núcleo de ingeniería biomédica
facultades de medicina e ingeniería
universidad de la república

Redes Neuronales: elementos para su implementación con circuitos eléctricos.

German Fierro[1]

[1]Estudiante de la Facultad de Ingeniería - Universidad de la República, Uruguay

Email: germanhfierro@hotmail.com;

Resumen

Este trabajo pretende mostrar los elementos para la implementación hardware de una red neuronal, en particular se estudian dispositivos especialmente diseñados con este fin. Se realiza un análisis de sus posibles tecnologías y principales características. Para entender como están dispuestos, es necesario entender los conceptos esenciales sobre redes neuronales, por esta razón se hace una presentación de todo el camino a recorrer para llegar a lo que nos concierne. Finalmente, se analiza la implementación hardware de una red neuronal en una de estas tecnologías, desde su entrenamiento hasta el resultado.

1. Introducción

Una red neuronal es una unidad de procesamiento de información, cuyo paradigma es el propio cerebro humano. Se pretende simular tanto el funcionamiento del sistema nervioso como el procesamiento de la información realizado por el cerebro. Nace aquí una interrogante, con lo poderosa que son las computadoras hoy en día, ¿por qué tomar el camino neuronal para procesar la información? Aunque las máquinas sean capaces de computar millones de operaciones por segundo, no son capaces de entender el significado de las formas visuales o de distinguir entre distintas clases de objetos. El cerebro cuenta con diminutos procesadores realizando pequeñas funciones que trabajando en conjunto con un fin común, permiten obtener respuestas significativamente más rápidas que computador en algunos contextos, además se caracteriza por:

1. .Robustez, tolerancia a fallas, desde que nacemos se mueren neuronas pero vemos que nuestro funcionamiento sigue intacto.
2. Flexibilidad, aprende en nuevos ambientes, es auto programable.
3. Procesa información difusa.

4. Funcionamiento en paralelo.

5. Es pequeño, compacto y consume poca energía.

Estas características fueron precursoras del estudio de las redes neuronales ya hace algún tiempo, ya que son el camino para entender a fondo el principio de funcionamiento del cerebro. Es de notar que el interés proviene de un público bastante amplio, desde la psicología a la ingeniería.

2. Neurona Artificial

Para reproducir estas habilidades en las redes neuronales artificiales(de ahora en más ANN) es necesario concebir conceptos esenciales de nuestro sistema nervioso, como lo son el cálculo paralelo, memoria distribuida y adaptabilidad. Es natural pensar que para este objetivo, el punto de partida es el modelado de la neurona, entidad elemental de nuestro sistema nervioso. Este archivo .tex fue elaborado de forma de permitir, con pequeños cambios, elaborar fácilmente un documento de documentación.

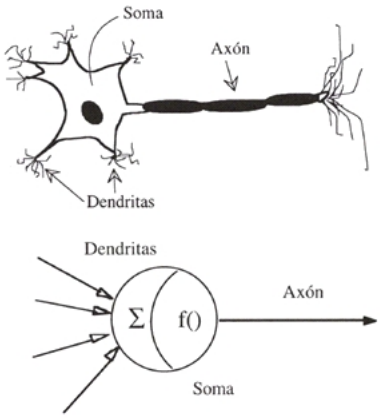

Figura 1: modelo de la neurona, las dendritas como receptor de información, el somo donde se procesa, y el axón transmisor.

Al momento del aprendizaje estas entidades son capaces de formar estructuras más complejas al interactuar entre sí a través de una región llamada sinapsis. "Se estima que existen alrededor de 1011 neuronas en el cerebro de un ser humano, y que cada una de ellas recibe entre 1000 y 10.000 contactos sinápticos". En definitiva, la neurona juega el rol de un ladrillo en la .estructura del saber". La región sináptica es donde entra en contacto el axón de una neurona transmisora con las dendritas de la neurona receptora estableciendo así una comunicación unidireccional. La información es integrada en la neurona receptora, produciendo una salida en esta. "La sinapsis puede poseer distinta eficiencia en su capacidad de transducción. Además, existen sinapsis que facilitan la respuesta (excitatorias), mientras que existen también sinapsis que dificultan la respuesta (inhibitorias). Esto está relacionado con el tipo de neurotransmisor utilizado por la sinapsis. Estas propiedades (eficiencia, estimulación o inhibición) son globalmente representadas en los modelos matemáticos mediante números llamados "pesos sinápticos", cuya magnitud mide la eficiencia de la transducción y su signo positivo o negativo señala si la sinapsis es excitatoria o inhibitoria".

Entonces el modelo matemático de una neurona contemplando lo que es la sinapsis neuronal, es el siguiente:

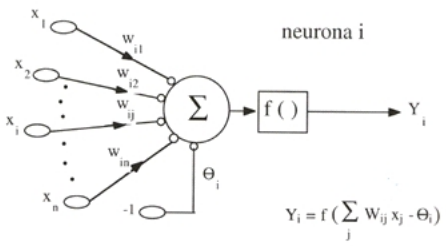

Figura 2: Modelo de interacción sináptica de una neurona.la salida como una función de las entradas.

Donde, x_j son las entradas de una neurona, ω_{ij} son pesos sinápticos de la i-ésima neurona, θ_i es el nivel de disparo de la i-ésima neurona y $f()$ función de transferencia.

Como muestra el gráfico, la función de transferencia se aplica a la cantidad: $h = \sum_j \omega_{ij} x_j - \theta_i$

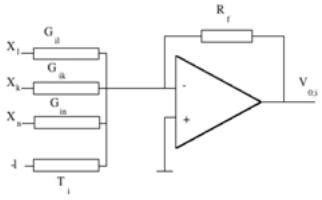

Figura 3: Analogia de la neurona con el amplificador operacional.

Este es el llamado potencial post-sináptico. Dos comentarios al respecto tienen lugar aquí, el primero, dando un primer acercamiento a lo que es la implementación en Hardware, el modelo es plausible con un simple amplificador operacional y resistores como muestra la figura .

Segundo las funciones de transferencia son de distintos tipos, en particular se buscan que sean continuas y monóticas crecientes porque es el comportamiento que las neuronas realmente tienen. En particular las transferencias más usadas son:

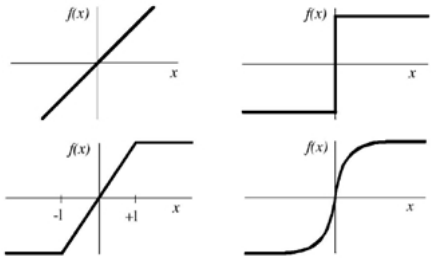

Figura 4: Funciones de transferencia principales que se aplican al potencia post-sináptico.

La neurona, es en realidad la entidad elemental de estructuras más complejas que se forman debido a contactos sinápticos entre distintas neuronas. Se identifica, a su vez que estas estructuras están formadas por

capas. Estas pueden ser modeladas como muestra la figura, por un bloque de "s"neuronas al que se le ingresa un vector p de r.[en]tradas, luego cada neurona produce su potencial post-sináptico haciendo una combinación lineal de las entradas y ajustando su nivel de disparo. Finalmente la salida del bloque se obtiene al aplicar l la función de transferencia a dicho potencial. En consecuencia una capa es bien definida dando su matriz de pesos y su función de transferencia.

$$\begin{pmatrix} a_1 \\ . \\ . \\ . \\ a_s \end{pmatrix} = \begin{pmatrix} \omega_{11} & ... & \omega_{1r} \\ . & & . \\ . & & . \\ . & & . \\ \omega_{1s} & ... & \omega_{sr} \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ . \\ . \\ . \\ p_s \end{pmatrix} + \begin{pmatrix} b_1 \\ . \\ . \\ . \\ b_s \end{pmatrix}$$

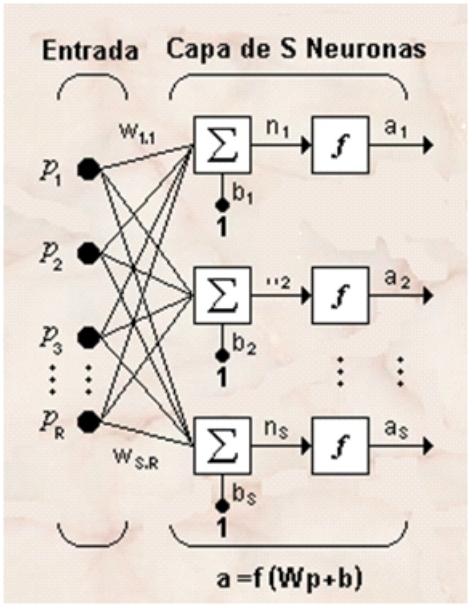

Figura 5: modelo de una capa de neuronas.

3. 3 Implementación de redes neuronales

Como se mencionaba anteriormente el estudio de redes neuronales ha abarcado muchas disciplinas, entre ellas a la que a nosotros concierne que es la ingeniería. Diversos problemas que un computador no ha sabido resolver, se han tratado con redes neuronales. Tengamos en cuenta que una máquina no es capaz de tomar una decisión en el momento pertinente, y menos aún cuando se presentan situaciones difíciles de cuantificar. En cambio una red puede ser entrenada y con base a experiencias similares que adquirió con anterioridad podría eventualmente tomar una decisión adecuada.

En principio la implementación de una red neuronal es posible tanto en Hardware como en Software, el inconveniente es que los computadores bajo la arquitectura von Neumann tienen un procesamiento secuencial. Esto hace que la implementación software presente limitaciones en aplicaciones en tiempo real, y hace inviable el ejecutar algoritmos de aprendizaje para redes de gran porte. En cambio, la implementación hardware utiliza procesamiento en paralelo de la misma forma que la propias redes neuronales lo hacen, consiguiendo superioridad frente al software por lo menos en lo que respecta a demoras.

3.1. Un poco de historia

"Los primeros intentos de realizar redes neuronales en hardware son tan antiguos como la propia área. En 1951 Marvin Minsky construyó la primera neuro-computadora denominada Snark. Aunque esta máquina operaba bien desde el punto de vista técnico (realizaba el ajuste automático de los pesos), nunca llegó a resolver ninguna aplicación con resultados importantes desde el punto de vista del procesamiento de la información suministrada."

."El primer prototipo exitoso fue el Mark I Perceptron desarrollado por Frank Rosenblatt, Charles Wightman y otros. El Mark I fue utilizado con éxito en el reconocimiento de caracteres (Hecht-Nielsen, 1991). "

En los 90's prolifera la realización electrónica de redes neuronales. Empresas como Intel, Siemens, Philips, Hitachi, AT&T, etc, desarrollan los primeros productos comerciales de importancia (procesador neuronal ETANN de Intel, neurocomputador CNAPS).

Carver Mead (Mead,1989), con el desarrollo de sistemas neuronales electrónicos que imitan la estructura neuronal de órganos sensoriales como es por ejemplo la retina.

3.2. Redes Hardware

En la actualidad existen dispositivos montados en un circuito integrado especialmente diseñados para implementar redes neuronales en Hardware. Entre ellos encontramos redes neuronales entrenables(de ahora en mas TNN), estos permiten implementar modelos de redes neuronales, y algoritmos de aprendizaje, sacando provecho de su funcionalidad de procesado en paralelo, el diagrama de bloques de la figura6 muestra su estructura interna. Básicamente son chips que se les puede aplicar algoritmos de entrenamiento y de esta manera almacenar en su memoria la matriz de pesos que caracteriza un capa de neuronas. Existen diferentes tipos, entre ellos encontramos Chips digitales , analógicos e híbridos.

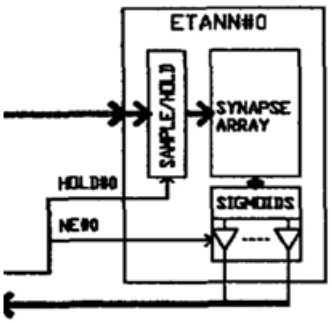

Figura 6: Diagrama de bloques de red entrenable ETANN

Los digitales se caracterizan por que son fácil de integrar a otros aplicaciones ya que están procesando información digital, el almacenamiento de la matriz de pesos es simple (en ram). El principal problema es que general son en más lentos que los analógicos, ya que para utilizarlos en una red con señales analógicas es necesario hacer conversiones y eso enlentece el procesamiento global.

Los analógicos son caracterizados por su velocidad, permiten configuraciones neuronales que los digitales no, ya que podría eventualmente sumar salidas en neuronas codificadas en corriente. En contrapartida están

expuestos a dependencias con la temperatura, a la manufacturación, al envejecimiento. Almacenaje de la matriz de peso es complicado, especialmente si se requiere almacenaje no volátil.

Finalmente los híbridos tratan de quedarse con los mejor de ambos mundos, el procesado interno es analógico pero el almacenado es digital. En el anexo se encuentran especificaciones de MD1220, ETANN, NeuroClassifier, que son chips representando tecnología digital, analógica e híbrida respectivamente.

3.3. Aplicación

Como aplicación de las redes hardware exponemos el trabajo M.Holler, AS- S.Tam, J.Brauch, sobre Neural Network Recognition of Objects Based On Impact Dynamics, donde se utilizan tres redes neuronales entrenables ETANN para el reconocimiento de tres objetos desconocidos. Un acelerómetro montado en una plataforma de madera produce distintas formas de onda, cuando distintos objetos golpean sobre esta. Después de una rutina de entrenamiento las redes neuronales entenderán los patrones fundamentales de las formas de onda para finalmente clasificar el origen de la colisión, esto es, se aprende a identificar la colisión de un material conocido con uno que no lo es. Se busca además obtener la clasificación en tiempo real, por lo que es aprovechada la arquitectura en paralelo de los ETANN para identificar el objeto.

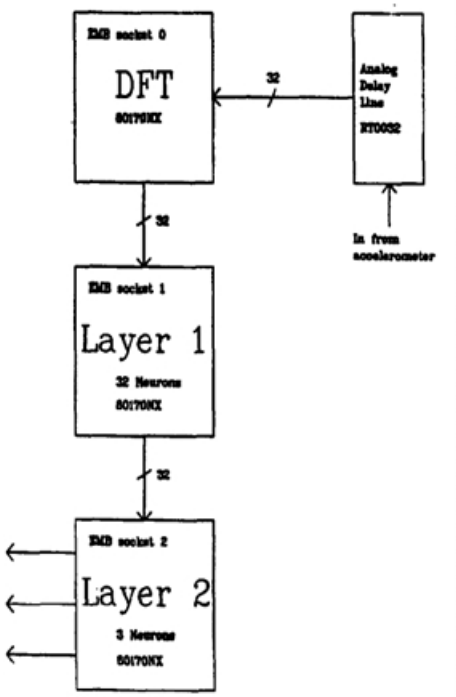

Figura 7: Diagrama de bloques de sistema capaz de reconocer objetos a traves del impacto con otro conocido, se utilizan 3 capas de neuronas, una para el bloque TFD, y otras 2 para el bloque de reconocimiento.

El sistema es el mostrado en la figura7 donde la onda obtenida del acelerómetro es muestreada a 5kHz, para que un bloque obtenga la transformada discreta de Fourier(TFD). Finalmente la representación en frecuencia es la información que procesa la unidad de reconocimiento, determinando así el objeto desconocido.

La vibración captada por el acelerómetro, es muestreada por el RT0032. Este simplemente envía por un bus paralelo de 32 canales la señal muestrea a 5kHz, reteniendo su salida hasta que nuevos 32 medidas sean obtenidos. Naturalmente esto introduce un retardo de 6ms, por lo que es el cuello de botella en lo que a tiempo se refiere. Este vector de 32 valores es la entrada a la unidad TFD.

La unidad de reconocimiento está compuesta por dos capas neuronales, la primera de 32 neuronas cuya entrada es la TFD hallada en el bloque antes analizado. Mientras que la segunda de tan solo 3 neuronas, tiene en sus salidas el resultado de la clasificación del objeto, cuando el objeto j golpea la plataforma la neurona j luego de 6ms envía una respuesta.

3.3.1. Implementación y entrenamiento

El entrenamiento del sistema es realizado con el Intel Neural Network Training System(iNNTS) que es una herramienta hardware para entrenar y simular el chip. Esta se comunica con el mismo a través del ETANN Multilayer chip Board(EMB) que básicamente es una placa con conectores donde introducir los chip ETANN, permite controlar las señales de entrada y salida entre capas. El kit de trabajo cuenta además con el iDynaMind que es un software también producido por intel, que se utiliza para simular la red y para bajar la matriz de pesos al chip. Veremos cómo se usan estas herramientas, para cumplir el cometido. Se utilizan dos EMB, uno en la instancia de entrenamiento como interface iNNTS-Hardware, y otro al momento de uso. Se debe tener en cuenta que antes de entrenar al sistema, se debe definir la topología de la red. Esto es logrado en el iDynaMind, a través de código de alto nivel, que inicializa las matrices de pesos en el chip, estos datos son guardados en la EPROM del chip por lo que representan memoria no volátil.

Es usual utilizar un TNN para implementar un bloque TFD, ya que el trabajo de una neurona artificial es en definitiva realiza una suma ponderada de entradas. Teniendo esto en cuenta, al ingresar las entradas y pesos adecuados se consigue hallar la TFD de una señal. Con lo que se consigue implementar el bloque con un ETANN, utilizando 32 de sus 64 neuronas de una sola capa de las que cuenta el chip. La matriz de pesos se carga a través del iDynaMind de acuerdo a la ecuación 1

$$f_l = \sum_{k=0}^{N-1} x_k e^{-\frac{2\pi j}{N} kl} \tag{1}$$

Las salidas son 16 números complejos (32 reales) que representan valores de la transformada en determinadas frecuencias equiespaciadas, estas van desde la continua hasta 15kHz.

Observemos que el bloque anterior en realidad no necesitó entrenamiento para cargar su matriz de pesos, sin embargo en los bloques siguientes si es necesario. Para realizarlo se crea un circuito para que el iNNTS detecte el impacto, una vez que esto ocurre espera 6ms para enviar un comando (HOLD) a la unidad TFD para que mantenga la salida, para respetar el tiempo que le llevaría al RT0032 obtener su salida. Luego, el iNNTS adquiere la salida del TFD con un ADC, y almacena en un archivo los datos obtenidos junto a la respuesta que se esperaría en la capa de salida. El procedimiento se repite sistemáticamente más de 1000 veces y se compila el archivo. En esta instancia el iDynaMind cuenta con una primera aproximación de la matriz de pesos del bloque reconocimiento, que son descargados en el chip. Hay que tener presente, que a priori los valores de los pesos en la computadora donde todo es ideal, son distintos a las que el chip realmente debería tener para que funcione correctamente, por eso es en definitiva una primera aproximación. Finalmente para obtener los ajustes "más finos"de la matriz, se implementa el entrenamiento çhip-in-loop". El simulador ingresa los datos relevados directamente en el chip, y censa la salida, luego ajusta infinitesimalmente los valores pertinentes de la matriz para obtener las salidas esperadas. Una vez corrida esta optimización el sistema está listo para utilizarse, con lo que se ingresan los chips a la EMB de aplicación y se obtienen los siguientes resultados.

3.3.2. Resultados

Los resultados fueron interesantes, golpearon la plataforma de madera con tres materiales distintos un total de 1500 veces, 500 con cada uno. El sistema respondió para un objeto con un cien por ciento de efectividad, mientras que para los otros dos cometió un solo error. En le figura se muestra las respuestas de las neuronas cuando el objeto uno colisiona contra la plataforma de madera.

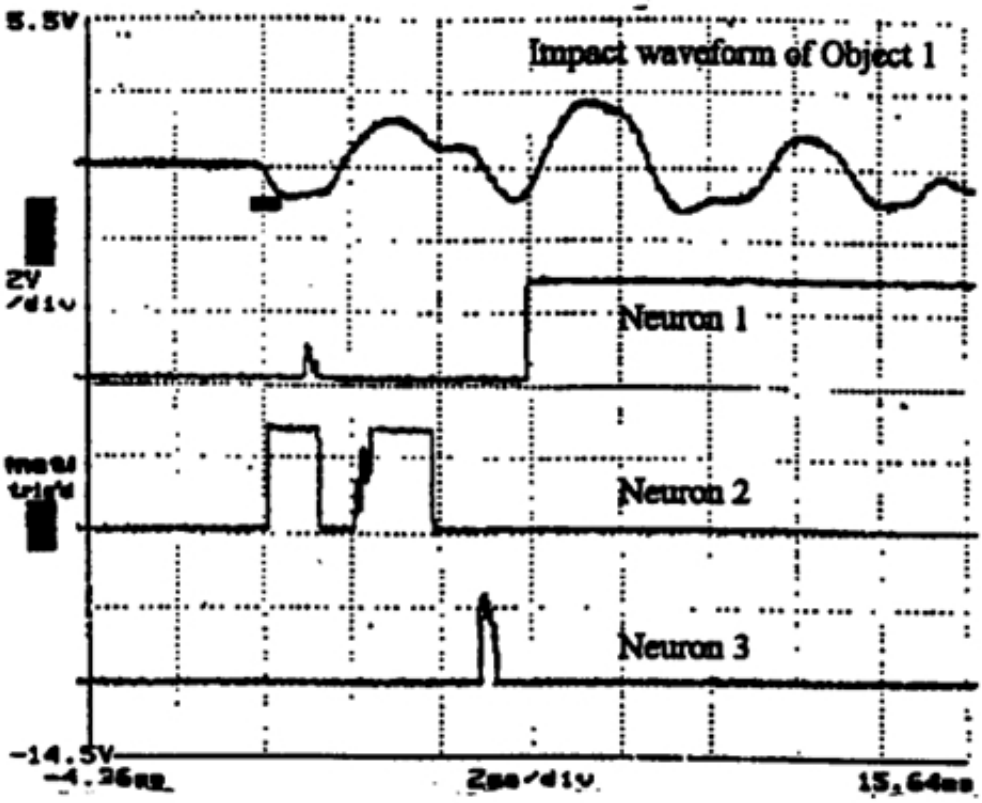

Figura 8: diagramas de tiempos, curva superior muestra la salida del acelerómetro al impactar el objeto 1, las tres restantes muestran las salidas de las respectivas neurones.

4. Conclusiones

En el trabajo se analizó la implementación de una red neuronal en hardware con un dispositivo especialmente diseñado para ella, comprobándose el gran desempeño del mismo. Sin embargo deben tenerse presente los costos al momento de elegir esta tecnología para nuestros fines. El principal costo es el tiempo empleado para realizar el entrenamiento necesario para la aplicación. Aplicaciones que además no son demasiado complejas, observemos que es factible tomar otro camino para implementar la aplicación presentada aquí. En general las aplicaciones logradas por estos sistemas son de ésta orden de complejidad, cuyos tiempo de aprendizaje son relativamente extensos [9]. El problema de la complejidad se podría subsanar utilizando chips con capas de mayor tamaño, sin embargo es razonable que los tiempos de aprendizaje crezcan considerablemente. Entonces existe este compromiso de complejidad-tiempo que el usuario debería tomar en

cuenta al momento de diseñar.

Agradecimientos

Quiero agradecer a Franco Simini, Gabriel Geido y Jorge Lobo, por brindarme información y guiarme en la elaboración. A Eduardo Mizraji por presentarme este increíble campo de redes Neuronales, y sobre todo a Nicolás Casaballe por ser un amigo incondicional, siempre aconsejando de la mejor manera.

Referencias

1. Juan P. Oliver, André Fonseca de Oliveira, Julio Pérez Acle, Roberto J. de la Vega, Rafael Canetti , *Implementation of Adaptive Logic Networks on an FPGA board*,

2. (1)Juan Carlos Moctezuma Eugenio, (2) César Torres Huitzil, *ESTUDIO SOBRE LA IMPLEMENTACIÓN DE REDES NEURONALES ARTIFICIALES USANDO XILINX SYSTEM GENERATOR*,

3. Bonifacio Martín del Brío, Carlos Serrano Cinca *Fundamentos de las redes neuronales artificiales: hardware y software*,

4. (Compiladas por Franco Simini, *Ingeniería Biomédica perspectivas desde el Uruguay*,

5. *http://www.particle.kth.se/ lindsey/HardwareNNWCourse/home.html*,

6. *http://electronica.com.mx/neural/*,

7. *http://ohm.utp.edu.co/neuronales/main.htm*,

8. M.Holler, AS- S.Tam, J.Brauch, *Neural Network Recognition of Objects Based On Impact Dynamics*,

9. S.Tam', M.Holler', J.Brauch', A.Pine2, A.Peterson3, S.Anderson', S.Deiss4, *A Reconfigurable Multi-Chip Analog Neural Network; Recognition and Back-Propagation Training*,