura 1.15.

Resumen del modelo ARIMA				
Parámetro	Estimado	Erro Est.	t	P -value
MA(1)	0.640346	0.0662336	9.668	0.000000

Figura 1.15. Estimación del modelo $ARIMA(0,1,1)$.

1.4.3.2 Condiciones de estacionaridad e invertibilidad

Si un modelo utiliza parámetros autorregresivos la suma de sus valores deben ser menor que uno. Si utiliza parámetros de media móvil, la suma de sus valores deben ser menor a uno. En el caso del modelo $ARIMA(0, 1, 1)$, no tiene parte autorregresiva, por lo tanto, la condición de estacionaridad ($\phi < 1$) se cumple, por otro lado, la parte de media móvil del modelo cumple con la condición de invertibilidad ($\theta_1 < 1$), en la figura 1.15 se observa que el valor del parámetro θ_1 es menor a la unidad.

1.4.3.3 Análisis de los residuales del modelo

Para que una serie de tiempo sea estacionaria en sentido amplio deben cumplir con las condiciones mencionadas en la subsección 1.2.1. Sin embargo, también es importante verificar los residuales de cada modelo propuesto, ya que si la serie es estacionaria sus residuales también deberán serlo [13]. En el periodograma integrado de la figura 1.16, se puede ver en los residuales del modelo $ARIMA(0, 1, 1)$ que no existe un valor cercano a cero, se mantienen dentro de los límites de confianza, además de que se ajusta adecuadamente a la línea, por lo tanto se puede decir, que se comportan como ruido blanco gaussiano.

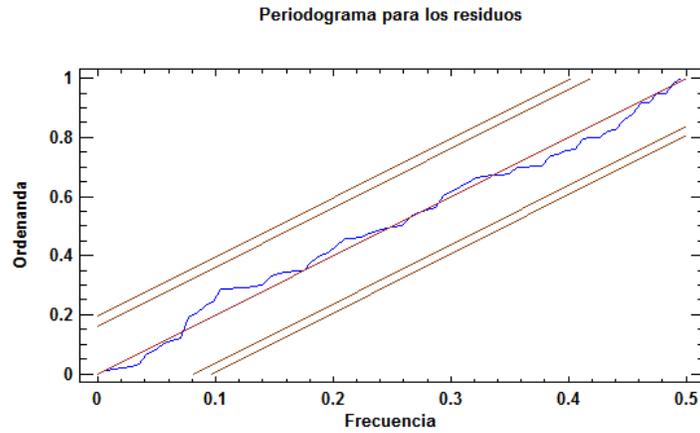


Figura 1.16. Periodograma integrado de los residuos del modelo $ARIMA(0,1,1)$.

Para que un modelo sea adecuado para una serie de tiempo, sus residuos deben comportarse como ruido blanco gaussiano, si es así el modelo es factible. Las funciones de correlación simple y parcial de los residuos del modelo no deben presentar valores significativos.

Podemos ver que tanto la ACF (figura 1.17) como la PACF (figura 1.18) de los residuos del modelo $ARIMA(0,1,1)$ no presentan algún valor significativo, por lo que se puede corroborar lo que se mencionó en el periodograma integrado del modelo, que los residuos del modelo se comportan como ruido blanco gaussiano y también se puede decir, que al modelo no le hace falta agregar algún parámetro.

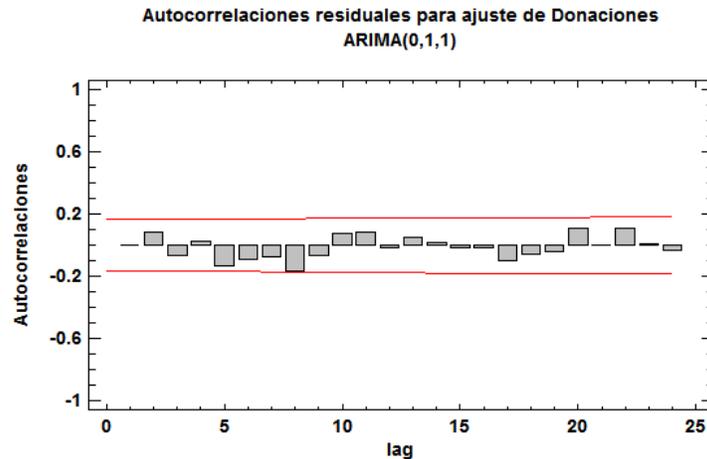


Figura 1.17. ACF de los residuos del modelo $ARIMA(0,1,1)$.

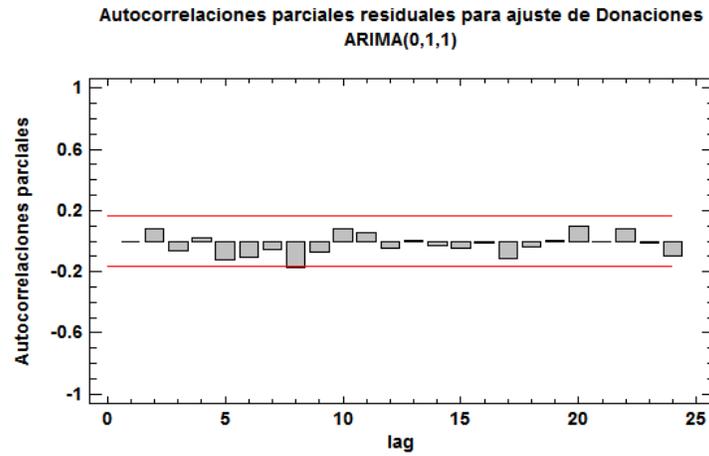


Figura 1.18. PACF de los residuales del modelo ARIMA(0,1,1).

1.4.3.4 Datos faltantes y atípicos

La serie original (figura 1.5) no presenta estacionalidad, sin embargo, presenta tendencia. En el diagrama de box-plot podemos corroborar esto, se observa en la figura 1.19 que la mediana de cada año no tiene valores parecidos y no mantiene una dirección constante.

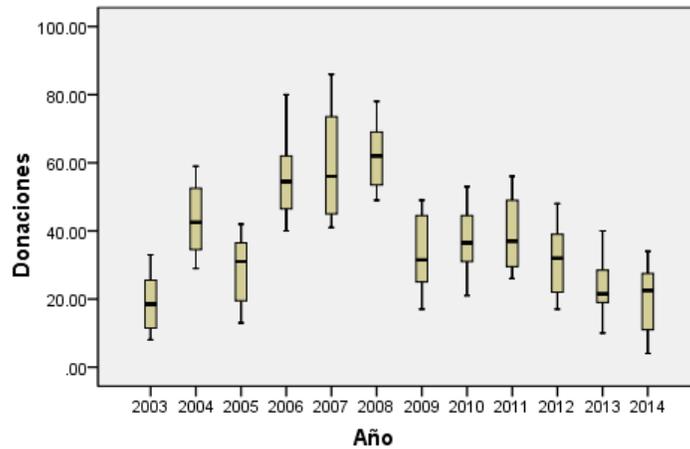


Figura 1.19. Diagrama de caja y bigotes de la serie de tiempo por año.

De manera más general, en la figura 1.20 se observa en diagrama de box-plot un punto alejado, este punto se encuentra en el mes de enero del año 2007, que se consideró dentro del pronóstico para obtener un resultado más cercano a la realidad. Con respecto a los cuartiles, la población está un poco más concentrada en el primer y segundo cuartil (25% y 50% de la población) que la población de el segundo y tercer cuartil (50% y 75% de la población).

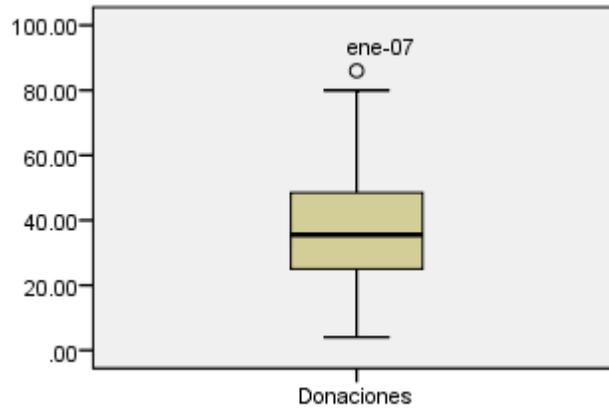


Figura 1.20. Diagrama de caja y bigotes.

1.4.3.5 Comparación de modelos

El primer modelo es un $ARIMA(0, 1, 1)$ sin constante. El parámetro $MA(1)$ es significativo, los valores de las gráficas ACF y PACF se mantienen dentro los límites de confianza, en la PACF se ve ligeramente un valor en $k = 8$. En el periodograma integrado no se presenta ningún escalón significativo, los residuales se ajustan a la línea recta, lo que indica que se trata de ruido blanco gaussiano. En el periodo de estimación (figura 1.21), en las pruebas de aleatoriedad, en RUNS aparece un asterisco, esto indica que es marginalmente significativo. En el periodo de validación (figura 1.22), tiene la menor suma de cuadrados de los residuales (RMSE).

El segundo modelo es un $ARIMA(2, 1, 0)$ sin contante. Los parámetros $AR(1)$ y $AR(2)$ son significativos, todos los valores de la ACF se encuentran dentro de los límites de confianza y en la PACF solo se observa un valor en $k = 8$, el resto se mantiene dentro de los límites de confianza. En el periodograma integrado parece indicar que los residuales se comportan como ruido blanco gaussiano, esto es porque no presenta algún escalón significativo y también se ajusta a la línea. En el periodo de estimación (figura 1.21), pasa todas las pruebas de aleatoriedad y es el modelo que tiene la mayor RMSE y el segundo en el periodo de validación (figura 1.22).

El tercer modelo es un $ARIMA(3, 1, 0)$ sin constante. El valor estadístico t de cada parámetro es mayor en valor absoluto a dos, por lo tanto son significativos. Al igual que los dos anteriores modelos, en la ACF no sobresale ningún valor y en la PACF se observa un valor en $k = 8$, todos los demás valores se mantienen dentro de los límites de confianza. En el periodograma integrado parece indicar que los residuales se comportan como ruido blanco gaussiano, esto es porque no presenta algún escalón significativo y también se ajusta a la línea. En el periodo

de estimación (figura 1.21), pasa todas las pruebas de aleatoriedad, sin embargo en el periodo de validación (figura 1.22) es el modelo que tiene la mayor RMSE.

El cuarto modelo es un $ARIMA(2, 1, 1)$ sin constante. Los parámetros tanto de la parte medias móviles como la autorregresiva son significantes. En las ACF y PACF no existe ningún valor significativo. En el en periodograma integrado no presenta ningún escalón, los residuales se mantienen dentro de los límites de confianza y se ajustan a la línea. En la figura 1.21, es el modelo con la menor RMSE y en la parte de las pruebas de aleatoriedad, aparece un asterisco en MEAN.

Modelos

- (A) $ARIMA(0,1,1)$
- (B) $ARIMA(2,1,0)$
- (C) $ARIMA(3,1,0)$
- (D) $ARIMA(2,1,1)$

Periodo de estimación

Modelo	RMSE	MAE	MAPE	ME	MPE
(A)	13.4019	10.6396	40.7791	-0.104985	-18.6949
(B)	13.6944	10.7454	41.1862	-0.0891434	-17.8437
(C)	13.514	10.5909	41.033	-0.116318	-18.2077
(D)	13.2644	10.3515	41.3471	-0.175508	-20.5246

Modelo	RMSE	RUNS	RUNM	AUTO	MEAN	VAR
(A)	13.4019	*	OK	OK	OK	OK
(B)	13.6944	OK	OK	OK	OK	OK
(C)	13.514	OK	OK	OK	OK	OK
(D)	13.2644	OK	OK	OK	*	OK

Figura 1.21. Periodo de estimación.

Periodo de validación

Modelo	RMSE	MAE	MAPE	ME	MPE
(A)	12.4484	11.3772	114.361	-1.18697	-79.2154
(B)	13.8851	12.7108	128.796	-0.920508	-87.3028
(C)	13.9045	12.8221	128.118	-1.10314	-85.8137
(D)	13.5686	11.9793	129.417	-2.33297	-96.2477

Figura 1.22. Periodo de validación.

1.4.3.6 Elección del modelo

Los parámetros de cada modelo son significantes esto quiere decir, que los modelos no están sobrestimados y las gráficas de ACF y PACF no indican que les haga falta agregar algún parámetro, además de que sus residuales se

CAPÍTULO 1 PRONÓSTICO DE LAS CÉLULAS MADRE USANDO BOX-JENKINS 40

comportan como ruido blanco gaussiano. Dos de los cuatro modelos propuestos pasan todas las pruebas de aleatoriedad (modelos 2 y 3), sin embargo, no son los modelos que tienen la menor RMSE, tanto en el periodo de validación como de estimación. Por otra parte, el modelo $ARIMA(2, 1, 1)$ es el que tiene la menor RMSE en el periodo de estimación, sin embargo, el modelo $ARIMA(0, 1, 1)$ es el que tiene la menor RMSE en el periodo de validación de los pronóstico. Por lo tanto el modelo que se seleccionará es el $ARIMA(0, 1, 1)$. En la figura 1.23 se muestra la estimación del parámetro $MA(1)$.

Resumen del modelo ARIMA

Parámetro	Estimado	Erro Est.	t	P-value
MA(1)	0.640346	0.0662336	9.668	0.000000

Figura 1.23. Parámetros del modelo $ARIMA(0,1,1)$.

Capítulo 2 Pronóstico con otro modelo matemático

Con frecuencia nos enfrentamos a problemas que implican la extracción de información a partir de datos muy complejos e inciertos, a los que no siempre se puede dar solución por medio de aproximaciones tradicionales. Sin embargo, existen técnicas que son capaces de realizar cálculos complejos para procesar información, una de estas técnicas son las redes neuronales artificiales (RNA).

A continuación se explicará brevemente, los conceptos básicos de las RNA, su arquitectura, principales modelos y la aplicación de esta metodología para la predicción de donaciones de células madre por cordón umbilical.

2.1 Qué son las redes neuronales artificiales

Una RNA es una estructura compuesta de un número de unidades interconectadas (neuronas artificiales). Cada una de ellas posee una característica de entrada/salida e implementa una función. Las RNA están inspiradas en el sistema biológico natural, persiguen imitar ciertas habilidades humanas atribuibles al cerebro y a millones de elementos interconectados llamados neuronas. Tratan de crear modelos artificiales que solucionen problemas que no se pueden resolver mediante técnicas algorítmicas convencionales. En la figura 2.1 se ve gráficamente la estructura y el funcionamiento de una red neuronal.

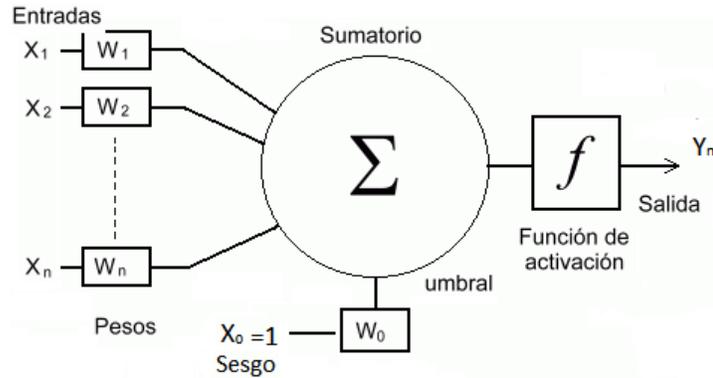


Figura 2.1. Estructura y funcionamiento de una RNA.

2.1.1 Elementos básicos de una RNA

Los elementos básicos de una red neuronal artificial son los encargados de realizar las operaciones de la misma. A continuación se presentará los elementos que la constituyen.

- Entradas o capas de entradas (X_1, X_2, \dots, X_n). Son escalares que se proporcionan a la red. El número que constituyen esta capa dependerá de los datos del problema.
- Salidas o capa de salidas (Y_1, Y_2, \dots, Y_m). Son los valores que arroja la red como el resultado del aprendizaje.
- Conexiones o pesos asociados a cada entrada (W_1, W_2, \dots, W_k). Son valores numéricos que resaltan la importancia de la entrada correspondiente. Estas conexiones pueden ser excitadoras (presentan un signo positivo), o inhibitoras (conexiones negativas).
- Punto de suma de entradas ponderadas. Se encarga de sumar todas las entradas multiplicadas por su respectivo peso (W_k). La suma de ponderaciones está dada por la siguiente ecuación: $V_k = \sum_{i=0}^n X_i W_{ik}$, donde X_1, X_2, \dots, X_n son las entradas de las neuronas k , W_1, W_2, \dots, W_k son los pesos asociados a cada entrada de la neurona y V_k es la suma de las entradas multiplicadas por los pesos correspondientes.
- Sesgo (X_0). Es un valor formado por una entrada fija e igual a 1, multiplicada por el peso W_0 .
- Valor umbral. Cada neurona tiene también un cierto valor umbral. En el interior de la neurona se suman las entradas, y a esta suma se le resta el valor umbral propio de la neurona. Esto genera la activación de la neurona, también conocida como **potencial post-sináptico** o **PSP**.

- **Función de activación.** Es una función lineal o no lineal, que limita la amplitud de la salida de la neurona. La función de activación está dada por la siguiente ecuación: $Y_k = \Gamma(V_k)$. Donde $\Gamma(V_k)$ es la función de activación y Y_k es la salida de la neurona k .

2.1.2 Funciones de activación

Al igual que las neuronas biológicas, las RNA también tienen grados de activación que pueden limitarse únicamente a dos, como en el caso de las biológicas o variar entre un rango de valores (definido entre $[0,1]$ o entre $[-1,1]$).

Para obtener la salida final de una neurona se debe aplicar a la entrada ponderada total una función de activación que calcula el valor de la salida o estado de actividad de la neurona. La función de activación combina la entrada total a la k -ésima neurona, obtenido a partir de los estímulos y pesos recibidos.

Existen diferentes funciones de activación, entre las más usadas están:

1. *La función lineal o identidad.* Devuelve directamente el valor de activación de la neurona. Este tipo de función se utiliza en redes de baja complejidad [1]. En forma matemática se expresa de la siguiente forma:

$$\begin{aligned} \Gamma(V_k) &= V_k \\ \text{Rango} &= (-\infty, \infty) \end{aligned} \tag{2.1}$$

Y gráficamente,

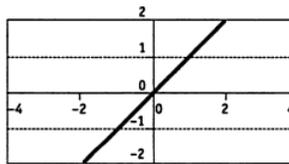


Figura 2.2. Función lineal [1].

2. *La función escalón o signo.* Presenta salidas binarias (habitualmente $\{0,1\}$ o $\{-1,1\}$). Si la activación de la neurona es inferior a un determinado umbral, la salida se asocia con un determinado output, y si es igual o superior al umbral se asocia con el otro output [1]. Su expresión matemática es de la siguiente forma:

Para un rango = $\{0, 1\}$

$$a)\Gamma(V_k) = \begin{cases} 1 & \text{si } V_k \geq 0 \\ 0 & \text{si } V_k < 0 \end{cases} \tag{2.2}$$

y gráficamente,

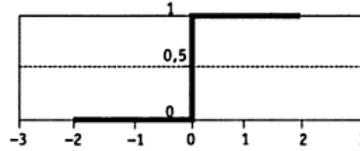


Figura 2.3. Función escalón (a) [1].

Para un rango = $\{-1, 1\}$

$$b)\Gamma(V_k) = \begin{cases} 1 & \text{si } V_k \geq 0 \\ -1 & \text{si } V_k < 0 \end{cases} \quad (2.3)$$

y gráficamente,

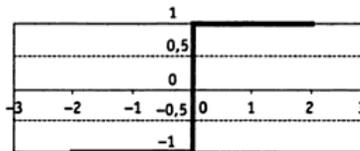


Figura 2.4. Función escalón (b) [1].

3. *La función mixta o lineal a tramos.* Es una variante progresiva de la función escalón. Si la activación de la unidad es menor que un límite inferior pre-establecido, la salida se asocia con un determinado valor; si la activación es igual o superior que un límite superior, la salida se asocia con otro valor. Si el nivel de activación se encuentra comprendido entre ambos límites, se aplica la función lineal [1]. Su expresión matemática es de la siguiente forma:

Para un rango = $[0, 1]$

$$a)\Gamma(V_k) = \begin{cases} 1 & \text{si } V_k \geq 1 \\ V_k & \text{si } -1 \leq V_k < 1 \\ 0 & \text{si } V_k < -1 \end{cases} \quad (2.4)$$

y gráficamente,

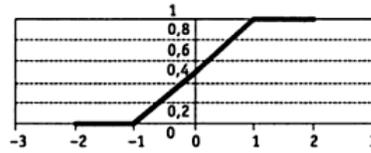


Figura 2.5. Función mixta (a) [1].

Para un rango = $[-1, 1]$

$$b)\Gamma(V_k) = \begin{cases} 1 & \text{si } V_k \geq 1 \\ V_k & \text{si } -1 \leq V_k < 1 \\ -1 & \text{si } V_k < -1 \end{cases} \quad (2.5)$$

y gráficamente,

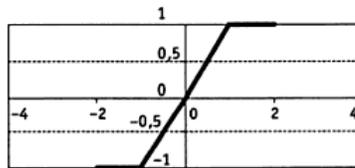


Figura 2.6. Función mixta (b) [1].

4. *La función sigmoidea.* Definida en un determinado intervalo monotónico con límites superiores e inferiores. Entre las funciones sigmoideas de transferencia más aplicadas destacan *la función sigmoide o logística y función tangente hiperbólica* [1]. Esta última es la más usada en redes neuronales y se recomienda para problemas de predicción. Su expresión matemática es:

Función sigmoide o logística.

$$\Gamma(V_k) = \frac{1}{1 + e^{-V_k}} \text{ Rango} = (0, 1) \quad (2.6)$$

y gráficamente,

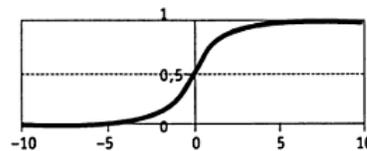


Figura 2.7. Función sigmoide o logística [1].

Función tangente hiperbólica.

$$\Gamma(V_k) = \frac{e^{V_k} - e^{-V_k}}{e^{V_k} + e^{-V_k}} \text{ Rango} = (-1, 1) \quad (2.7)$$

y gráficamente,

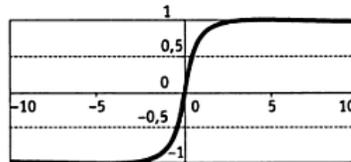


Figura 2.8. Función tangente hiperbólica [1].

5. *La función gaussiana.* Las funciones gaussianas se suelen aplicar en redes complejas con m capas ocultas ($m \geq 2$) que requieren reglas de propagación basadas en el cálculo de distancias cuadráticas entre los vectores de entrada y los pesos de la red [1]. Su expresión matemática es:

$$\Gamma(V_k) = e^{-V_k^2/\sigma^2} \text{ Rango} = (0, 1) \quad (2.8)$$

y gráficamente,

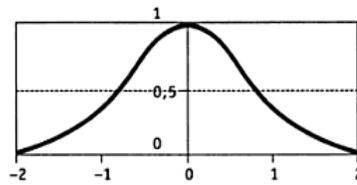


Figura 2.9. Función gaussiana [1].

6. *La función sinusoidal.* Genera salidas continuas en el intervalo $[-1, 1]$. Estas funciones suelen emplearse en los casos en los que se requiere explícitamente una periodicidad temporal [1]. Su expresión matemática es:

$$\Gamma(V_k) = \text{sen}(V_k+1) \text{ Rango} = (-1, 1) \quad (2.9)$$

y gráficamente,

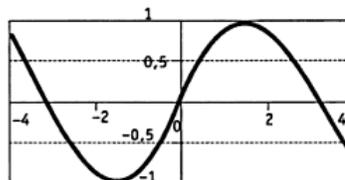


Figura 2.10. Función sinusoidal [1].

2.2 Arquitectura de una RNA

Los elementos de una RNA se pueden conectar entre sí. Estas se agrupan en capas, cada una de ellas con un conjunto de neuronas de número variable y comportamiento similar.

Cada capa está conectada a la inmediata posterior total o parcialmente, excepto la última capa, que constituye la salida total de la red neuronal. Existen tres tipos de capas [3]:

1. *Capa de entrada.* El número y tipo de neuronas que construyen esta capa, depende de los datos del problema.
2. *Capas intermedias.* Pueden ser más de una, dependiendo del tipo y complejidad del problema a resolver con la red. Mediante el tratamiento adecuado de estas capas se consiguen las propiedades de generalización, extracción de características, adaptación, etc., que hacen muy interesante el trabajo de las RNA.
3. *Capa de salida.* El número de neuronas de esta capa depende del formato esperado de salida de la red.

La figura 2.11 representa los tres elementos que toda RNA debe tener.

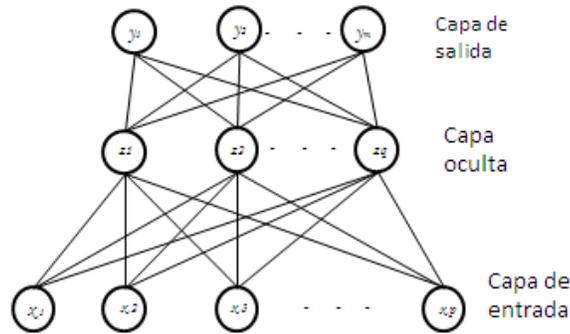


Figura 2.11. Arquitectura de una red neuronal .

Las diferentes formas de distribuir, conectar e interrelacionar estos tres tipos de capas, junto con el tipo de neuronas que constituyen cada una de ellas, nos va a definir los diferentes paradigmas de red existentes. Podríamos clasificar las estructuras neuronales o modelos conexionistas de diferentes formas.

2.2.1 Clasificación según el número de capas

Existen dos tipos de clasificaciones de las RNA según su número de capas [2]:

1. *Redes monocapa.* Las redes monocapa solo cuentan con una capa de neuronas de entrada y una de salida, que intercambian señales con el exterior y que constituyen a un tiempo capa oculta y capa de salida de sistema. Las redes monocapa se utilizan típicamente en tareas relacionadas con lo que se conoce como auto-asociación, por ejemplo, para regenerar informaciones de entrada que se presentan a las incompletas o distorsionadas, o para el reconocimiento de patrones.

2. *Redes multicapa.* Las redes multicapa están compuestas por al menos una capa oculta entre la capa de entrada y la de salida de la RNA. Las redes multicapa pueden resolver problemas más complicados que las redes monocapa. Normalmente la dificultad del problema a resolver va asociado con el tamaño de las redes multicapa, que está definido por el número de capas ocultas y las neuronas en cada una de ellas.

2.2.2 Clasificación según el tipo de conexiones

La conectividad entre las neuronas canaliza la información de salida de una neurona como entrada de otra. A continuación se presentan los diferentes tipos de conexiones que existen en una RNA [2]:

1. *Conexiones hacia delante.* El sentido de transmisión de la información en este tipo de conexiones se realiza desde neuronas de capas inferiores (más cercanas a la capa de entrada) hacia neuronas de capas superiores (más cercanas a la salida). También se denominan *feedforward*.

2. *Conexiones hacia atrás.* Estas conexiones llevan los datos de las neuronas de una capa superior, a otras de una capa inferior. Son también llamadas *feedbackward*.

3. *Conexiones laterales.* Son conexiones que se hacen entre neuronas de una misma capa. Se diseñan como excitadoras (con signo positivo), permitiendo la cooperación (con peso positivo) o la inhibición (con peso negativo) entre neuronas, lo que establece una competencia entre neuronas de la misma capa.

4. *Conexiones con retardo.* Los elementos de retardo se incorporan en las conexiones para implementar modelos dinámicos temporales (redes dinámicas). Se almacenan los datos antiguos para después procesarlos junto con la información actual.

5. *Conexiones auto-recurrentes.* La salida de una neurona se conecta con su propia entrada.

2.2.3 Clasificación según tipo de información

1.- *Redes estáticas.* El tipo de información que manejan las redes estáticas tiene carácter estacionario, es decir, el valor de la salida solo depende del valor de las entradas en el mismo momento en el que esta se produce.

Estas redes se pueden caracterizar estructuralmente por la inexistencia de bucles de realimentación y de elementos de proceso que las forman. Normalmente van asociadas a las redes con conexiones hacia adelante (*feedforward*).

2.- *Redes dinámicas.* La información que se maneja en este tipo de redes incluye la variable tiempo, es decir, la salida depende de la entrada en el momento actual y de los valores de entrada y/o salida en momentos anteriores.

Para incluir la variable tiempo, se pueden incorporar en la red retardos temporales o conexiones de tipo *feedbackward* entre las capas de neuronas [2].

2.3 Aprendizaje y recuerdo

Los datos de entrada de la RNA se procesan mediante un conjunto de operaciones internas, que proporcionan una salida de la red. Este conjunto de operaciones internas se define como aprendizaje, es decir, el proceso por el que una red neuronal crea, modifica, o destruye sus conexiones en respuesta a una información de entrada. Las RNA son capaces de generalizar un determinado procesamiento a partir de un conjunto de patrones de aprendizaje. Por ello, el aprendizaje es necesario, para que la red pueda extraer conocimiento a partir de los patrones de entrada que se le presentan, por lo que es habitual que la RNA sea entrenada previamente a la ejecución de la tarea que debe llevar a cabo.

La mayoría de los modelos neuronales (redes off-line) tienen dos modos o fases de funcionamiento: la fase aprendizaje o entrenamiento y la fase recuerdo, ejecución u operación. En estos modelos es necesario ejecutar inicialmente la fase de entrenamiento para establecer los pesos de la red y, posteriormente utilizar el modelo en la fase recuerdo, manteniendo los pesos fijos. Por otra parte, existen modelos neuronales en los que las fases de aprendizaje y recuerdo coinciden (redes on-line), de forma que la red puede aprender y modificar sus conexiones

durante el modo de operación, por lo que los pesos varían de forma dinámica cada vez que se presentan al sistema una nueva información [1]. A continuación se explicará brevemente en qué consiste cada una de las fases.

2.3.1 Fase de aprendizaje o entrenamiento

Para la construcción de una red neuronal, se parte de un modelo determinado de neuronas y de una determinada arquitectura, estableciendo los pesos iniciales aleatorio o nulos. A partir de este modelo empieza la fase de aprendizaje o entrenamiento de la red para dar solución al objeto de estudio. Este ajuste se puede llevar a cabo mediante dos procesos diferentes pero complementarios.

- *Modelado de la sinapsis (pesos) de la red*

A través de una regla de aprendizaje, construida con el objetivo de minimizar una determinada función de error. Se denomina $w_{ij}(t)$ al peso que contenga la neurona pre-sináptica i con la post-sináptica j en el momento t (iteración t -ésima), el estado de dicha sinapsis en el momento $t + 1$ está dada por la siguiente expresión:

$$w_{ij}(t + 1) = w_{ij}(t) + \Delta w_{ij}(t) \quad (2.10)$$

donde $\Delta w_{ij}(t)$ es la variación generada en el peso por la regla de aprendizaje, considerando las señales que llegan a la j -ésima neurona en el instante t . El proceso de aprendizaje suele ser iterativo, actualizándose los pesos sucesivamente hasta que la red alcanza el rendimiento deseado (error máximo permitido) o hasta que se ha ejecutado un conjunto determinado de iteraciones.

- *Creación y/o destrucción de neuronas de la red*

Se denomina construcción o creación a la introducción de nuevas neuronas en el sistema, con los pesos asociados resultantes (modelos de redes constructivas), mientras que la destrucción implica la eliminación de la neurona de la red, con la desaparición de los pesos a ella asociadas (modelos de poda).

2.3.1.1 Algoritmos de aprendizaje

Existen dos sistemas de aprendizaje básico que influyen en la estructura de la red y en sus resultados: aprendizaje supervisado y el aprendizaje no supervisado, cuya distinción procede del campo de reconocimiento de patrones (kosko, 1992). La diferencia fundamental entre ambos tipos radica en la existencia de un agente externo o supervisor, que controla el sistema de aprendizaje de la red.

1. *Aprendizaje supervisado*

El aprendizaje supervisado se caracteriza por la presencia de un agente externo (supervisor o maestro) que controla el proceso de entrenamiento,

estableciendo la respuesta que deberá generar la red (output del sistema) a partir de una entrada determinada [1]. En cada iteración, el supervisor comprueba la salida de la red, la compara con la salida deseada, si existen diferencias, se procede a modificar los pesos de las conexiones con el fin de conseguir que la salida obtenida se aproxime más a la deseada. La red realizará varias iteraciones modificando los pesos de sus conexiones hasta alcanzar la condición de paro del entrenamiento, dejando a partir de ese momento los pesos fijos [2].

Desde el punto de vista formal, sea $E(w)$ la función que representa el error esperado de la red, expresado en función de sus pesos sinápticos. El aprendizaje supervisado trata de estimar una función multivariante desconocida, $f: R^N \rightarrow R^M$ a partir de submuestras de patrones de entrada-salida (x, y) , $x \in R^N$ y $y \in R^M$, tomadas aleatoriamente de la muestra total [2].

El modelo de la función multivariante se basa en la minimización iterativa de $E(w)$ mediante algún algoritmo de aproximación, por el cual se pueden distinguir tres tipos de aprendizaje supervisado: por corrección de error, por refuerzo o por tipo estocástico.

- *Aprendizaje por corrección de error.* Su funcionamiento se basa en el ajuste de los pesos de las conexiones de la red a partir de la diferencia entre los valores deseados y los obtenidos por el sistema, esto es en función del error cometido en la salida.
- *Aprendizaje por refuerzo.* También denominado “premio-castigo”. Consiste en indicar mediante una señal de refuerzo (éxito=+1 o fracaso=-1), si la salida obtenida se ajusta a la salida deseada, y en función de ello, se procede al ajuste de los pesos utilizando un mecanismo basado en probabilidades. Resulta más lento que el aprendizaje por corrección de error.
- *Aprendizaje Estocástico.* Se basa en la introducción de cambios aleatorios en los valores de los pesos de la red, evaluando su efecto a partir de la salida deseada y de una distribución de probabilidad. Suele establecerse una analogía en términos termodinámicos, asociando la red con un sólido físico que tiene cierto grado energético. La energía de la red representa su grado de estabilidad, por lo que el aprendizaje consiste en minimizar la energía del sistema a través del ajuste de los pesos.

La figura 2.12 muestra de forma breve los principales modelos de redes con aprendizaje supervisado.

Tipo de aprendizaje supervisado	Modelo de red
Aprendizaje por corrección de error.	<ul style="list-style-type: none"> • Perceptrón • Adaline/Madaline • Backpropagation • Brain-State-in-a-Box
Aprendizaje por refuerzo.	<ul style="list-style-type: none"> • Linear Reward Penalty • Adaptive Reward Penalty • Adaptive Heuristic Critic
Aprendizaje estocástico.	<ul style="list-style-type: none"> • Boltzmann Machine • Cauchy Machine

Figura 2.12. Principales modelos de RNAs con aprendizaje supervisado [1].

2. Aprendizaje no supervisado

Las redes con aprendizaje no supervisado o autosupervisado no requieren de información externa para ajustar los pesos de las conexiones neuronales. En este aprendizaje se presenta a la red un conjunto de patrones sin contar con la salida deseada. La red, por medio de un algoritmo de aprendizaje, estima la función de densidad probabilista $p(x)$ que describe la distribución de patrones x , $x \in R^p$.

Para que la red obtenga resultados de calidad, es necesario disponer de un número suficiente de patrones de aprendizaje [1]. La red aprende a adaptarse basándose en las experiencias recogidas de los patrones de entrenamiento, encontrando correlaciones o categorías entre sí. La red modifica los pesos de forma que los vectores de entrada más similares sean asignados a la misma unidad de salida, en los que se conoce como mapeo de características [2].

Hay varias posibilidades en cuanto a la interpretación de la salida de estas redes, que dependen de su estructura y del algoritmo de aprendizaje empleado. En algunos casos, la salida representa el grado de familiaridad o similitud entre la información que se le está presentando en la entrada y las informaciones que se le han mostrado hasta entonces. En otros casos, se podría realizar un establecimiento de categorías, indicando la red a la salida a qué categoría pertenece la información presentada a la entrada, siendo la propia quien debe encontrar las categorías apropiadas a partir de las correlaciones entre las informaciones presentadas.

Por lo general se suelen considerar dos tipos de aprendizaje no supervisado: aprendizaje hebbiano y aprendizaje competitivo y comparativo.

- *Aprendizaje hebbiano*. Es la base de muchas otras reglas de aprendizaje, cuyo objetivo es extraer características de los datos de entrada. La base es una suposición: si dos neuronas N_i y N_j toman el mismo estado simultáneamente, es decir, que ambas estén activas o inactivas, el peso de la conexión de ambas se incrementa. Las entradas y las salidas de las neuronas solo presentan dos estados: activa o inactiva.

- *Aprendizaje competitivo y comparativo.* Se enfoca a la clasificación de los datos. Como característica principal del aprendizaje competitivo se puede decir que, si un patrón nuevo se determina que pertenece a una clase reconocida previamente, entonces la inclusión de este nuevo patrón a esta clase matizará la representación de la misma. Si el patrón de entrada se determinó que no pertenece a ninguna de las clases reconocidas anteriormente, entonces la estructura y los pesos de la red neuronal serán ajustados para reconocer la nueva clase.

2.3.1.2 Resumen de algoritmos más utilizados

En la figura 2.13 se muestra un resumen de los algoritmos que más se utilizan para el entrenamiento de una red:

Paradigma	Arquitectura	Algoritmo de aprendizaje	Tareas
Supervisado Tipos de aprendizaje: • Aprendizaje por corrección de error • Aprendizaje por refuerzo • Aprendizaje estocástico	Perceptrón o Perceptrón multicapa	Algoritmo de aprendizaje perceptrón, retropropagación del error, ADALINE, MADALINE	Clasificación de patrones, aproximación de funciones, predicción, control, ...
	Elman y Jordan recurrentes	Retropropagación del error	Síntesis de series temporales
	Recurrentes	Algoritmo de aprendizaje Boltzman	Clasificaciones de patrones
	Competitivo	LVQ	Categorización intra-clase, compresión de datos
	Red ART	ARTMap	Clasificación de patrones, categorización intra-clase
No supervisado Tipos de aprendizaje: • Aprendizaje hebbiano • Aprendizaje competitivo y comparativo	Red de Hopfield	Aprendizaje de memoria asociativa	Memoria asociativa
	Multicapa sin realimentación	Proyección de Sannon	Análisis de datos
	Competitiva	VQ	Categorización, Análisis de datos
	SOM	Kohonen SOM	Categorización, análisis de datos
	Redes ART	ART1, ART2	Categorización

Figura 2.13. Algoritmos de aprendizaje más conocidos.

2.3.1.3 Terminología

En la etapa de entrenamiento se utilizan términos que se definirán a continuación:

- *Patrón.* Un patrón es un pequeño grupo de observaciones, formado a partir de un conjunto de datos.
- *Patrones de entrenamiento.* Es un conjunto de datos, formado en grupos pequeños suministrados a la red para su aprendizaje y están compuestos por patrones de entrada y patrones de salida.

- *Ciclo*. Es un barrido de todos los patrones del conjunto o patrones de entrenamiento.
- *Criterios de paro*.
 - Se alcanza un número máximo de ciclos.
 - Se repiten los ciclos hasta que el error alcance un valor máximo permisible.
- *Regla de aprendizaje*. La regla de aprendizaje es la que se encarga de modificar los pesos sinápticos de las RNA para que su respuesta sea correcta.
- *Tasa de aprendizaje (η)*. Controla el tamaño del cambio de los pesos en cada iteración. En el entrenamiento de la red se debe evitar dos cosas: tener un ritmo de aprendizaje demasiado pequeño, ya que esto puede ocasionar una disminución importante en la velocidad de convergencia y la posibilidad de acabar atrapado en un mínimo local; por otra parte, tener un ritmo de aprendizaje demasiado grande puede conducir a la inestabilidad en la función error, esto evitará que se produzca la convergencia debido a que se darán saltos en torno al mínimo sin alcanzarlo. Por lo tanto se recomienda tomar valores entre 0.05 y 0.5 o similares [15].
- *Factor momento (α)*. Acelera la convergencia de los pesos. Toma un valor próximo a 1. Después de calcular el error sobre cada patrón de entrenamiento, cada peso es ajustado en proporción al gradiente del error calculado en el paso hacia atrás. Los cambios en los pesos reducen el error total de la red neuronal. Para evitar que en la superficie del error los pesos estimen una minimización del error con un óptimo local, se aplica el momento como un mecanismo alternativo para cambiar la dirección de los pesos y eventualmente logre un mínimo global. Esto mejorará los tiempos de entrenamiento al algoritmo backpropagation [14].

Cabe mencionar, que es opcional aplicar el factor momento, en caso de hacerlo, en lugar de calcular los pesos para el conjunto de entrenamiento mediante:

$$\Delta_p w_{ji} = \eta \delta_{pj} Z_{pi} \tag{2.11}$$

se afecta la expresión a través de un término que indica la contribución proporcional de los pesos previos, esto es mediante la siguiente ecuación:

$$\Delta_p w_{ji} = \eta \delta_{pj} Z_{pi} + \alpha \cdot \Delta_p w_{ji}(\textit{anterior}) \tag{2.12}$$

2.3.1.4 Algoritmo de entrenamiento Backpropagation o retropropagación

Paso hacia adelante

1. Seleccionar un vector de entrada desde el conjunto de entrenamiento con su correspondiente salida.
2. Calcular el valor de entrada de las neuronas de la capa oculta:

$$v_{pj} = \sum_{j=1}^N w_{jp} x_{pj} \quad (2.13)$$

3. Calcular el valor de salida de los nodos de la capa oculta, aplicando la función de activación:

$$F(v_{pj}) = y_{pj} \quad (2.14)$$

4. Calcular la entrada de la neurona de la capa de salida usando la ecuación del paso 2.
5. Calcular el valor de salida de la neurona de la capa de salida usando la ecuación del paso 3.

Paso hacia atrás

6. Calcular el término de error para la neurona de la capa de salida:

$$\delta_{pj} = y_{pj}(1-y_{pj})(d_{pj}-y_{pj}) \quad (2.15)$$

7. Calcular los términos de error para las neuronas de la capa oculta:

$$\delta_{pj} = y_{pj}(1-y_{pj})\left(\sum \delta_{pk} w_{kj}\right) \quad (2.16)$$

8. Actualizar el peso de la neurona de la capa de salida:

$$\Delta_p w_{ji} = \eta \delta_{pj} y_{pi} \quad (2.17)$$

9. Actualizar los peso de las neuronas de la capa de oculta mediante la ecuación 2.16.
10. Repetir del paso 1 al 9 hasta que el error sea aceptable:

$$E_p = \frac{1}{2} \sum_{j=1}^N (d_{pj} - y_{pj})^2 \quad (2.18)$$

2.3.2 Evaluación del aprendizaje

Los procesos de entrenamiento de las RNA se basan en la minimización del error de un conjunto de patrones. Después de lograr esto, el objetivo final de un buen entrenamiento es mejorar la capacidad de generalización de la red, de manera que las predicciones para unos datos no usados durante el proceso de entrenamiento reproduzcan correctamente el fenómeno a modelar. De esta forma, la capacidad de generalización se basa en una buena interpolación a partir de los patrones de entrenamiento disponibles.

Para mejorar la capacidad de generalización de las RNA es deseable que el conjunto de patrones de entrenamiento sea de calidad y no presente demasiado ruido. Otros aspectos que se deben considerar para mejorar la generalización de la RNA son el número de patrones disponibles para el entrenamiento, es decir, contar con un conjunto de patrones de entrenamiento, los más numerosos posibles.

Cuando se evalúa una red neuronal no solo es importante evaluar si la red ha sido capaz de aprender los patrones de entrenamiento, sino también, evaluar el comportamiento de la red ante nuevos patrones; a este proceso se le conoce como generalización de un modelo de RNA y se adquiere en la fase de entrenamiento [Sanger, 1989]. Para ello, es necesario que durante el proceso de aprendizaje la red extraiga las características de las muestras, para poder luego responder correctamente a nuevos patrones.

En el proceso de entrenamiento de la red se debe considerar, junto con el error cometido respecto al conjunto de patrones de aprendizaje (“error de aprendizaje”), el denominado “error de generalización”, que se puede medir utilizando un conjunto de test distinto a la muestra de entrenamiento. El error de aprendizaje, indica la calidad de respuesta de la red a los patrones de entrenamiento, y el error de generalización, indica la calidad de respuesta de la red a nuevos patrones. Para poder obtener una medida de ambos errores es necesario dividir el set de datos disponibles en dos, el set de datos de entrenamiento, y el set de datos de evaluación. El primero se utiliza durante la fase de entrenamiento para que la red pueda extraer las características de los mismos y, mediante el ajuste de sus pesos sinápticos, para que la red logre una representación interna de la función. El segundo se utiliza para evaluar la capacidad de generalización de la red.

La causa más común de la pérdida de capacidad de generalización es el sobre aprendizaje. Esto sucede cuando la cantidad de ciclos de entrenamientos tiende a ser muy alta. Se observa que la respuesta de la red a los patrones de entrenamiento es muy buena, mientras que la respuesta a nuevos patrones tiende a ser muy pobre. Al aumentar el número de ciclos la red tiende a sobre ajustar la respuesta a los patrones de entrenamiento, a expensas de una menor capacidad

de generalización. Otra razón de que la RNA pierda la capacidad de generalización es por el uso excesivo de neuronas ocultas en la red neuronal. Esto hace que la red tienda a ajustar con mucha exactitud los patrones de entrenamiento, evitando que la red extraiga las características del conjunto. Este problema se ve agravado cuando los patrones de entrenamiento poseen ruido, ya que la red ajusta también el ruido de los mismos [1].

2.4 Modelos de RNA aplicados para predicción

Algunos de los modelos de redes neuronales que se aplican para hacer predicciones sobre valores futuros, se explican brevemente a continuación.

2.4.1 Red neuronal perceptrón multicapa

En 1986, Rumelhart y otros autores, presentan la "regla delta generalizada" para adaptar los pesos propagando los errores hacia atrás, es decir, propagar los errores hacia las capas ocultas inferiores. De esta forma se consigue trabajar con múltiples capas y con funciones de activación no lineales. Se demuestra que el perceptrón multicapa es un aproximado universal. Un perceptrón multicapa puede aproximar relaciones no lineales entre los datos de entrada y salida. Esta red se ha convertido en una de las arquitecturas más utilizadas en el momento [20].

2.4.1.1 Características del perceptrón multicapa

- Es una de las arquitecturas más utilizadas para resolver problemas reales.
- Se evalúa un conjunto de datos de entradas y se obtienen valores reales o vectores con valores reales.
- Todas las neuronas se relacionan con todas las neuronas, incluyendo las neuronas de la capa oculta.

2.4.1.2 Componentes del perceptrón multicapa

- Capa de entrada: solo se encarga de recibir las señales de entrada y propagarla a la siguiente.
- Capa de salida: proporciona al exterior la respuesta de la red para cada patrón de entrada.
- Capas ocultas: realizan un procesamiento no lineal de los datos de entrada.

En la figura 2.14 se muestra un ejemplo general de la arquitectura perceptrón multicapa:

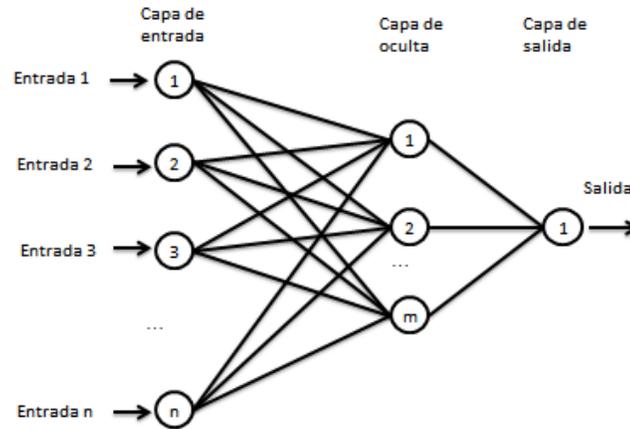


Figura 2.14. Arquitectura general perceptrón multicapa.

2.4.2 Red neuronal adaline/madaline

La arquitectura de adaline (adaptive linear neuron) fue creada por Bernard Widrow en 1959. La estructura del adaline es prácticamente idéntica al perceptrón simple, pero es un mecanismo físico capaz de realizar aprendizaje. Utiliza un dispositivo lógico que realiza una suma lineal de las entradas y genera una función umbral para el resultado de dicha suma. Se compone de una sola capa de n neuronas con m entradas [20].

El entrenamiento se realiza presentando repetidamente una serie de parejas de entradas y salidas. La regla de aprendizaje en la arquitectura de la red adaline es la regla de Widrow-Hoff expresada en la siguiente ecuación [21]:

$$\Delta w_i = \eta a_i (t * x) \tag{2.19}$$

siendo η la constante de aprendizaje, a_i la salida de la unidad i , t la salida deseada y por último x la salida de la unidad adaline.

En la figura 2.15 se muestra un ejemplo general de la arquitectura de la red adaline. La unidad procesador representada por un círculo sumatorio implementa una función umbral. Las conexiones de cada una de las entradas tienen asociado un valor de ponderación llamado también peso w_i . La salida de la unidad adaline es ± 1 .

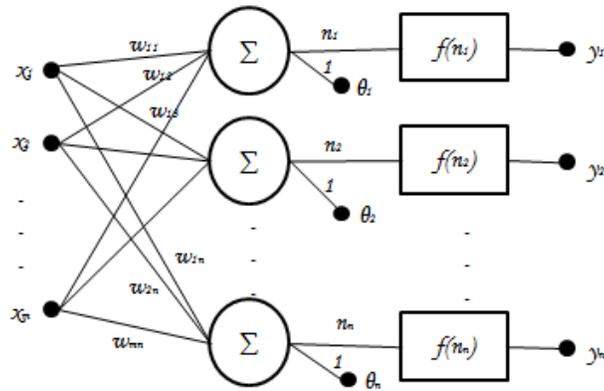


Figura 2.15. Arquitectura general adaline [21].

Existe una versión multicapa de la adaline denominada madaline (multilayer adaline) que consiste en una red neuronal con neuronas similares a las de adaline, pero contiene capas de neuronas ocultas. Fue creada también por Widrow, presenta una configuración constituida por dos o más unidades adaline. Una de las características más llamativas de la red madaline es la ausencia de pesos asociados a las conexiones entre las unidades adaline y madaline.

La red neuronal Madaline es una red neuronal que incluye una capa de entrada i , una capa intermedio u oculta formada por j neuronas o unidades adaline y una capa de salida con una única neurona denominada unidad madaline. En la figura 2.16 se muestra un ejemplo general de la arquitectura de la red madaline [21].

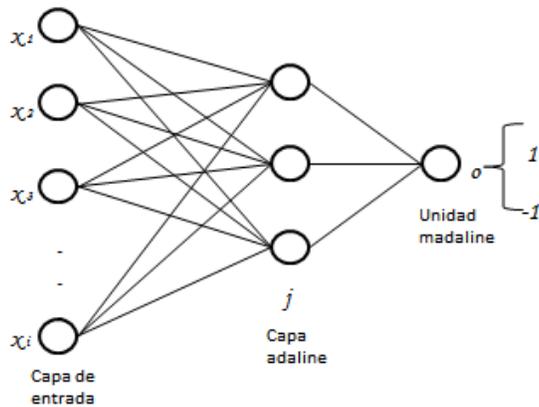


Figura 2.16. Arquitectura general madaline [21].

2.4.3 Red neuronal de función de base radial (RBF)

Las redes neuronales de función de base radial son uno de los modelos más utilizados debido a su rápida información, generalidad y simplicidad. Son modelos híbridos, esto es porque incorpora tanto el aprendizaje supervisado como no supervisado.

La parte no supervisada se encuentra desde el vector de las entradas a la capa oculta. En esta parte se decide el número de nodos radiales que cubran una gran parte del espacio definido por las entradas, mientras que la parte supervisada empieza desde la capa oculta a la capa de salida. La arquitectura de los modelos neuronales de función de base radial tiene tres capas, de entrada, oculta (realiza una transformación local sobre las señales que se reciben en la capa de entrada) y salida, cuya expresión es [19]:

$$y = \beta_0 + \sum_{j=1}^q \beta_j g_j(x_i) + \varepsilon \tag{2.20}$$

$$X = (x_1, x_2, \dots, x_p)$$

$$\{\beta_j, j = 1, 2, \dots, q\}$$

En la figura 2.17 se muestra de un modelo neuronal de función de base radial.

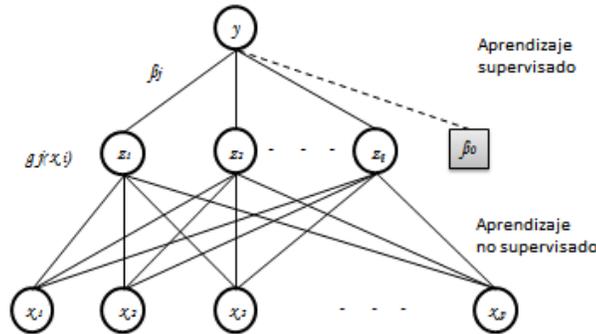


Figura 2.17. Modelo neuronal RBF [19].

2.5 Metodología para predicción con RNA

En este capítulo se explicó de manera breve que son las redes neuronales, los elementos básicos de una red neuronal, las principales funciones de activación, así como el entrenamiento o aprendizaje y generalización de una red neuronal y algunos modelos que son utilizados para pronosticar valores futuros.

A continuación se definirá el proceso previo a la implementación del modelo, donde se describirá cada uno de los pasos que se utilizarán y que abarcan los conceptos que se han explicado previamente.

1. *Escalamiento de los datos:*

$$Z_t = \frac{y_t - Min}{Max - Min} \quad (2.21)$$

donde:

y_t = *datos originales*

Min = *valor mínimo de la serie de tiempo*

Max = *valor máximo de la serie de tiempo*

2. *Dividir los datos en dos conjuntos:*

- El primer conjunto está compuesto por el 80% de datos, que serán utilizados para el entrenamiento de la red.
- El 20% de los datos de la serie se utilizará para evaluar la capacidad de generalización o predicción de la red.

3. *Definición de la arquitectura de la RNA.*

Un perceptrón multicapa está compuesto por una capa de entrada, una capa de salida y una o más capas ocultas, sin embargo, se ha comprobado que en la mayoría de los problemas es suficiente con una capa oculta. Se definirá de la siguiente forma en número de nodos que se tendrá en cada capa. En los siguientes pasos definirán en número de entradas y el número de nodos en la capa oculta que se tendrán en los modelos.

- Número de entradas: p .
- Número de nodos en la capa oculta: q .
- Número de nodos en la capa de salidas: 1.

Las conexiones entre neuronas que son siempre hacia delante, van desde las neuronas de la capa de entrada hacia la siguiente capa. Por lo tanto, la dirección de la información que utilizará es *alimentación adelantada* y el tipo de interconexión será *totalmente conectada*.

La función de activación para los nodos de la capa oculta y la capa de salida es la siguiente:

- Función de activación de los nodos de la capa oculta. Función logística $y = \frac{1}{1+e^{-v}}$
- Función de activación de la salida. Función logística $y = \frac{1}{1+e^{-v}}$

4. *Determinar las entradas (p)*

Para definir el número de entradas para el modelo se pueden seguir alguna de las siguientes recomendaciones:

- La periodicidad de los datos. La temporalidad de los datos se refiere al periodo en el que fue medida cada una de las observaciones de cada variable, esto es con el fin de determinar el número de entradas que tendrá la RNA. Tratándose de una serie de tiempo mensual, se tomará la idea de la temporalidad para definir el número de entradas en la red, de forma que para este ejemplo se considerarán 12 entradas más el sesgo.
- Prueba por ensayo y error. Se pueden probar con diferentes periodos, por ejemplo, 1,2,3,4 periodos, 2,4,6,8,10,12 periodos, etc.
- Las funciones de correlación de la serie de tiempo. Las funciones de correlograma simple y parcial de la serie estacionaria pueden considerarse como herramientas para determinar las entradas. Las entradas a la red, serán aquellos datos cuya correlación sea significativamente diferente de cero.

5. *Determinar el número de nodos en la capa oculta (q)*

El número de neuronas ocultas determina la capacidad de aprendizaje de la red neuronal. Para evitar el problema de sobre ajuste, se debe usar el mínimo número de neuronas ocultas con las cuales la red rinda de forma adecuada. Para determinar el número de nodos en la capa oculta existen dos alternativas:

- Pruebas por ensayo y error. Agregar más nodos y comparar los errores de ajuste y predicción.
- Otra forma, es la suma del número de nodos en la capa entrada y el número de nodos en la capa de salida, y dividir entre dos es decir: $(\#Entradas + \#salida)/2$. Si el valor que se obtuvo es decimal, se redondea. Es una regla ad hoc, que se ha utilizado en experimentos previos y ha resultado de utilidad [1].

6. *Algoritmo de entrenamiento: Backpropagation*

- Determinar los pesos iniciales aleatoriamente. Se generan aleatoriamente los pesos iniciales, en un rango de valores entre -0.5 y 0.5 o algo similar. Se recomienda probar con diferentes conjuntos de valores iniciales para obtener mejores resultados. Se seleccionará el modelo que obtenga el menor promedio entre la suma de cuadrados de los errores de ajuste y predicción.
- El valor de la tasa de aprendizaje (η) controla el tamaño de pesos en cada iteración. Toma valores entre 0.05 y 0.5 .
- El factor momento (α) acelera la convergencia de pesos. Toma un valor cercano a 1 .

- Condición de paro. Existen dos formas o dos criterios de parada en el entrenamiento de la red. Definir el número máximo de ciclos o definir el error permitido.
- Se calcula el error de entrenamiento mediante la ecuación 2.18. Error total sobre el conjunto de entrenamiento:

$$E = \sum_p^N E_p \quad (2.22)$$

7. Predicción

Evaluar la capacidad de la red con el 20% de los datos establecidos en el paso 2.

- Por medio de la ecuación de predicción definida por la RNA se obtiene el valor futuro $t + 1$. Para hacer predicciones para los periodos $t + n$, se utiliza como entrada a $t + 1$, para obtener el siguiente valor futuro $t + 2$ y así sucesivamente para todo el conjunto de predicción.
- Calcular el error de generalización mediante la ecuación la ecuación 2.18.

Error total sobre el conjunto de prueba (ecuación 2.22).

8. Comparación entre RNA con distintas entradas

Se seleccionara a la red, cuyos errores de entrenamiento y generalización sean mínimos. Si el error de entrenamiento es demasiado pequeño comparado con el error de generalización, indica que hay un sobre ajuste o memorización.

9. *Medidas de ajuste y predicción.* En el anexo C se muestran los criterios que se utilizaron para las medidas de ajuste y predicción.

2.6 Modelado con RNA

Para pronosticar el número de donaciones que se obtendrán en los próximos años usando RNA, se sigue la metodología que se describió en la sección 2.5. Se utilizará el lenguaje R para el desarrollo de modelo (entrenamiento y pruebas de las RNA).

1. Escalamiento de los datos

Los datos de la serie de tiempo son transformados a valores que se encuentren entre 0 y 1.

2. Dividir los datos en dos conjuntos (entrenamiento y prueba)

Los valores de la serie de tiempo se encuentran desde enero del 2003 a diciembre del 2014, es decir, 12 años, los datos se dividen de la siguiente forma:

- Patrones de entrenamiento: enero 2003 - diciembre 2012, es decir, 120 meses, que equivale al 80% de los datos.
- Patrones de prueba: enero 2013 - diciembre 2014, es decir 24 meses, que equivale al 20% de la población.

3. *Arquitectura de la RNA*

Las redes neuronales que se probarán tendrán la misma arquitectura, es decir, que serán de alimentación adelantada, totalmente conectada, se utilizará la misma función de activación en la capa oculta y en la capa de salida, con una capa oculta.

4. *Determinación de las entradas*

Para seleccionar las entradas se seguirán las recomendaciones de la sección 2.5 en el paso 3. Por ello tomaremos como entradas para las RNA las siguientes:

- Al tratarse de una serie temporal mensual se considerará 12 entradas y el sesgo.
- Una sugerencia de Faraway y Chatfield (1988) es el análisis de las matrices de pesos para seleccionar las entradas. Una propuesta para seleccionar las entradas de la siguiente red es obtener las medias de la matriz final de pesos de cada entrada de la primera red. Se consideran las tres medias mayores en valor absoluto. De la red de 13 entradas con medias mayores son 2, 3 y 5.
- De las 13 entradas se tomaron solo las pares: 2, 4, 6, 8, 10 y 12.
- Las entradas 1, 2, 3 y 4, son las entradas que tienen una correlación significativamente diferente de cero.

5. *Número de nodos en la capa oculta*

El número de nodos de la capa oculta de cada RNA se calculó con el promedio entre el número de entradas (p) y el número de salidas (1). En la figura 2.18 se muestra el número de nodos que se definieron para cada RNA.

RNA N°	Entradas	N° de entradas (p)	N° de nodos en la capa oculta (q)
RNA 1	1 a 13	13	7
RNA 2	2,3,5	3	2
RNA 3	2,4,6,8,10,12	6	3
RNA 4	1,2,3,9	4	2

Figura 2.18. Número de entradas y de nodos en la capa oculta.

6. *Entrenamiento de la RNA*

Las cuatro RNA fueron entrenadas con el algoritmo Backpropagation.

- Selección de los pesos iniciales. Los pesos iniciales se generaron para cada red neuronal haciendo uso del lenguaje R utilizando la función “runif”. Se probaron con diferentes conjuntos de pesos para cada red neuronal.
- El factor momento (α): 0.9
- Valor de la tasa de aprendizaje (η): 0.55. Se hicieron pruebas variando los valores de cada uno de los parámetros y se seleccionaron aquellos valores que obtuvieron la menor suma de cuadrados.
- Condición de paro. 120 ciclos.
- Se calcula el error de entrenamiento.

En la figura 2.19 se muestran los errores que se obtuvieron en el entrenamiento de cada una de las RNA. La RNA que obtuvo el menor error fue de 13 entradas.

RNA N°	N° de entradas (p)	Error de entrenamiento
RNA 1	13	1.84209
RNA 2	3	1.875485
RNA 3	6	1.950702
RNA 4	4	1.980089

Figura 2.19. Tabla de error de entrenamiento.

7. *Predicción*

Después de finalizar el entrenamiento con cada una de las RNA, se generaron las predicciones utilizando los patrones de prueba, para evaluar la capacidad de generalización de cada una de las RNA. En la figura 2.20 se muestra los errores de generalización de las tres RNA.

RNA N°	N° de entradas (p)	Error de generalización
RNA 1	13	0.4878023
RNA 2	3	0.5830511
RNA 3	6	0.5475792
RNA 4	4	0.5745004

Figura 2.20. Tabla de errores de generalización.

8. *Comparación y elección de la RNA*

Se comparan los errores de ajuste y de predicción de las cuatro RNA, en la figura 2.21 se muestran los errores de entrenamiento y generalización de cada una.

RNA N°	N° de entradas (p)	Error de entrenamiento	Error de generalización
RNA 1	13	1.842093	0.4878023
RNA 2	3	1.875485	0.5830511
RNA 3	6	1.950702	0.5475792
RNA 4	4	1.980089	0.5745014

Figura 2.21. Tabla de errores de entrenamiento y generalización.

En las figuras 2.22 y 2.23 se muestran los errores de ajuste y predicción de cada una de las RNA:

RNA N°	Ajuste			
	RMSE	MAPE	MAE	DER
RNA 1	14.9011	35.21611	11.8441	12.00181
RNA 2	15.6807	37.68146	12.9813	15.4235
RNA 3	15.393	37.01717	12.28964	14.69501
RNA 4	15.5936	37.55309	12.48692	15.26955

Figura 2.22. Tabla de errores de ajuste.

RNA N°	Predicción				
	RMSE	MAPE	MAE	RMSPC	CORR
RNA 1	23.38084	54666.38	21.11689	3.644983	0.6110085
RNA 2	25.56181	65340.59	23.50666	3.894353	-0.209841
RNA 3	24.77204	61365.38	22.64755	3.80376	0.3946441
RNA 4	25.37368	64382.35	23.30408	3.871921	-0.139484

Figura 2.23. Tabla de errores de predicción.

En las figuras 2.22 y 2.23, podemos ver que la RNA 1 tiene los menores errores de ajuste y predicción, al igual que los errores de entrenamiento y generalización, por lo tanto se selecciona la RNA 1.

Capítulo 3 Comparación de modelos

3.1 Análisis comparativo

Después de haber obtenido los mejores modelos de cada metodología y los pronósticos mediante los dos modelos, se realizará un análisis comparativo de cada modelo.

3.1.1 Ajuste y predicción

En la figura 3.1 se presentan los errores de ajuste y predicción de los mejores modelos, RNA y ARIMA seleccionados:

Modelo	Ajuste			Predicción		
	RMSE	MAPE	MAE	RMSE	MAPE	MAE
RNA 1 (13 Ent.)	14.9011	35.21611	11.8441	23.38084	54666.38	21.11689
ARIMA(0,1,1)	13.4019	40.7791	10.6396	12.4484	114.361	11.3772

Figura 3.1. Tabla de errores de ajuste y predicción.

Se observa que los errores de ajuste y predicción del modelo ARIMA son menores que los del la RNA. Para verlos de manera más clara, a continuación se presentarán las gráficas de la serie y de los modelos.

3.1.2 Gráficas de ajuste y predicción de los modelos

En la gráfica que se muestra en la figura 3.2 se observa la serie de tiempo contra el ajuste de los modelos ARIMA y RNA. El modelo ARIMA tiene un mejor ajuste a los datos reales de la serie de tiempo, así como también los pronósticos del modelo ARIMA se acercan más a los datos reales que los pronósticos de la RNA del año 2014.

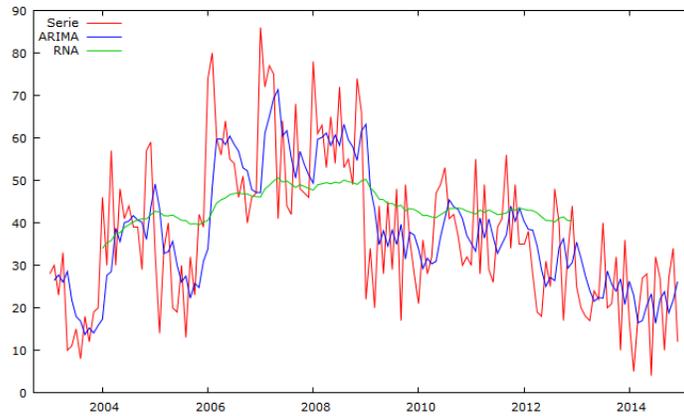


Figura 3.2. Gráfica de ajuste de la serie de tiempo con RNA y ARIMA.

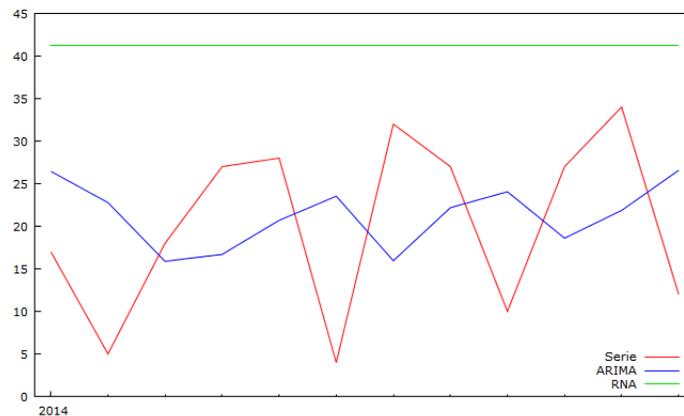


Figura 3.3. Gráfica de validación de pronósticos de los dos modelos.

3.2 Discusión del análisis

Loss errores de ajuste y predicción del modelo ARIMA son menores a los de la RNA, esto se corrobora en las figuras 3.2, en donde se observa que se tiene un buen ajuste a los datos de la serie original, lo mismo se destaca en la figura 3.3, en la gráfica de la validación de los pronósticos del año 2014 de cada modelo.

3.3 Pronósticos del modelo seleccionado

Los pronósticos que genera el modelo seleccionado $ARIMA(0, 1, 1)$ se muestran en la figura 3.4.

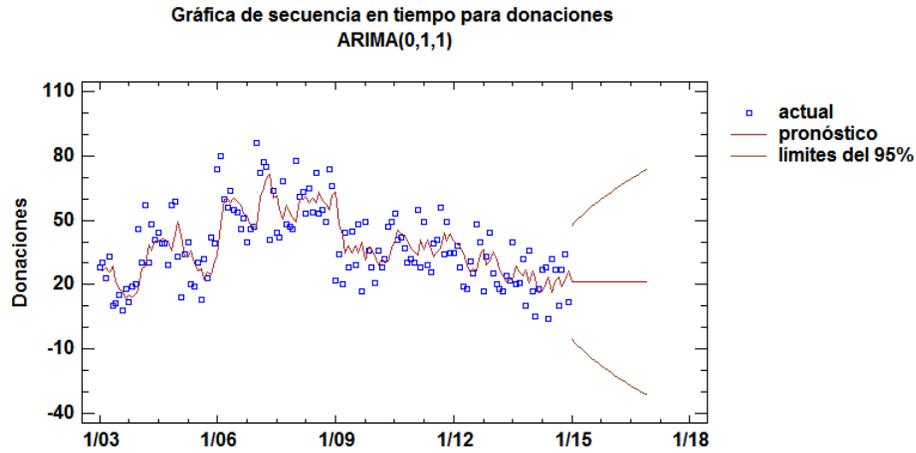


Figura 3.4. Ajuste del modelo ARIMA(0,1,1).

En la figura 3.5 se observa la gráfica de los pronósticos que genera el modelo, con sus correspondientes intervalos de confianza del 95% de predicción para los pronósticos. Los límites muestran en donde se podría ubicar el valor real del dato pronosticado.

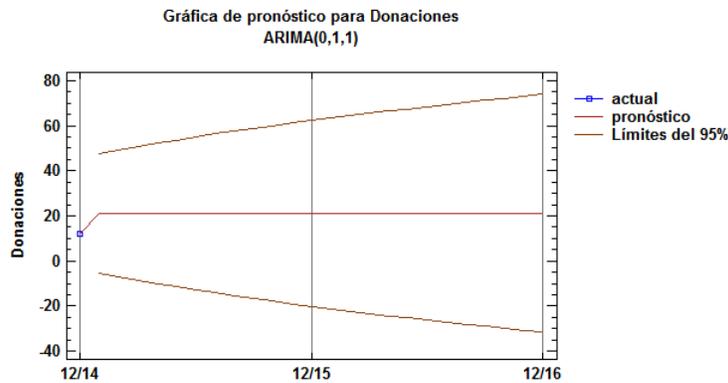


Figura 3.5. Pronósticos de donaciones del CNTS.

En la figura 3.6 se observa la tabla de los pronósticos de las próximas donaciones de células madre por cordón umbilical, a partir del año 2015 al año 2016 y sus límites, superior e inferior con un 95% de confiabilidad. Se observa que las donaciones de células madre por cordón umbilical se mantienen de manera constante.

<i>Periodo</i>	<i>Pronóstico</i>	<i>Inferior 95.0% límite</i>	<i>Superior 95.0% Límite</i>
1/15	21.0705	-5.42554	47.5666
2/15	21.0705	-7.08708	49.2281
3/15	21.0705	-8.6559	50.797
4/15	21.0705	-10.146	52.287
5/15	21.0705	-11.5681	53.7092
6/15	21.0705	-12.9308	55.0719
7/15	21.0705	-14.2409	56.382
8/15	21.0705	-15.5042	57.6452
9/15	21.0705	-16.7252	58.8663
10/15	21.0705	-17.908	60.0491
11/15	21.0705	-19.056	61.1971
12/15	21.0705	-20.172	62.3131
1/16	21.0705	-21.2586	63.3997
2/16	21.0705	-22.318	64.4591
3/16	21.0705	-23.3522	65.4932
4/16	21.0705	-24.3628	66.5038
5/16	21.0705	-25.3514	67.4925
6/16	21.0705	-26.3194	68.4605
7/16	21.0705	-27.268	69.4091
8/16	21.0705	-28.1984	70.3394
9/16	21.0705	-29.1115	71.2525
10/16	21.0705	-30.0083	72.1493
11/16	21.0705	-30.8896	73.0306
12/16	21.0705	-31.7562	73.8973

Figura 3.6. Tabla de pronósticos del año 2015 al año 2016.

Conclusiones

El objetivo de este trabajo de investigación fue obtener un modelo que fuera capaz de pronosticar el número de donaciones de células madre de cordón umbilical del CNTS en los próximos dos años, por medio del uso de dos metodologías: la metodología Box-Jenkins y por medio de redes neuronales artificiales, para comparar el resultado de ambas.

Para obtener un modelo que se ajuste mejor a los datos por medio de la metodología Box-Jenkins, la serie presentaba varianza constante, por lo cual no fue necesario aplicar alguna transformación; sin embargo, sí presentaba tendencia, por lo que se realizó una diferencia ordinaria. Una vez obtenida una serie estacionaria, se construyeron varios modelos tentativos, siguiendo los pasos de la metodología Box-Jenkins, y se seleccionó el modelo que mejor se ajustó a los datos para pronosticar.

En cuanto a la metodología empleada para elaborar una RNA, fue muy distinta. A diferencia de la metodología usada para tener un modelo ARIMA, que por medio de una serie de pasos proporciona un modelo específico, en las RNA no se tiene eso, no existen reglas como tales para construir un problema dado. Hay muchos factores a tomar en cuenta: el algoritmo de aprendizaje, la arquitectura, el número de neuronas por capa, el número de capas, la representación de los datos, entre otras.

Cabe destacar que aunque el modelo ARIMA tiene un mejor ajuste que el modelo de la RNA (figura 3.2) y la validación de los pronósticos se alejan de los valores reales (figura 3.3), los pronósticos que proporcionó la RNA se mantuvieron dentro de los intervalos de confianza de los pronósticos del modelo ARIMA. Sin embargo, los errores tanto de predicción, como de ajuste estaban por encima de los errores de ajuste y predicción del modelo ARIMA.

Los resultados expuestos en el capítulo 3, muestran que para el caso de aplicación, el modelo ARIMA tiene un mejor ajuste y por ende un mejor pronóstico. Obtener un pronóstico por medio de un modelo de una RNA, los datos no fueron suficientes. En el CNTS solo se tienen los datos desde el año 2003 al año 2014 y no se han actualizado los registros del último año.

El problema de predecir el número de donaciones de células madre del cordón umbilical, se justificó, dado el éxito y la importancia que han tenido en otros países para combatir enfermedades mortales; esto también podría aplicarse en México para mejorar la salud de los mexicanos, y no solo eso, también la manera de obtención de las células madre del cordón umbilical es muy fácil, sin dolor y sin ningún riesgo, tanto para el bebé como para la madre.

Es importante hacer la siguiente observación: en la serie de tiempo, el número de donaciones va disminuyendo y los pronósticos no indican que vaya a aumentar los siguientes meses, lo cual, no debería ser un problema en el país, donde cada minuto nacen cuatro personas, de acuerdo con la información del INEGI. Entonces, ¿por qué no va en aumento el número de donaciones de células madre del cordón umbilical?, este planteamiento puede ser el inicio de otra investigación, cuyo objetivo sería llegar a conocer estas causas.

Anexos

Anexo A

Tabla de datos de la serie de donaciones de células madre del cordón umbilical.

Forecast Table for Donaciones

Model: ARIMA(0,1,1)

<i>Period</i>	<i>Data</i>	<i>Forecast</i>	<i>Residual</i>
1/03	28.0		
2/03	30.0	26.47	3.53003
3/03	23.0	27.7396	-4.73956
4/03	33.0	26.035	6.96504
5/03	10.0	28.54	-18.54
6/03	11.0	21.872	-10.872
7/03	15.0	17.9618	-2.96184
8/03	8.0	16.8966	-8.89661
9/03	18.0	13.6969	4.30309
10/03	12.0	15.2445	-3.24453
11/03	19.0	14.0776	4.92238
12/03	20.0	15.848	4.15203
1/04	46.0	17.3413	28.6587
2/04	30.0	27.6485	2.35152
3/04	57.0	28.4942	28.5058
4/04	30.0	38.7464	-8.74642
5/04	48.0	35.6007	12.3993
6/04	41.0	40.0602	0.939821
7/04	44.0	40.3982	3.60181
8/04	39.0	41.6936	-2.69359
9/04	39.0	40.7248	-1.72483
10/04	29.0	40.1045	-11.1045
11/04	57.0	36.1107	20.8893
12/04	59.0	43.6236	15.3764

<i>Period</i>	<i>Data</i>	<i>Forecast</i>	<i>Residual</i>
1/05	33.0	49.1538	-16.1538
2/05	14.0	43.344	-29.344
3/05	34.0	32.7903	1.20966
4/05	40.0	33.2254	6.7746
5/05	20.0	35.6619	-15.6619
6/05	19.0	30.029	-11.029
7/05	30.0	26.0624	3.93759
8/05	13.0	27.4786	-14.4786
9/05	32.0	22.2713	9.72869
10/05	23.0	25.7703	-2.77027
11/05	42.0	24.7739	17.2261
12/05	39.0	30.9693	8.03065
1/06	74.0	33.8576	40.1424
2/06	80.0	48.295	31.705
3/06	60.0	59.6978	0.302208
4/06	56.0	59.8065	-3.80648
5/06	64.0	58.4375	5.56253
6/06	55.0	60.4381	-5.43805
7/06	54.0	58.4822	-4.48224
8/06	46.0	56.8702	-10.8702
9/06	51.0	52.9607	-1.96068
10/06	40.0	52.2555	-12.2555
11/06	46.0	47.8478	-1.84778
12/06	47.0	47.1832	-0.183216
1/07	86.0	47.1173	38.8827
2/07	72.0	61.1016	10.8984
3/07	77.0	65.0213	11.9787
4/07	75.0	69.3295	5.67054
5/07	41.0	71.3689	-30.3689
6/07	64.0	60.4466	3.55339
7/07	44.0	61.7246	-17.7246
8/07	42.0	55.3499	-13.3499
9/07	68.0	50.5485	17.4515

<i>Period</i>	<i>Data</i>	<i>Forecast</i>	<i>Residual</i>
10/07	48.0	56.825	-8.82503
11/07	47.0	53.6511	-6.65107
12/07	46.0	51.259	-5.25899
1/08	78.0	49.3676	28.6324
2/08	61.0	59.6653	1.33467
3/08	63.0	60.1453	2.85465
4/08	53.0	61.172	-8.17203
5/08	65.0	58.2329	6.76707
6/08	54.0	60.6667	-6.66673
7/08	72.0	58.269	13.731
8/08	53.0	63.2074	-10.2074
9/08	55.0	59.5363	-4.53628
10/08	49.0	57.9048	-8.90479
11/08	74.0	54.7022	19.2978
12/08	66.0	61.6427	4.35731
1/09	22.0	63.2098	-41.2098
2/09	34.0	48.3886	-14.3886
3/09	20.0	43.2137	-23.2137
4/09	44.0	34.8648	9.13522
5/09	28.0	38.1503	-10.1503
6/09	45.0	34.4997	10.5003
7/09	29.0	38.2762	-9.27617
8/09	48.0	34.94	13.06
9/09	17.0	39.6371	-22.6371
10/09	49.0	31.4956	17.5044
11/09	36.0	37.7911	-1.79109
12/09	28.0	37.1469	-9.14692
1/10	21.0	33.8572	-12.8572
2/10	36.0	29.2331	6.76694
3/10	28.0	31.6668	-3.66681
4/10	32.0	30.348	1.65197
5/10	47.0	30.9422	16.0578
6/10	49.0	36.7174	12.2826

<i>Period</i>	<i>Data</i>	<i>Forecast</i>	<i>Residual</i>
7/10	53.0	41.1349	11.8651
8/10	41.0	45.4022	-4.40222
9/10	42.0	43.8189	-1.81895
10/10	37.0	43.1648	-6.16476
11/10	30.0	40.9476	-10.9476
12/10	32.0	37.0102	-5.01024
1/11	30.0	35.2083	-5.20829
2/11	55.0	33.3351	21.6649
3/11	28.0	41.127	-13.127
4/11	49.0	36.4058	12.5942
5/11	29.0	40.9354	-11.9354
6/11	26.0	36.6428	-10.6428
7/11	39.0	32.8151	6.18495
8/11	41.0	35.0395	5.96051
9/11	56.0	37.1832	18.8168
10/11	34.0	43.9507	-9.95074
11/11	49.0	40.3719	8.62808
12/11	35.0	43.475	-8.47504
1/12	35.0	40.427	-5.42696
2/12	38.0	38.4751	-0.475135
3/12	28.0	38.3043	-10.3043
4/12	19.0	34.5983	-15.5983
5/12	18.0	28.9883	-10.9883
6/12	31.0	25.0363	5.96368
7/12	25.0	27.1812	-2.18118
8/12	48.0	26.3967	21.6033
9/12	40.0	34.1664	5.83359
10/12	17.0	36.2645	-19.2645
11/12	33.0	29.3359	3.66406
12/12	44.0	30.6537	13.3463
1/13	25.0	35.4538	-10.4538
2/13	20.0	31.694	-11.694
3/13	18.0	27.4882	-9.48823

<i>Period</i>	<i>Data</i>	<i>Forecast</i>	<i>Residual</i>
4/13	17.0	24.0758	-7.07575
5/13	24.0	21.5309	2.46907
6/13	22.0	22.4189	-0.418942
7/13	40.0	22.2683	17.7317
8/13	20.0	28.6455	-8.64555
9/13	21.0	25.5361	-4.53615
10/13	32.0	23.9047	8.0953
11/13	10.0	26.8162	-16.8162
12/13	36.0	20.7682	15.2318
1/14	17.0	26.2464	-9.24637
2/14	5.0	22.9209	-17.9209
3/14	18.0	16.4756	1.52443
4/14	27.0	17.0238	9.97616
5/14	28.0	20.6118	7.3882
6/14	4.0	23.269	-19.269
7/14	32.0	16.3388	15.6612
8/14	27.0	21.9714	5.02857
9/14	10.0	23.78	-13.78
10/14	27.0	18.824	8.17605
11/14	34.0	21.7645	12.2355
12/14	12.0	26.165	-14.165

Anexo B

1) Programa de entrenamiento del modelo RNA

```

#-----Lectura de datos-----
RNAMod<- read.csv(file="ModRNA.csv", header = TRUE)
RNAMod <- as.matrix(RNAMod[,2])
#-----
#pesos de la capa de entrada y oculta
pesos <- read.csv(file = "pesos.csv", header = T)
pesos <- as.matrix(pesos)
pesos <- pesos[,-1]
pesosEntradaOriginales<-pesos

#pesos de la capa oculta y salida
pesosCapOc <- read.csv(file = "pesosCapOc.csv", header = T)
pesosCapOc <- as.matrix(pesosCapOc)
pesosCapOc <- pesosCapOc[,-1]
pesosCapOcOriginal <- pesosCapOc
#-----
#Num. patrones de entrenamiento
PatEnt<-108
#Taza de aprendizaje
n<-0.55
#Momento
a<-0.9
#-----
DatosEntrada <- matrix(nrow=PatEnt, ncol = 12)
sesgo <- matrix(1,nrow = PatEnt,ncol = 1)
SalidasDes<-matrix(nrow=PatEnt, ncol = 1)
ErrorLamdaOc<-matrix(nrow = 7, ncol = 1)
ErrorEntrenamiento<-matrix(nrow = PatEnt, ncol = 1)
#patrones de entrenamiento
K<-0
for(i in 1:PatEnt)
{
  k<-i
  for(j in 1:13)
  {
    if(j!=13)
    {
      DatosEntrada[i,j]<-RNAMod[k]
      k<-k+1
    }
    else
    {
      SalidasDes[i,1]<-RNAMod[k]
      k<-0
    }
  }
}

Salidaestimada<-matrix(nrow = PatEnt, ncol = 1)
#agregar el sesgo a la matriz de patrones
DatosEntrada <- cbind(DatosEntrada,sesgo)

```

```

#-----Entrenamiento de la red-----
#-----Paso hacia adelante-----
for(patron in 1:PatEnt){
  for(i in 1:7){
    for(j in 1:13){
      pesosEntradaOriginales[i,j] <- pesos[i,j]
      pesos[i,j] <- pesos[i,j]*DatosEntrada[patron,j]
    }
  }

  SumaP <- rowSums(pesos)
  SumaP <- as.matrix(SumaP)

  for(i in 1:7){
    SumaP[i,1] <- (1/(1+exp(-1*SumaP[i,1])))
  }
  for(i in 1:7){
    pesosCapocOriginal[i]<- pesosCapoc[i]
    pesosCapoc[i]<-pesosCapoc[i]*SumaP[i,1]
  }
  pesosCapoc <- as.matrix(pesosCapoc)
  SumaNesa <- colSums(pesosCapoc)
  SumaNesa <- (1/(1+exp(-1*SumaNesa)))
#-----
Salidaestimada[patron,1]<-SumaNesa
ErrorEntrenamiento[patron]<-((SalidasDes[patron]-SumaNesa)^2)/2
#-----Paso hacia atrás-----

ErrorLamdaS<-SumaNesa*(1-SumaNesa)*(SalidasDes[patron]-SumaNesa)
for(i in 1:7)
{
  pesosCapoc[i]<-(n*ErrorLamdaS*SumaP[i,1])+(a*pesosCapocOriginal[i])
}

for(i in 1:7)
{
  ErrorLamdaOc[i,1]<-SumaP[i,1]*(1-SumaP[i,1])*ErrorLamdaS*pesosCapocOriginal[i]
}

for(i in 1:7)
{
  for(j in 1:13)
  {
    pesos[i,j]<-(n*ErrorLamdaOc[i]*DatosEntrada[patron,j])+(a*pesosEntradaoriginales[i,j])
  }
}
}#fin for general
#-----
#Error de conjunto de entrenamiento
SumErrorEntr<- colSums(ErrorEntrenamiento)

```

2) Programa de generalización del modelo RNA

```

#-----13 entradas-----
#-----

setwd("c:/Users/Rubi/Desktop/Excel/RNA/Obtener modelo RN")

#-----Datos de entrada-----
RNAMod<- read.csv(file="ModRNA.csv", header = TRUE)
RNAMod <- as.matrix(RNAMod[,2])
#MaFiPe13Momento.csv
PesosEntra<-read.csv(file="MaFiPe12Momento.csv", header = TRUE)
PesosEntra <- PesosEntra[,-1]
PesosEntra <- as.matrix(PesosEntra)

pesosOriginalEntradas<-PesosEntra
#MaFiPesOc13Momento.csv
PesosOculta<-read.csv(file="MaFiPesOc12Momento.csv", header = TRUE)
PesosOculta <- PesosOculta[,-1]
PesosOculta <- as.matrix(PesosOculta)
pesosCapOcOriginal<- as.numeric(PesosOculta)
max<-86
min<-4
PatGen<-24-12
DatosEntrada <- matrix(nrow=PatGen, ncol = 12)
sesgo <- matrix(1,nrow = PatGen,ncol = 1)
SalidasDes<-matrix(nrow=PatGen, ncol = 1)
MaPronosticos<-matrix(nrow=PatGen, ncol=1)

#Obtener patrones
h<-121
for(i in 1:PatGen)
{
  k<-0
  k<-h
  for(j in 1:12)
  {
    DatosEntrada[i,j]<-RNAMod[k]
    k<-k+1
  }
  h<-h+1
}

k<-133
for(i in 1:PatGen)
{
  SalidasDes[i,1]<-RNAMod[k]
  k<-k+1
}

#Errores de generalización
ErrorGeneralizacion<-matrix(nrow =PatGen, ncol = 1)
SalidasEstimadasPron<-matrix(nrow =PatGen, ncol = 1)
#agregar el sesgo a la matriz de patrones
DatosEntrada <- cbind(DatosEntrada,sesgo)

```

```

#-----Generalización de la red-----
#-----Paso hacia adelante-----
ContadorPr <- 1
for (patron in 1:PatGen)
{
  for(i in 1:7){
    for(j in 1:13){
      PesosEntra[i,j] <- pesosoriginalEntradas[i,j]*DatosEntrada[patron,j]
    }
  }
  #sumamos las entradas de cada neurona
  SumaPesos <- rowSums(PesosEntra)
  SumaPesos <- as.matrix(SumaPesos)
  #aplicamos la función de activación
  for(i in 1:7)
  {
    SumaPesos[i,1] <- (1/(1+exp(-1*SumaPesos[i,1])))
  }
  #se multiplica la salida de la neurona por el peso
  for(i in 1:7)
  {
    Pesosoculta[i]<-(as.numeric(pesosocoriginal[i]))*SumaPesos[i,1]
  }

  Pesosoculta <- as.matrix( as.numeric(Pesosoculta))
  SumaNeSa <- colSums(Pesosoculta)
  SumaNeSa <- (1/(1+exp(-1*SumaNeSa)))
  #-----Error de generalización de la red-----

  ErrorGeneralizacion[ContadorPr]<-((SalidasDes[patron]-SumaNeSa)^2)/2
  SalidasEstimadasPron[ContadorPr]<-SumaNeSa
  #-----Transformación de los datos-----
  MaPronosticos[ContadorPr,1]<-(((max-min)*SumaNeSa)+min)
  ContadorPr <- ContadorPr + 1
}

write.csv(SalidasEstimadasPron, file="SalidasGeneralizacionRed12Entradas.csv")
#-----Pronosticos
MaPronosticos
#-----Error total-----
SuErrorGeneralizacion<- colSums(ErrorGeneralizacion)
ErrorGener<-SuErrorGeneralizacion

```

3) Programa de ajuste

```

#-----
#Ajuste del modelo
#-----
n<-108
max<-86
min<-4
np<-(12*6)+6
#salidas deseadas
salidaestimada<-read.csv(file = "SalidasEstimadasRed12Entradas.csv", header = T)
salidaestimada<- as.matrix(salidaestimada[,2])
salidasDeseadas <- read.csv(file = "SalidasDeseadas.csv", header = T)
salidasDeseadas <- as.matrix(salidasDeseadas[,1])
salidasestimadas<-matrix(nrow = n, ncol = 1)
#residuos
Residuos<-matrix(nrow = n, ncol = 1)

#calcular Medidas de evaluación de ajuste

for(i in 1:n)
{
  salidasestimadas[i,1]<-(((max-min)*salidaestimada[i,1])+min)

  Residuos[i,1]<-(salidasestimadas[i,1]-salidasDeseadas[i,1])
}

ValoresRMSE<-matrix(nrow = n, ncol = 1)
ValoresMAPE<-matrix(nrow = n, ncol = 1)
ValoresRMSPC<-matrix(nrow = n, ncol = 1)
ValoresMAE<-matrix(nrow = n, ncol = 1)
ErrorEntrenamiento<-matrix(nrow = n, ncol = 1)
for(i in 1:n)
{
  ErrorEntrenamiento[i,1]<-((Residuos[i,1])^2)/2
  ValoresRMSE[i,1]<-(Residuos[i,1])^2
  ValoresMAPE[i,1]<-abs(Residuos[i,1]/SalidasDeseadas[i,1])
  ValoresRMSPC[i,1]<-(Residuos[i,1]/SalidasDeseadas[i,1])^2
  ValoresMAE[i,1]<-abs(Residuos[i,1])
}
ErrorTotalPatrones<-colSums(ErrorEntrenamiento)
SumaRMSE<-colSums(ValoresRMSE)
SumaMAPE<-colSums(ValoresMAPE)
SumaRMSPC<-colSums(ValoresRMSPC)
SumaMAE<-colSums(ValoresMAE)

RMSE<-sqrt(SumaRMSE/n)
MAPE<-(SumaMAPE/n)*100
RMSPC<-sqrt(SumaRMSPC/n)
MAE<-SumaMAE/n
DER<-sqrt(SumaRMSE/n-np)
Var<-SumaRMSE/n

```

4) Programa de validación de pronósticos

```

#-----
#Validación de pronosticos
#-----

n<-12
max<-86
min<-4
np<-(12*6)+6
#salidas deseadas
Salidaestimada<-read.csv(file = "SalidasGeneralizacionRed12Entradas.csv", header = T)
Salidaestimada<- as.matrix(Salidaestimada[,2])
SalidasDeseadas <- read.csv(file = "SalidasdeseadasPronosticos.csv", header = T)
SalidasDeseadas <- as.matrix(SalidasDeseadas[,1])
Salidasestimadas<-matrix(nrow = n, ncol = 1)
#residuos
Residuos<-matrix(nrow = n, ncol = 1)
#calcular Medidas de evaluación de ajuste

for(i in 1:n)
{
  Salidasestimadas[i,1]<-(((max-min)*Salidaestimada[i,1])+min)
}
for(i in 1:n)
{
  Residuos[i,1]<-(Salidasestimadas[i,1]-SalidasDeseadas[i,1])
}

ValorRMSE<-matrix(nrow = n, ncol = 1)
ValorMAPE<-matrix(nrow = n, ncol = 1)
ValorRMSPC<-matrix(nrow = n, ncol = 1)
ValorMAE<-matrix(nrow = n, ncol = 1)
ErrorGeneraizacion<-matrix(nrow = n, ncol = 1)

for(i in 1:n)
{
  ErrorGeneraizacion[i,1]<-((Residuos[i,1])^2)/2
  ValorRMSE[i,1]<-(Residuos[i,1])^2
  ValorMAPE[i,1]<-abs(Residuos[i,1]/SalidasDeseadas[i,1])
  ValorRMSPC[i,1]<-(Residuos[i,1]/SalidasDeseadas[i,1])^2
  ValorMAE[i,1]<-abs(Residuos[i,1])
}
ErrorTotalPatrones<-colSums(ErrorGeneraizacion)
SumRMSE<-colSums(ValorRMSE)
SumMAPE<-colSums(ValorMAPE)
SumRMSPC<-colSums(ValorRMSPC)
SumMAE<-colSums(ValorMAE)

RMSE<-sqrt(SumRMSE/n)
MAPE<- (SumRMSE/n)*100
RMSPC<-sqrt(SumRMSPC/n)
MAE<-SumMAE/n
Var<-SumRMSE/n
Corr<-cor(Salidasestimadas,SalidasDeseadas)#correlación

RMSE
MAPE
MAE
RMSPC #o RECOMP
Corr

```

5) Programa para obtener los pronósticos de los próximos dos años

```

#-----Pronosticos-----
#-----Datos de entrada-----
DatosEntrada<- read.csv(file="pronTreceEntradas.csv", header = TRUE)
DatosEntrada <- as.matrix(DatosEntrada)
DatosEntrada

PesosEntra<-read.csv(file="MaFiPe12Momento.csv", header = TRUE)
PesosEntra <- as.matrix(PesosEntra)
PesosEntra <- PesosEntra[,-1]
pesosOriginalEntradas<-PesosEntra

PesosOculta<-read.csv(file="MaFiPesoc12Momento.csv", header = TRUE)
PesosOculta <- as.matrix(PesosOculta)
PesosOculta <- PesosOculta[,-1]
pesosCapocOriginal<- as.numeric(PesosOculta)
max<-86
min<-4
MaPronosticos<-matrix(nrow=24, ncol=1)
...

-----Entrenamiento de la red-----
-----Paso hacia adelante-----
contadorPr <- 1

r(k in 1:24){

for(i in 1:6){
  for(j in 1:12){

    PesosEntra[i,j] <- pesosOriginalEntradas[i,j]*DatosEntrada[j]
  }
}
#sumamos las entradas de cada neurona
SumaPesos <- rowSums(PesosEntra)
SumaPesos <- as.matrix(SumaPesos)

#aplicamos la función de activación
for(i in 1:6)
{
  SumaPesos[i,1] <- (1/(1+exp(-1*SumaPesos[i,1])))
}

#se multiplica la salida de la neurona por el peso
for(i in 1:6)
{
  PesosOculta[i]<-(as.numeric(pesosCapocOriginal[i]))*SumaPesos[i,1]
}
PesosOculta <- as.matrix(as.numeric(PesosOculta))

SumaNeSa <- colSums(PesosOculta)
SumaNeSa <- (1/(1+exp(-1*SumaNeSa)))
#neuron
MaPronosticos[contadorPr,1]<-(((max-min)*SumaNeSa)+min)

for(h in 1:12){
  if(h==1){
    DatosEntrada[h,1] <- MaPronosticos[contadorPr,1]
  }
  else{
    DatosEntrada[h,1] <- DatosEntrada[h+1,1]
  }
}

  contadorPr <- contadorPr + 1
}
MaPronosticos

```

Anexo C

Medidas de ajuste y predicción

Raíz cuadrada del error medio (RMSE).

$$\sqrt{\frac{1}{n} \sum_{t=1}^n (\hat{y}_t - y_t)^2}$$

Error medio absoluto porcentual (MAPE).

$$100 * \frac{1}{n} \sum_{t=1}^n \left| \frac{\hat{y}_t - y_t}{y_t} \right|$$

Error medio absoluto (MAE).

$$\frac{1}{n} \sum_{t=1}^n |\hat{y}_t - y_t|$$

Desviación Estándar Residual (DER).

$$\sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n - r}}$$

Raíz del Error Cuadrado Porcentual (RMSPC).

$$100 * \sqrt{\frac{1}{n} \sum_{t=1}^n \left(\frac{\hat{y}_t - y_t}{y_t} \right)^2}$$

Coefficiente de correlación Pearson (Corr).

$$\frac{\frac{\sum \hat{y}_t * y_t}{n} - \bar{\hat{y}}_t * \bar{y}_t}{S_{\hat{y}_t} S_{y_t}}$$

Donde:

n = número de observaciones.

\hat{y}_t = valor estimado por el modelo en el periodo t .

y_t = valor observado de la variable en el periodo t .

r = número de parámetros estimados por modelo.

$S_{\hat{y}_t}$ = desviación típica de la variable \hat{y}_t .

S_{y_t} = desviación típica de la variable y_t .

Bibliografía

- [1] López, Raquel Flórez, and José Miguel Fernández Fernández. 2008. Las Redes Neuronales Artificiales.
- [2] López, Leonor Hernández. 2006. Predicción Y Optimización de Emisores Y Consumo Mediante Redes Neuronales En Motores Diesel. Reverte.
- [3] Díez, Raúl Pino, Alberto Gómez Gómez, and Nicolás de Abajo Martínez. 2001. Introducción a La Inteligencia Artificial: Sistemas Expertos, Redes Neuronales Artificiales Y Computación Evolutiva. Universidad de Oviedo.
- [4] “Células Madre de Cordón Umbilical. Obtención, Conservación Y Bancos.” <http://lascelulasmadre.es/cordon-umbilical>
- [5] Libro de La Salud Cardiovascular Del Hospital Clínico San Carlos Y La Fundación BBVA. 2007. Fundación BBVA.
- [6] “La Sangre Del Cordón Umbilical Nacersano.” <http://nacersano.marchofdimes.com/parto-y-nacimiento/la-sangre-del-cordon-umbilical.aspx>
- [7] “CNTS: México.” http://cnts.salud.gob.mx/interior/bscu_estadi.html (February 28, 2014).
- [8] “Cordones Para Salvar Vidas - QUO Mx.” <http://quo.mx/revista-quo/2014/05/29/cordones-para-salvar-vidas>.
- [9] “Donación de Células Madre de Cordón Umbilical.” http://www.medicable.com.mx/index.php?option=com_content&view=article&id=3340:donacion-de-celulas-madre-de-cordon-umbilical&catid=10&Itemid=168

- [10] “España Es El Segundo País Del Mundo En Donaciones de Cordón Umbilical.”
<http://www.lavanguardia.com/vida/20150213/54427204939/espana-es-el-segundo-pais-del-mundo-en-donaciones-de-cordon-umbilical.html>
- [11] “Memoria Anual 2014 Registro de Donantes de Médula Ósea (REDMO) by Fundación Josep Carreras Contra La Leucemia / José Carreras Leukaemia Foundation - Issuu.” http://issuu.com/fundacionjosepcarreras/docs/memoria_anual_2014_castella_web_/1?e=2896830/12423749
- [12] “Programa de Donantes de Médula Ósea – Estados Unidos de América.”
<http://www.celvitae.es/exito-creciente/%20gr%C3%A1fica%201.7>
- [13] González Videgaray, Mari Carmen. 2001. Pronósticos: Metodología de Box-Jenkins.
- [14] Colmenares, Gerardo. “TLU(s) MULTICAPAS Se Pueden Implementar Funciones En TLU Con Más de Una Capa.”
- [15] Marín Diazaraque, Juan Miguel. 2007. “Introducción a Las Redes Neuronales Aplicadas.” Manual Data Mining: 1–31.
halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema3dm.pdf.
- [16] STATGRAPHICS: «Statagraphics Centurion XVI.I para Windows Versión 16.01 » <http://www.statgraphics.com/>
- [17] SPSS: «IBM SPSS 21.0 para Windows» Software. <http://www-01.ibm.com/software/analytics/spss/>
- [18] R Studio: «R Studio 0.98» Software. <https://www.rstudio.com/>
- [19] Porras, Salvador Torra, and Enric Monte Moreno. 2013. Modelos Neuronales Aplicados En Economía: Casos Prácticos Mediante Mathematica / Neural Networks. Addlink Software Científico.
- [20] “Multicapa Sobreaprendizaje Perceptron Neuronas.”
<http://www.lab.inf.uc3m.es/~a0080630/redes-de-neuronas/perceptron-multicapa.html>
- [21] Basogain Olabe, Xabier. 2005. “Redes Neuronales Artificiales Y Sus Aplicaciones.”
Medicina Intensiva 29(1): 13–20. <http://linkinghub.elsevier.com/retrieve/pii/S021056910574198X>.

[22] Bruce L Bowerman, Richard T O'Connell, Anne B Koehler, Miguel Balderas Lozada. 2007. Pronósticos, series de tiempo y regresión : un enfoque aplicado.

[23] John E. Hanke and Dean W. Wichern. 2006. Pronósticos en los negocios.

[24] González Casimiro, María Pilar. 2009. Análisis de series temporales: Modelos ARIMA.