UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN

DISEÑO Y SIMULACIÓN ELÉCTRICA
DE UNA NEURONA PULSANTE
EN TECNOLOGÍA CMOS DE 0.6 MICRONES

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN TELECOMUNICACIONES, SISTEMAS Y ELECTRÓNICA

P R E S E N T A:

GONZÁLEZ FLORES JESÚS SALVADOR

UNAM
CUAUTITLÁN

ASESOR: ING. JOSÉ LUIS BARBOSA PACHECO

DR. VÍCTOR HUGO PONCE PONCE

Cuautitlán Izcalli, Estado de México, 2021.

ÍNDICE

Capítulo 1: Introducción

El progreso en tecnologías de hardware y software ofrece nuevas opciones cada año, la mayoría de las mejoras logradas hasta hoy en día se deben al avance tecnológico del hardware. Las tres primeras generaciones de computadoras fueron potenciadas con estas tecnologías (tubos de vacío, transistores, circuitos integrados de mediana escala (MSI), circuitos integrados de larga escala (LSI)) [1]. Desde la creación del transistor este ha servido como unidad fundamental de desarrollo en diversas arquitecturas de computadoras entre las que destacan las arquitecturas clásicas: Harvard y Von Neumann, además de arquitecturas avanzadas: sistemas de multiprocesadores, arquitecturas de flujo de datos, entre otras. Esto para conseguir un desempeño específico en la resolución de problemas que pueden no ser computables o consumir gran cantidad de potencia lo cual los vuelve costosos. Las arquitecturas modernas pueden ejecutar cálculos numéricos y operaciones simbólicas con rapidez, pero no se pueden comparar positivamente con el desempeño del cerebro humano, que puede manejar tareas perceptuales como el lenguaje y el reconocimiento de imágenes. Por estas razones se han desarrollado algoritmos que permiten explorar las capacidades actuales de cómputo en el ámbito de la *Inteligencia Artificial (IA)*. La *IA* es un área de estudio en las ciencias de la computación que tiene como objetivo crear sistemas que exhiben comportamientos que relacionamos con inteligencia como el razonamiento, toma de decisiones, comprensión del lenguaje, aprendizaje y resolución de problemas. La *IA* usa como herramientas algoritmos computacionales, cálculo multivariable, estadística, probabilidad y álgebra lineal. Esta área es una colección de algoritmos, los cuales poseen diferentes grados de complejidad, siendo el más sencillo un sistema experto, que logra tomar decisiones a partir de un conjunto de reglas de relaciones previamente definidas. Existe un área dentro de la inteligencia artificial denominada *Machine Learning*, la cual permite realizar la actualización de reglas de aprendizaje mediante la ejecución del cómputo de sus algoritmos. El estudio de los algoritmos permite construir modelos matemáticos basados en datos para generar los datos de salida o el comportamiento deseado, sin concebir el funcionamiento de las reglas que operan las relaciones del sistema.

Algunos de los algoritmos más comunes de *Machine Learning* son los siguientes:

Análisis de regresión: Es un conjunto de métodos estadísticos que permite examinar relaciones entre dos o más variables de datos y la influencia entre ellas en cuestión de dependencia e independencia.

Árboles de decisión: Son métodos de aprendizaje usados para clasificar y hacer regresión de datos. El objetivo es crear un modelo que realice predicciones de una variable, basada en reglas inferidas por las características de los datos usando estadística u otros algoritmos de machine learning. Pueden ser usados para representar gráficamente la toma de decisiones, ya que clasifica la información en clases y sus relaciones entre ellas en forma análoga a un árbol.

Algoritmos genéticos: Son usados en problemas de funciones discontinuas, no diferenciables y estocásticas, el cual se rige por reglas de selección, de modificación y de recombinación, con el objetivo de generar grandes grupos de nuevas aproximaciones en cada iteración acercándose a una solución óptima y seleccionar las mejores aproximaciones para repetir el proceso de cómputo con valores aleatorios.

Redes Neuronales Artificiales (*ANN*): Inspiradas en redes neuronales biológicas las cuales logran, mediante el uso de datos muestra, resolver tareas específicas asignadas mediante reconocimiento de patrones, sin ser programadas mediante reglas. Su relación con el sistema nervioso es su característica fundamental: un perceptrón que es el equivalente a una célula de neurona biológica, representada por funciones matemáticas la cual logra propagar la evaluación de los datos mediante sus conexiones con otros perceptrones a través de la red de perceptrones, cada uno evalúa continuamente los nuevos datos hasta la salida de la red artificial, donde se decide si el proceso de aprendizaje ha terminado.

Las *ANN* son algoritmos que con poca o nula supervisión humana se acercan a la solución del problema del usuario sin necesidad de pre-procesar los datos. Actualmente nos permiten realizar procesos de identificación y toma de decisiones con gran precisión y rapidez, estos han impulsado desarrollos tecnológicos que se han integrado a nuestras actividades

cotidianas, desde recomendaciones de noticias en un dispositivo móvil, hasta aplicaciones de gran importancia hoy en día; como poder asistir la creación de nuevos fármacos para tratamiento de enfermedades. Dichas redes neuronales artificiales operan sobre sistemas de cómputo basados en transistores de silicio que forman plataformas digitales masivas de cómputo, que comprenden billones de transistores en un solo sustrato de silicio.

La inteligencia artificial permite: optimización de software y hardware, desempeñar tareas complejas de visión por computadora, procesamiento de lenguaje natural, transcripción de voz y síntesis. Estas son áreas que han mejorado en los años recientes con el uso de la inteligencia artificial. Los algoritmos de aprendizaje de máquina y redes neuronales requieren de un avance continuo en software y hardware particularmente que sea de arquitectura paralela, debido a las operaciones de multiplicación de matrices del aprendizaje profundo de redes neuronales artificiales multicapa.

Existen una gran variedad de plataformas de hardware disponible en las que se desarrollan algoritmos de inteligencia artificial entre las que se encuentran CPU's, GPU's, servidores dedicados y FPGA's. Un CPU es un conjunto reducido de núcleos de procesamiento aritmético que realiza la ejecución de instrucciones en serie, que es flexible optimizando los procesos de cada núcleo en redes o conjuntos de datos pequeños con un consumo de energía muy reducido. Las plataformas GPU son arreglos de cientos hasta miles de procesadores más sencillos que los que componen un CPU, poseen la capacidad de poder realizar cómputos en paralelo, adecuado para aplicaciones de IA, esto se aprovecha en cómputos de grandes conjuntos de datos y redes neuronales artificiales con grandes cantidades de capas, además libera ciclos de trabajo al CPU a costa de un mayor consumo de energía. Dadas estas ventajas se adopta como acelerador de *IA* al GPU. Para permitir el uso de plataformas dedicadas existen servicios de cómputo en la nube, algunos de ellos diseñados específicamente para aplicaciones de IA, que pueden ser alquilados los recursos en horas para entrenar los modelos y probar novedosos algoritmos, para posteriormente distribuirlos en las plataformas que los utilizarán. Por esta razón se usan aceleradores de inteligencia artificial para mejorar el tiempo de entrenamiento de algoritmos de AI. Además, es común el uso de FPGA's por sus

características de hardware reconfigurable para implementar arquitecturas personalizadas al algoritmo en cuestión ya que los modelos siempre están en constante desarrollo.

1.1 Planteamiento del problema

Si bien es cierto que aún queda mucho trabajo por hacer en el campo de la *IA*, persiste el objetivo de lograr emular las características del cerebro, que desempeña tareas simultáneas de control, reconocimiento y movimiento con un consumo de energía del orden de los 20W, en contraste con un ordenador estándar reconociendo objetos entre un conjunto de 1,000 diferentes tipos de objetos consume alrededor de 250W [2], la operación de la redes neuronales artificiales en las arquitecturas modernas de cómputo se debe a los principios de la computación digital basada en silicio: la segregación de unidades de procesamiento y de almacenamiento, en contraste con las células corticales donde se lleva a cabo el cómputo co-localizado y almacenamiento en las sinapsis neuronales. Estos son mecanismos observados en el cerebro por la neurociencia. [3] Los transistores en sistemas digitales son ampliamente usados como *swtiches* para construir lógica booleana, mientras que el cómputo basado en pulsos de manejo de eventos en el cerebro es inherentemente estocástico. No existe una plataforma de hardware que cubra ampliamente las expectativas de cómputo biológico. Un segundo aspecto es la demanda de recursos computacionales de la inteligencia artificial en la que se generan redes y algoritmos neuronales más precisos. Un acercamiento es el cómputo neuromórfico introducido por *Carver Mead* en *Caltech* en 1980, para emular arquitecturas biológicas de cómputo presentes en el sistema nervioso a través de transistores, evolucionando rápidamente hasta el cómputo de manejo de eventos discretos (*pulsos neuronales*) durante los primeros años de la década de los años 2000. Los esfuerzos resultaron en el desarrollo de los denominados chips neuromórficos de alta escala de integración. Hoy en día el término neuromórfico se usa para describir, sistemas analógicos, digitales o mixtos en sistemas de muy alta escala de integración (*VLSI*). Podemos describir el área del cómputo neuromórfico como un esfuerzo sinérgico entre los dominios tecnológicos de hardware-software y la neurociencia para construir redes neuronales artificiales verosímiles. Por esta razón, un aspecto clave es entender los mecanismos cualitativos de las células corticales individuales para diseñar circuitos y arquitecturas que sean capaces de emular el comportamiento observado en sus contrapartes biológicas

operando como neuronas pulsantes. Los algoritmos aplicados en sistemas digitales son prácticos y convenientes, pero no son ideales para implementar sistemas en tiempo real o detalladas simulaciones de sistemas de muy alta densidad, incluso se necesitan recursos de una supercomputadora para obtener un desempeño deseable simulando sistemas robustos. Como, por ejemplo en la simulación de áreas del córtex cerebral de mamíferos. De esta manera, basados en modelos artificiales neuronales se incorporan las características cualitativas del sistema biológico del cerebro llevando a una computación *in-situ* donde memoria y procesamiento sean parte del mismo proceso, siendo un proceso analógico inherentemente estocástico y paralelo, compartiendo las mismas características cualitativas del cerebro, esto se logra trasladando el concepto de redes neuronales artificiales verosímiles en software a un circuito en silicio. El desarrollo de una plataforma de cómputo neuromórfico podría reducir la carga de procesamiento de otras arquitecturas de ordenadores, siendo una arquitectura dedicada para algoritmos de redes neuronales artificiales verosímiles, además de desempeñarse como una interfaz analógica-digital adquiriendo datos estocásticos y procesarlos sin algún acondicionamiento digital, para lograr reducir los sistemas necesarios para el procesamiento digital de datos entregando los resultados directamente a un ordenador común.

1.2 Justificación

Las neuronas implementadas en silicio son circuitos de muy alta escala integración (*VLSI*) híbridos analógicos/digitales que emulan el comportamiento electrofisiológico de las células corticales biológicas. Estas emulaciones del cerebro operan en tiempo real. La velocidad de la red es independiente del número de neuronas que se integran. Estos circuitos permiten emular directamente en hardware la dinámica neuronal individual de una célula o de una red en su conjunto. Este sistema permite la integración de otras soluciones prácticas que son inspiradas biológicamente como señales de cócleas y retinas neuromórficas. Existen diversos modelos neuronales que debido a su simplicidad, se implementan en estas plataformas, siendo un circuito de neurona el bloque fundamental de una red neuronal, hasta formar núcleos de cientos de neuronas formando redes de neuronas interconectadas por elementos resistivos con memoria. En el diseño de estos circuitos se deben tomar en cuenta aspectos ligados a la alta densidad de neuronas presentes en un chip neuromórfico, así como del

consumo global de energía. Es necesario también enfocar el esfuerzo de diseño para poder plantear un procesamiento de información paralelo, con memoria y procesamiento en el mismo circuito integrado.

Existen investigaciones en el área de nanotecnología que tienen como objetivo desarrollar exitosamente un nuevo dispositivo electrónico primitivo que sea compatible con representaciones de información no binaria. El *memristor* posee ventajas no lineales para una posible aplicación en tecnologías nanoelectrónicas, este logra variar su resistencia eléctrica y mantener almacenado ese valor, haciendo la función de una memoria no volátil. Este dispositivo puede escalarse hasta sub-10 nm, posee una alta velocidad de conmutación y compatibilidad CMOS. El *memristor* es considerado una probable memoria estándar para circuitos neuromórficos, lo cual elimina la necesidad de circuitos adicionales de memoria entre los diversos núcleos de neurona en silicio. Ejemplos de estos estudios: el reciente impacto de dispositivos memristores postulados por Leon Chua en 1971 **[3]**, debido a investigación en nanotecnología electrónica, la cual permite el mecanismo conocido como *Plasticidad de pulsos dependientes del tiempo* (*STDP*) que redefine la plasticidad Hebbiana sináptica propuesta por D.O Hebb **[4]** estas reglas de aprendizaje en una plataforma neuromórfica en la que el sistema de memoria esté conformado por memristores, permite llevar a cabo procesamiento de datos dentro de la memoria donde cada memristor es un peso sináptico ajustable. Este mecanismo sináptico es observado en sistemas nerviosos biológicos lo cual permite redes neuronales asíncronas en las cuales los impulsos neuronales son enviados hacia adelante y hacia atrás. Mediante distintos patrones de pulsos neuronales se puede manipular y ajustar las reglas de aprendizaje STDP para sinapsis excitatorias e inhibitorias. El memristor y un circuito de neurona usando las reglas STDP permite crear grandes redes neuronales artificiales verosímiles en silicio ya sea como un procesador, un sistema de autoaprendizaje visual artificial, hasta sistemas de reconocimiento de lenguaje.

Se resaltan los siguientes sistemas neuromórficos: el chip Truenorth diseñado para resolver problemas de clasificación y reconocimiento, Neurogrid para asistir computacionalmente a investigaciones neurocientíficas. Se espera que estas plataformas de procesamiento neuromórfico sean capaces de generar modelos de predicción basados en pequeños conjuntos de datos de muestras, que en componentes digitales computacionales no son capaces. Esto se

traduciría en una nueva clase de dispositivos que pueden ser entrenados mediante bajo consumo de energía y menor tamaño físico. El chip Loihi de Intel, el cual busca generar una plataforma neuromórfica que emule la red neuronal biológica que existe en el cerebro mediante redes pulsantes artificiales para aplicar dispositivos neuromórficos como sensores, y cámaras, además de implementar algoritmos de tercera generación de IA en la que se exploren soluciones para la ambigüedad, contradicción e interferencia en adoptaciones e interpretaciones de cognición autónoma. Hasta Marzo del año 2020 existen 4 aplicaciones de la plataforma Loihi con densidad 262k a 100M de neuronas.

1.3 Hipótesis

Diseñar un circuito analógico que emule el comportamiento observado en las neuronas biológicas, para demostrar la posibilidad de integrar un conjunto de neuronas pulsantes en un solo circuito integrado y desarrollar una plataforma de cómputo neuromórfico.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar un circuito de neurona analógica de tipo pulsante, mediante tecnología CMOS de 0.6 micrones, aplicable en el desarrollo de futuras plataformas de procesamiento neuromórfico.

1.4.2 Objetivos específicos

- Seleccionar una celda de neurona pulsante implementada como base de diseño y proyectar el circuito en tecnología CMOS de 0.6 micrones.

- Simular mediante el simulador de circuitos integrados (*SPICE*) algunos de los patrones de disparo observados en las neuronas biológicas.

- Desarrollar el diseño geométrico (*layout*) del circuito propuesto, conducir las simulaciones y depuraciones al circuito que permitan determinar su correcto funcionamiento, considerando los elementos parásitos extraídos.

Capítulo 2: Fundamentos de redes neuronales artificiales pulsantes

Las redes neuronales artificiales se caracterizan por sus capacidades de ser entrenadas bajo supervisión humana o sin ninguna supervisión y generar un aprendizaje automático, logrando describir patrones o reglas que no son identificables en conjuntos de datos, que de otra forma puede ser observado como un comportamiento aleatorio natural. Si un modelo matemático convenientemente describe un conjunto de datos ya conocidos, no es necesario el uso de estas herramientas de inteligencia artificial, pero si el conjunto de datos es parcialmente conocido puede usarse para descubrir relaciones entre los datos conforme se usan los datos entregados; sin embargo, estas herramientas son conocidas como *"solucionadores de cajas negras",* si necesitamos conocer las relaciones cualitativas de muestras de una base de datos, en las *ANN* no es posible, ya que no describe un modelo matemático, sino sólo aproximaciones numéricas. Pueden existir dos redes neuronales artificiales con diferentes pesos simpáticos y mismos resultados. Otro aspecto que dificulta extraer una función matemática definida para un problema dado, es que para el entrenamiento de los modelos neuronales se usan grandes cantidades de datos, parámetros y funciones de activación no lineales que lo convierte en un modelo difícil de interpretar. En proyectos relacionados con inteligencia artificial el objetivo es la fiabilidad de la predicción generada por las reglas de relación de los datos y no el conocimiento del modelo matemático, dejando de lado la naturaleza del algoritmo generado por la red neuronal mientras demuestre que es confiable. **[4]**

2.1 Redes neuronales biológicamente inspiradas

Existe gran cantidad de algoritmos de ANN desarrollados con aplicaciones definidas. Se describirán un par de estos algoritmos: la red neuronal artificial multicapa de retropropagación y red neuronal artificial convolutiva (*CNN*).

Una red neuronal artificial multicapa de retro propagación puede ser representada como un grafo cuyos nodos son unidades de cómputo que se conectan para transmitir la información de nodo a nodo. Cada unidad de cómputo es capaz de evaluar una función para

cada entrada. El grafo representa una cadena de funciones compuestas que transforma un vector entrada en un vector de salida. Uno de los objetivos del algoritmo de red artificial neuronal de retropropagación multicapa es corregir los errores de los pesos sinápticos en las capas entre la entrada y la salida, que se denotan como capas ocultas, empezando desde la capa de salida, con esto se dota de optimización al sistema. El algoritmo es usado para encontrar un mínimo local en una función de error estimado. En una red neuronal artificial de retropropagación como en la ***Fig.1*** se representan las entradas de señales, que se prolongarán desde la capa de entrada hasta la capa de salida. En este caso las salidas de las neuronas de la primera capa serán la entrada de la segunda capa escondida y así sucesivamente. Sin importar el número de capas que posea la red siempre seguirá un mismo flujo de dirección, propagación hacia delante, de la capa de entrada a la capa de salida. Se puede observar que un proceso de ***n*** salidas posee ***n*** neuronas en su última capa, las capas intermedias ocultas extraen la mayoría de la información relacionada al comportamiento del sistema y lo codifica usando pesos sinápticos. Los pesos sinápticos se usan como parámetros de ajuste en cada neurona, son determinados en procesos supervisados de entrenamiento o pueden asignarse aleatoriamente sin supervisión. Para cada muestra de entrada se realizan operaciones como lo denota la **Eq. (2.1)**.

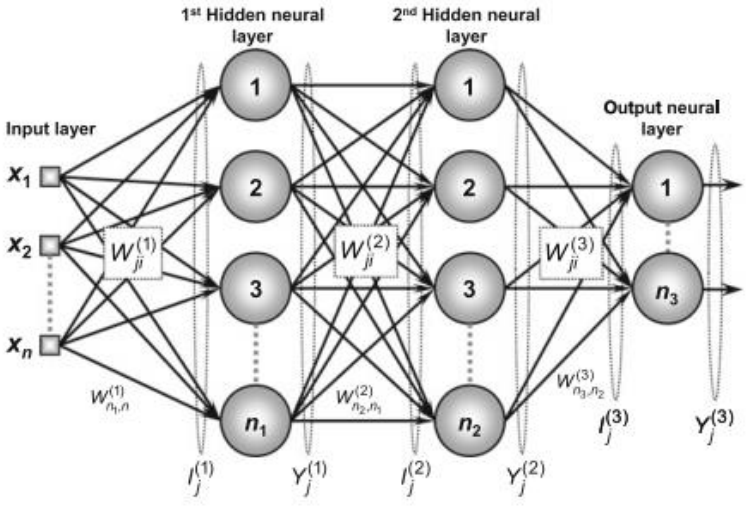

Fig.1 Red neuronal artificial multicapa de retropropagación [5].

- $\boldsymbol{W_{ji}^{(L)}}$ son los pesos sinápticos de las matrices, que conectan la neurona $\boldsymbol{j-}$ é$\boldsymbol{sima}$ de la capa $\boldsymbol{L}$ con la neurona$\boldsymbol{j-sima}$ de la capa ($\boldsymbol{L}$ -1).

- $\boldsymbol{I_j^{(L)}}$ Vectores cuyos elementos denotan los pesos de las entradas de la neurona $\boldsymbol{j-}$ é$\boldsymbol{sima}$ de una capa $\boldsymbol{L}$.
- $\boldsymbol{Y_j^{(L)}}$ son vectores cuyos elementos denota la salida de una neurona $\boldsymbol{j-}$ é$\boldsymbol{sima}$ relacionada a una capa $\boldsymbol{L}$.
- Cada neurona ***(j)*** pertenece a una capa ***(L)***.

El entrenamiento inicia propagando hacia adelante las entradas de la red neuronal. La etapa de entrada comienza obteniendo un conjunto de datos de entrenamiento X y se inicializan los vectores $\boldsymbol{W_{ji}^{(1)}}$, $\boldsymbol{W_{ji}^{(2)}}$, $\boldsymbol{W_{ji}^{(3)}}$ y empleando una función de activación $g(\cdot)$ para la salida de cada neurona. En la ***Fig.2*** se muestra que las neuronas de la capa de entrada realizan una sumatoria sobre los pesos sinápticos y los datos de entrada, como describe la **Eq. (2.1)**.

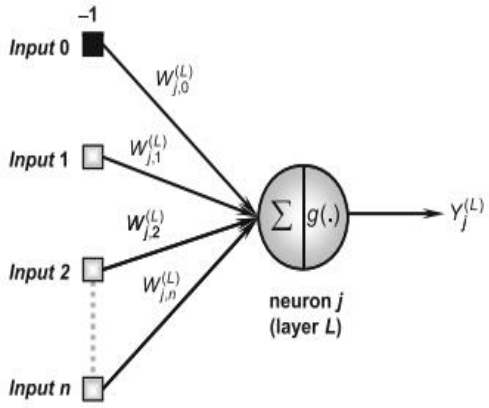

Fig.2 Función de activación y multiplicación de entradas y pesos sinápticos [5].

$$\boldsymbol{I_j^{(1)} = \sum_{i=0}^{n} W_{ji}^{(1)} \cdot x_i \leftrightarrow I_j^{(1)} = W_{1.0}^{(1)} \cdot x_0 + W_{1.1}^{(1)} \cdot x_1 + W_{1.n}^{(1)} \cdot x_n} \qquad (2.1)$$

Se observa que la neurona en la ***Fig.2*** realiza una operación de muestreo y una operación g*(·)* representa una *función de activación* que debe ser continua y diferenciable en todo el dominio, al cual se aplica en la **Eq. (2.2)**.

$$Y_j^{(1)} = g\left(I_j^{(1)}\right) \tag{2.2}$$

Una de las funciones de activación más comunes para redes de retropropagación es la función *sigmoide* definida por la **Eq. (2.3)**.

$$s_c(x) = \frac{1}{1+e^{-cx}} \tag{2.3}$$

Se muestran diversas formas de sigmoide en la cual se aprecia que entre mayor es la constante C, más se acerca a una función escalón unitario.

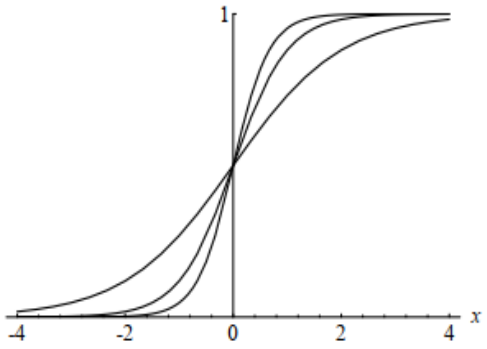

Fig.3 Tres sigmoides (par c=1, c=2 y c=3) [6].

Definir una *función de activación* permite aproximar el error para calcular la desviación producida por las neuronas de salida. El propósito de usar una función de activación es introducir una no linealidad, la función sigmoide convierte diferentes entradas de valores entre -∞ y +∞, a valores de salida entre 0 y 1. Una vez completada la propagación hacia delante en la capa de salida, se mide el error cuadrático. La *función de error cuadrado* es usada para medir el desempeño local producido por la capa de salida.

$$E(K) = \frac{1}{2}\sum_{j=1}^{n_3}\left(d_j(k) - Y_j^{(3)}(k)\right)^2 \tag{2.4}$$

Para un conjunto de ***p*** muestras, el desempeño total del algoritmo puede ser calculado usando la estimación error cuadrado.

$$E_M = \frac{1}{P}\sum_{K=1}^{P} E(k) \tag{2.5}$$

Donde ***E(K)*** es el error cuadrado obtenido en ***Eq. (2.4).*** Una vez medido el error, inicia la etapa de retropropagación del error, este es el procedimiento que distingue a este algoritmo, consiste en el ajuste de los pesos de entrada con respecto al ajuste de salida. Se logra medir el error local con los pesos sinápticos empleando la definición de gradiente descendiente y aplicando la regla de la cadena recursivamente para conectar los pesos de salida con la capa de entrada. La **Eq. (2.6)** para las neuronas de capa de salida muestra la relación del error de la salida de la capa de salida con su entrada.

$$\nabla E^{(3)} = \frac{\partial E}{\partial W_{ji}^{(3)}} = \frac{\partial E}{\partial Y_j^{(3)}} \cdot \frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}} \cdot \frac{\partial I_j^{(3)}}{\partial W_{ji}^{(3)}} \tag{2.6}$$

De la relación del error de neurona de salida con la entrada se deriva la regla generalizada de δ. Esto facilita el manejo aritmético del entrenamiento generalizando.

$$\delta_j^{(3)} = \frac{\partial E}{\partial W_{ji}^{(3)}} = \frac{\partial E}{\partial Y_j^{(3)}} \cdot \frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}} = \frac{\partial E}{\partial I_j^{(3)}} \tag{2.7}$$

Desarrollando los términos dentro de $\eth$**:**

$$\frac{\partial E}{\partial Y_j^{(3)}} = -\left(d_j - Y_j^{(3)}\right)$$

$$\frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}} = g'\left(I_j^{(3)}\right)$$

$$\delta_j^{(3)} = -\left(d_j - Y_j^{(3)}\right) \cdot g'\left(I_j^{(3)}\right)$$

Donde $\boldsymbol{g'}$ denota la primera derivada de la función de activación.

Para el último término de $\frac{\partial E}{\partial W_{ji}^{(3)}}$.

$$\frac{\partial I_j^{(3)}}{\partial W_{ji}^{(3)}} = Y_j^{(2)}$$

Resolviendo.

$$\frac{\partial E}{\partial W_{ji}^{(3)}} = -\left(d_j - Y_j^{(3)}\right) \cdot g'\left(I_j^{(3)}\right) \cdot Y_j^{(2)}$$

Una vez ajustada la matriz de pesos sinápticos $W_{ji}^{(3)}$ el gradiente debe tener dirección opuesta para minimizar el error:

$$\Delta W_{ji}^{(3)} = -\eta \cdot \frac{\partial E}{\partial W_{ji}^{(3)}} \leftrightarrow \Delta W_{ji}^{(3)} = \eta \cdot \delta_j^{(3)} \cdot Y_i^{(2)} \tag{2.8}$$

Donde $\boldsymbol{\delta}$ define el gradiente local relacionado a la enésima neurona en la capa de salida y η es el rango de aprendizaje al cual se asigna un valor y una precisión de error ($\boldsymbol{\varepsilon}$). La expresión puede ser simplificada en notación logarítmica para el ajuste o actualización del nuevo peso sináptico.

$$W_{ji}^{(3)} \leftarrow W_{ji}^{(3)} + \eta \cdot \delta_j^{(3)} \cdot Y_i^{(2)} \tag{2.9}$$

La ***Eq. (2.7)*** es el ajuste y actualización de los pesos sinápticos en la capa de salida de la red tomando en cuenta la diferencia observada entre los valores de la capa de salida y los valores deseados. El siguiente procedimiento ajusta los pesos sinápticos en las capas ocultas, ajustan mediante estimaciones de error las capas adyacentes, estas no pueden obtener estimaciones de error directamente de la capa de salida, así que las obtienen de la capa anterior.

$$\nabla E^{(2)} = \frac{\partial E}{\partial W_{ji}^{(2)}} = \frac{\partial E}{\partial Y_j^{(2)}} \cdot \frac{\partial Y_j^{(2)}}{\partial I_J^{(2)}} \cdot \frac{\partial I_j^{(2)}}{\partial W_{ji}^{(2)}} \tag{2.10}$$

La derivación de la regla de la cadena es como se muestra:

$$\frac{\partial E}{\partial Y_j^{(2)}} = \sum_{k=1}^{n_3} \frac{\partial E}{\partial I_k^{(3)}} \cdot \frac{\partial E I_k^{(3)}}{\partial Y_j^{(2)}} = \sum_{k=1}^{n_3} \frac{\partial E}{\partial I_k^{(3)}} \cdot \frac{\partial E I_k^{(3)}}{\partial Y_j^{(2)}}$$

$$= \sum_{k=1}^{n_3} \frac{\partial E}{\partial I_k^{(3)}} \cdot \frac{\partial \left(\sum_{k=1}^{n_3} W_{ki}^{(3)} \cdot Y_j^{(2)} \right)}{\partial Y_j^{(2)}} = \sum_{k=1}^{n_3} \frac{\partial E}{\partial I_k^{(3)}} \cdot W_{kj}^{(3)}$$

$$\frac{\partial E}{\partial Y_j^{(2)}} = -\left(d_j - Y_j^{(3)} \right) \cdot g'\left(I_j^{(3)} \right) \cdot W_{kj}^{(3)}$$

$$\frac{\partial E}{\partial Y_j^{(2)}} = -\left(\sum_{k=1}^{n_3} \delta_k^{(3)} \cdot W_{kj}^{(3)} \right)$$

$$\frac{\partial Y_j^{(2)}}{\partial I_J^{(2)}} = g'\left(I_j^{(2)} \right)$$

$$\frac{\partial I_j^{(2)}}{\partial W_{ji}^{(2)}} = Y_i^{(1)}$$

De donde $\delta_j^{(2)}$ para la segunda capa oculta está dada por:

$$\delta_j^{(2)} = -\left(\sum_{k=1}^{n_3} \delta_k^{(3)} \cdot W_{kj}^{(3)} \right) \cdot g'\left(I_j^{(2)} \right)$$

La expresión indica actualización del peso sináptico de las neuronas de la segunda capa oculta, tomando en cuenta los valores de error de retropropagación originados en la capa de salida.

$$\boldsymbol{W_{ji}^{(2)} \leftarrow W_{ji}^{(2)} + \eta \cdot \delta_j^{(2)} \cdot Y_i^{(1)}} \tag{2.11}$$

De esta forma se puede retropropagar hacia las capas previas a la salida siguiendo la misma metodología. Para la capa de entrada se corrige el peso sináptico tomando en cuenta los errores originado en las capas de salida.

$$\boldsymbol{\nabla E^{(1)} = \frac{\partial E}{\partial W_{ji}^{(1)}} = \frac{\partial E}{\partial Y_j^{(1)}} \cdot \frac{\partial Y_j^{(1)}}{\partial I_J^{(1)}} \cdot \frac{\partial I_j^{(1)}}{\partial W_{ji}^{(1)}}}$$

$$\boldsymbol{\frac{\partial E}{\partial Y_j^{(1)}} = -\left(\sum_{k=1}^{n_3} \delta_k^{(2)} \ . \ W_{kj}^{(2)}\right)}$$

$$\boldsymbol{\frac{\partial Y_j^{(1)}}{\partial I_J^{(1)}} = g'\left(I_j^{(1)}\right)}$$

$$\boldsymbol{\delta_j^{(1)} = -\left(\sum_{k=1}^{n_3} \delta_k^{(2)} \ . \ W_{kj}^{(2)}\right) \ . \ g'\left(I_j^{(1)}\right)}$$

$$\boldsymbol{\frac{\partial I_j^{(2)}}{\partial W_{ji}^{(2)}} = x_i}$$

$$\boldsymbol{W_{ji}^{(1)} \leftarrow W_{ji}^{(1)} + \eta \cdot \delta_j^{(1)} \cdot x_i} \tag{2.12}$$

Este procedimiento es generalizado para todas las capas.

Se inicia un contador de épocas de entrenamiento y se repite hasta que el error cumpla con las condiciones del valor de precisión ε deseado. La combinación de pesos para la cual se minimice la función de error es considerada la solución del problema de aprendizaje.

$$\left|E_M^{Acutal} - E_M^{Anterior}\right| \leq \varepsilon \qquad (2.13)$$

Actualizando los datos de corrección de errores y repitiendo el proceso descrito, hasta que se considere el modelo lo suficientemente entrenado cuando el peor error cuadrado entre dos *época*s sucesivas es menor o igual al *factor de precisión (*$\mathcal{E}$) que se requiere para mapear el problema. La variable *época* puede ser usada como criterio de término de entrenamiento cuando se llega a un número de épocas predeterminado **[5].** En el ***anexo*** se presenta un ejemplo numérico de una red neuronal artificial de retropropagación con 2 neuronas de entrada, dos capas ocultas y una capa de salida.

Un problema con *ANN* de retropropagación es que en tareas de procesamiento de imágenes la red puede ser demasiado densa para algunas arquitecturas y deja de ser una implementación práctica. En este aspecto es conveniente usar las *Redes convolutivas* que logra reducir el número de parámetros en la red.

Las redes convolutivas (ConNets sobresalen en tareas de reconocimiento y clasificación de grandes cantidades de datos, pueden tomar imágenes como entradas de datos y no solo listas de atributos, las *redes convolucionales* pueden identificar bordes, esquinas y texturas mediante la aplicación de capas de convolución usadas como filtros de imagen y capas de agrupación para reescalar la imagen reduciendo el número de parámetros. El objetivo es generar mapas de características aprendidas por dichas capas de neuronas conocidas como *mapas característicos,* poseen los mismos pesos y umbrales de activación compartidas (para minimizar las variables de ajuste) para cada diferente parte de la imagen. Los *mapas característicos* son locales en dos dimensiones de un vector de entrada, es decir que es un plano seccionado de un volumen.

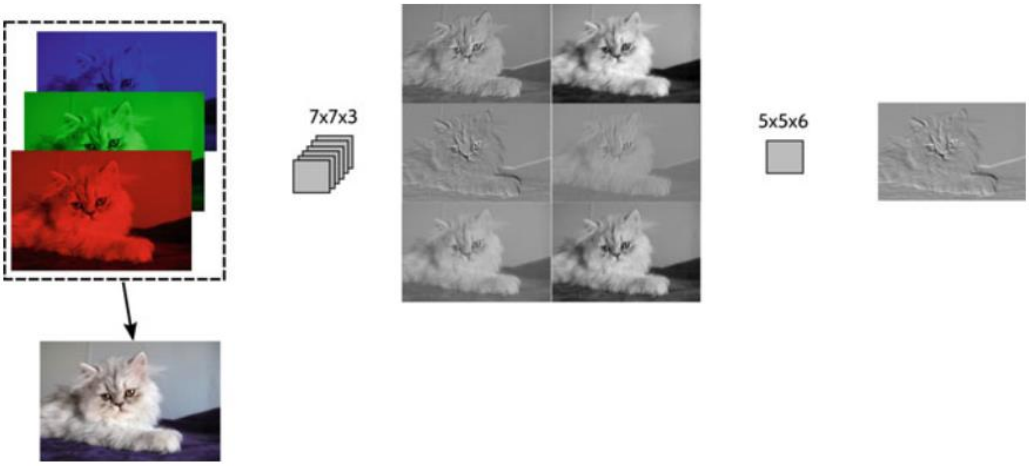

Fig.4 Filtro de convolución aplicado a una imagen [7] .

Una imagen a color posee ancho (W), alto (H) y tres canales de color R, G y B, una representación geométrica es un volumen. Debido a la gran cantidad de variables que se involucran se debe reducir el tamaño de la imagen. Generalmente se usan imágenes pequeñas de tamaño 32x32 RGB ya que al aplicar los filtros de convolución se seccionan bordes de la imagen. La idea es que puedan entrenarse a partir identificar características abstractas de dicha imagen **[5].** Cada neurona está conectada mediante la capa predecesora local, compartiendo las características locales.

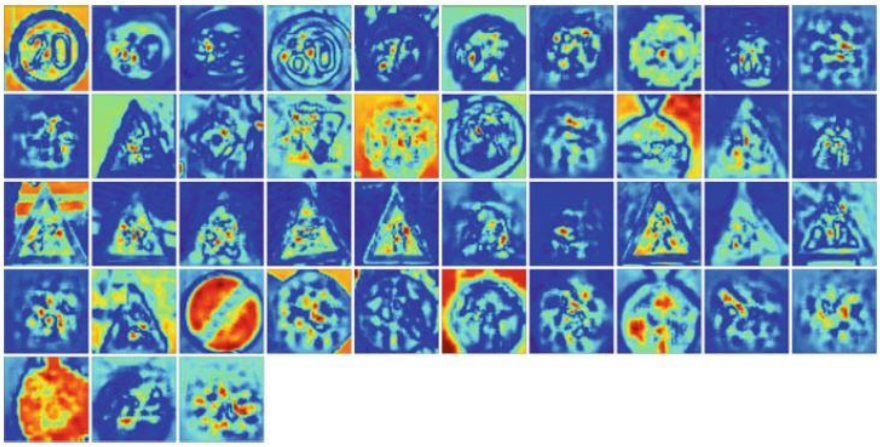

Fig.5 *Mapas característicos* de señales de tránsito [7].

El proceso de entrenamiento es que una imagen se introduce a una serie de capas de procesamiento, de la cual, la capa de operación convolución es de tamaño definido por el

diseñador, divide la imagen en diferentes secciones omitiendo en algunos casos los bordes, la operación convolución es la modificación de la imagen aplicando un filtro, con el propósito de resaltar características de una imagen como son los bordes, líneas, patrones y texturas, conforme más capas de convolución se aplique reconoce partes de objetos. Los filtros extraen características específicas locales de la imagen, conservando los canales de color y recorre toda la imagen aplicando un filtro a cada segmento de la imagen en cada canal de color, realizando el mismo procedimiento por cada canal de color aumentando la dimensión de volumen. El resultado de aplicar una capa de convoluciones es un mapa de mismas dimensiones WxH y con mayor número de canales, para esto después de una capa de convolución se aplica una *capa de agrupación* que toma de una región de convolución un valor característico, ya sea un valor máximo o un valor promedio por cada región, esto genera un escalamiento que puede reducir una imagen a mitad de tamaño.

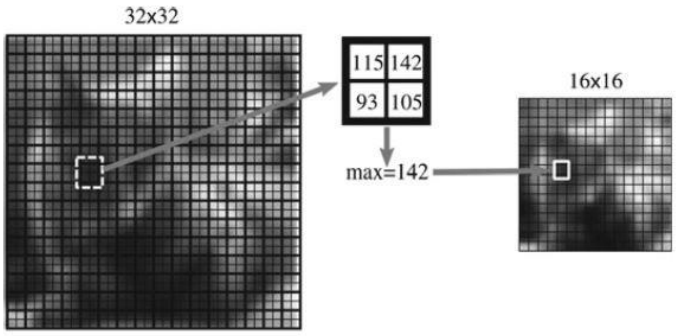

Fig.6 Aplicación de capa de agrupación "*pooling layer*" [7].

El siguiente paso en el entrenamiento es una capa de normalización en la que los valores de la capa de agrupación normalizan a un valor 0 valores negativos. La aplicación sucesiva de capas de convolución resalta y extrae características acumuladas de la imagen, no existe un método o regla de cuantos ciclos o capas, por esta razón se busca los filtros que sean adecuados para diferentes tipos de imágenes. Las capas finales son las capas de neuronas *completamente conectadas*, en las cuales las matrices de varias dimensiones se ordenan en un vector de una dimensión de varios elementos. La capa de salida mediante clasificación de probabilidades determina cuál categoría es la que más se acumulan clasificaciones. Este aprendizaje puede hacer uso de la técnica de retropropagación en la que el aprendizaje es

supervisado, se determina si la clasificación fue correcta y se ajustan los pesos mediante gradiente descendiente para minimizar el error.

El punto débil de las *ConNets* es que solo son útiles en módulos de clasificación y no directamente en tareas de detección, en tiempo real ya que cualquier falso positivo por el módulo detección entra al módulo de clasificación y es tomado en cuenta **[6] .** Para el entrenamiento de estos modelos de redes neuronales se usan conjuntos de datos de entrenamiento de imágenes clasificadas como MNIST que consiste de 60,000 imágenes 28x28x1 de dígitos escritos a mano entre 0 y 9 para clasificarlas en 10 categorías. Las redes neuronales artificiales con varias capas ocultas son llamadas redes de aprendizaje profundo, estos algoritmos pueden ser entrenados para clasificar imágenes con alta precisión, en algunos casos las redes convolucionales son mejores que los humanos ya que poseen un bajo error en tareas de clasificaciones.

Los algoritmos de redes neuronales están inspirados por la arquitectura y dinámica de las redes de células corticales en el cerebro, estos algoritmos son modelos abstractos altamente simplificados, no es un modelo verosímil. Las células corticales son de diferentes tipos, las dendritas pueden desempeñar cómputos complejos no lineales y las conexiones sinápticas no poseen solo un parámetro de ajuste sino un comportamiento de sistema dinámico por eso se dice que están "inspiradas" biológicamente, pero distan de tener esas características, es necesario describir un modelo verosímil.

2.2 Redes neuronales verosímiles

El córtex cerebral incluye una población de billones de neuronas y cada una posee cientos de contactos sinápticos lo cual brinda una complejidad inherente. Se han realizado esfuerzos para caracterizar electrofisiológicamente el sistema nervioso. Nuestro entendimiento de la dinámica cortical se amplía cuando agregamos propiedades intrínsecas del sistema biológico. Por ejemplo, los diferentes tipos de células corticales pueden generar una variedad de patrones de disparo en respuesta a una misma entrada, estos pulsos se propagan a velocidades variables. Las observaciones empíricas indican que la eficacia de que una sinapsis cambie rápidamente en respuesta a un preciso intervalo de patrón de disparo *pre* y *post sináptico* se

debe a un fenómeno referido como plasticidad de pulso neuronal dependiente del tiempo. Como resultado este sistema demuestra amplios patrones espacio temporales de actividad con la posibilidad de que las células corticales espontáneamente se reorganicen en grupos. [7] En la figura siguiente se observan diferentes caracterizaciones electrofisiológicas de interneuronas inhibitorias, divididas en cinco clases.

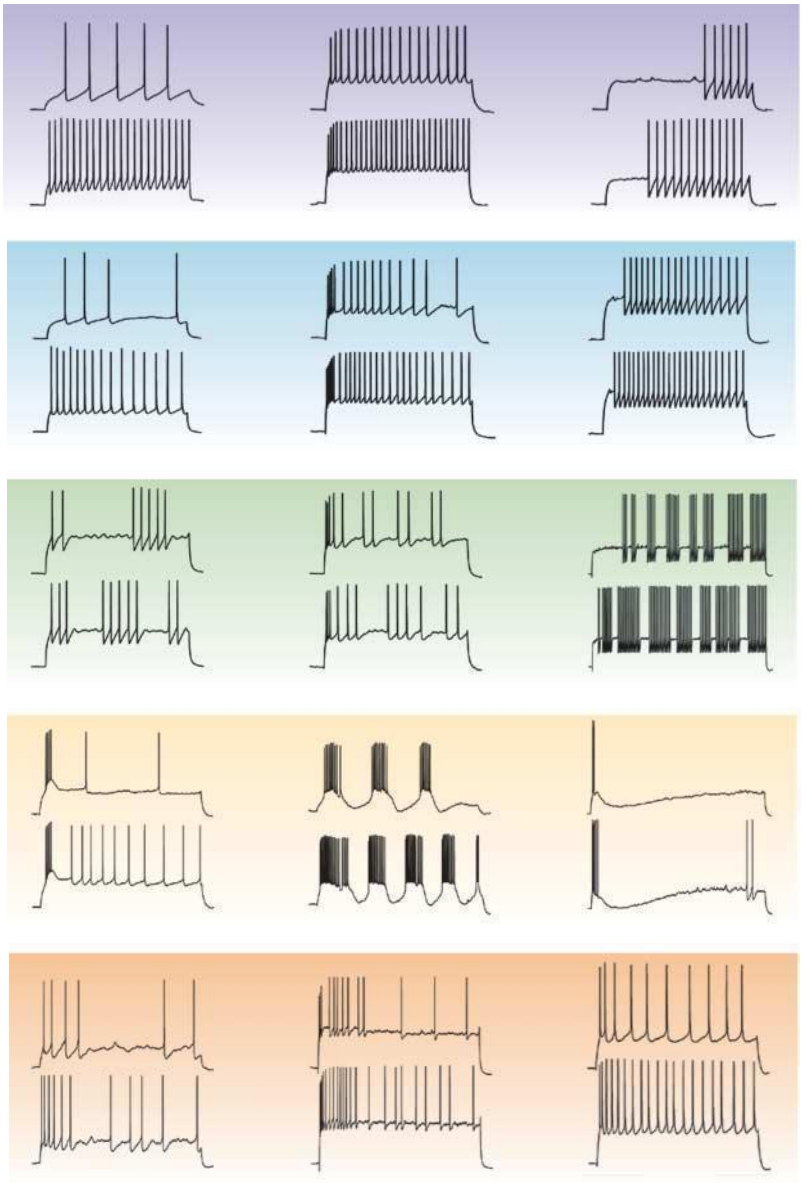

Fig.7 Caracterización electrofisiológica de interneuronas inhibitorias [8] .

El modelo de neurona debe de ser computacionalmente simple y capaz de producir una amplia cantidad de patrones de disparo, las cuales son propiedades que parecen mutuamente excluyentes. Un acercamiento a esto es el modelo generalizado simplificado llamado de *integración y disparo* que es computacionalmente posible y efectivo.

Los modelos fenomenológicos de neuronas pulsantes son populares para estudios de codificación neuronal, memoria en redes neuronales dinámicas, estos modelos de *integración y disparo* son modelos neuronales de umbrales de disparo en la que la forma del potencial de acción no es usada para transmitir información y pasan a ser considerados como eventos que caracterizan la llegada de un pulso a una sinapsis.

2.2.1 Modelo Generalizado de Integración y disparo

En el modelo generalizado de integración y disparo, los pulsos son generados cuando el potencial $\boldsymbol{u}$ cruza por el umbral $\boldsymbol{\theta_{reset}}$ por debajo. El instante en que se cruza el umbral define el tiempo de disparo $\boldsymbol{t^f}$.

$$\boldsymbol{t^f}: \mathrm{u}\left(\boldsymbol{t^f}\right) = \boldsymbol{\theta} \text{ y } \frac{\boldsymbol{du}}{\boldsymbol{dx}} \mathrm{t}=\boldsymbol{t^f} > \boldsymbol{0} \tag{2.14}$$

El modelo posee solamente una variable $\boldsymbol{u}$ que describe en el tiempo el potencial de membrana, el cual sigue la ecuación lineal diferencial ***Eq. (2.15)*****.**

$$\boldsymbol{\tau \frac{d}{dt} u = f(u) + R(u) I} \tag{2.15}$$

El sistema dinámico se detiene cuando el voltaje $\boldsymbol{u}$ alcanza el umbral $\boldsymbol{\theta_{reset}}$, $\boldsymbol{t^f}$ es definido y la ecuación de potencial ajusta una condición inicial de reinicio $\boldsymbol{u_r}$ en un tiempo $\boldsymbol{t^f + \Delta^{abs}}$.

El modelo *leaky integrate and fire* ***Eq. (2.16)*** es una recta geométrica de la cual se pueden modelar comportamientos sin estimulación, cuadráticos y exponenciales.

$$\boldsymbol{\tau \frac{d}{dt} u = -(u - urest) + RI} \tag{2.16}$$

Se muestra que la función no lineal ***R(u)*** puede ser interpretada como dependiente de voltaje de entrada mientras que una función no lineal ***f(u)*** reemplaza al término $-(\boldsymbol{u}-\boldsymbol{urest})$. Típicas funciones ***f(u)*** son la función exponencial y una función cuadrática.

En la **Fig.8** se observa el estado momentáneo de una ecuación diferencial unidimensional, descrita por una sola variable ***u***. La variable es graficada en el plano horizontal donde un incremento de voltaje representa una flecha con desplazamiento a la derecha y una disminución de voltaje con una flecha desplazándose a la izquierda. Para cada ***u*** en el eje horizontal corresponde $\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u})$,siendo ***f(u)*** una función arbitraria exponencial, se puede leer directamente el valor de flujo.

El cambio $\frac{\boldsymbol{d}}{\boldsymbol{dt}}\boldsymbol{u}$ del voltaje es graficado como función ***f(u)*** del voltaje ***u***, en la ausencia de estimulación I=0 las intersecciones con el eje horizontal definen el potencial de reposos ***urest*** y el umbral de disparo (V) del modelo no lineal de integración y disparo. Un cambio positivo en el potencial de membrana $\frac{\boldsymbol{d}}{\boldsymbol{dt}}\boldsymbol{u}$ ***f(u)>0*** implica que el voltaje aumenta, mientras que $\frac{\boldsymbol{d}}{\boldsymbol{dt}}\boldsymbol{u}$ ***=f(u)<0*** implica que el voltaje decrece, cuando el voltaje alcanza el valor ***θrest*** el voltaje se reinicia a un valor dado. En la **Fig.8** se observa que existen valores críticos fijos **f(u)=0** para la curva $\frac{\boldsymbol{du}}{\boldsymbol{dt}}$. Que son ***urest*** y (V).

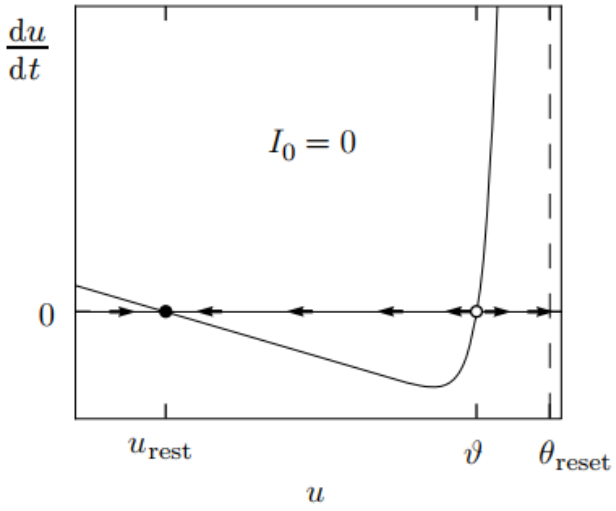

Fig.8 Umbrales en el modelo de integración y disparo para ausencia de estimulación [9] .

Existe un valor para $\boldsymbol{u}$ en la función, en el caso de una corriente nula sigue existiendo un flujo, un punto fijo ***urest*** es un punto estable de potencial de reposo, cualquier valor menor a (𝕍) pero mayor a cero lo llevará a ***urest***, si existe un valor igual a (𝕍) no fluirá hacia el reposo sino hacia el umbral, en el umbral será *restablecido* a un valor más bajo que puede ser ***urest***.

Existe otra dinámica en la que la corriente es diferente de cero, se explica a continuación. Supongamos que el sistema en un tiempo $\boldsymbol{t_0}$ se perturba ligeramente alrededor del punto fijo a un nuevo valor $\boldsymbol{u_0 + x(t_0)}$, en el cual el comportamiento de $\boldsymbol{x(t)}$, sigue la ecuación diferencial $\frac{\boldsymbol{dx}}{\boldsymbol{dt}} = \boldsymbol{f(u_0)} + \frac{\boldsymbol{df}}{\boldsymbol{du}}\boldsymbol{(u_0)}$ que se desprende de la ***Eq. (2.15)*** en un punto fijo $\boldsymbol{f(u_0)}$ y su solución es $\boldsymbol{x(t_0)exp[b(t - t_0)]}$. Si la curva **Fig.9** es igual a $\frac{\boldsymbol{df}}{\boldsymbol{du}}\boldsymbol{(u_0)}$ entonces es negativa la amplitud de perturbación x(t), descae hasta cero, indicando estabilidad. El valor negativo de la curva $\frac{\boldsymbol{df}}{\boldsymbol{du}}\boldsymbol{(u_0)} < 0$ implica estabilidad en un punto fijo.

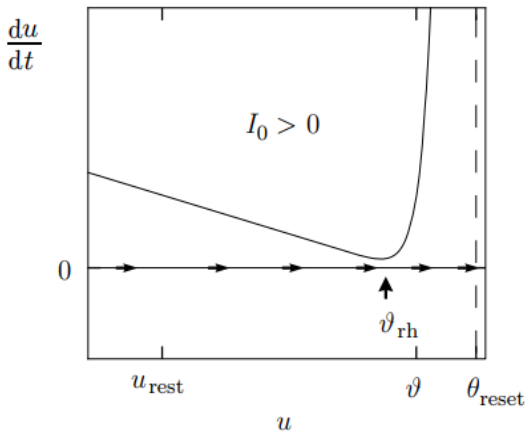

Fig.9 Umbrales en el modelo de integración y disparo para una constante positiva [9].

Si la corriente es positiva, la curva se desplaza verticalmente a un nuevo valor $\boldsymbol{f(u) + RI_0}$. Si la corriente es suficientemente grande los puntos fijos convergen y desaparecen y $\frac{\boldsymbol{du}}{\boldsymbol{dt}}$ siempre es positiva. El voltaje fluye directamente hacia el umbral $\boldsymbol{\theta reset}$ y es reiniciado.
La corriente necesaria en el régimen de disparos continuos corresponde al punto donde la intersección de voltaje desaparece $\boldsymbol{v_{rh}(ICr)}$ *reobase* o punto de bifurcación.

El modelo exponencial de integración es un caso especial del modelo general no lineal de integración y disparo, está definido por:

$$\boldsymbol{\tau \frac{d}{dt} u = -(u - urest) + \Delta_T exp\left(\frac{u - v_{rh}}{\Delta_T}\right) + RI} \qquad (2.17)$$

En el cual el término izquierdo describe el potencial de membrana, del lado derecho se nota que existe una componente lineal del modelo de *leaky integrate-and*-fire y un término no lineal exponencial con un umbral ($\boldsymbol{v_{rh}}$).

En la ausencia de una entrada externa (I=0) la ecuación diferencial del modelo exponencial posee dos intersecciones con el eje cero, el punto inestable fijo actúa como umbral para el pulso ubicado en el lado derecho ($\boldsymbol{v_{rh}}$).

Cuando la entrada externa aumenta se desplazan los puntos de intersección hasta que crean un punto de bifurcación determinado $\boldsymbol{\frac{d}{dt} u = 0 = u} = (\boldsymbol{v_{rh}})$, esta constante se nombra *reobase*.

Cuando el potencial de membrana es igual al umbral numérico $\boldsymbol{\theta reset}$ se define el tiempo de disparo. Después de un pulso el potencial es reajustado a un valor $\boldsymbol{ur}$ y la integración se reinicia a un tiempo $\boldsymbol{t^f}$. $+\boldsymbol{\Delta^{abs}}$. Donde $\boldsymbol{\Delta^{abs}}$. Es un tiempo de refracción.

Un punto de partida para describir modelos que puedan ser lo suficientemente simplificados para poder implementarlos en circuitos de alta densidad es el modelo de Hodgking-Huxley, en el cual se describen desde un punto de vista biofísico los potenciales de acción como resultado del paso de una corriente en canales iónicos de una membrana celular.

2.2.2 Modelo de Huxley and Hodgkin

El modelo de *H-H* es reconocido por modelar la dinámica de canales iónicos mediante su conductancia, originalmente se modelaron 3 canales iónicos, en la actualidad se pueden modelar alrededor de 200 canales iónicos que han sido descritos por la neurociencia. Las ecuaciones están basadas en detallados modelos neuronales para diferentes tipos de sinapsis. [9]

Las células corticales están separadas del medio por una membrana celular que consiste de capas de lípidos que forman un aislante eléctrico, la cual está formada de proteínas específicas que actúan como compuertas de iones. Dentro de la membrana existe una concentración de iones diferente a la del medio acuoso exterior. Esta diferencia de iones genera un potencial eléctrico que desempeña un papel principal en la dinámica neuronal.

El medio está conformado mayormente por sodio fuera de la célula y dentro existe mayor concentración de potasio. En equilibrio, la diferencia de concentración causa un potencial de Nerst $\boldsymbol{E_{Na}}$, este equilibrio genera un potencial de voltaje dentro de la célula más positivo que fuera de ella.

$$\boldsymbol{\Delta u = \frac{kT}{q} In \frac{n_2}{n_1}} \tag{2.18}$$

Siempre existe un contacto entre el medio interior de la célula y el exterior a través de los canales iónicos, donde los iones de Na+ pasan de un lado a otro.

Si la diferencia de voltaje Δu es más pequeño que el valor del potencial de nerst $\boldsymbol{E_{Na}}$, más iones de sodio fluyen hacia dentro de la célula y decrece la diferencia de concentración. Si el voltaje generado es mayor que el potencial de nerst los iones fluyen hacia fuera de la célula, esta dirección de corriente será inversa cuando el voltaje Δu sobrepase $\boldsymbol{E_{Na}}$, a esto se le conoce como potencial de inversión.

De esta forma los elementos que conforman el sistema pueden ser modelados. La membrana que separa el medio interior del medio acuoso actúa como un capacitor, la corriente de

entrada que es aplicada a la célula, puede agregar más carga al capacitor o fugarse a través de los canales de la membrana. Cada tipo de canal es representado por un resistor, existe un canal no especificado que posee fuga con una resistencia R_E, las resistencias son variables dependientes del voltaje, modelando el flujo del canal. La diferencia de potencial en la membrana es representada por una batería, el potencial de Nerst es diferente para cada tipo de ion, así que se presentan diferentes baterías para los canales.

El modelo se compone:

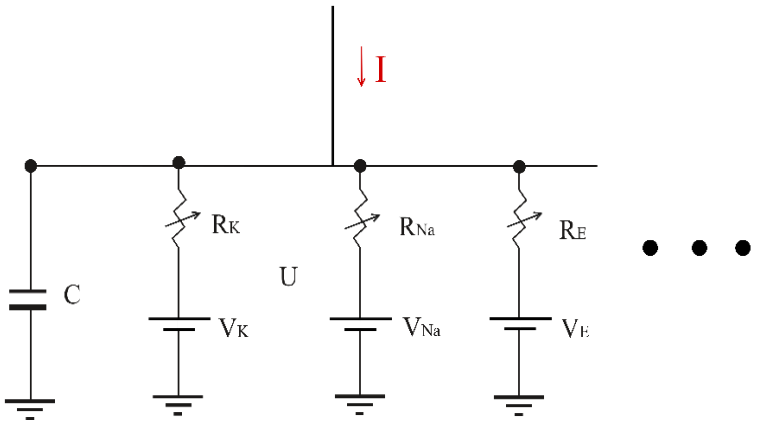

Fig.10 Representación circuital de modelo de Hodking and Huxley.

$$\boldsymbol{I(t) = I_c(t) + I_k + I_{Na} + I_E} \tag{2.19}$$

Donde sí se expresa en términos del voltaje de membrana

$$\boldsymbol{C\frac{du}{dt} = -\sum_I I_i \cdot (t) + I(t)} \tag{2.20}$$

Se indica que $\boldsymbol{u}$ es el voltaje a través de la membrana y la sumatoria $\sum_i \boldsymbol{I_i \cdot (t)}$ es la suma de las corrientes iónicas a través de la membrana. El modelo está escrito en términos de la resistencia eléctrica, puede escribirse como una conductancia dependiente de voltaje $\boldsymbol{G = \frac{1}{R}}$.

Siendo $\boldsymbol{u = R(I_R)}$

$$\boldsymbol{I_R = \frac{1}{R}(u)}$$

$$\boldsymbol{I_R = \frac{1}{R}(u)}$$

El voltaje total $\boldsymbol{u}$ es la diferencia entre el voltaje a través del capacitor y el voltaje de inversa de cada canal iónico calculado a través de la ecuación de Nerst, ya que describe el cambio de voltaje entre el exterior de la célula y su interior. Para el canal de potasio el voltaje total es:

$$\boldsymbol{I_k = \frac{1}{R_k}(u - E_k)} \tag{2.21}$$

Si se aplica para cualquier otra corriente se puede reescribir el voltaje de membrana, se obtiene la primera ecuación de Hodking and Huxley para la conservación de corriente.

$$\boldsymbol{C\frac{du}{dt} = -\frac{1}{R_k}(u - E_k) - \frac{1}{R_{Na}}(u - E_{Na}) - \frac{1}{R_E}(u - E_E) + I(t)} \tag{2.22}$$

Los parámetros $\boldsymbol{E_{Na}}$, $\boldsymbol{E_k}$ y $\boldsymbol{E_E}$ son las potencias de voltaje de inversa. Para describir la conductancia dependiente del voltaje y el tiempo si los canales están abiertos, entonces existe un flujo de corriente máximo, aunque normalmente algunos canales están bloqueados impidiendo el paso de la corriente, la propuesta matemática incluye variables de control de compuertas $\boldsymbol{m, n}$ y $\boldsymbol{h}$ que modelan la probabilidad de que el canal esté abierto en un tiempo dado, sea Na+ controlado por $\boldsymbol{m}$ y $\boldsymbol{h}$, K+ controlado por $\boldsymbol{n}$.

La conductancia efectiva de los canales de sodio se modela:

$$\boldsymbol{\frac{1}{R_{Na}} = (g_{Na}m^3)}$$

La variable $\boldsymbol{m}$ describe la activación del canal mientras $\boldsymbol{h}$ la bloquea.

$$\boldsymbol{C\frac{du}{dt} = -g_{Na}m^3h(u - E_{Na}) - g_k n^4(u - E_k) - g_E(u - E_E) + I(t)} \tag{2.23}$$

Las tres variables $\boldsymbol{m}$, $\boldsymbol{m}$ y $\boldsymbol{h}$ evolucionan de acuerdo con la ecuación diferencial:

$$\boldsymbol{\dot{x} = -\frac{1}{\tau_x(u)}\left[x - x_{0(u)}\right]} \tag{2.24}$$

Existe una ecuación diferencial para cada variable de canal, donde $\dot{x}$ son las variables m, n y h. Basado en las observaciones empíricas de H-H. Lo que forma un conjunto de 4 ecuaciones que formulan el modelo de H-H nos permite medir para cada variable de compuerta los valores estacionarios en cada canal.

EL modelo de H-H no es adecuado para modelos computacionales de gran densidad, pero nos permite simular una sola neurona en tiempo real, en contraste el modelo de I&F es incapaz de producir una amplia variedad de pulsos y comportamientos dinámicos reportados en células corticales.

2.2.3 Modelo simple de neuronas pulsantes de Izhikevich

El modelo simple de Izhikevich reproduce una amplia variedad de pulsos de neurona como post inhibitorios continuos, oscilaciones y resonancias, este modelo simple consta de dos ecuaciones diferenciales con una variable no lineal, es un modelo canónico en el sentido de que es preciso como el modelo de H-H que toma en cuenta la información de las corrientes iónicas. Debido a su sencillez este modelo simple puede simular redes corticales consistentes de decenas de miles de neuronas en tiempo real en una resolución de 1ms usando una computadora de 1Ghz. [10]

El trabajo de Izhikevich reduce modelos fisiológicos en modelos bidimensionales ordenándolos en ecuaciones diferenciales.

$$v' = 0.004v^2 + 5v + 140 - u + I \tag{2.25}$$

$$u' = a(bv - u) \tag{2.26}$$

$$\text{si } v \geq 30mV, \text{entonces } \{v \leftarrow c \;\; u \leftarrow u + d \tag{2.27}$$

En donde v y u son adimensionales como las variables a, b, c y d. La variable v representa el potencial de membrana de la neurona y u representa la variable de recuperación de

membrana, que activa el canal iónico de K+ y desactiva canal iónico de Na+ después de un pulso, la variable de reset se describe por la ***Eq.(2.27)*** Las corrientes sinápticas son entregadas por la variable $\boldsymbol{I}$. Los términos $\mathbf{0.004}\boldsymbol{v}^{\mathbf{2}} + \mathbf{5}\boldsymbol{v} + \mathbf{140}$ fueron determinados para poder lograr escalas de tiempo en el potencial de membrana de hasta ms. Dependiendo de los comportamientos anteriores de la membrana, el potencial de umbral puede ser tan bajo como $-\mathbf{55}\boldsymbol{mV}$ hasta $-\mathbf{40}\boldsymbol{mV}$. El parámetro $\boldsymbol{a}$ describe la escala de tiempo de la variable de recuperación, un valor pequeño resulta en una lenta recuperación. El parámetro $\boldsymbol{b}$ describe la sensibilidad de la variable de recuperación $\boldsymbol{u}$ en las fluctuaciones del potencial de membrana. El parámetro $\boldsymbol{c}$ describe después de un pulso el valor de reset del potencial de membrana $\boldsymbol{V}$ causado por un alto y rápido umbral de las conductancias de K +. El parámetro $\boldsymbol{c}$ describe un *reset* después de un pulso en la variable de recuperación de la variable y causado por un umbral alto y lento en las conductancias de Na + y K +. La variación de los parámetros muestra diferentes patrones de pulso intrínsecos algunos exhibidos por tipos de células corticales. Las células corticales de los mamíferos pueden clasificarse de acuerdo con el patrón de pulsos y ráfagas de pulsos observados. Se muestran algunos de los patrones de disparo excitatorios que genera este modelo, la versión digital de las figuras y su reproducción y permisos están libremente disponibles en *www.izhikevich.com.*

RS (*pulsos regulares*) los pulsos más típicos en el córtex, la neurona pulsa pocas veces en periodos cortos, entre pulsos y después los periodos incrementan. Es también llamada frecuencia de adaptación de pulsos.

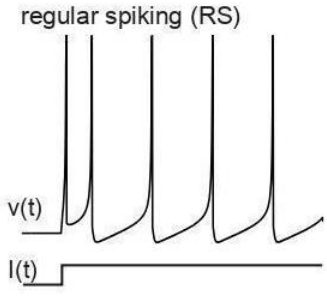

Fig.11 Patrón de disparo regular [10].

IB (*ráfagas intrínsecas*) las neuronas pulsan ráfagas en disparos simples y repetitivos. Durante la ráfaga inicial la variable $\boldsymbol{u}$ se incrementa y eventualmente la dinámica evoluciona de rafa a pulsos simples.

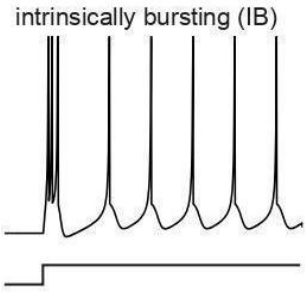

Fig.12 Patrón de disparo de ráfaga intrínseca [10].

CH (*ráfaga disparos irregulares*) las neuronas disparan rafas y los pulsos que lo componen son cercanos, describen frecuencias altas de hasta 40hz.

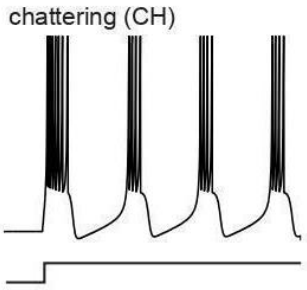

Fig.13 Patrones de disparo irregulares [10].

El modelo reproduce además patrones de disparo de células corticales inhibitorias como son los siguientes:

FS (*pulsos rápidos*) son pulsos periódicos a frecuencias muy altas prácticamente sin disminución o adaptación de frecuencia de pulsos.

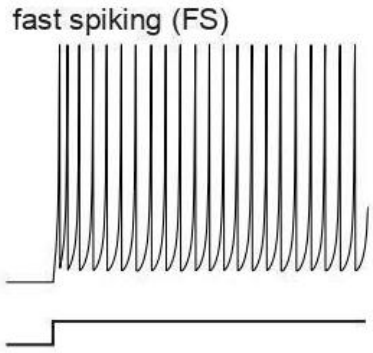

Fig.14 Patrones de disparo rápidos [10].

LTS (*pulsos de bajo umbral*) trenes de pulsos a frecuencias altas con pulsos intervalos breves de adaptación, bajo umbral de disparo.

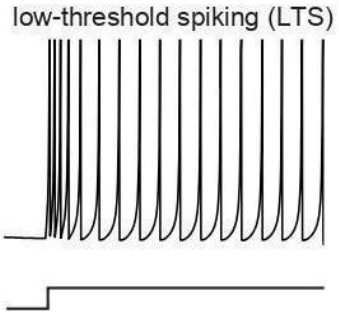

Fig.15 Patrones de disparo de bajo umbral [10].

TC (*talamocorticales*) poseen dos regímenes de disparo: cuando están en reposo y cuando exhiben disparos tónicos.

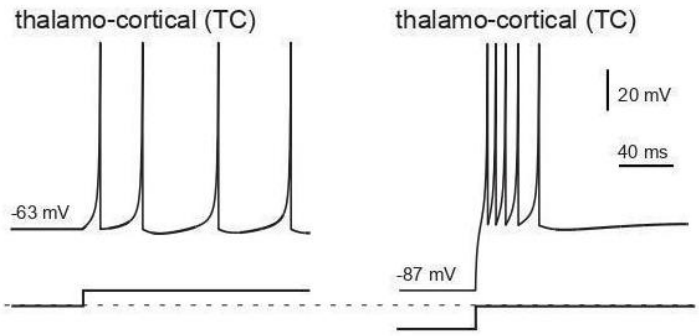

Fig.16 Patrones de disparo tálamo corticales [10].

Además, el modelo presenta un comportamiento dinámico adicional.
RZ (*resonador*) pulsos con un umbral elevado sostenido que produce oscilaciones. Puede cambiar de estado después de un estímulo adecuado en un periodo de tiempo.

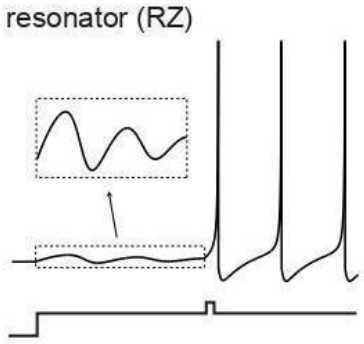

Fig.17 Patrones de disparo resonadores [10].

El modelo resulta de gran utilidad cuando se simulan redes neuronales pulsantes de alta densidad.

El modelo híbrido de redes neuronales pulsante de Izhikevich ha sido aplicado para reconocimiento de patrones mediante memorias asociativas, simulaciones realistas de modelos del cerebro, estas simulaciones no hubieran sido posibles en modelos basados en conductancias a menos que se realizarán con supercomputadoras.

Basados en los modelos y algoritmos anteriores el procesamiento de comunicación de información basado en pulsos es una característica de los sistemas en hardware que tienen como objetivo implementar una plataforma computación para redes neuronales artificiales pulsantes, usando las características físicas inherentes de los dispositivos Metal-óxido-silicio (MOS) donde el voltaje posee características no-lineales para simular la dinámica neuronal. Estos dispositivos forman parte del área del cómputo neuromórfico, integrando a su estudio materiales emergentes y dispositivos asociativos.

2.3 Circuitos integrados neuromórficos

Los laboratorios de Intel, desarrollaron en noviembre del 2017 un chip llamado "Loihi" de quinta generación de aprendizaje automático con propósitos de investigación y pruebas. Es un chip de 60$\boldsymbol{mm^2}$ fabricado en un proceso de 14nm FinFET implementa modelos de redes neuronales pulsantes en silicio. Integra conectividad jerárquica, tiempos de retraso sinápticos y posee reglas de aprendizaje sináptico programables. Se ha demostrado que el chip puede resolver mediante redes convolucionales pulsantes problemas de optimización de tipo LASSO con una superioridad en términos de eficiencia energética tres veces mayor que CPU'S convencionales del mismo proceso, voltaje y área.

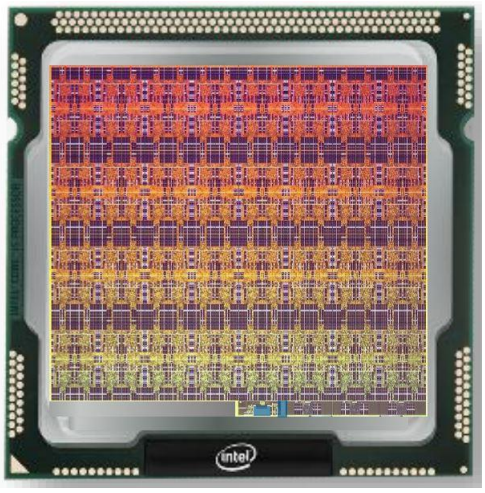

Fig.18 Chip de aprendizaje automático neuromórfico Loihi [11].

El chip está comprendido de un total de 2,070 millones de transistores y una memoria de 33MB de SRAM además de una malla multinúcleo de 128 núcleos de neurona, en cada núcleo están implementados 1,024 unidades neuronales pulsantes, agrupadas en conjuntos de árboles, lo que permite 130 millones de sinapsis, tres procesadores embebidos x86 e interfaces de comunicación fuera del chip que expanden jerárquicamente la red. El dispositivo es funcional con una fuente de voltaje de entre 0.50V a 1.25V. La comunicación entre los diferentes núcleos es manejada a través de una red asíncrona embebida en el chip (NoC), esta red NoC puede escribir, leer mensajes del núcleo de administración, además de comunicarse con la red pulsante directamente **[12]**. Debido a que el hardware está diseñado como plataforma para redes neuronales pulsantes, soporta aprendizaje acelerado en ambientes para sistemas que requieren de operación autónoma y de aprendizaje continuo. Actualmente el chip es de uso de investigación, existen pocas unidades de prueba.

Un chip superior a Loihi es el chip TrueNorth reconfigurable de IBM puede desempeñarse como una interfaz de red neuronal de ultra-bajo consumo, es una arquitectura no-Von Neumann que emula la arquitectura neuronal del cerebro. Diseñado en un proceso de 28nm, con alrededor de 256 millones de sinapsis en 4,096 núcleos de neurona, el chip consume apenas 70mW en una tarea de redes neuronales biológicas en tiempo real, capaz de producir 46 billones de operaciones sinápticas por segundo, implementa un millón de neuronas. TrueNorth permite una distribución paralela, modular y escalable, en una arquitectura flexible que integra comunicación computacional y memoria. El chip es capaz de detección multi objetos y aplicaciones de clasificación con una entrada de en 240x400 pixeles de resolución en 3 canales de video a 30 cuadros por segundo el chip consume 65mW.El ecosistema actualmente está en uso en 30 laboratorios de universidades y gobierno. La plataforma se usa en aplicaciones de celular, embebida en cómputo en la nube y supercomputadoras.

En ambos sistemas se implementan redes neuronales pulsantes en las que el circuito básico es una celda de neurona. Existen diversos modelos implementados que usan como base teórica los modelos de Izhikevich. Deben de ser lo suficientemente simples para poder implementarlos mediante dispositivos electrónicos y ser escalables para poder implementarse

en celdas de hasta millones de circuitos de neuronas, compatibles con tecnología de memristores, es deseable que pueden ser implementados juntos, para poder realizar el procesamiento de información dependiente de un circuito de memoria sináptica, y memoria RAM en la que se almacenen instrucciones de aprendizaje o cómputo en la nube y formar celdas y ampliar el sistema. La tecnología CMOS es comúnmente usada en estos circuitos de neurona ya que permite aprovechar sus características no lineales y de alta integración electrónica, además que la mayoría de los circuitos electrónicos sino se basan en esta tecnología, son compatibles eléctricamente con ella.

2.4.1 Modelo de neurona de Wijekoon

Un modelo de neurona pulsante que se inspira en el modelo Izhikevich fue introducido por Wijekoon **[13].** Representa una versión simplificada del modelo original de Izhikevich, es posible su implementación en tecnología de circuitos integrados, lográndose reproducir algunas de los distintos modos de disparo que han sido observados en las neuronas biológicas, como la generación de: impulsos regulares, impulsos rápidos, impulsos en ráfaga, entre otros.

El diseño del circuito integrado de Wijekoon se realizó mediante transistores de efecto de campo metal óxido semiconductor (MOSFET, por sus siglas en inglés). Se asume que la operación de estos transistores se mantiene en todo momento en la región de saturación.

El modelo de corriente de drenador para un transistor MOSFET de canal N, en la región de saturación se describe en la ***Eq.(3.1)*.**

En donde, donde μ_o es la movilidad superficial de los portadores minoritarios en el canal de los transistores MOSFET canal N o P, en $(cm^2/V\text{-}s)$, C_{ox} es la capacitancia por unidad de área en (F/cm^2) debido al óxido delgado, W es el ancho efectivo del canal, L es la longitud efectiva del canal, λ es el parámetro de modulación del canal en (V^{-1}) y, V_T, es el voltaje de umbral del transistor. En este trabajo se consideró a λ=0, con fines de simplificar el análisis de la neurona

Como se puede ver, en la ***Eq. (2.25)*** el primer término eleva al cuadrado el potencial de la membrana, V, y se multiplica por un factor (0.004). Esta operación quedaría representada por

la diferencia de potenciales entre el voltaje, V_{GS} y el V_{th}, en ***Eq. (3.1)*** multiplicada por el término, $\frac{\mu_o C_{ox} W}{2L}$, en equivalencia al factor 0.004, descrito en el modelo de Izhikevich. Los términos: 5v y 140, del modelo de Izhikevich no se consideraron en el modelo de Wijekoon.

Los equivalentes circuitales de los modelos de neuronas pulsantes se desprenden del trabajo de Hodgkin-Huxley **[14]** .

En la **Fig. 20** se presenta el circuito de carga del capacitor Cv que representa el potencial de membrana de una neurona, la configuración del circuito es un espejo de corriente.

La terminal de compuerta del transistor M1, está conectada al potencial del capacitor, *Vc*, que almacena el potencial de membrana *V*, en el modelo de Izhikevich. Asumiendo unas condiciones iniciales de V_C= 0V en *t*= 0s en dicho capacitor y si se aplica un escalón de corriente a la entrada de la neurona *I*, se tiene que el voltaje en el capacitor queda definido por

$$V_c = \frac{1}{c}\int_0^t I(t)\, dt$$

Dado que la corriente, I_{in}, es constante, entonces el voltaje en el capacitor, inicialmente incrementa de manera lineal

$$V_c = \frac{1}{c_v} \;.\; I(t)$$

cómo se presenta en la **Fig.19.**

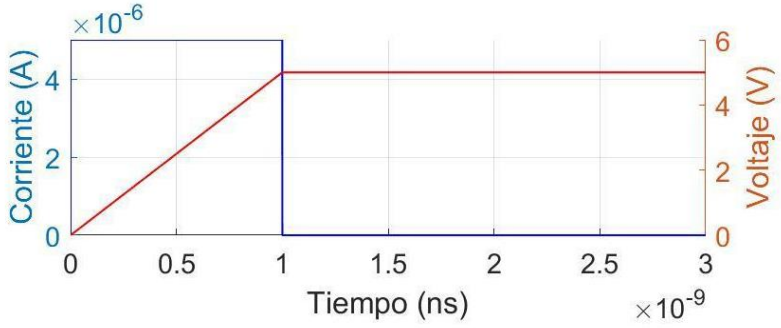

Fig.19 Escalón de corriente aplicado a un capacitor en condiciones iniciales.

El transistor M1 con $\boldsymbol{C_v}$ en condiciones iniciales, se encuentra en estado apagado, se asume que no circula corriente entre la terminal de drenador y fuente. Dado que el voltaje en la compuerta de M1, $\boldsymbol{V_{GS1} = V_c}$ una vez que $\boldsymbol{V_{GS1}}$, supera el voltaje de umbral $\boldsymbol{V_t}$, M1 se enciende, fluyendo una corriente extra, hacia el capacitor $\boldsymbol{C_v}$, debido a que la corriente de M1, se copia en el transistor M3, por efecto del circuito espejo de corriente M2-M3. La corriente que circula en M3, se incrementa de manera cuadrática en función del tiempo y se suma a la corriente I_{in}, resultando en un súbito incremento en el voltaje a través de $\boldsymbol{C_v}$, justo a partir del encendido del transistor M1, ver **Fig. 20**.

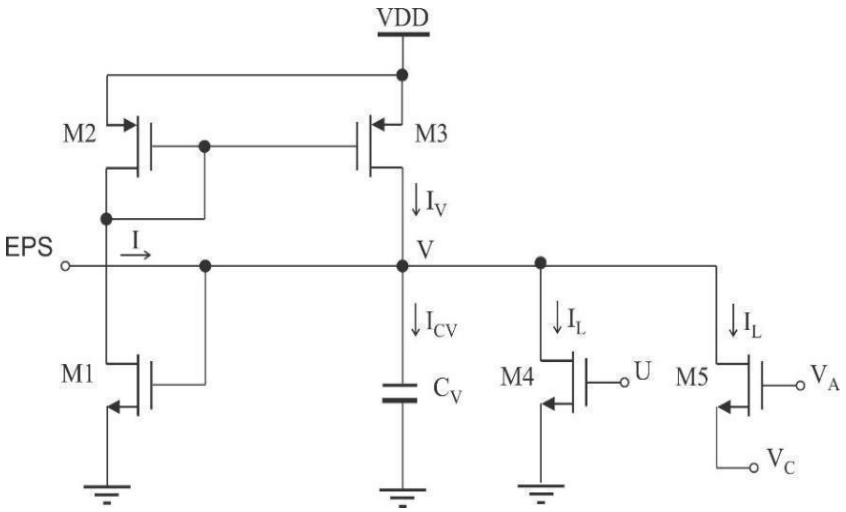

Fig. 20 Circuito que emula el potencial de la membrana de una neurona pulsante propuesto por Wijekoon, para la variable rápida (V).

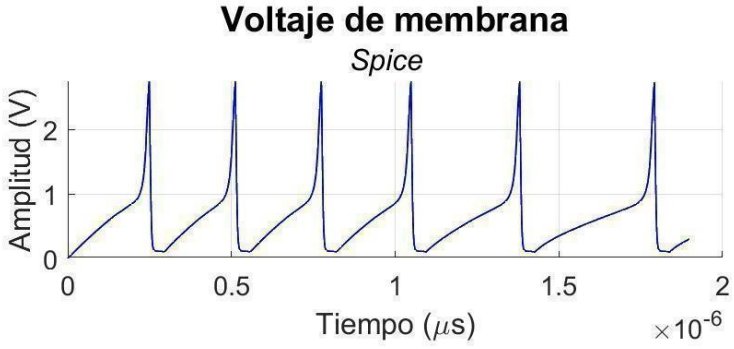

Fig. 21 Simulación del potencial de la membrana, partiendo de un voltaje inicial en la membrana de 0V.

Se muestran impulsos en un periodo de tiempo de 2 ms. Se aprecia como el potencial de la membrana crece de manera lineal, desde *t*=0, hasta un instante t= 2 ms, en donde comienza a crecer más rápidamente, por efecto del encendido del transistor M1, cuando $\boldsymbol{V_{GS}}$ alcanza el valor de V_T del transistor.

En la **Fig.20** el transistor M4 modela la corriente de fuga, por efecto de las conductancias asociadas al modelo. Está corriente de fuga es a su vez función de la segunda variable de estado *U* denominada variable lenta y se explica a continuación. Finalmente, el transistor M5, como interruptor, se colocó para permitir el restablecimiento del potencial de reposo de la neurona *Vc*. Dependiendo de la magnitud del voltaje *Vc*, que constituye un parámetro de ajuste de la neurona, diversos modos de disparo se pueden lograr.

En la **Fig. 22** conformado por un espejo de corriente derivado del circuito de membrana, se muestra el circuito que emula el potencial de membrana para la variable lenta *U*. El potencial U, se almacena en las terminales del capacitor C_U, se elige que $C_V >> C_U$. Considerando que el transistor M8 se encuentre apagado, la diferencia de los voltajes *V* e *U* se refleja en el nodo *U*, en el dominio de la corriente, como I_{VU}-I_U. El resultado de esta diferencia es la corriente I_{CU}, que circula a través de C_U, que a su vez establece el valor del voltaje *U* entre las terminales de C_U, que se retroalimenta al bloque de circuito de la variable rápida **Fig. 20** y define el nivel de corriente de fuga de la membrana I_L por medio de M4.

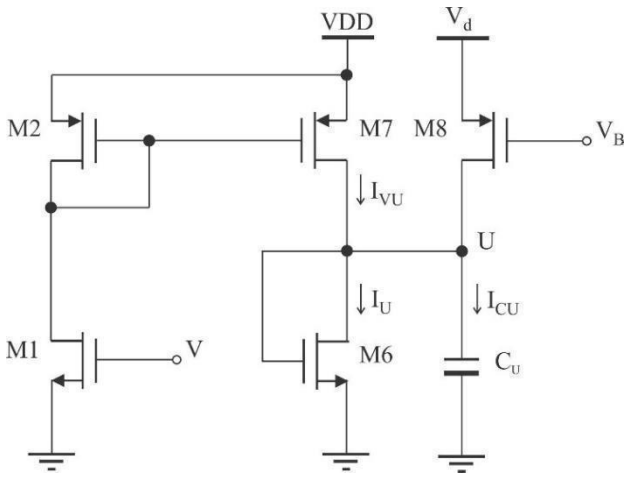

Fig. 22 Circuito que la variable lenta U, en el modelo de neurona de Wijekoon.

Como resultado, mediante el circuito presentado en la **Fig. 22** se logra emular el potencial U, de la ecuación auxiliar en el modelo de Izhikevich, como función del potencial de la membrana V, debido a que I_{CU}, es una corriente cuya magnitud es una proporción de I_{VU}, al ser ponderada por el efecto del espejo de corriente formado por los transistores M2-M3. El producto de los términos a y b, en el modelo de Izhikevich ***Eq. (2.25)*** **y** ***Eq. (2.26)*** estarían definidos por los parámetros de proceso de los transistores, así como del diseño de las geometrías W y L de los transistores. El cálculo y análisis de las geometrías quedan fuera del alcance de este trabajo. El potencial U tiende a incrementarse en una determinada proporción en cada disparo de la neurona, por efecto de la corriente que circula por M7 y que alcanza un valor máximo en la cima del impulso de la neurona. Al mismo tiempo, el valor de U, tiende a descargarse a través del transistor M6, el cual opera como diodo MOS. M8 por otra parte, es un transistor que solo se enciende al momento en que el voltaje de la membrana supera el voltaje de disparo V_{th} definido para la neurona, y permite precargar el valor del potencial U, a un nivel de voltaje cercano a V_d, el cual constituye un segundo parámetro para conmutar entre los diversos modos de disparo de la neurona.

La condición para el restablecimiento del voltaje de reposo de la neurona, se emula cuando el potencial de la membrana alcanza un determinado voltaje de umbral de disparo V_{th}. Lo que sigue a continuación, es definir un circuito comparador que brinde un pulso de voltaje con un nivel igual a V_{DD}, cuando V supere a V_{th}, que se indica como V_A, y al mismo tiempo, genere un pulso complementario V_{SS}, indicado como V_B. Esto dará como resultado que $V = V_c$, y $U = V_d$, ver **Fig. 23** y **Fig. 24.**

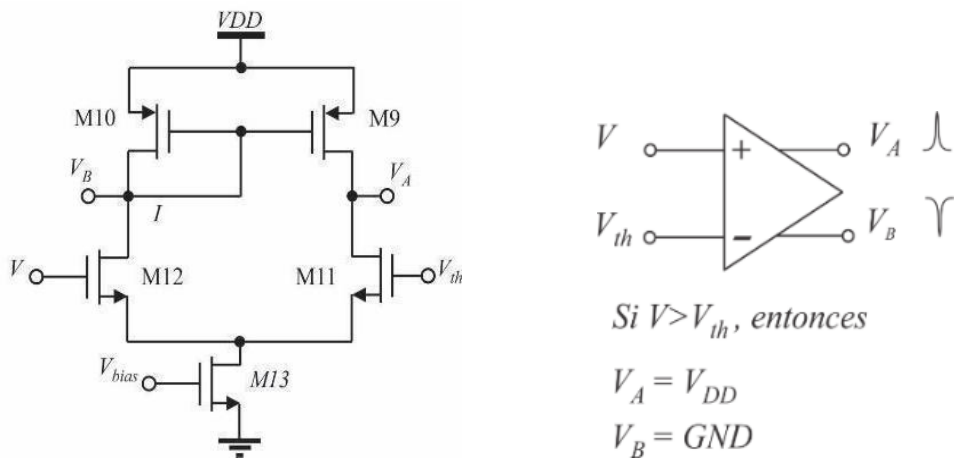

Fig. 23 Circuito comparador.

Las ecuaciones de corriente en saturación y en la región lineal aplicadas a los circuitos presentados en las **Fig. 20 , Fig.22 - 24.** se definen mediante ***Eq. (3.1)*.**

Finalmente, el circuito eléctrico completo integrando las variables lenta y rápida, así como el bloque de comparación se presenta en la **Fig. 24.**

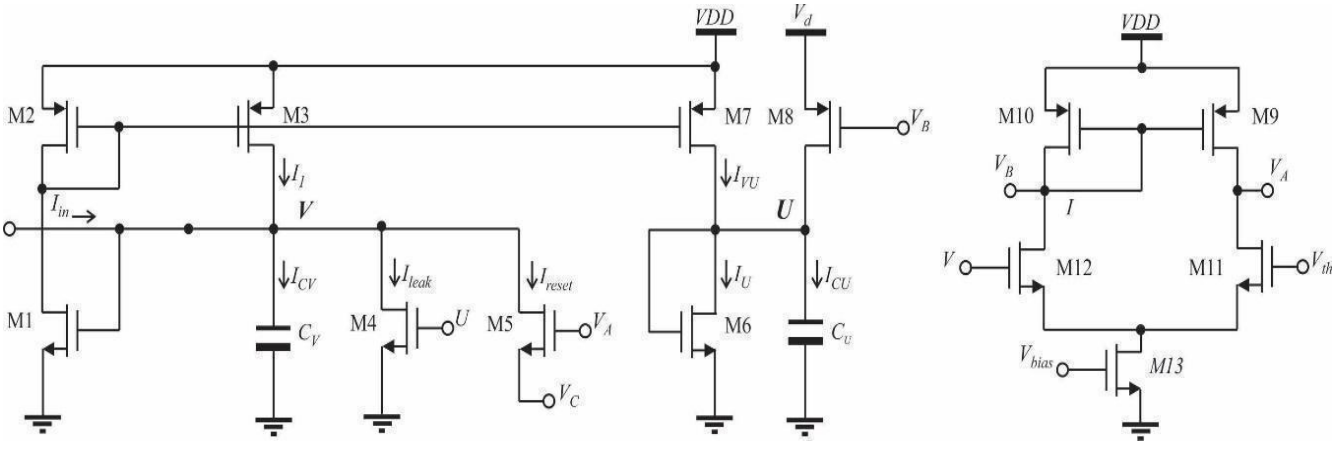

Fig. 24 Neurona pulsante implementada en tecnología CMOS según Wijekoon [13].

Capítulo 3: Diseño de circuitos integrados

La idea de múltiples dispositivos electrónicos en un mismo substrato surge alrededor de 1950, desde ese entonces se ha consolidado y expandido una industria que desarrollo la producción de chips simples que contienen unas decenas de componentes hasta dispositivos de memorias con millones de transistores, así como microprocesadores compuestos por algunos cientos de millones de dispositivos y componentes. La dimensión mínima para un transistor se redujo desde 25µm en 1960 hasta 12nm en el año 2015, logrando integrar mayor cantidad de dispositivos, disminución del consumo de energía y mejora en las velocidades en los circuitos integrados.

Los transistores de efecto de campo metal óxido semiconductor (*MOSFET*) son anteriores a la invención del transistor bipolar, debido a sus limitaciones de fabricación se introdujeron tiempo después en las tecnologías. Durante las primeras generaciones solo se producían transistores tipo-n. Los transistores CMOS capturaron rápidamente el mercado de tecnologías digitales, ya que las compuertas de los transistores solo disipan energía durante los periodos en que cambia de estado lógico y se requieren pocos dispositivos en los circuitos. Pronto se descubrió que las dimensiones de dispositivos MOS pueden ser escalados a bajas dimensiones más fácilmente que otros tipos de transistores. La introducción en el diseño de circuitos analógicos se caracterizó por el bajo costo de fabricación y la posibilidad de colocar circuitos digitales y analógicos en un mismo chip, el mejoramiento del desempeño a la vez que se reducía el costo del encapsulado para la tecnología CMOS, los convirtió en una tecnología muy atractiva.

Los CMOS son usados en la mayoría de los circuitos integrados de muy alta escala (VLSI) y de ultra escala (ULSI). El término VLSI está asociado a chips que contienen miles o millones de transistores metal óxido semiconductor de efecto de campo (MOSFET). La integración de los circuitos en un solo chip es posible debido al constante escalamiento de los transistores, haciéndolos más pequeños y reduciendo su consumo de energía. La tecnología del proceso de la manufacturación del semiconductor se refiere a características como la longitud de la compuerta y la distancia mínima entre dos dispositivos idénticos

(“*Half pitch”*). Conforme se avanza en el escalamiento nanométrico, las compañías fundidoras se refieren a las medidas de procesos en angstroms.

El cofundador de Intel en 1965 Gordon Moore nota una relación empírica en la que el número de transistores en un área de un circuito integrado se escala al doble cada dos años, esto puede observarse no en todos los casos, en la **Fig.25** TSMC describe el escalamiento del proceso de fabricación para circuitos integrados.

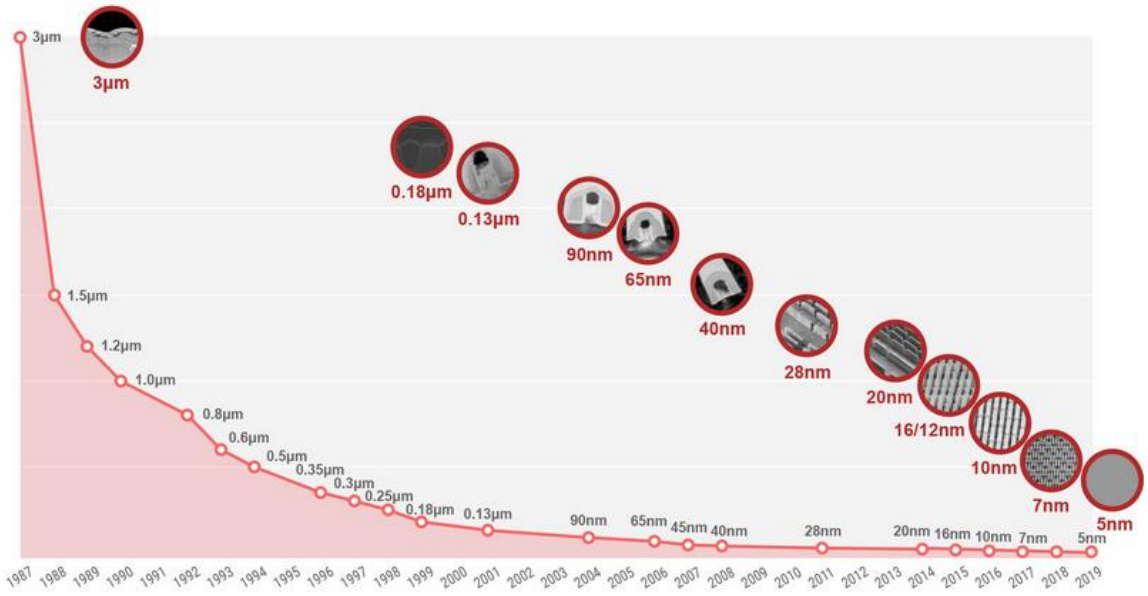

Fig.25 Desarrollo del escalamiento de los circuitos integrados[15].

Se desarrollan constantemente nuevos procesos de manufacturación para desarrollar nuevas características en dispositivos más pequeños, en general siguen un proceso básico que se describe a continuación.

3.1 Tecnología de fabricación de semiconductor complementario de óxido metálico (CMOS)

El proceso de manufactura toma entre seis y trece semanas para completarse, se lleva a cabo en fábricas de semiconductores comúnmente referidas como fundidoras, dentro de estas se lleva a cabo en cuartos limpios alguno completamente automatizados, algunos procesos en

atmósferas de distintos gases como nitrógeno. Los circuitos CMOS son fabricados en obleas circulares llamados *wafers.* En cada wafer se pueden construir cientos o incluso miles de chips individuales. Dentro del wafer se encuentran estructuras de prueba y monitores de proceso. El tamaño más común de wafer en producción tiene un diámetro de 200mm(12in).

El diseño del circuito integrado y de dibujo geométrico *(layout)* puede ser fabricado a través de MOSIS en un wafer multiproyecto, en el cual se combinan proyectos de múltiples chips de diversos sectores: educación, privados o de gobierno, logrando reducir el costo de fabricación entre la variedad de diseños implementados.

Se describe el proceso de fabricación de una estructura básica CMOS, en procesos modernos consiste en más de 200 pasos de fabricación **[16]**, pero puede ser descrito como una combinación de operaciones básicas:

- Procesamiento del wafer para producir el tipo de sustrato adecuado.
- Fotolitografía para definir cada región del semiconductor.
- Oxidación, deposito e implementación iónica para agregar contaminantes al wafer.
- Grabado para remover materiales del wafer.

Estas operaciones requieren de varios procesos en horno.

El wafer con el que se empieza debe ser producido con una alta calidad. Se conforma de un cuerpo de cristal de silicio que es producido con un muy bajo número de defectos e impurezas, debe contener niveles y tipos apropiados de dopaje para adquirir la resistencia eléctrica necesaria, se agregan dopantes sumergiéndolo en silicio fundido para adquirir la resistividad deseada y gradualmente se extrae mientras este gira, el resultado es un cilindro de cristal en una sola pieza con un diámetro típico de 10 cm a 30 cm del cual se cortan las obleas de silicón las cuales se llaman “wafer”. El wafer es pulido y grabado químicamente, para remover las alteraciones en la superficie durante el proceso

El segundo proceso es la fotolitografía, en el que se transfiere la información del diseño geométrico del circuito al wafer, se descomponen las capas del diseño geométrico, cada una de las cuales para ser creadas sobre el wafer con muy alta precisión. Los patrones de las capas son grabados en cristal transparente referidos como “máscaras” con láser de electrones

controlado con alta precisión, posteriormente el wafer es cubierto con una delgada capa de un material fotorresistor, el cual cambia sus propiedades de grabado dependiendo de la exposición a la luz ultravioleta. El material fotorresistente se endurece en las regiones expuestas a la luz y las que permanecen descubiertas son suaves. Se lleva a una grabadora química que disuelve las superficies suaves eliminando las áreas del fotorresistor que no se adhirieron, exponiendo la superficie de silicio.

Estos son los tres procesos básicos de litografía (1) cubrir la superficie con un material fotorresistor; (2) alinear la máscara con el diseño geométrico circuito a la superficie y exponerlo a luz ultravioleta; (3) grabar las áreas expuestas al fotorresistor. Se repite la secuencia al menos 5 veces en procesos de fabricación de un solo MOSFET.

El número de máscaras en un proceso vuelven altamente costosa la fabricación, afectando el precio unitario de un chip. Cada máscara tiene un costo de decenas de miles de dólares, añadiendo el costo de los procesos como la litografía que son procesos lentos, en los que modernos procesos de fabricación de CMOS incluyen 30 de estos procesos. Los costos de fabricación unitarios de circuito integrado actualmente permanecen constantes debido a que se mantiene bajo el número de transistores por área y el tamaño del *wafer* ha incrementado.

Una propiedad única del silicio es que puede crearse una capa uniforme de óxido en la superficie con poca deformación, permitiendo la fabricación de capas de compuertas de oxido tan delgadas de unas decenas de angstroms. Las compuertas de silicio de óxido sirven como cubierta protectora en varios pasos de la fabricación. También en áreas entre dispositivos, en esos casos se denomina campo de oxido (FOX). El desarrollo del área de las compuertas de silicio es una etapa crítica debido a que el ancho de esta determina cuánta corriente puede manejar un transistor.

En la implantación de iones los dopantes deben introducirse selectivamente en el *wafer*, dependiendo de la región que se desee formar. El método de implantar iones consiste en acelerar átomos donantes mediante haces concentrados de alta energía dirigidos a la superficie del *wafer,* estos penetran las áreas expuestas. Los niveles de dopaje se determinan por la intensidad y duración de la implantación. Esta implantación daña severamente el silicio, por esta razón se calienta aproximadamente a 1,000°C durante 15 a 30 minutos para volver a formar las superficies uniformes de silicio.

En la etapa de grabado el dispositivo necesita que se depositen varios materiales como polisilicio, materiales dieléctricos para separar entre sí las capas de materiales como los metales que sirven como conexiones entre otras capas.

Un método común para formar polisilicio en las capas de dieléctrico es mediante la técnica de depósito químico en fase de vapor (CVD), donde el wafer es colocado en un horno lleno de gas, que genera en el material reacciones químicas. Se graban y cortan los materiales para generar aberturas de pequeñas dimensiones mediante procesos líquidos, mediante plasma o bombardeando el material con iones producidos por gas. En estas aberturas se insertan contactos (*vias*) de metal para interconectar las diversas capas de metal. Existe un paso adicional en dispositivos activos: se cubre el wafer de cristal o capas de pasivación, protegiendo la superficie contra golpes en la manipulación del circuito, se remueven área de la máscara de pasivación para permitir colocar conexiones a las superficies de conexión externa (*pads*).

3.2 El transistor MOSFET

Dentro del estudio del transistor MOSFET se puede abordar el estudio desde la física del estado sólido y la mecánica cuántica del dispositivo. También es posible considerar el dispositivo semiconductor como una caja negra en la que el comportamiento es descrito en términos de cada voltaje y corriente en sus terminales, además de diseñar el circuito con poca atención a su operación interna.

Un dominio considerable del dispositivo semiconductor es esencial, en electrónica analógica los transistores no son sólo considerados como simples switches, varios efectos de segundo orden afectan directamente el desempeño. Con cada nueva generación de tecnologías de circuitos integrados y su constante reducción de área, estos efectos se vuelven más significativos.

Un MOSFET es un dispositivo de 4 terminales, de las cuales dos (la fuente y el drenador) son intercambiables. Cuando el sustrato está conectado a un potencial fijo se usan los modelos simplificados de tres terminales. La corriente fluye de la parte superior de la figura hacia la parte inferior. Los dispositivos NMOS y PMOS son complementarios.

En las **Fig. 26-27** se muestra cómo se definen los voltajes y corrientes, para cada terminal de un MOSFET tipo N y tipo P.

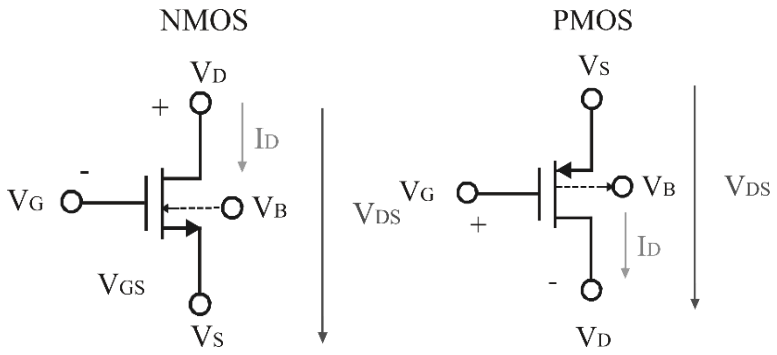

Fig. 26 Símbolo de MOSFET N y P con terminal de sustrato.

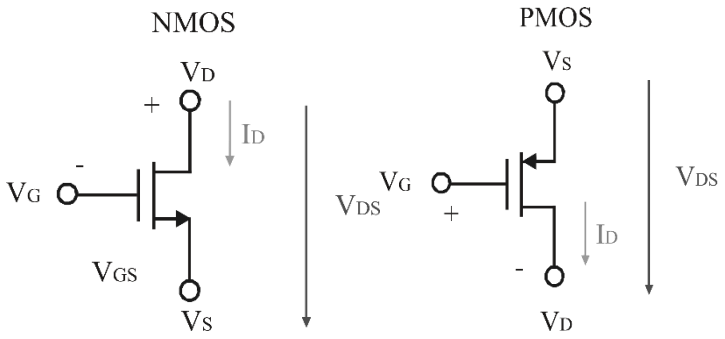

Fig. 27 Símbolo MOSFET N y P sin terminal de sustrato.

Se puede simplificar el dispositivo PMOS como un NMOS con los dopajes invertidos incluido el sustrato, pero en la fabricación ambos son construidos sobre un mismo wafer.

En el dispositivo la compuerta, el dieléctrico y el sustrato forman un capacitor, conforme VG se vuelve más positivo y el voltaje en las terminales de fuente y drenador son iguales a cero, los huecos del sustrato-P son repelidos del área de la compuerta separándose de los iones negativos. Bajo esta condición no existe flujo de corriente debido a que no hay portadores disponibles. Esta zona diferenciada de portadores y donadores recibe el nombre de zona de deflexión, el transistor este encendido, pero no hay flujo de carga. Se muestra en la ***Fig.28*.**

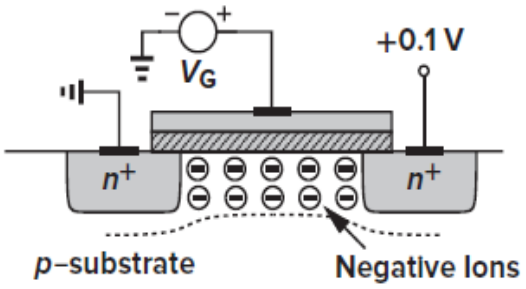

Fig.28 Zona de deflexión NMOS [17] .

$$V_{GS} > 0, V_s = V_D = 0$$

Conforme aumenta VGS positivamente y el voltaje en el drenador es mayor a cero como se muestra en la ***Fig.29*** el área de deflexión aumenta y el potencial en los extremos del canal también, este alcanza un valor suficientemente positivo, los electrones logran fluir de la fuente al drenador a través de la interfaz. Este comportamiento es llamado capa de inversión, debido a que el transistor este encendido y existe un flujo de portadores entre la fuente y el drenador. Solo la corriente de drenador puede indicar si el dispositivo este encendido o apagado. El valor para el cual VGS forma un área de inversión es llamado *Voltaje de umbral (Vth).* En el caso que el voltaje VGS siga aumentando, la carga en el área de deflexión permanece relativamente constante, mientras la densidad de la carga del canal incrementa, permitiendo mayor corriente entre S y D.

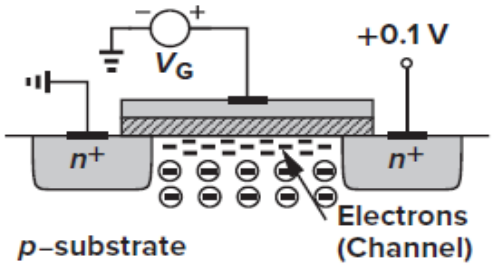

Fig.29 Formación de canal de corriente NMOS [17].

$$V_{GS} > 0, V_s = 0,\ V_D \neq 0$$

El fenómeno de encendido y apagado es similar en un dispositivo PMOS pero con las polaridades inversas.

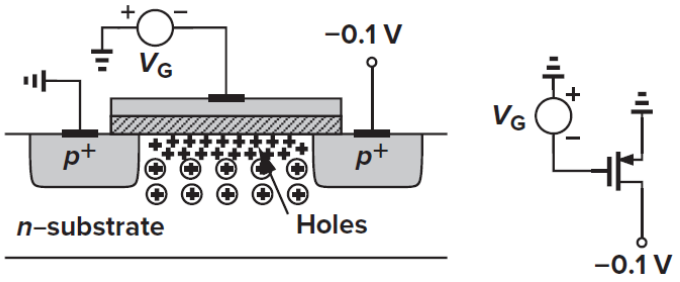

Fig.30 Formación de canal de corriente PMOS [17] .

Sí la compuerta es lo suficientemente negativa, se forma la capa de invasión formada de huecos creando un canal para conducir la corriente de fuente al drenador. El umbral de voltaje es negativo para PMOS.

El diseñador solo tiene control sobre el ancho del dispositivo y la longitud. El ancho de la capa de óxido no es controlable, es un parámetro generado por defecto del proceso en la fabricación.

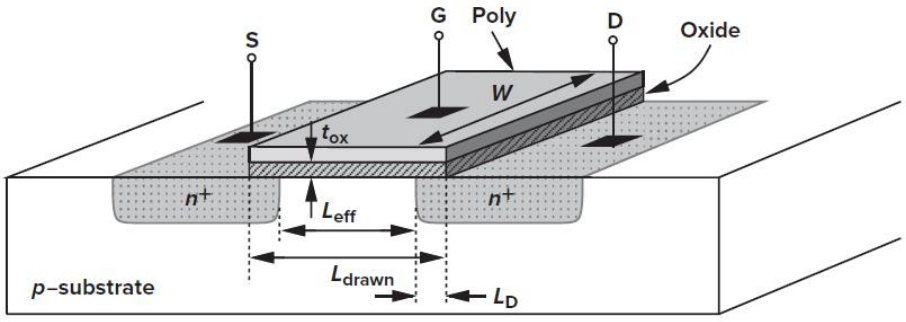

Fig.31 Vista transversal sobre un NMOS [17].

3.2.1 Modelo de la ecuación de corriente del transistor MOSFET

El modelo de corriente en un MOSFET está definido por los siguientes parámetros:

ID = Corriente que fluye a través del canal del dispositivo.

μn = Movilidad superficial de la carga de los portadores minoritarios (cm^2/V-s), mayor movilidad significa mayor velocidad y genera mayor corriente).

Cox = Capacitancia por unidad de área en (F/cm^2) debido al óxido delgado

W = Ancho efectivo del canal

L= Longitud efectiva del canal

Lambda (λ) = Parámetro de modulación del canal en (V^{-1})

VGS – VTH= Condición para generar un canal de corriente en el dispositivo.

Las ecuaciones de corriente para un transistor MOSFET de canal N se describen a continuación:

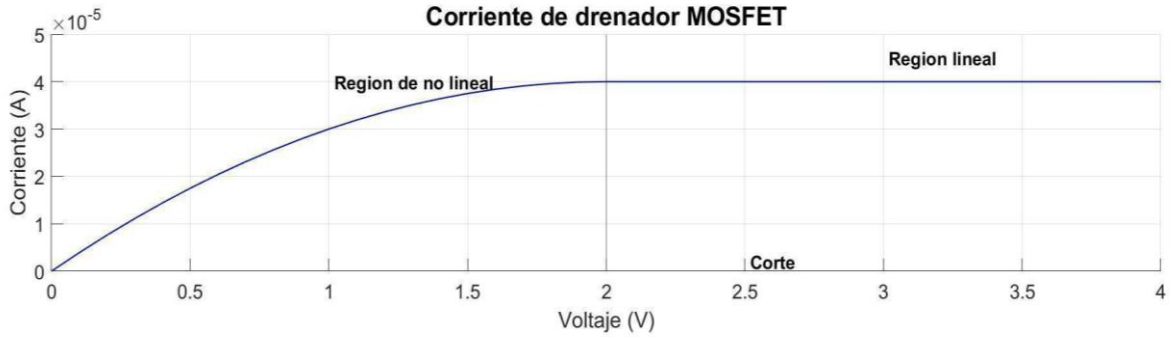

Fig.32 Región de operación MOSFET.

Saturación

$$\boldsymbol{I_D = \frac{\mu_o C_{ox} W}{2L}[(V_{GS} - V_T)^2]\ (1 + \lambda V_{DS}),\quad 0 < (V_{GS} - V_T) \leq V_{DS}} \qquad (3.1)$$

No saturación

$$\boldsymbol{I_D = \frac{\mu_o C_{ox} W}{L}\left[(V_{GS} - V_T) - \frac{VDS}{2}\right] VDS(1 + \lambda V_{DS}), \quad 0 < V_{DS} \leq (V_{GS} - V_T)} \quad (3.2)$$

3.2.2 Modelo de pequeña señal del MOSFET

Para simplificar los cálculos del circuito en términos de ganancia e impedancias, se puede usar un modelo de pequeña señal. La transconductancia de la compuerta puede ser determinada de la siguiente manera:

$$\boldsymbol{g_m = \frac{\partial i_D}{\partial V_{GS}} = \mu_n C_{ox} \frac{W}{L}(V_{GS} - Vth) = \sqrt{2\mu_n C_{ox} \frac{W}{L} I_D}} \quad (3.3)$$

La transconductancia es proporcional a la raíz cuadrada de la corriente de polarización y depende de la geometría del dispositivo.

La corriente de drenador es función del voltaje compuerta-fuente además de los voltajes del sustrato-fuente. El voltaje de compuerta-fuente controla el campo eléctrico vertical, el cual controla la conductividad del canal y por ende la corriente del drenador. Por otro lado, el voltaje del sustrato-fuente afecta al umbral, con el que cambia la corriente de drenador, cuando el voltaje de compuerta-fuente es fijo. Este efecto convierte al sustrato en una segunda compuerta, el efecto recibe el nombre de "efecto de sustrato". Cuando el voltaje en el sustrato-fuente no es constante, se agregan dos términos de transconductancia requeridos para el modelo de transistor MOS, uno asociado a la terminal de compuerta y el otro termino al sustrato con efecto como se muestra a continuación:

$$\boldsymbol{g_{mbs} = \frac{\partial i_D}{\partial V_{BS}} = \eta gm} \quad (3.4)$$

La conductancia $\boldsymbol{g_{bd}}$ y $\boldsymbol{g_{bs}}$ son conductancias equivalentes del sustrato al drenado y del drenador a la fuente, estas son pequeñas y están definidas en las siguientes ecuaciones respectivamente:

$$\boldsymbol{g_{bs} = \frac{\partial I_{BS}}{\partial V_{BS}} \cong 0} \tag{3.5}$$

$$\boldsymbol{g_{bd} = \frac{\partial I_{BD}}{\partial V_{BD}} \cong 0} \tag{3.6}$$

La transconductancia de canal en pequeña señal se vuelve importante cuando el valor en corriente alterna de la fuente-sustrato $\boldsymbol{V_{SB}}$ no es cero denotada en la siguiente ecuación:

$$\boldsymbol{g_{bs} = g_o = \frac{I_D \lambda}{1+\lambda V_{DS}} \cong I_D \lambda} \tag{3.7}$$

La conductancia de canal es dependiente de L a través de lambda ya que son inversamente proporcionales.

3.3 Celdas de circuitos analógicos CMOS

3.3.1 Compuerta de transmisión

Un circuito MOSFET en configuración de compuerta de transmisión (Transmission Gate) consta de dos MOSFETS en configuración Compuerta de paso (Pass Gate).

Un MOSFET en compuerta de paso consta de un solo MOSFET en el que a la terminal S se le asigna una conexión a tierra y al drenador se le etiqueta como salida de “ceros lógicos” y a la compuerta se le asigna VDD “uno lógico”. En esta configuración un NMOS es bueno transfiriendo ceros de la fuente al drenador.

Para un NMOS cuando la compuerta está fija a un voltaje VDD en ese caso la configuración Pass Gate está activa, si la compuerta está fija a tierra la configuración está desactivada y la salida se encuentre en un estado de alta impedancia. La configuración puede ser útil cuando se comparte un bus o un circuito lógico, además que las entradas y salidas son intercambiables.

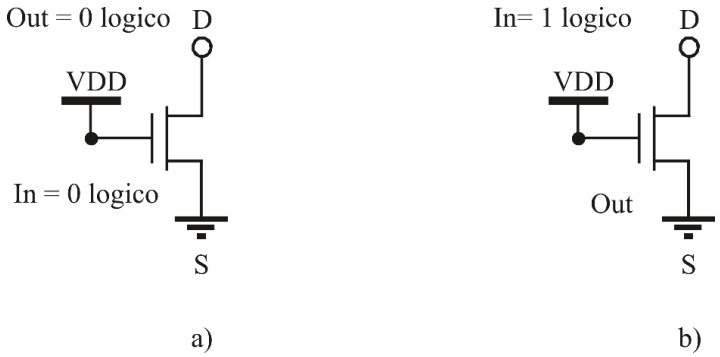

Fig.33 MOSFET configuración *transmisión gate*.

En la figura ***Fig.34*** se muestra la operación de un PMOS en configuración Pass gate. Se observa que la operación es complementaria al dispositivo NMOS. La configuración PG PMOS se enciende cuando la compuerta está controlada por una conexión a tierra, si está conectada a VDD el dispositivo se apaga y la salida se encontraría en alta impedancia. Se puede observar que la configuración con un PMOS es buena pasando un "uno lógico" pero mala pasando un cero, ya que en la salida se obtiene un voltaje igual a V_{th}.

Existe un retraso entre la entrada y la salida en la configuración de compuerta de paso, es útil en electrónica digital para corregir formas de onda de otros circuitos.

$$C_{tot} = C_L + \frac{C_{ox}}{2}$$

$$t_{delay} = 0.7 \cdot R_n C_{tot} = 0.7 \cdot R_n \cdot \left(C_L + \frac{C_{ox}}{2}\right) \quad (3.8)$$

Si se estiman con cálculos a mano los tiempos de retraso se obtendrán valores diferentes a los simulados, esto varía debido a los niveles de voltaje y las formas de onda que se miden en el tiempo de retraso.

La compuerta de transmisión es un circuito conformado de un NMOS y un PMOS conectados en paralelo con señales de control complementarias en configuración *PG.* Esta configuración cumple la función de un *relay* digital con señales de control digitales, conservando las velocidades de los dispositivos CMOS.

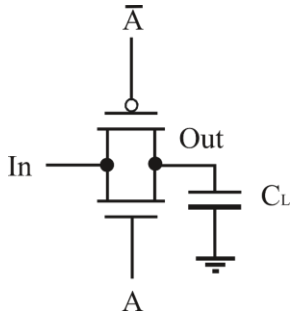

Fig.34 MOSFET en configuración *passgate*.

Mediante una señal de control se pude controlar el dispositivo ya que cuando el selector "A" se encuentra en alto, $\overline{\mathbf{A}}$ se encontrará en un estado bajo, la compuerta de transmisión está encendida y la entrada es transmitida a la salida. De esta forma se pueden transmitir los niveles lógicos cero y uno en un solo circuito teniendo control de qué estado lógico se desea transmitir.

El tiempo de retardo está dado por:

$$\boldsymbol{t_{delay} = 0.7 \cdot \left(R_n || R_p\right) \cdot C_L} \tag{3.9}$$

Es común usar varios circuitos en configuración Pass Gate para generar tiempos de retardo en circuitos

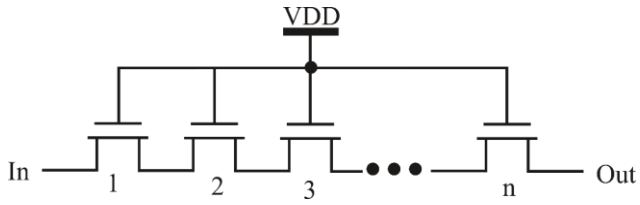

Fig.35 MOSFET en configuración Passgate que genera tiempos de retardo.

3.3.2 Inversor CMOS

La configuración fuente común con carga activase también conocida como inversor MOSFET, en la figura ***Fig.37*** se observa como el dispositivo CMOS es una fuente de corriente constante. El dispositivo M2 es un amplificador, se aplica una señal de entrada a la compuerta de M1-M2, se suponen ambos transistores en saturación. Vin se incrementa, Id1 se incrementa y el valor Vout disminuye y M2 entrega menos corriente a la salida lo que ocasiona que Vout disminuya. Ambos transistores operan en paralelo.

$$\boldsymbol{A_v = -(g_{m1} + g_{m2})(r_{01}||r_{o2})} \tag{3.10}$$

El circuito presenta una impedancia de salida de $\boldsymbol{Z_{out} = r_{01}||r_{o2}}$. El circuito presenta dos problemas principales, la variación en VDD o en los voltajes de umbral afectan directamente las corrientes de drenador. El circuito amplifica las variaciones en las fuentes de alimentación, es decir el ruido de la fuente VDD. Con una longitud de canales la configuración inversora posee baja ganancia.

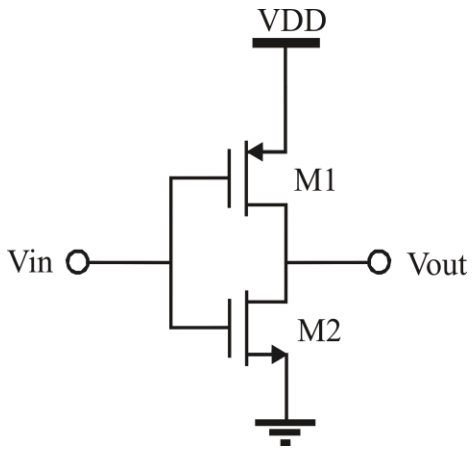

Fig.36 **Configuración inversor CMOS.**

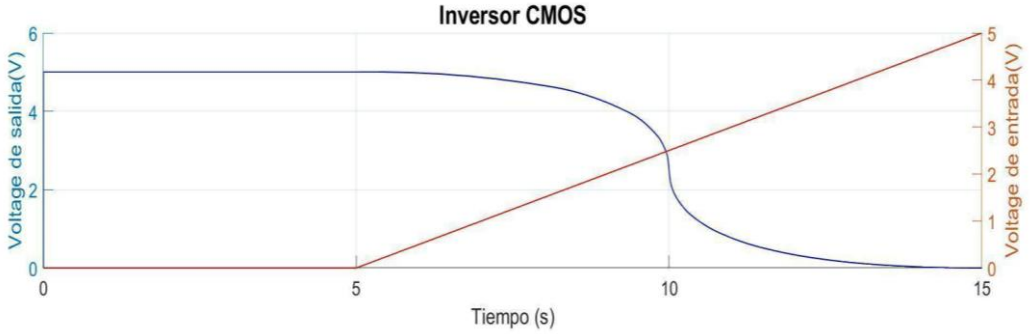

***Fig.37* Dinámica inversor CMOS.**

3.3.3 Espejo de corriente

Existen aspectos a tomar en cuenta en el desarrollo de fuentes de corriente como son: la fuente de alimentación, la dependencia a la temperatura, las variaciones de corriente en la salida o que el tamaño de las fuentes de corriente sea adecuado para agregar otras fuentes de corriente. El diseño de fuentes de corriente en circuitos analógicos está basado en copiar corriente de referencia asumiendo que será de una fuente precisa y definida. Un circuito relativamente complejo requiere ajustes externos para generar una corriente de referencia estable, de la cual es clonada para crear otras fuentes de corriente del sistema.

En la figura **Fig.38** se observa que un espejo de corriente se conforma de M1 y M2. En un caso general los transistores no necesitan ser idénticos. Considerando a $\boldsymbol{\lambda = 0}$.

$$\boldsymbol{I_{REF} = \frac{1}{2}\mu_n C_{ox}\left(\frac{W}{L}\right)_1 (V_{GS} - V_{TH})^2}$$

$$\boldsymbol{I_{out} = \frac{1}{2}\mu_n C_{ox}\left(\frac{W}{L}\right)_2 (V_{GS} - V_{TH})^2}$$

$$\boldsymbol{I_{out} = \frac{(W/L)_2}{(W/L)_1} I_{REF}} \qquad (3.11)$$

La propiedad de esta topología es que permite hacer una copia precisa de una corriente sin depender de la temperatura. La copia de $\boldsymbol{I_{REF}}$ a $\boldsymbol{I_{out}}$ involucra solamente las dimensiones, un parámetro que puede ser controlado con relativa precisión.

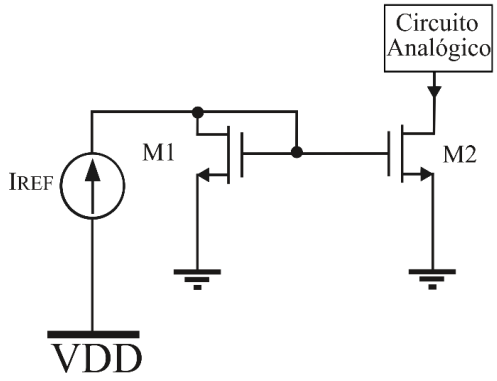

Fig.38 Configuración espejo de corriente CMOS.

Para lograr duplicar la magnitud de la corriente, es preferible crear un espejo de corriente unitario y repetirlo para conseguir el valor deseado

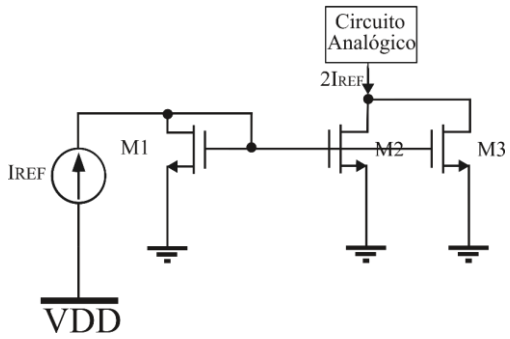

Fig.39 Copia de un espejo de corriente CMOS.

De igual manera para conseguir la mitad de la corriente Iref.

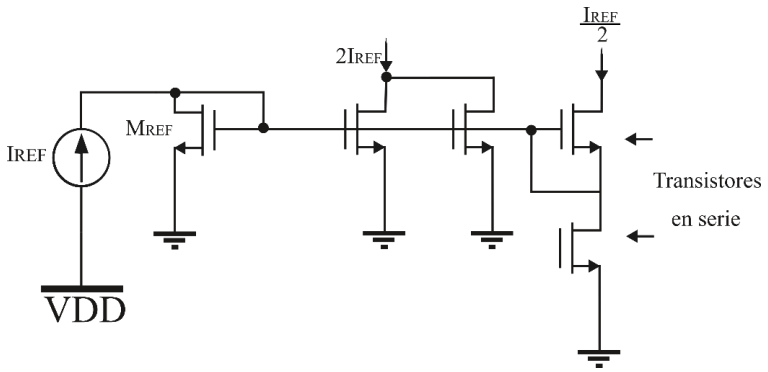

Fig.40 Copias sucesivas de un espejo de corriente CMOS.

3.3.4 Par diferencial

La carga en un amplificador diferencial con carga activa no necesita ser implementado mediante resistencias lineales, en cambio se hace con MOSFETS en configuración de diodo.

El circuito superior funciona como espejo de corriente, pero no como un circuito para polarizar, se debe a que la corriente no es constante en el tiempo, esa corriente la suministra M1 es dependiente de una señal y cambia con el tiempo al ser una señal aplicada entre M1 y M2.

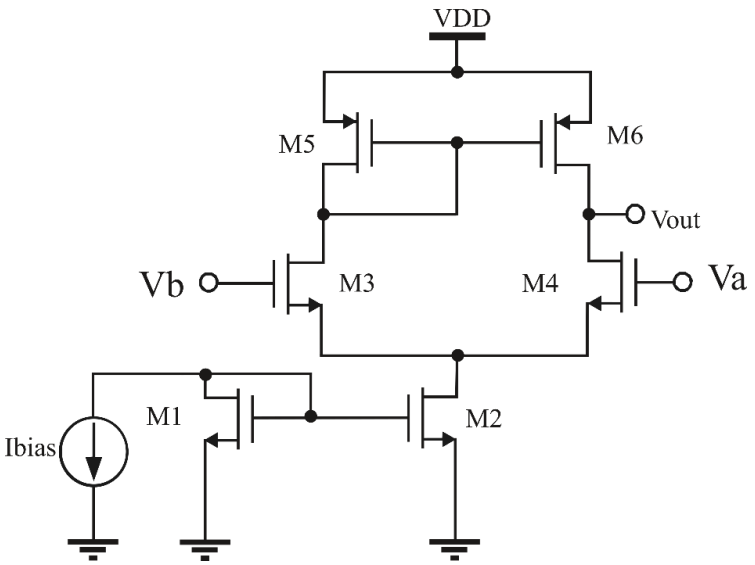

Fig.41 Configuración par diferencial con carga activa CMOS.

La ganancia a pequeña señal está representada por la ***Eq. (3.12)*.**

$$A_v = -g_m N\left(g_m^{-1}||r_{ON}||r_{OP}\right) \approx -\frac{g_{mN}}{g_{mp}} \tag{3.12}$$

El diodo MOSFET consume el margen de tensión, generando en la salida un *swing*, el cual puede ser visto como un comparador, ya que los nodos de salida cambian en direcciones opuestas diferenciales. Si el voltaje en V_{in} en M1 sube, I_{ss} fluye más a M1 y menos M2, si la corriente en M2 disminuye, el voltaje en el nodo de salida aumenta, se comporta como un circuito en configuración fuente común o inversor, en el cual, si la corriente baja, la salida incrementa. Esto por efecto de la corriente suministrada por la fuente de corriente conformada por M3 y M4, el flujo de la corriente es hacia dentro del nodo de salida.

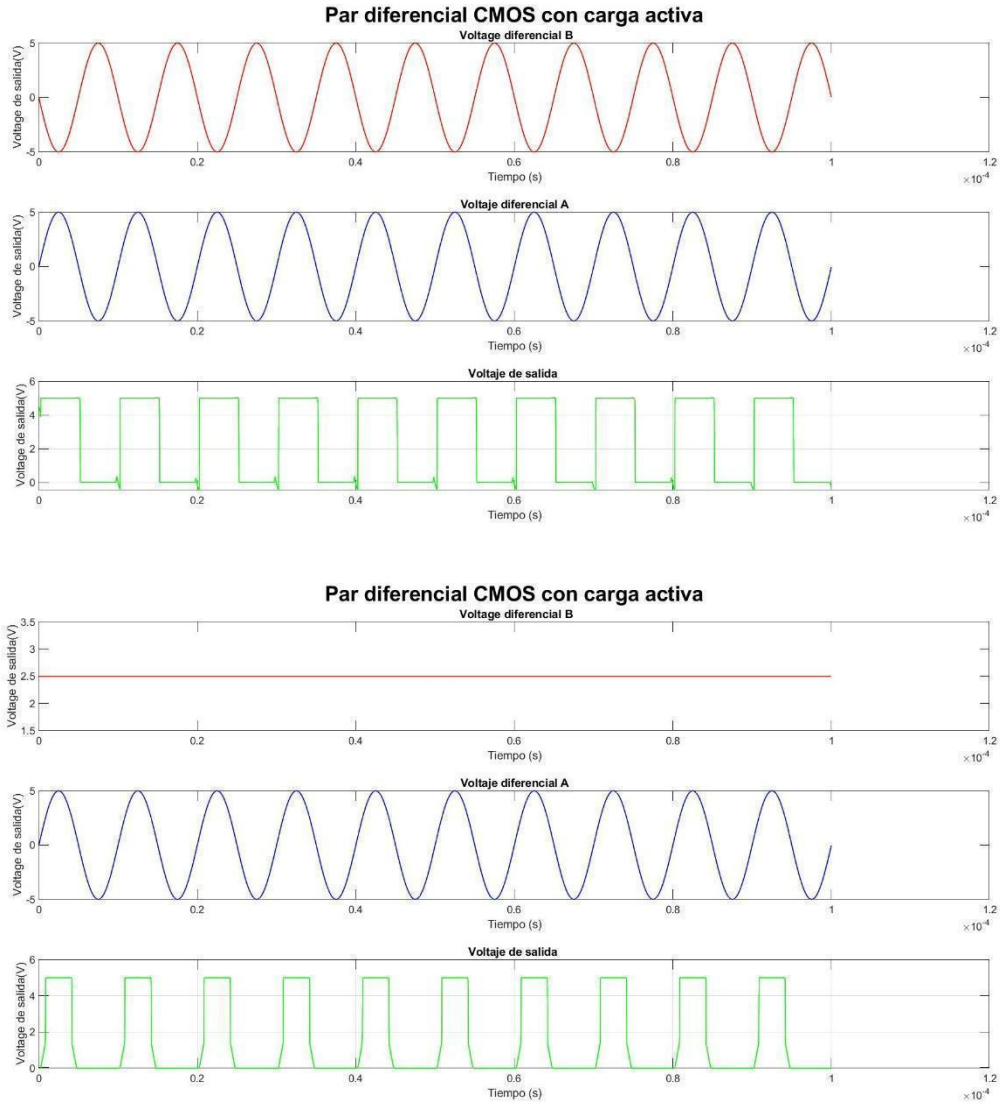

Fig.42 Dinámica par diferencial con carga activa.

3.4 Flujo de diseño típico CMOS

El proceso de diseño de los circuitos CMOS consiste en definir entradas y salidas, cálculos a mano, simulaciones en computadora, diseño geométrico, simulaciones del diseño geométrico incluyendo elementos parasíticos, reevaluación de las entradas y salidas, fabricación y etapas de pruebas.

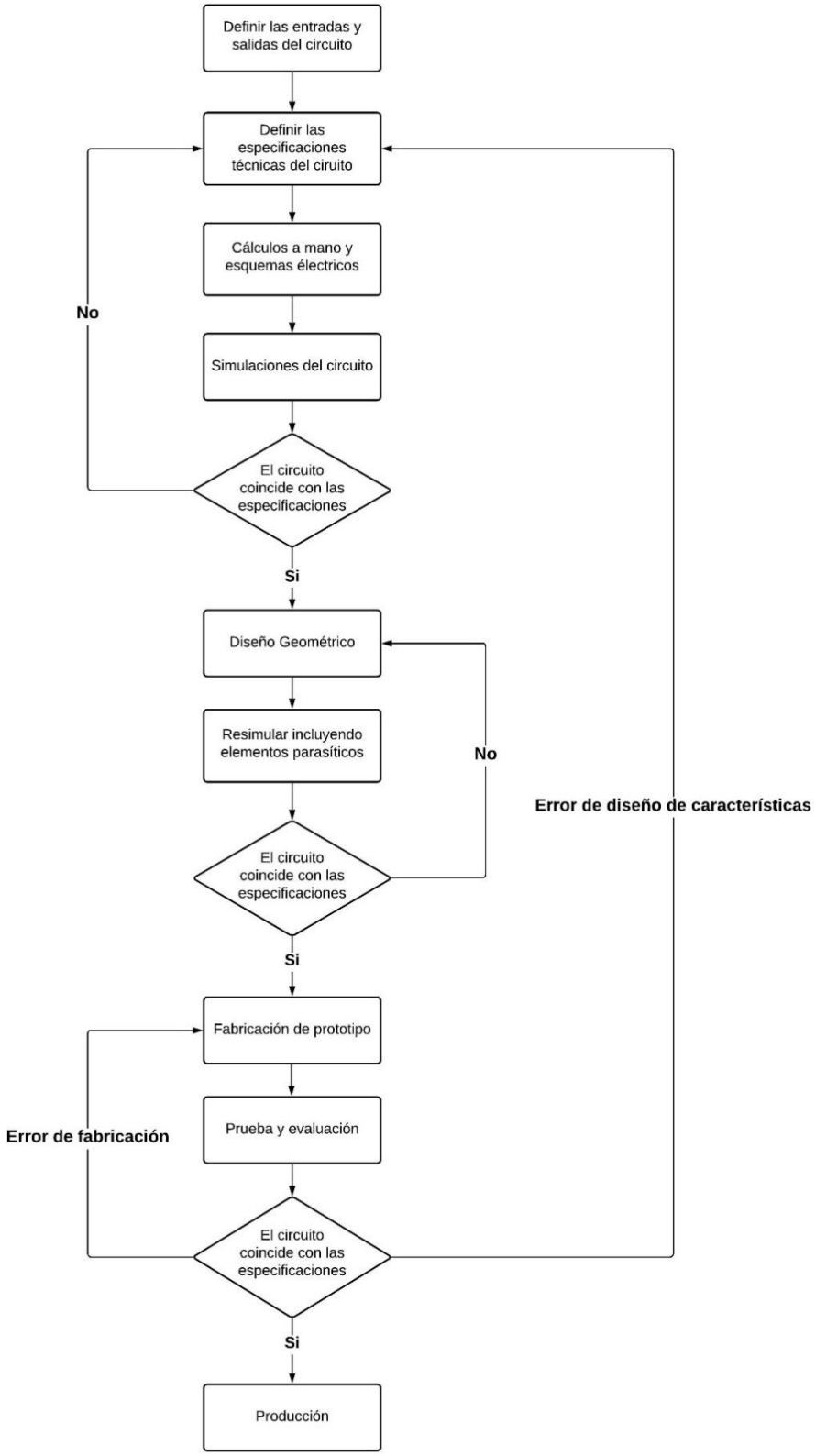

Fig.43 Flujo de diseño típico CMOS.

Las especificaciones del circuito cambian durante el avance del proyecto, como resultado de intercambios entre el costo y el desempeño del producto, cambios en el sector al que va dirigido, o las necesidades del cliente. Finalizando los cambios cuando el producto está en etapa de producción.

3.4.1 Especificaciones

Consiste en una representación de alto nivel del circuito, entradas y salidas detalladamente, esto con el objetivo de crear un panorama general de los circuitos que conforman el chip y los factores a considerar como el desempeño, las dimensiones físicas, el tamaño del chip, la tecnología de fabricación y las técnicas de diseño. Generalmente es un balance entre los requerimientos de la tecnología del mercado en el cual se introducirá y su viabilidad económica. Estos determinan la funcionalidad del circuito. Seguido a esto se inician los cálculos a mano para definir parámetros generales de operación, ya que se analizarán en un circuito esquemático dentro de un ambiente de diseño asistido

3.4.2 Diseño

Se diseñan a mano esquemas eléctricos detallados en base a los cálculos realizados para integrar las diferentes configuraciones de las que se realizaron los cálculos, esto para visualizar una distribución del espacio e implementar las correcciones e integraciones al circuito en base a la topología, como reemplazo de capacitores debido al área que ocupan, el tiempo de retraso de algunas configuraciones de circuitos. Es un proceso general del circuito sobre el que se trabaja en base a los requerimientos antes de llevarlo a una simulación eléctrica en computadora.

3.4.3 Simulación

El análisis y diseño de circuitos integrados depende fuertemente en la selección de modelos adecuados para los componentes, en un análisis a mano se usan modelos simples, pero en simulaciones y análisis de modelos complejos es necesario usar herramientas de análisis computacional. La precisión del análisis depende del modelo usado, es necesario entender las bases de los modelos más comunes.

SPICE (programa de simulación con énfasis en circuitos integrados) es un simulador de circuitos y modelado que es ampliamente usado en la industria para verificar el diseño del circuito para predecir matemáticamente el comportamiento de los componentes electrónicos, mediante el análisis del circuito por leyes de Kirchoff asigna valores de voltaje de nodos, como elementos pasivos y dispositivos semiconductores como MESFETS, MOSFETS.

La forma más común de iniciar un circuito es mediante un *netlist*, que es una descripción atreves de comandos de los elementos del circuito en referencias a nodos del circuito, elementos como transistores y capacitores y las propiedades de cada uno. El circuito puede dibujarse como un esquema eléctrico por el usuario mediante herramientas que le permiten seleccionar el componente y una vez finalizado se genera automáticamente el *netlist*. Existen diversos programas de diseño geométrico que permiten integrar comandos *SPICE* y exportar el modelo geométrico para su simulación. El simulador es capaz de analizar comportamientos complejos en un circuito como:

Análisis DC: el programa calcula el voltaje y la corriente del circuito basado en un rango de voltajes en directa.

Punto de operación: un punto de operación en la simulación sobre los datos de salida, no es gráfica, es una lista de voltajes de nodo, o análisis con diferentes corrientes.

Análisis de función de transferencia: El análisis puede ser usado para encontrar las impedancias de entrada y salida del circuito.

Se pueden importar modelos de circuitos proporcionados por el fabricante para modelar con el circuito específico durante todo el proceso. Los modelos de los transistores MOSFET son complejos, existen parámetros que se distribuyen por niveles para calcular los efectos de señal grande y señal pequeña como se observa en la **Tabla.1** además de formatos de modelo SPICE como el modelo BSIM3. Es un modelado de moderada inversión, y depende de los parámetros geométricos del dispositivo.

Parámetro SPICE	Descripción
VTO	Umbral de voltaje
UO	Movilidad de portadores en sustrato
TOX	El ancho de ácido de la compuerta.
LD	Difusión lateral
GAMMA	Efecto de sustrato
NSUB	Dopado del sustraído
PHI	Potencial de inversión

Tabla.1 Ejemplos de parámetros SPCICE del dispositivo.

Las tareas del diseño geométrico implican conocer los elementos parasíticos, como inductancias, capacitancias, uniones de materiales P y N asociados a problemas como cargas almacenadas indeseables o que afecten al desempeño de velocidad o precisión del diseño. El diseñador del circuito geométrico debe considerar cuidadosamente los elementos sensibles y críticos que llevan un sistema robusto a un desempeño predecible. Se debe tener un conocimiento básico de cómo se fabrican, encapsulan, prueban e incluso se implementan en PCB los circuitos.

Realizar el circuito geométrico es codificar las capas de material que componen a los elementos eléctricos y conectarlo con los demás elementos hasta forma los dispositivos y estos interconectarlos como en el esquema eléctrico, como se vio en las etapas de fabricación de CMOS el resultado del proceso no es perfectamente reproducido en el *wafer*. Se deben cumplir en el dibujo geométrico una serie de reglas de manufacturación en aspectos de densidad, de área, de espaciamiento como son el ancho del polisilicio, el espaciamiento entre el polisilicio el sustrato activo, la extensión del polisilicio con respecto del activo, el espaciamiento de polisilicio dentro del activo, el espaciamiento del campo de polisilicio entre otras, estas pertenecen a procesos de verificación en el diseño geométrico.

El proceso de determinar las geometrías de las máscaras del semiconductor se denomina "*Layout*" es realizado en un programa de diseño asistido por computadora ("*CAD*"). Las herramientas *CAD* son usadas para comprobar diseños esquemáticos eléctricos, validar diseños geométricos, simularlos y verificarlos. Se usó en el diseño del circuito de neurona el software "*Electric VLSI Design System*" que posee licencia *GNU*.

3.4.5 Verificación

Las reglas de verificación se aseguran de que el diseño sea funcional en términos de las dimensiones del *wafer* a pesar de los defectos en los procesos de fabricación, entre las cuales se encuentran reglas *DRC*, en estas reglas a la par de la realización del diseño geométrico se verifica que se cumplan dentro de la herramienta *CAD* como puede ser el tamaño y el espacio entre diferentes difusiones. Las fundidoras definen miles de reglas DRC, en cada tecnología se vuelven más complicadas y son más numerosas. Otro tipo de reglas son las LVS en las que se verifica la funcionalidad del diseño al generar el programa *CAD* un *netlist* para el diseño geométrico y el esquema eléctrico los compara, ya que las reglas DRC no se asegura que los diseños eléctricos y geométricos sean representaciones mutuas.

3.4.6 Depuración

Una vez que las reglas de diseño fueron validadas, la etapa de depuración consiste en simular la representación geométrica y examinar los resultados obtenidos optimizando los parámetros de los dispositivos o en caso de que el resultado no sea el correcto con las especificaciones del diseño modificarlo iterando las etapas de diseño hasta que una depuración no sea necesaria.

Capítulo 4: Desarrollo y propuesta de solución

El circuito propuesto en este trabajo está inspirado por el modelo de neurona de Wijekoon [13]. La oscilación de pulsos neuronales se logra mediante las ecuaciones diferenciales descritas por Izhikevich [18], dos variables de estado denotadas con los capacitores C*u* para *U (variable lenta)* y C*v*. para *V (potencial de membrana)* que es el voltaje de membrana celular del cual se signó una capacitancia diez veces mayor a Cv con respecto de Cu para mantener una carga lenta en U con respeto de V, y una condición de *reset* implementada con dos señales de *reset* Va y Vb, cada una con distintos tiempos de retraso para asegurar la carga de los circuitos de membrana y de variable lenta.

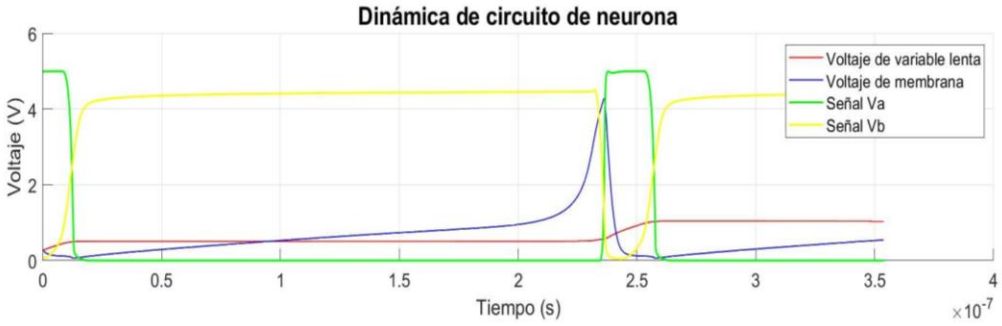

Fig.44 Dinámica de circuito propuesto de neurona en silicón.

Se consiguió un circuito que consta de 21 transistores para un diseño VLSI. El circuito se muestra en la Fig.45 en el cual se muestran los valores completos de ancho (W) y largo (L) de los canales del drenador y la fuente para cada transistor.

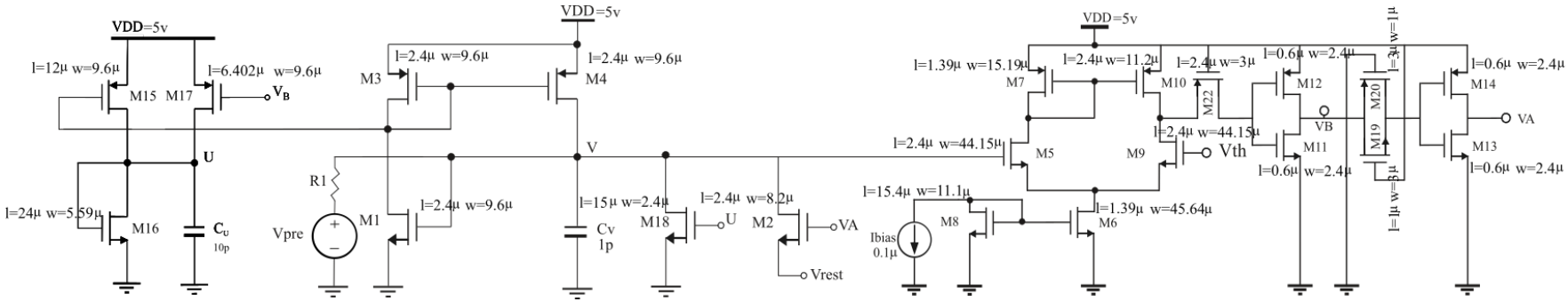

Fig.45 Esquema del circuito eléctrico del núcleo de neurona.

El circuito está formado por tres bloques o configuraciones principales: Circuito de potencial de membrana (M1-M4 y M18) **Fig.46,** Circuito de variable lenta (M15-M17) **Fig.48** y circuito comparador (M5-M7, M8-M14 y M22) **Fig.50.** La experimentación y simulación eléctrica se realizó en *LTspiceVII* es un software de tipo *SPICE* licencia GNU distribuido por Analog Devices.

4.1.1 Circuito para desarrollar el potencial de membrana (V)

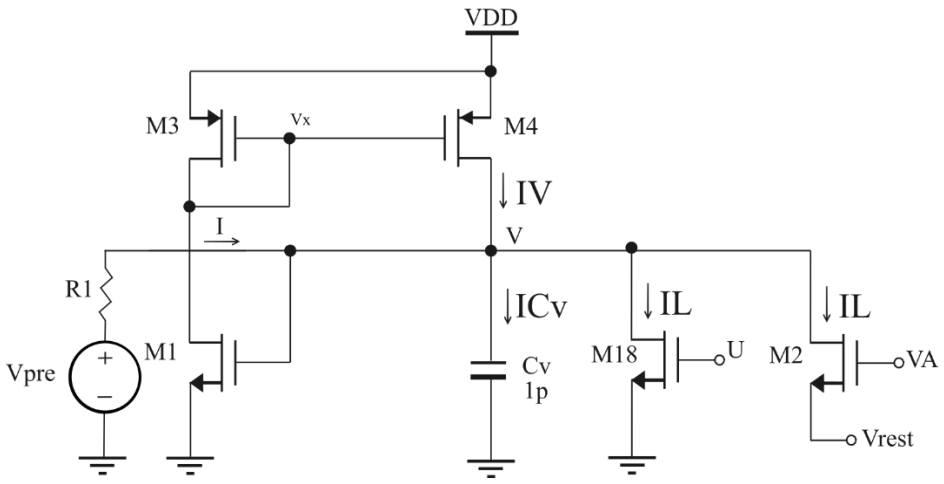

Fig.46 Circuito de potencial de membrana.

En la **Fig.46** se muestra el circuito de membrana, donde la magnitud de la corriente que entrega M4 es controlada por el potencial de membrana en la compuerta de M1. Los transistores M1, M2 y M3 forman una configuración de espejo de corriente. Las corrientes IL son corrientes de fuga, en el caso del transistor M18 es controlada por la variable *U*. La corriente que genera la fuente Vpre a través de R1 es una corriente postsináptica que es externa (inhibitoria o excitatoria) es decir que puede aportar al incremento de potencial de acción del pulso neuronal o puede prolongar el tiempo de relajación, siendo también un mecanismo inhibitorio. La suma de las corrientes integradas en el capacitor resulta:

$$C\frac{dV}{dt} = Iv - IL + I \qquad (4.1)$$

Esta ecuación no tiene condición de umbral para generar un pulso de neurona, el proceso es dependiente de la variable lenta, pero más fuertemente de la variable de reset *VA* de M2.
Una vez que el comparador detecta un pulso de neurona, el comparador genera un pulso *VA*, el cual abre el drenador en M2 y genera una corriente de fuga, como si hiperpolariza la membrana hasta el voltaje de reposo *reset*.

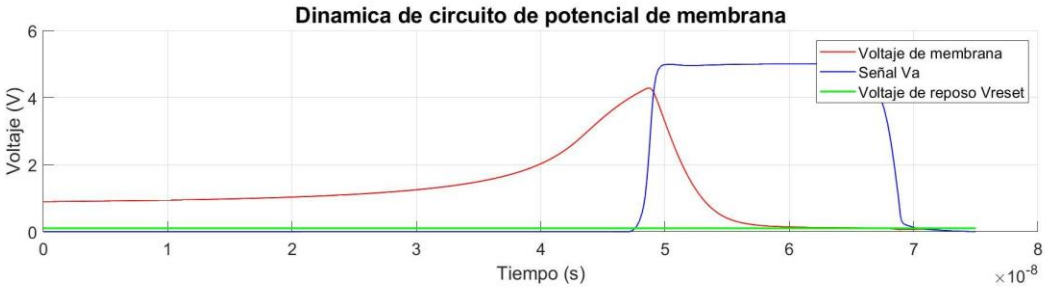

Fig.47 Dinámica de circuito de potencial de membrana.

4.1.2 Circuito para desarrollar el potencial auxiliar (U)

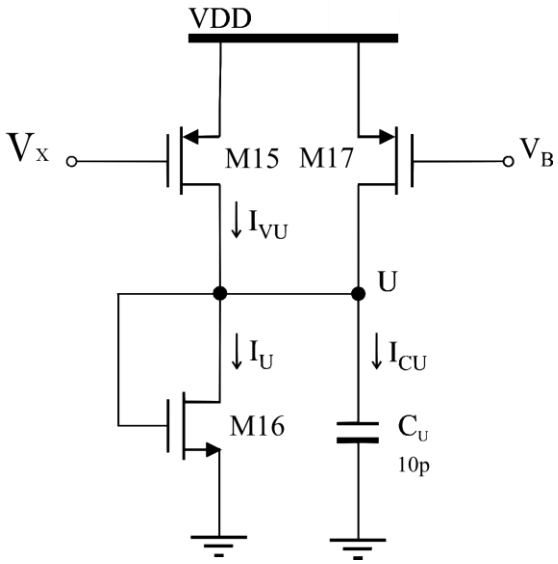

Fig.48 Circuito de variable lenta.

El circuito se compone de un espejo de corriente formado por M1 y M15 controlado por el voltaje de membrana el cual suministra al capacitor *Cu*, este capacitor está en paralelo con M16 el cual está configurado como diodo. La suma de las corrientes en el capacitor:

$$\boldsymbol{C\frac{du}{dt} = Iuv - Iu} \tag{4.2}$$

Una vez que el circuito de membrana genera un pulso y es detectado por el comparador, activa una señal de corta duración *VB* en la compuerta del M17, la cual abre el transistor y descarga el capacitor a través del diodo M16, el tamaño de los canales de M16 y la corta duración de VB asegura que el valor de voltaje de *Cu* no sea reseteado totalmente al valor de VB, sino que se agrega carga por medio de M8 cuando se desactiva la señal de *VB*.

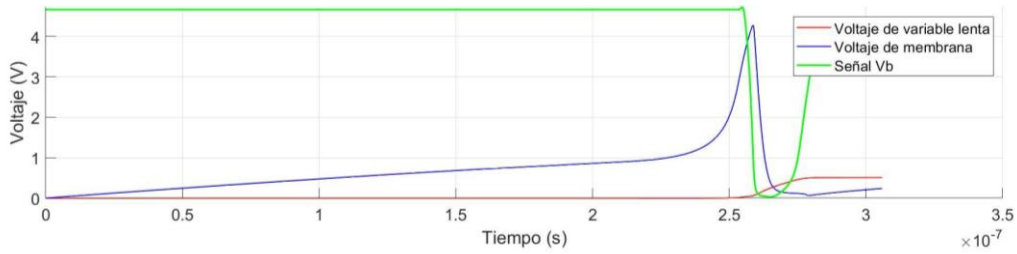

Fig.49 Dinámica circuito variable lenta.

El mecanismo con cada pulso en la membrana V incrementa el valor de voltaje en *Cu* y se reduce el tiempo de hiperpolarización de la membrana, y así genera una dinámica de patrones de disparo en la membrana celular V.

4.1.3 Circuito comparador y condición de *reset* de la neurona

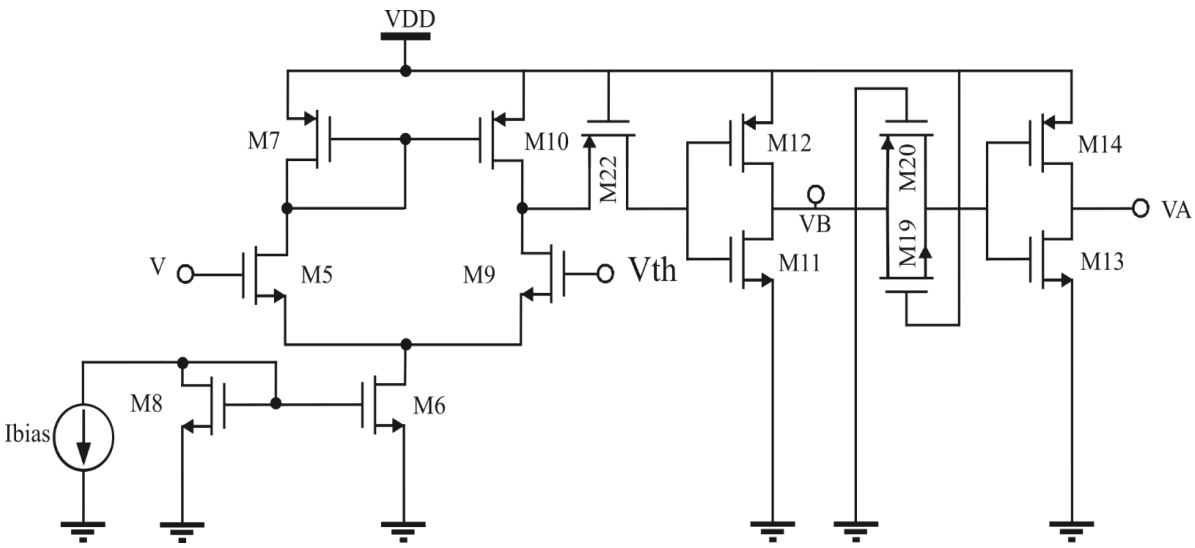

Fig.50 Circuito comparador.

Está conformado por un amplificador diferencial con carga activa siendo M8 y M6 la fuente de corriente, M5 y M9 el amplificador diferencial, M7 y M10 la carga activa de espejo de corriente. El circuito compara el voltaje de membrana con el voltaje de umbral *Vth*, cuando el voltaje de membrana incrementa y supera *Vth* produce una señal de *reset* en la salida, el transistor M10 determina la duración de la señal.

Fig.51 Dinámica de circuito comparador.

Debido a la velocidad del circuito se agregó a la salida del comparador un circuito de retardo para integrar los tiempos de señales VB y VA, formado por los transistores M11, M12, M13, M14, M19, M20 Y M21.

Lo componen dos circuitos inversores (M13, M14, M11 Y M12), un MOSFET en configuración Pass Gate (M22) y dos MOSFET en configuración Transsmision Gate (M19 Y M20). Con la única función de acondicionar los tiempos y polarizaciones de las señales de activación del comparador, ya que de estos depende la frecuencia de pulsos del circuito y tipo de MOS que sea tipo P o tipo N.

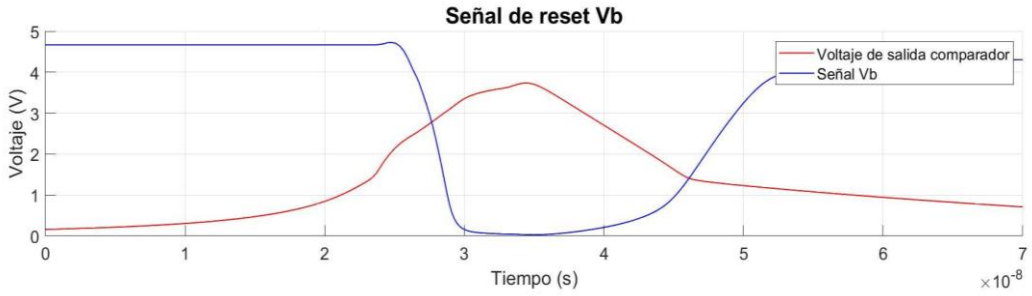

Fig.52 Señal de reinicio Vb.

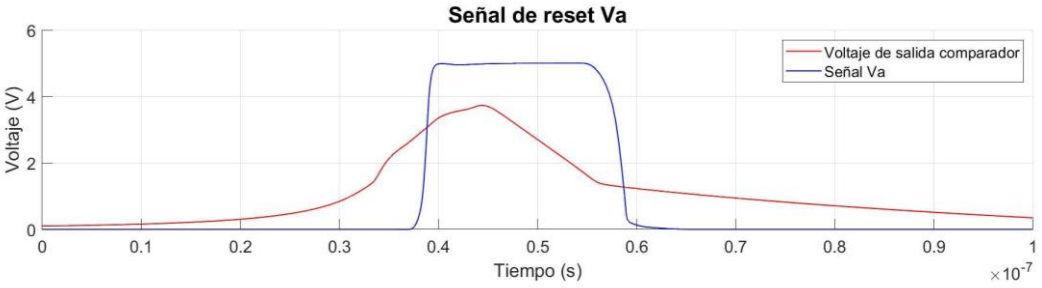

Fig.53 Señal de reinicio Va.

El tamaño de los canales de los transistores que ajustan el tiempo de retardo de las señales de *reset* fueron asignados arbitrariamente de modo que pueda reproducir los patrones de disparo cualitativamente como se ha descrito en la **Fig.7** y **Fig.11-17.**

4.1.4 Diseño geométrico de circuito de neurona pulsante

Se usaron las reglas de diseño del proceso C5 para el diseño geométrico, es un proceso de 0.6µm con un factor de escalamiento lambda de 0.3µm, con una L mínima de 0.6µm. Es un proceso optimizado para señales mixtas de 5V, ofrece aplicaciones de mediana densidad, y alto desempeño, señales complejas digitales y analógicas de hasta 20V, está disponible el diseño de transistores de bajo consumo de potencia. El proceso de 600nm el cual posee un dispositivo NMOS de largo de canal de 500nm fabricado por un equipo en IBM T.J Watson Research Center en 1975 y fue comercializado entre los años 1990-1005, por las compañías Mitsubishi Electric, Toshiba, NEC, Intel e IBM.

Para el diseño geométrico se usó el software de diseño asistido por computadora licencia *GNU Electric VLSI*, el cual cubrió todas las necesidades de diseño.

El diseño *layout* comienza trasladando el esquema eléctrico que se diseñó en *SPICE* a *Electric VLSI* definiendo los tamaños de los transistores, instanciando los modelos que proporciona el fabricante del proceso que contienen los parámetros de material de fabricación para ser simulados dentro del diseño, etiquetando en el esquema eléctrico las señales y el tipo de señal que es, los códigos de SPICE en la **Fig.54** y **Fig.56** instancian fuentes de corriente y voltaje, como es la fuente de corriente Ibias, una fuente de voltaje conectado al nodo de compuerta y drenador de M9 , la fuente de voltaje V6 y la resistencia en serie que se conecta al nodo *V*.

La capacitancia de *Cu (C1) y Cv (C2)* fueron aproximadas y corregidas mediante un circuito RC del que se hizo un esquema eléctrico y diseño *layout* del circuito para ajustar la curva de tiempo de carga similar a la que se obtiene en las simulaciones de SPICE, dado que los datos del proceso del fabricante no indica los valores de las capas de polisilicio a usar en el diseño de los capacitores.

Los transistores M5, M6 y M9 se dividen en *fingers* es decir en tres transistores en paralelo para distribuir su ancho en tres partes iguales. El diseño geométrico del circuito consta de tres partes fundamentales, la parte superior como el área de PMOS, área inferior NMOS y el área a la derecha del circuito dedicada a los capacitores.

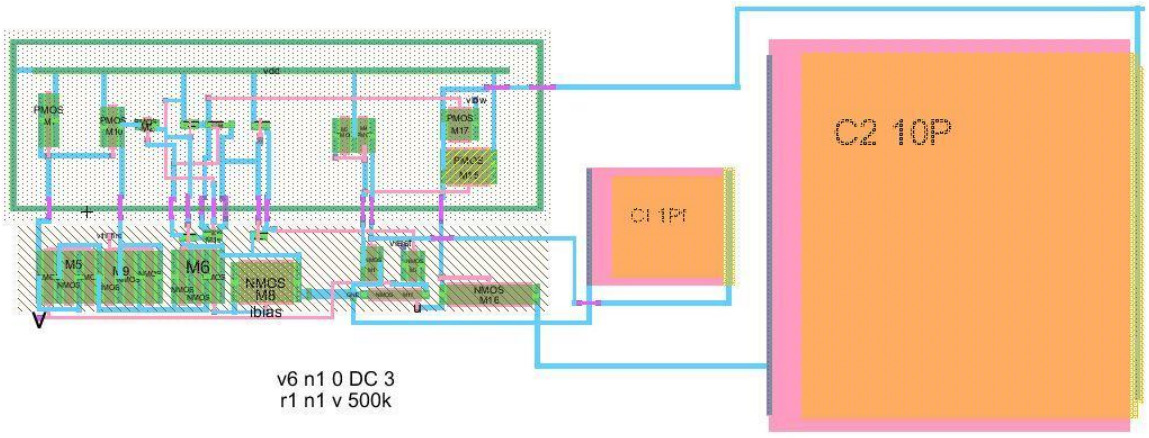

Fig.54 Diseño geométrico del circuito de neurona.

La segmentación de la topología entre transistores N y P se debe a que los PMOS son construidos sobre un sustrato N (polarizado al voltaje más positivo en el circuito), el cual está sobre el sustrato P(polarizado al voltaje más bajo del circuito), qué es el sustrato base, esto requiere que los transistores de diferente tipo P y N estén separados ya que estarán construidos sobre diferentes sustratos o "pozos" esto requiere implantar un *anillo de guarda*, ya que este evita que se genere un diodo entre el sustrato N y el sustrato P en el pozo PMOS, ya que cada sustrato está polarizado a un potencial fijo de voltaje como se observa en la **Fig.55.** Una vez definido la disposición del circuito se dibujaron los transistores P y N instanciando el modelo del fabricante etiquetando qué tipo de transistor es y colocándolos verticalmente de modo que los canales de drenador y fuente sean paralelos al eje Y, de este modo la corriente fluye horizontalmente. A medida que se hacían conexiones se verificaban las reglas de diseño (DRC) corrigiendo los errores paso por paso, de este modo si existe una conexión sobrepuesta a una distancia que viole las reglas de diseño del proceso de fabricación se indica, se corrige evitando que se acumulen los errores, ya que estos errores simples a vista no pueden identificarse.

Una vez terminada la conexión de los elementos se verificaron las reglas de diseño (DRC) y que el esquema eléctrico coincida con el *layout* (NCC o LVC), una vez que se corrigieron los errores se implementó el anillo de guarda polarizado con el voltaje más alto que es VDD, se verifica DRC, NCC y se comprueban los sustratos (*Wells*) de cada tipo de transistor, y se depuran los errores.

Se incluyo el código SPICE del esquema eléctrico y se creó un espacio de trabajo de SPICE dentro de *Electric*, una vez abierto el archivo. spi en LTSPICE, se incluyeron condiciones iniciales a los capacitores C1 y C2 para indicar condiciones iniciales a cero volts en un tiempo cero. Se eligen las señales que se desean observar y se verifican las señales obtenidas del *layout* contra las del esquema eléctrico.

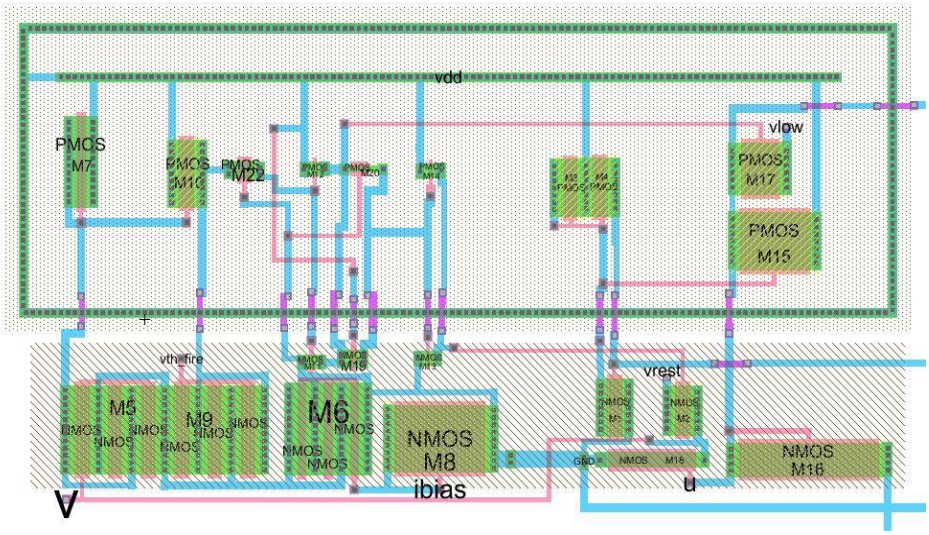

Fig.55 Diseño geométrico dentro de un anillo de guarda.

Se introdujo el circuito de neurona en una plantilla de *pads* de tamaño de 3x3mm y se re - simuló desde las entradas del *pad*.

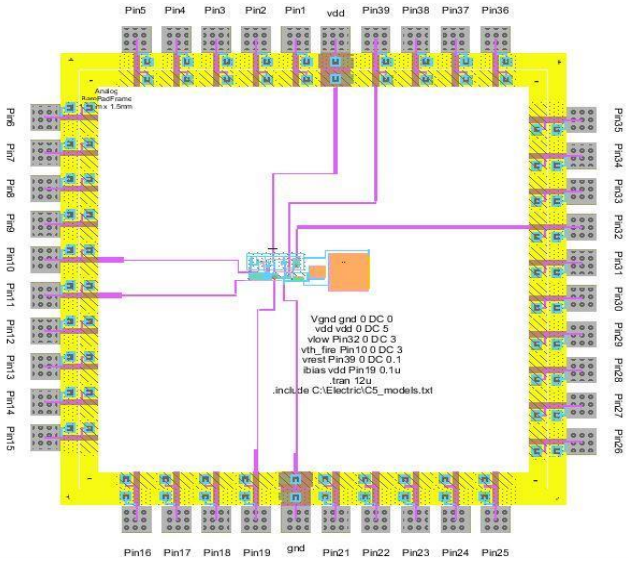

Fig.56 Diseño geométrico dentro de una plantilla de *pad's* de 3x3mm.

Capítulo 5: Resultados experimentales y discusiones

Los resultados obtenidos representan patrones de disparo observados en el modelo hibrido de Izikevich [19] y observados en células corticales [3] estas son generadas un solo circuito de núcleo de mediante una corriente presináptica. Se obtuvieron once diferentes patrones de disparo al igual que el circuito propuesto por Wijekoon [13], tales como RS1-2 que es un pulso lento con tiempo de relajación y patrones FS3 que son patrones de disparo sin tiempo de relajación en ráfagas ininterrumpidas, esto es necesario para implementar mecanismo de plasticidad neuronal necesarios para implementar algoritmos en la plataforma neuromórfico. Posteriormente se muestra una comparación entre los patrones de pulsos simulados eléctrica y geométricamente *postlayout*, se hace notar que las variaciones sen los resultados simulados pueden ser inducidas por el valor de los capacitores en el esquema geométrico ya que esta esta representa las corrientes de variables rápida y lenta en el circuito, representando el tiempo de relajación de disparo del circuito.

5.1Curva de integración del potencial de la membrana

Se observa en la Fig.57 la curva de integración generada a partir de una excitación presináptica para el potencial de membrana en un patrón de disparo RS1-2 hasta que alcanza la carga máxima antes de recibir un pulso de hiperpolarización.

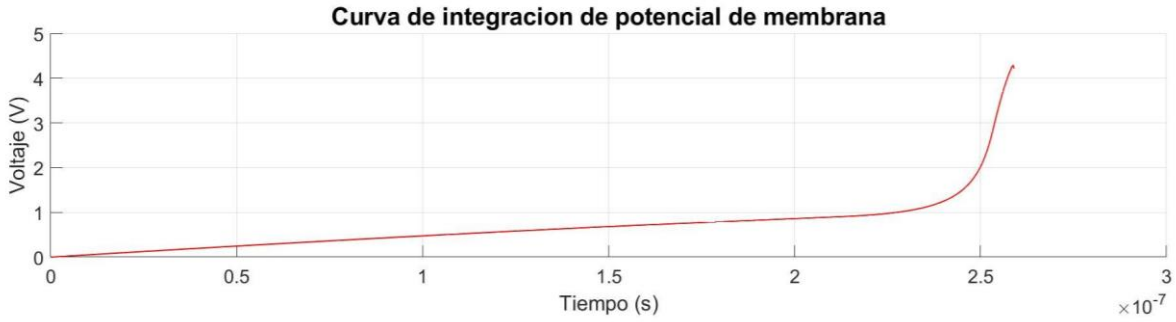

Fig.57 Curva de integración de potencial de membrana.

5.2 Dinámica de potencial de membrana simulado

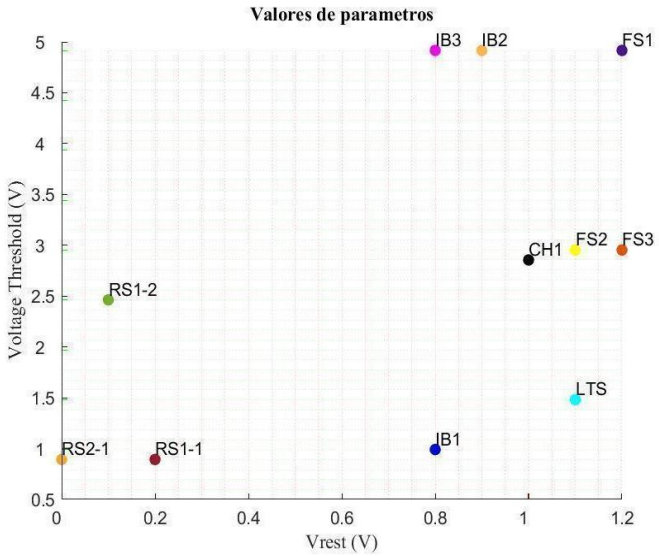

Fig.58 Dinámica de potencial de membrana simulado.

En la **Fig.58** se muestran las magnitudes de los parámetros eléctricos ajustables que generan la dinámica neuronal.

Los resultados del esquema eléctrico comparados con el diseño *layout* se muestran a continuación.

5.2.1 Generación de patrones de disparos regulares (*Regular Spiking*)

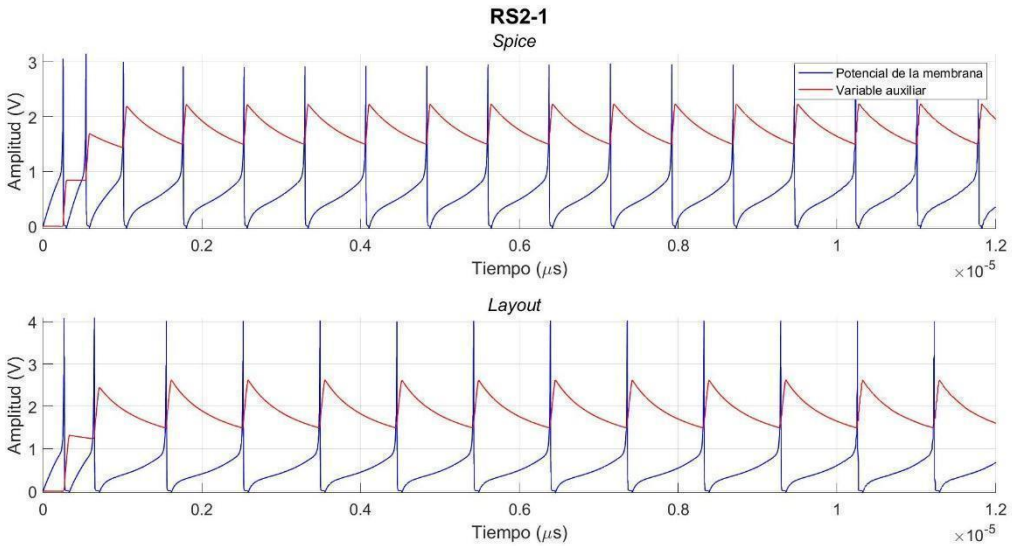

Fig.59 Generación de patrones de disparos regulares (RS2-1).

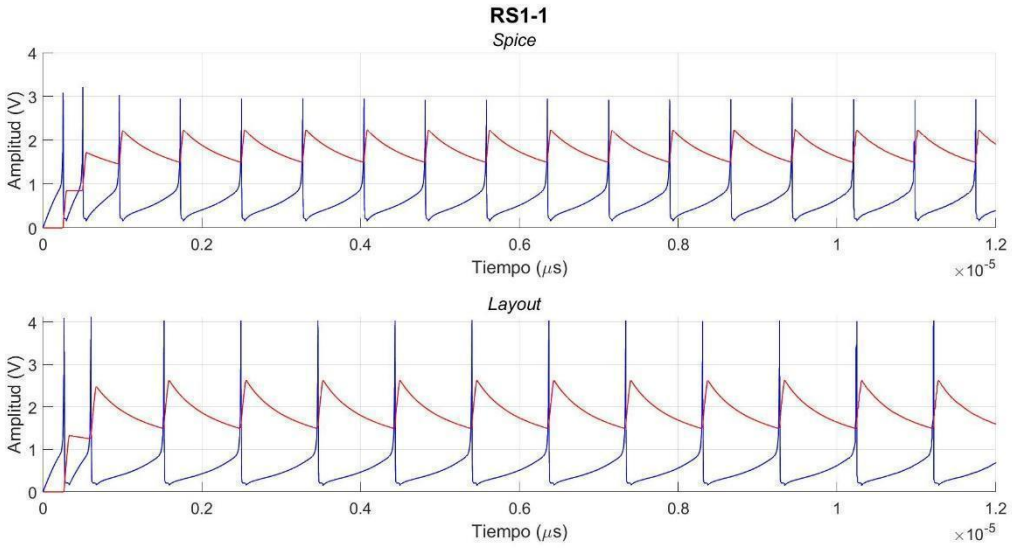

Fig.60 Generación de patrones de disparos regulares (RS1-1).

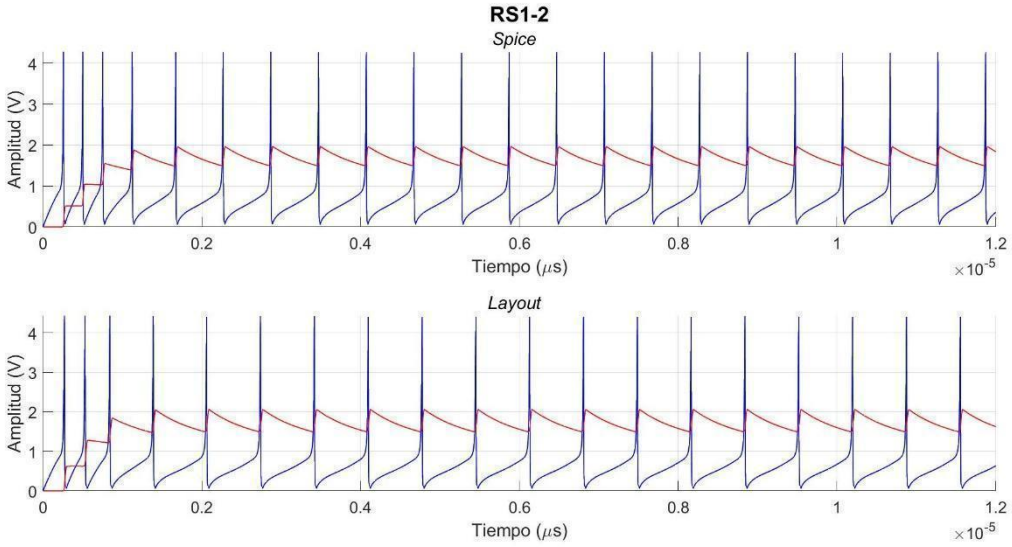

Fig.61 Generación de patrones de disparos regulares (RS1-2).

5.2.2 Generación de patrones de disparos rápidos (*Fast Spiking*).

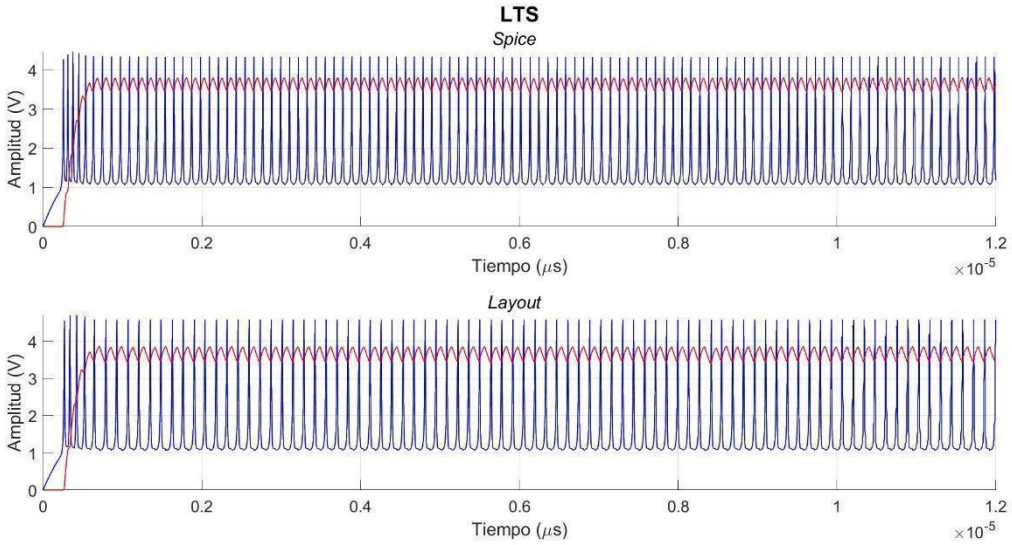

Fig.62 Generación de patrones de disparos rápidos (LTS).

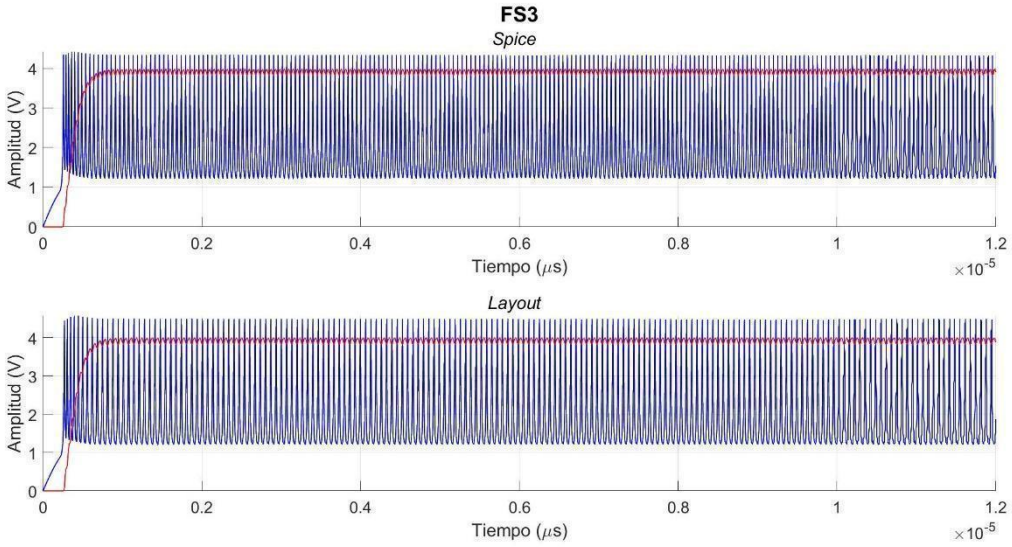

Fig.63 Generación de patrones de disparos rápidos (FS3).

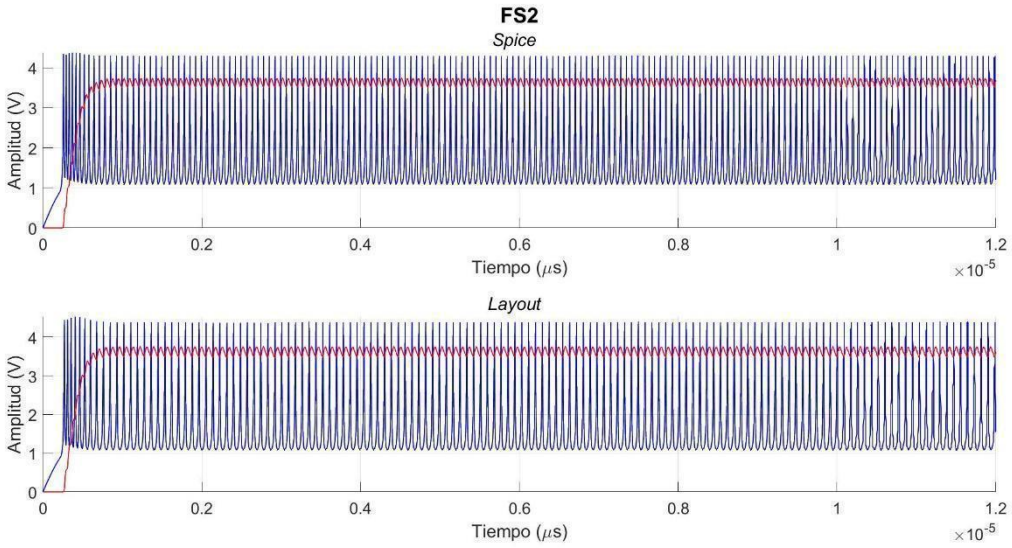

Fig.64 Generación de patrones de disparos rápidos (FS2).

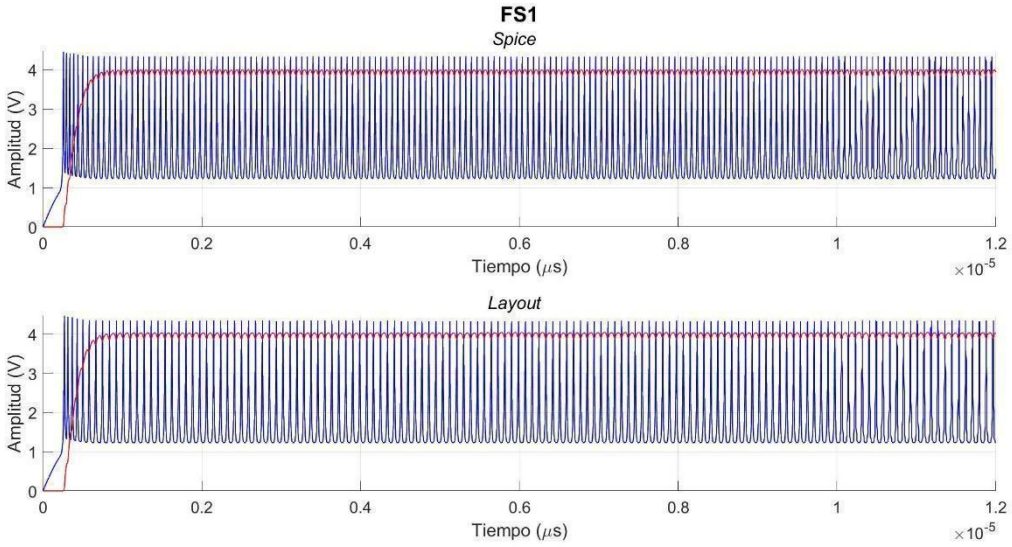

Fig.65 Generación de patrones de disparos rápidos (FS1).

5.2.3 Generación de patrones de disparo irregulares (*Chattering*).

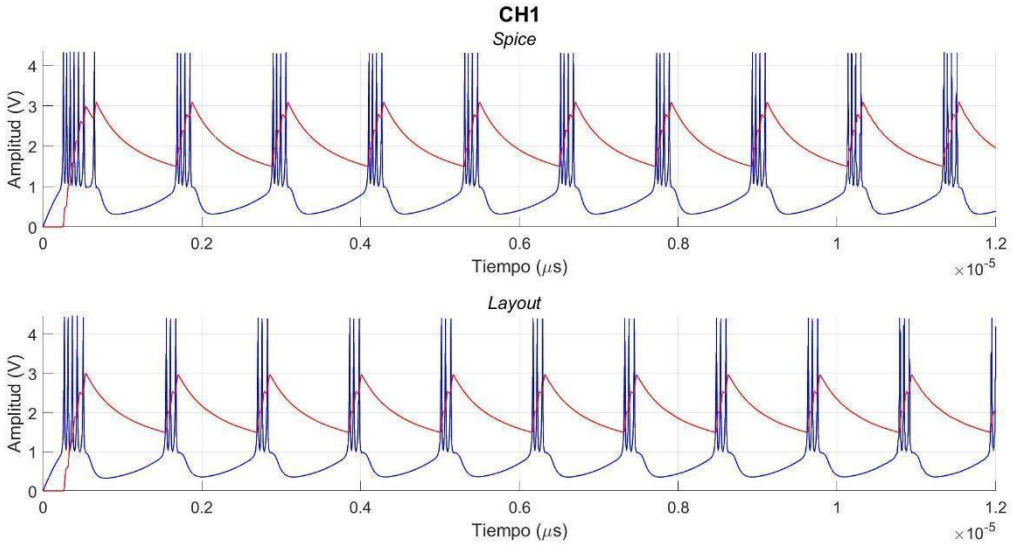

Fig.66 Generación de patrones de disparos irregulares (CH1).

5.2.4 Generación de patrones en ráfagas (*Intrinsic Bursting*)

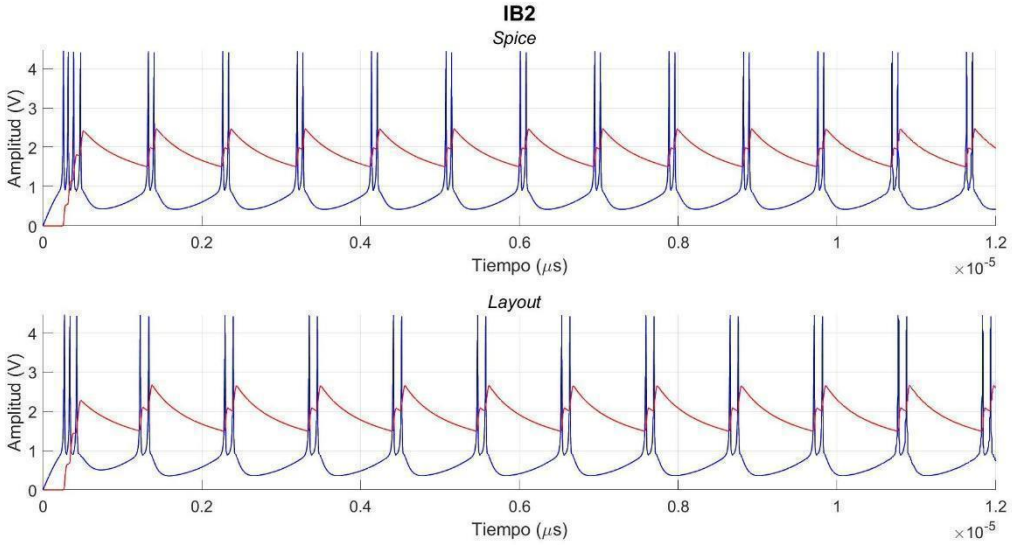

Fig.67 Generación de patrones de disparos en ráfagas (IB2).

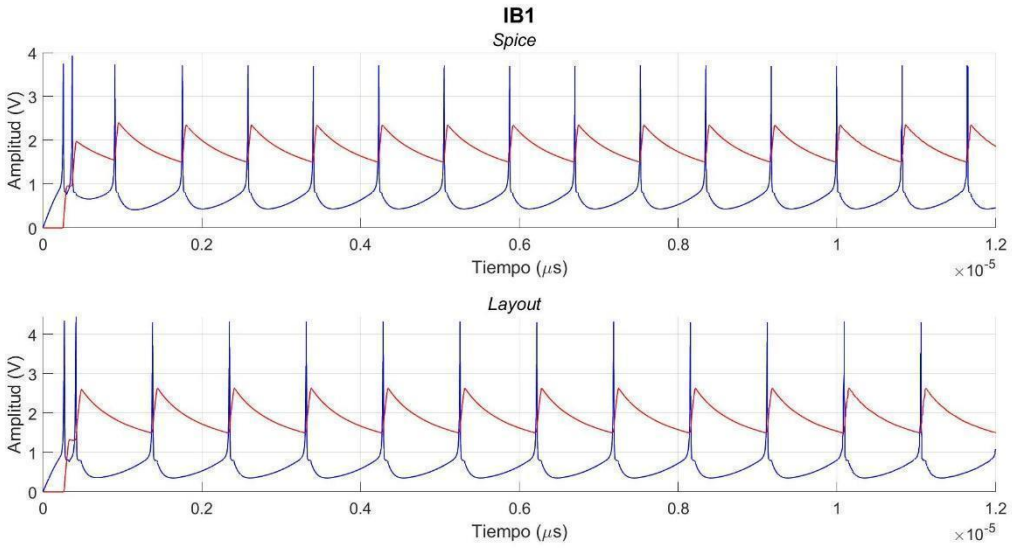

Fig.68 Generación de patrones de disparos en ráfagas (IB1).

Los pulsos de neurona presentan grandes variaciones en los patrones RS1-1, RS1-2, RS2-1 entre la simulación post-layout y la simulación eléctrica, pero no lo suficiente para afectar el desempeño del circuito ya que el objetivo es obtener las características cualitativas de los patrones de disparo.

Capítulo 6: Conclusiones y trabajo futuro

Se presentó un circuito CMOS el cual es capaz de generar patrones verosímiles de impulsos neuronales, mediante el ajuste de dos parámetros eléctricos. Estos comportamientos se han verificado en simulaciones SPICE y simulaciones post-layout, comprobando la plausibilidad de un circuito de neurona en tecnología de 0.6µm usando 21 transistores MOSFET. Dada el área del chip se pueden introducir aproximadamente 16 neuronas y se puede estimar que si se usan tecnologías más finas puede aumentar el número de circuitos de neurona en un área de 3x3mm hasta 128 neuronas. El trabajo futuro se enfocará en integrar núcleos de neurona interconectándolos a través de un bus compartido y posteriormente integrar memristores para simular las conexiones sinápticas en las que se lleva a cabo el ajuste de los parámetros de aprendizaje en el sistema de un chip neuronal.

Anexo

Ejemplo numérico entrañamiento de red neuronal artificial de retropropagación. Se desarrolla numéricamente las dos primeras iteraciones de la siguiente red neuronal de retropropagación.

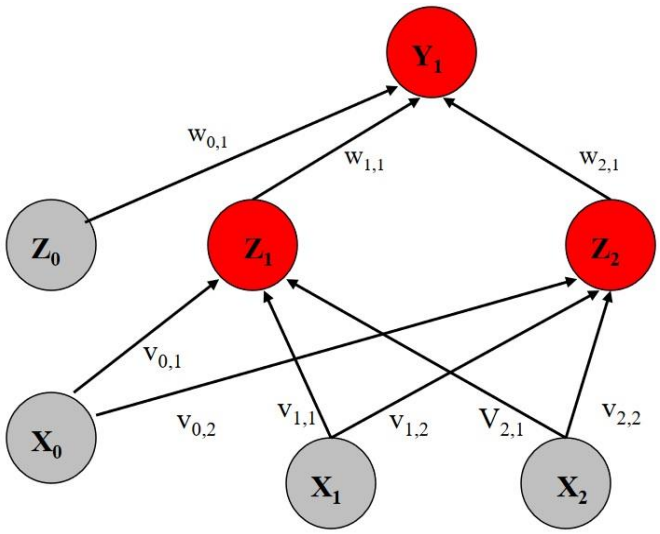

Fig.69 Red neuronal de retropropagación con una capa oculta.

$\boldsymbol{Para\ P1 = [0\ 1\ ]\quad T1 = [0]}$ Épocas = 0

Propagación hacia adelante.

Capa de entrada.

$$\boldsymbol{I_{Z1}^{(1)} = v_{0.1}^{(1)} \cdot\ x_0 +\ v_{1.1}^{(1)} \cdot x_1 + v_{2.1}^{(1)} \cdot x_2} = 0.9$$

$$\boldsymbol{Y_{Z1}^{(1)} = g\left(I_{Z1}^{(1)}\right) =} 0.71$$

$$\boldsymbol{I_{Z2}^{(1)} = v_{0.2}^{(1)} \cdot\ x_0 +\ v_{1.2}^{(1)} \cdot x_1 + v_{2.2}^{(1)} \cdot x_2} = 0.9$$

$$\boldsymbol{Y_{Z2}^{(1)} = g\left(I_{Z2}^{(1)}\right) =} 0.71$$

Capa de salida.

$$\boldsymbol{I_{Y1}^{(2)} = w_{0.1}^{(2)} \cdot z_0 + w_{1.1}^{(2)} \cdot Y_{Z1}^{(1)} + w_{2.1}^{(2)} \cdot Y_{Z2}^{(1)}} = 1.43$$

$$\boldsymbol{Y_{Y1}^{(2)} = g\left(I_{Y1}^{(2)}\right) =} 0.8084$$

Cálculo del error cuadrático.

$$\boldsymbol{E = \frac{1}{2}\left(0 - Y_{YI}^{(2)}\right)^2 = 0.3267}$$

Retropropagación del error.

Cálculo de error capa de salida.

$$\boldsymbol{\nabla E^{(2)} = \delta_{Y1}^{(2)} \cdot Y_I^{(1)}}$$

$$\boldsymbol{\delta_{Y1}^{(3)} = \left(0 - Y_{Y1}^{(2)}\right) \cdot g'\left(Y_{Y1}^{(2)}\right) = \left(0 - Y_{Y1}^{(2)}\right) \cdot \left(1 - Y_{Y1}^{(2)}\right)\left(Y_{Y1}^{(2)}\right) = -0.12520}$$

Actualización de peso sináptico.

$$\boldsymbol{W_{01}^{(2)} \leftarrow W_{01}^{(2)} + \eta \cdot \delta_{Y1}^{(2)} \cdot 1 = 0.8 + (0.5)(1)(-0.12520) = 0.7373}$$

$$\boldsymbol{W_{11}^{(2)} \leftarrow W_{11}^{(2)} + \eta \cdot \delta_{Y1}^{(2)} \cdot Y_{Z1}^{(1)} = 0.3 + (0.5)(0.71)(-0.12520) = 0.2554}$$

$$\boldsymbol{W_{21}^{(2)} \leftarrow W_{21}^{(2)} + \eta \cdot \delta_{Y1}^{(2)} \cdot Y_{Z2}^{(1)} = 0.6 + (0.5)(0.71)(-0.12520) = 0.5554}$$

Cálculo de error de capa oculta.

$$\boldsymbol{\nabla E^{(2)} = \delta_{Y1}^{(2)} \cdot \delta_{Z1}^{(1)} \cdot x_0}$$

$$\boldsymbol{\nabla E^{(2)} = \delta_{Y1}^{(2)} \cdot \delta_{Z2}^{(1)} \cdot x_1}$$

$$\boldsymbol{\delta_{Z1}^{(1)} = \delta_{Y1}^{(2)} \cdot \left(1 - Y_{Z1}^{(2)}\right)\left(Y_{Z1}^{(2)}\right) \cdot W_{11}^{(2)} = -0.0077}$$

$$\boldsymbol{\delta_{Z2}^{(1)} = \delta_{Y1}^{(2)} \cdot \left(1 - Y_{Z2}^{(2)}\right)\left(Y_{Z2}^{(2)}\right) \cdot W_{21}^{(2)} = -0.0154}$$

Actualización de peso sináptico.

$$\boldsymbol{v_{01}^{(1)} \leftarrow v_{01}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_0 = 0.2 + (0.5)(-0.0077)(1) = 0.1961}$$

$$\boldsymbol{v_{02}^{(1)} \leftarrow v_{02}^{(1)} + \eta \cdot \delta_{Z2}^{(1)} \cdot x_0 = 0.2 + (0.5)(-0.0154)(1) = 0.1922}$$

$$\boldsymbol{v_{11}^{(1)} \leftarrow v_{11}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_1 = 0.5 + (0.5)(-0.0154)(0) = 0.5}$$

$$\boldsymbol{v_{12}^{(1)} \leftarrow v_{12}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_1 = 0.5 + (0.5)(-0.0154)(0) = 0.5}$$

$$\boldsymbol{v_{21}^{(1)} \leftarrow v_{21}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_2 = 0.7 + (0.5)(-0.0077)(1) = 0.6961}$$

$$\boldsymbol{v_{22}^{(1)} \leftarrow v_{22}^{(1)} + \eta \cdot \delta_{Z2}^{(1)} \cdot x_2 = 0.7 + (0.5)(-0.0154)(1) = 0.6922}$$

Épocas = 1.

Propagando hacia delante con nuevos pesos.

Capa de entrada.

$$\boldsymbol{I_{Z1}^{(1)} = v_{0.1}^{(1)} \cdot x_0 + v_{1.1}^{(1)} \cdot x_1 + v_{2.1}^{(1)} \cdot x_2} = 0.8922$$

$$\boldsymbol{Y_{Z1}^{(1)} = g\left(I_{Z1}^{(1)}\right)} = 0.7093$$

$$I_{Z2}^{(1)} = v_{0.2}^{(1)} \cdot x_0 + v_{1.2}^{(1)} \cdot x_1 + v_{2.2}^{(1)} \cdot x_2 = 0.8844$$

$$Y_{Z2}^{(1)} = g\left(I_{Z2}^{(1)}\right) = 0.7077$$

Capa de salida.

$$I_{Y1}^{(2)} = w_{0.1}^{(2)} \cdot z_0 + w_{1.1}^{(2)} \cdot Y_{Z1}^{(1)} + w_{2.1}^{(2)} \cdot Y_{Z2}^{(1)} = 1.3115$$

$$Y_{Y1}^{(2)} = g\left(I_{Y1}^{(2)}\right) = 0.7878$$

Cálculo del error cuadrático.

$$E = \frac{1}{2}\left(0 - Y_{YI}^{(2)}\right)^2 = 0.31023$$

Retropropagación del error.

Cálculo de error capa de salida.

$$\nabla E^{(2)} = \delta_{Y1}^{(2)} \cdot Y_I^{(1)}$$

$$\delta_{Y1}^{(3)} = \left(0 - Y_{Y1}^{(2)}\right) \cdot g'\left(Y_{Y1}^{(2)}\right) = \left(0 - Y_{Y1}^{(2)}\right) \cdot \left(1 - Y_{Y1}^{(2)}\right)\left(Y_{Y1}^{(2)}\right) = -0.1317$$

Actualización de peso sináptico.

$$W_{01}^{(2)} \leftarrow W_{01}^{(2)} + \eta \cdot \delta_{Y1}^{(2)} \cdot 1 = 0.7373 + (0.5)(1)(-0.1317) = 0.6714$$

$$W_{11}^{(2)} \leftarrow W_{11}^{(2)} + \eta \cdot \delta_{Y1}^{(2)} \cdot Y_{Z1}^{(1)} = 0.2525 + (0.5)(0.7093)(-0.1317) = 0.2087$$

$$W_{21}^{(2)} \leftarrow W_{21}^{(2)} + \eta \cdot \delta_{Y2}^{(2)} \cdot Y_{Z2}^{(1)} = 0.5554 + (0.5)(0.7077)(-0.1317) = 0.5087$$

Cálculo de error de capa oculta.

$$\boldsymbol{\nabla E^{(2)} = \delta_{Y1}^{(2)} \cdot \delta_{Z1}^{(1)} \cdot x_0}$$

$$\boldsymbol{\nabla E^{(2)} = \delta_{Y1}^{(2)} \cdot \delta_{Z2}^{(1)} \cdot x_1}$$

$$\boldsymbol{\delta_{Z1}^{(1)} = \delta_{Y1}^{(2)} \cdot \left(1 - Y_{Z1}^{(2)}\right)\left(Y_{Z1}^{(2)}\right) \cdot W_{11}^{(2)} = -0.006935}$$

$$\boldsymbol{\delta_{Z2}^{(1)} = \delta_{Y1}^{(2)} \cdot \left(1 - Y_{Z2}^{(2)}\right)\left(Y_{Z2}^{(2)}\right) \cdot W_{21}^{(2)} = -0.015130}$$

Actualización de peso sináptico.

$$\boldsymbol{v_{01}^{(1)} \leftarrow v_{01}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_0 = 0.1961 + (0.5)(-0.006935\ )(1) = 0.1926}$$

$$\boldsymbol{v_{02}^{(1)} \leftarrow v_{02}^{(1)} + \eta \cdot \delta_{Z2}^{(1)} \cdot x_0 = 0.1922 + (0.5)(-0.015130)(1) = 0.1846}$$

$$\boldsymbol{v_{11}^{(1)} \leftarrow v_{11}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_1 = 0.5 + (0.5)(-0.006935\ )(0) = 0.5}$$

$$\boldsymbol{v_{12}^{(1)} \leftarrow v_{12}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_1 = 0.5 + (0.5)(-0.015130)(0) = 0.5}$$

$$\boldsymbol{v_{21}^{(1)} \leftarrow v_{21}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_2 = 0.6961 + (0.5)(-0.006935\ )(1) = 0.6926}$$

$$\boldsymbol{v_{22}^{(1)} \leftarrow v_{22}^{(1)} + \eta \cdot \delta_{Z2}^{(1)} \cdot x_2 = 0.6922 + (0.5)(-0.015130)(1) = 0.6846}$$

Épocas = 2.

$$\boldsymbol{Para\ \varepsilon \leq 0.001\ en\ P1 = [0\ 1\ ]\quad T1 = [0.04457]}$$

$$\boldsymbol{\text{É}pocas = 278.}$$

$$\boldsymbol{v_{01}^{(1)} = 0.39012 \qquad W_{01}^{(2)} = -1.378799}$$

$$v_{02}^{(1)} = 0.263463 \qquad W_{11}^{(2)} = -1.283647$$

$$v_{11}^{(1)} = 0.5 \qquad W_{21}^{(2)} = -0.9259748$$

$$v_{12}^{(1)} = 0.5$$

$$v_{21}^{(1)} = 0.890124$$

$$v_{22}^{(1)} = 0.763463$$

$Para\ P2 = [1\ 0\ ]\quad T2 = [1]$ Épocas = 0

Propagación hacia adelante.

Capa de entrada.

$$I_{Z1}^{(1)} = v_{0.1}^{(1)} \cdot x_0 + v_{1.1}^{(1)} \cdot x_1 + v_{2.1}^{(1)} \cdot x_2 = 0.7$$

$$Y_{Z1}^{(1)} = g\left(I_{Z1}^{(1)}\right) = 0.6681$$

$$I_{Z2}^{(1)} = v_{0.2}^{(1)} \cdot x_0 + v_{1.2}^{(1)} \cdot x_1 + v_{2.2}^{(1)} \cdot x_2 = 0.7$$

$$Y_{Z2}^{(1)} = g\left(I_{Z2}^{(1)}\right) = 0.6681$$

Capa de salida.

$$I_{Y1}^{(2)} = w_{0.1}^{(2)} \cdot z_0 + w_{1.1}^{(2)} \cdot Y_{Z1}^{(1)} + w_{2.1}^{(2)} \cdot Y_{Z2}^{(1)} = 1.40$$

$$Y_{Y1}^{(2)} = g\left(I_{Y1}^{(2)}\right) = 0.8024$$

Cálculo del error cuadrático.

$$\boldsymbol{E = \frac{1}{2}\left(1 - Y_{YI}^{(2)}\right)^2 = 0.019522}$$

Retropropagación del error.

Cálculo de error capa de salida.

$$\boldsymbol{\nabla E^{(2)} = \delta_{Y1}^{(2)} \cdot Y_I^{(1)}}$$

$$\boldsymbol{\delta_{Y1}^{(3)} = \left(1 - Y_{Y1}^{(2)}\right) \cdot g'\left(Y_{Y1}^{(2)}\right) = \left(1 - Y_{Y1}^{(2)}\right) \cdot \left(1 - Y_{Y1}^{(2)}\right)\left(Y_{Y1}^{(2)}\right) = 0.031}$$

Actualización de peso sináptico.

$$\boldsymbol{W_{01}^{(2)} \leftarrow W_{01}^{(2)} + \eta \cdot \delta_{Y1}^{(2)} \cdot 1 = 0.8 + (0.5)(1)(0.031) = 0.81}$$

$$\boldsymbol{W_{11}^{(2)} \leftarrow W_{11}^{(2)} + \eta \cdot \delta_{Y1}^{(2)} \cdot Y_{Z1}^{(1)} = 0.3 + (0.5)(0.031)(0.8024) = 0.31}$$

$$\boldsymbol{W_{21}^{(2)} \leftarrow W_{21}^{(2)} + \eta \cdot \delta_{Y1}^{(2)} \cdot Y_{Z2}^{(1)} = 0.6 + (0.5)(0.031)(0.8024) = 0.61}$$

Cálculo de error de capa oculta.

$$\boldsymbol{\nabla E^{(2)} = \delta_{Y1}^{(2)} \cdot \delta_{Z1}^{(1)} \cdot x_0}$$

$$\boldsymbol{\nabla E^{(2)} = \delta_{Y1}^{(2)} \cdot \delta_{Z2}^{(1)} \cdot x_1}$$

$$\boldsymbol{\delta_{Z1}^{(1)} = \delta_{Y1}^{(2)} \cdot \left(1 - Y_{Z1}^{(2)}\right)\left(Y_{Z1}^{(2)}\right) \cdot W_{11}^{(2)} = 0.0020}$$

$$\boldsymbol{\delta_{Z2}^{(1)} = \delta_{Y1}^{(2)} \cdot \left(1 - Y_{Z2}^{(2)}\right)\left(Y_{Z2}^{(2)}\right) \cdot W_{21}^{(2)} = 0.0041}$$

Actualización de peso sináptico.

$$\boldsymbol{v_{01}^{(1)} \leftarrow v_{01}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_0 = 0.2 + (0.5)(0.0020)(1) = 0.2010}$$

$$\boldsymbol{v_{02}^{(1)} \leftarrow v_{02}^{(1)} + \eta \cdot \delta_{Z2}^{(1)} \cdot x_0 = 0.2 + (0.5)(0.0041)(1) = 0.2020}$$

$$\boldsymbol{v_{11}^{(1)} \leftarrow v_{11}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_1 = 0.5 + (0.5)(0.0020)(1) = 0.5010}$$

$$\boldsymbol{v_{12}^{(1)} \leftarrow v_{12}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_1 = 0.5 + (0.5)(0.0041)(1) = 0.5020}$$

$$\boldsymbol{v_{21}^{(1)} \leftarrow v_{21}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_2 = 0.7 + (0.5)(0.0020)(0) = 0.7}$$

$$\boldsymbol{v_{22}^{(1)} \leftarrow v_{22}^{(1)} + \eta \cdot \delta_{Z2}^{(1)} \cdot x_2 = 0.7 + (0.5)(0.0041)(0) = 0.7}$$

Épocas = 1.

Propagando hacia delante con nuevos pesos.

Capa de entrada.

$$\boldsymbol{I_{Z1}^{(1)} = v_{0.1}^{(1)} \cdot\ x_0 +\ v_{1.1}^{(1)} \cdot x_1 + v_{2.1}^{(1)} \cdot x_2} = 0.7020$$

$$\boldsymbol{Y_{Z1}^{(1)} = g\left(I_{Z1}^{(1)}\right)} = 0.6686$$

$$\boldsymbol{I_{Z2}^{(1)} = v_{0.2}^{(1)} \cdot\ x_0 +\ v_{1.2}^{(1)} \cdot x_1 + v_{2.2}^{(1)} \cdot x_2} = 0.704$$

$$\boldsymbol{Y_{Z2}^{(1)} = g\left(I_{Z2}^{(1)}\right)} = 0.6690$$

Capa de salida.

$$\boldsymbol{I_{Y1}^{(2)} = w_{0.1}^{(2)} \cdot\ z_0 +\ w_{1.1}^{(2)} \cdot Y_{Z1}^{(1)} + w_{2.1}^{(2)} \cdot Y_{Z2}^{(1)}} = 1.4125$$

$$\boldsymbol{Y_{Y1}^{(2)} = g\left(I_{Y1}^{(2)}\right) =} 0.8061$$

Cálculo del error cuadrático.

$$\boldsymbol{E = \frac{1}{2}\left(1 - Y_{YI}^{(2)}\right)^2 = 0.01879}$$

Retropropagación del error.

Cálculo de error capa de salida.

$$\boldsymbol{\nabla E^{(2)} = \delta_{Y1}^{(2)} \cdot\ Y_I^{(1)}}$$

$$\boldsymbol{\delta_{I1}^{(3)} = \left(1 - Y_{Y1}^{(2)}\right) \cdot g'\left(Y_{Y1}^{(2)}\right) = \left(1 - Y_{Y1}^{(2)}\right) \cdot \left(1 - Y_{Y1}^{(2)}\right)\left(Y_{Y1}^{(2)}\right) =\ 0.0302}$$

Actualización de peso sináptico.

$$\boldsymbol{W_{01}^{(2)} \leftarrow W_{01}^{(2)} + \eta \cdot \delta_{Y1}^{(2)} \cdot 1 = 0.8156 + (0.5)(0.0302)(1) =\ 0.8356}$$

$$\boldsymbol{W_{11}^{(2)} \leftarrow W_{11}^{(2)} + \eta \cdot \delta_{Y1}^{(2)} \cdot Y_{Z1}^{(1)} = 0.3104 + (0.5)(0.0302)(0.6686) =\ 0.3204}$$

$$\boldsymbol{W_{21}^{(2)} \leftarrow W_{21}^{(2)} + \eta \cdot \delta_{Y1}^{(2)} \cdot Y_{Z2}^{(1)} = 0.6104 + (0.5)(0.0302)(0.6690) =\ 0.6205}$$

Cálculo de error de capa oculta.

$$\boldsymbol{\nabla E^{(2)} = \delta_{Y1}^{(2)} \cdot \delta_{Z1}^{(1)} \cdot x_0}$$

$$\boldsymbol{\nabla E^{(2)} = \delta_{Y1}^{(2)} \cdot \delta_{Z2}^{(1)} \cdot x_1}$$

$$\boldsymbol{\delta_{Z1}^{(1)} = \delta_{Y1}^{(2)} \cdot \left(1 - Y_{Z1}^{(2)}\right)\left(Y_{Z1}^{(2)}\right) \cdot W_{11}^{(2)} = 0.0020}$$

$$\boldsymbol{\delta_{Z2}^{(1)} = \delta_{Y1}^{(2)} \cdot \left(1 - Y_{Z2}^{(2)}\right)\left(Y_{Z2}^{(2)}\right) \cdot W_{21}^{(2)} = 0.0040}$$

Actualización de peso sináptico.

$$\boldsymbol{v_{01}^{(1)} \leftarrow v_{01}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_0 = 0.2010 + (0.5)(0.0020)(1) = 0.2020}$$

$$\boldsymbol{v_{02}^{(1)} \leftarrow v_{02}^{(1)} + \eta \cdot \delta_{Z2}^{(1)} \cdot x_0 = 0.2020 + (0.5)(0.0040)(1) = 0.2041}$$

$$\boldsymbol{v_{11}^{(1)} \leftarrow v_{11}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_1 = 0.510 + (0.5)(0.0020)(1) = 0.5020}$$

$$\boldsymbol{v_{12}^{(1)} \leftarrow v_{12}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_1 = 0.5020 + (0.5)(0.0040)(1) = 0.5041}$$

$$\boldsymbol{v_{21}^{(1)} \leftarrow v_{21}^{(1)} + \eta \cdot \delta_{Z1}^{(1)} \cdot x_2 = 0.7 + (0.5)(0.0020)(0) = 0.7}$$

$$\boldsymbol{v_{22}^{(1)} \leftarrow v_{22}^{(1)} + \eta \cdot \delta_{Z2}^{(1)} \cdot x_2 = 0.7 + (0.5)(0.0040)(0) = 0.7}$$

Épocas = 2.

$$\boldsymbol{Para\ \varepsilon \leq 0.001\ \ Para\ \ P2 = [1\ 0\ ]\ \ \ T2 = [0.9554414542921429]}$$

$$\boldsymbol{\text{É}pocas = 261}$$

$$\boldsymbol{v_{01}^{(1)} = 0.298962 \qquad W_{01}^{(2)} = 1.608346}$$

$$v_{02}^{(1)} = 0.348283 \qquad W_{11}^{(2)} = 0.854760$$

$$v_{11}^{(1)} = 0.598962 \qquad W_{21}^{(2)} = 1.63359$$

$$v_{12}^{(1)} = 0.648283$$

$$v_{21}^{(1)} = 0.7$$

$$v_{22}^{(1)} = 0.7$$

Referencias bibliográficas

[1] S. G. Shiva, Advanced Computer Architecures, Boca Raton, FL: Taylor & Francis Group, 2006, pp. 1-2.

[2] A. J. &. P. P. Kaushik Roy, «Towards spike-based machine intelligence,» *Nature,* vol. 607–617, n° 575, p. 11, 27 Noviembre 2019.

[3] Leon O. Chua, «Memristor The missing circuit element,» *IEEE Transactions on circuit theory,* vol. 18, n° 5, pp. 507-519, 1971.

[4] D. Hebb, The organization of behavior, New York: McHill University, 1949.

[5] I. N. d. Silva, Artificial Neural Networks a practical course, Switzerland: Springer, 2017.

[6] R. Rojas, Neural Networks, Berlin: Springer-Verlag, 1996.

[7] H. H. Aghdam, Guide to convolutional Neural Networs, Cham: Springer, 2017.

[8] M. T. R. W. Henry Markram, «Interneurons of the neocortical inhibitory system,» *Nature reviews,* vol. 5, pp. 793-807, 2004.

[9] W. GERSTNER, «Adaptation and firing patterns,» de *NEURONAL DYNAMICS*, Cambridge, Cambridge University Press, 2014, pp. 136-165.

[10] E. M. Izhikevich, «Simple Model of Spiking Neurons,» *IEEE Transactions of neural networks,* vol. 14, n° 6, pp. 1569-1572, 2003.

[11] Intel, «www.intel.com,» [En línea]. Available: https://www.intel.com/content/www/us/en/research/neuromorphic-computing.html. [Último acceso: 14 05 2020].

[12] M.Davis, «Loihi: a neuromorphic manicore processor with on-chip learning,» *IEEE Micro,* vol. 38, pp. 82-99, 2018.

[13] P. D. Jayawan H.B. Wijekoon, «Compact silicon neuron circuit with spiking and bursting behaviour,» *Neural Networks,* vol. 21, n° 2-3, pp. 524-534, 2008.

[14] H. Hodgkin AL, « A quantitative description of membrane current and its application to conduction and excitation in nerve.,» *J Physiol,* vol. 117, n° 4, pp. 500-544, 1952.

[15] tsmc, «www.tsmc.com,» [En línea]. Available: https://www.tsmc.com/english/dedicatedFoundry/technology/logic.htm#l_3-micron_technology. [Último acceso: 20 5 2020].

[16] R. J. Baker, «Introduction to CMOS Design,» de *CMOS: Circuit Design, Layout, and Simulation*, New Jersey, IEEE PRESS, 2010, pp. 1-2.

[17] B. Razavi, «Introduction to analog design,» de *Design of Analog CMOS*, Los Angeles, Mc.Graw Hill, 2017, pp. 1-19.

[18] R. J. Baker, «VLSI Layout Examples,» de *CMOS: Circuit Design, Layout, and Simulation*, New Jersey, IEEE Press , 2010, pp. 411-433.

[19] W. Gerstner, «Networks of neurons and population activity,» de *NEURONAL DYNAMICS*, Cambridge, Cambridge University Press, 2014, pp. 287-417.

[20] P. R. Gray, Analysis and desing of analog integrated circuits, John Wiley and Sons: John Wiley and Sons, 2001.

[21] E. M., «Spike-timing Dynamics of Neuronal Groups,» *Cerebral cortex,* vol. 14, nº 8, pp. 933-944, 2004.

[22] I. E. M., «Phil. Trans. R. Soc.,» 2010. [En línea]. Available: https://doi.org/10.1098/rsta.2010.0130. [Último acceso: 12 June 2020].

[23] M. Davis, «nice work shop,» 2 May 2018. [En línea]. Available: https://niceworkshop.org/wp-content/uploads/2018/05/02-MDavies-Loihi-Plenary.pdf. [Último acceso: 2020 June 12].