idetec

Libro de Actas

Estudiantes de Grado y Posgrado

Secretaría de Ciencia, Tecnología y Posgrado

UTN VILLA MARIA

Compilación:
Ing. Marcelo Cejas, Ing. Javier Gonella, Ing. Fabián Sensini

Congreso de Investigaciones y Desarrollos en Tecnología y Ciencia, IDETEC 2020 : Libro de Actas - Estudiantes de Grado y Posgrado / Micaela Mariel Achetoni ... [et al.] ; compilado por Marcelo Oscar Cejas ; Javier Nicolás Gonella ; Fabián Marcelo Sensini. - 1a ed. - Ciudad Autónoma de Buenos Aires : edUTecNe, 2021.
268 p. ; 240 x 150 cm.

ISBN 978-987-4998-69-9

1. Ingeniería. 2. Tecnologías. 3. Medio Ambiente. I. Achetoni, Micaela Mariel. II. Cejas, Marcelo Oscar, comp. III. Gonella, Javier Nicolás, comp. IV. Sensini, Fabián Marcelo, comp.

CDD 607.3

Edición y Diseño:

Universidad Tecnológica Nacional – República Argentina
Rector: Ing. Héctor Eduardo **Aiassa**
Vicerrector: Ing. Haroldo **Avetta**
Secretaria Académica: Ing. Liliana Raquel **Cuenca Pletsch**

Universidad Tecnológica Nacional – Facultad Regional Villa María
Decano: Ing. Pablo Andrés **Rosso**
Vicedecano: Ing. Franco **Salvático**

edUTecNe – Editorial de la Universidad Tecnológica Nacional
Coordinador General a cargo: Fernando H. **Cejas**
Director Colección Energías Renovables, Uso Racional de Energía, Ambiente: Dr. Jaime **Moragues.**

Queda hecho el depósito que marca la Ley Nº 11.723

Sarmiento 440, Piso 6 (C1041AAJ)
Buenos Aires, República Argentina
Publicado Argentina – Published in Argentina

ISBN 978-987-4998-69-9

PARALELIZACIÓN DE REDES NEURONALES EN EL ÁMBITO DE LA VISIÓN COMPUTACIONAL

Mariela F. Galdamez[1], Pamela A. Chirino[1], Paola G. Caymes-Scutari[1,2], Germán Bianchini[1]
[1]Laboratorio de Investigación en Cómputo Paralelo/Distribuido Departamento de Ingeniería en Sistemas de Información Facultad Regional Mendoza/Universidad Tecnológica Nacional Rodríguez 273 (M5502AJE) Mendoza, +54 261 5244579
[2]Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
mariela.galdamez.16@gmail.com, pamelaachirino@gmail.com, pcaymesscutari@frm.utn.edu.ar, gbianchini@frm.utn.edu.ar

Resumen

En este artículo, se presenta un estudio inicial sobre las redes neuronales aplicadas en la visión computacional y sobre el objetivo de estudiar su paralelización. La visión computacional puede utilizarse en diversos campos de la ciencia y la ingeniería, y las redes neuronales, como una herramienta de la inteligencia artificial, permiten resolver problemas complejos que surgen en éstas. Sin embargo, la complejidad de la resolución de un problema se traslada a su aprendizaje, puesto que se necesitará una red neuronal con una estructura de tamaño importante para resolverlo. La complejidad de las redes neuronales yace en la velocidad de cómputo necesaria para conseguir una red neuronal funcional en un tiempo razonable, y es y es aquí donde se introduce la necesidad de potenciar la velocidad de aprendizaje. El paradigma de cómputo paralelo constituye una posibilidad para atender esta demanda al habilitar la distribución de tareas o la carga de procesamiento en diversos procesadores. En este trabajo, se estudian las principales características de las redes neuronales involucradas en la visión computacional, y las características y potencialidades más importantes del paradigma paralelo. A partir de ello, se proponen algunas posibilidades de paralelización sobre las redes neuronales, bajo la hipótesis de que será posible expresar estadísticamente los beneficios del paralelismo en el aprendizaje de redes neuronales, y comprobar que, al paralelizar, se consigue en un tiempo menor una convergencia de los pesos a algo cercano al óptimo global en cada neurona de la red neuronal.

Introducción

Existen diversas aplicaciones donde puede ser de gran utilidad la visión computacional. Entre ellas se encuentran, según mencionan García y Caranqui [1], sistemas de visión que permiten la navegación o el guiado automático de máquinas en la robótica, el reconocimiento de objetos inmersos en imágenes y su posterior clasificación, realizar vigilancia para detectar la presencia y movimiento de cuerpos extraños, entre muchas otras. Por lo tanto, sabiendo sus diversas utilizaciones, es importante saber en qué consiste la misma. La visión computacional es una de las disciplinas de la inteligencia artificial que tiene como objetivo tratar con computadoras la información obtenida a partir de imágenes, recibidas por dispositivos como cámaras, para así reconocer objetos y la posición de estos en el ambiente capturado [2]. En pocas palabras, la visión computacional busca automatizar las tareas que puede realizar el sistema visual humano, y en este proyecto se optó por utilizar redes neuronales como modelo computacional a emplear.

Una red neuronal es capaz de aprender a realizar distintas tareas y de seguir aprendiendo de los datos que recibe ya una vez puesta en funcionamiento; pero conseguir un aprendizaje adecuado a su utilidad es una tarea muy compleja. La razón de su complejidad es la cantidad de cómputo necesario para procesar los ejemplos elegidos como datos de entrada para su entrenamiento. Mientras más grande sea el tamaño de una red neuronal, aumenta el tiempo necesario para llegar a un funcionamiento óptimo de la misma, y si la cantidad de ejemplos es de gran volumen, el tiempo es mucho mayor. Por esto mismo, se presenta el paralelismo como una herramienta de reducción de tiempo de ejecución [3], ya que el mismo permite obtener una mayor velocidad de cómputo puesto que las tareas de procesamiento de datos se dividen en una mayor cantidad de procesadores. Una red neuronal de gran tamaño, que busca resolver un problema concreto de

dificultad importante, se le hace indispensable la utilización del paralelismo para conseguir un desempeño adecuado en un tiempo razonable.

Redes neuronales

Una red neuronal está caracterizada por su estructura, su función de activación y su aprendizaje. Según Fausett en [4], "la estructura de una red neuronal consiste en un cierto número de elementos simples de procesamiento llamados neuronas". Una neurona artificial [5], consta de dos etapas: en una primera etapa, se combinan las entradas provenientes de otras neuronas teniendo en cuenta los pesos de las sinapsis, obteniendo como resultado la entrada neta o excitación de la neurona; y en la segunda etapa, la entrada neta se utiliza como valor para aplicar en una función de activación, cuya responsabilidad es la generación de la salida de la neurona, que se propagará a otras neuronas. La mayoría de los modelos de redes neuronales artificiales utilizan un sesgo externo, o bias, el cual es un parámetro adicional de la neurona, igual que los pesos. En determinados modelos de neuronas binarias, el sesgo servirá para determinar el umbral de activación de la neurona, el punto a partir del cual la neurona activa su salida. Si el nivel de excitación de la neurona queda por debajo del umbral, se mantiene inactiva su salida. En cuanto su entrada neta supera el umbral de activación, se activa la salida de la neurona. En la Figura 1 se representa una neurona artificial, y su equivalencia a una neurona biológica. Cabe mencionar que el modelo matemático utilizado en redes neuronales se abstrajo del funcionamiento de un cerebro humano.

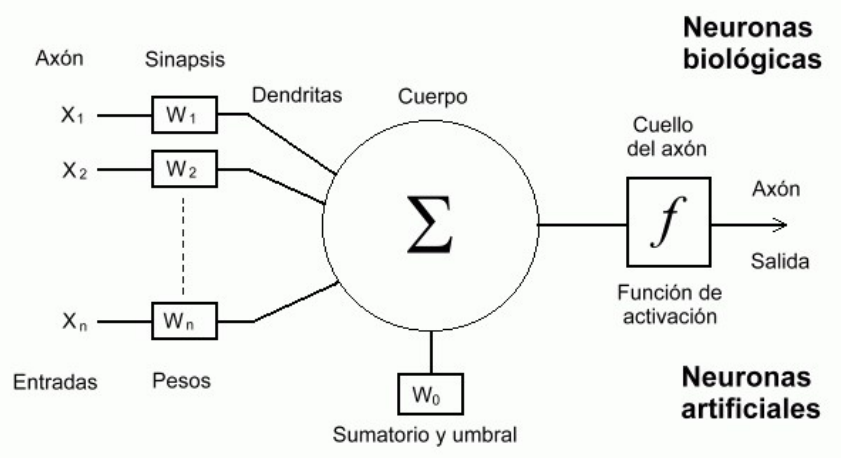

Fig.1. Relación entre neurona artificial y biológica

Habitualmente, las redes neuronales artificiales están formadas por conjuntos de neuronas que se agrupan en una única capa de entrada, una única capa de salida y en capas ocultas, posicionadas entre las dos mencionadas anteriormente [4], como se ilustra en la Figura 2. Por convención, se denota por x_i a cada una de las n entradas recibidas por una capa de neuronas formada por m neuronas [5]. A menudo se asume que el sesgo b_j no es más que un peso adicional vinculado a una entrada fija con valor 1, es decir, $w_{0j}=bj$ y $x_0=1$. La salida de cada neurona la representaremos por y_j , habitualmente un valor binario o real, que podemos considerar análogo al estado de activación (binario) o a la frecuencia de activación (real).

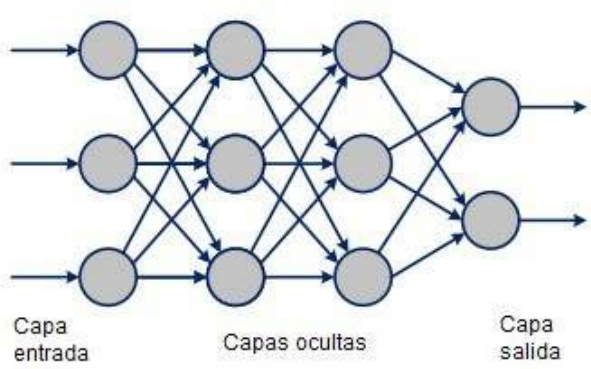

Fig.2. Red neuronal multicapa

En la etapa de integración de entradas, una neurona artificial combina las diferentes entradas x_i con sus pesos w_{ij} con la neurona j para determinar su entrada neta net_j,

$$net_j = \sum_i x_i w_{ij} \tag{1}$$

En la etapa de activación, una neurona artificial utiliza el valor asociado a su entrada neta para generar una salida y_j . El modelo más habitual genera una salida de tipo numérico a partir de la entrada neta de la neurona,

$$y_j = f\left(net_j\right) = f\left(\sum_i x_i w_{ij}\right) \tag{2}$$

donde la función f es la función de activación de la neurona, usualmente no lineal.

La clasificación de una red neuronal depende de su estructura, específicamente de la cantidad de capas ocultas y de la forma de conexión entre las neuronas de las mismas [5]. Para este proyecto, se escogió trabajar con redes neuronales multicapas, que se caracterizan por tener una o varias capas ocultas, y por las conexiones inexistentes entre neuronas de la misma capa.

Por último, lo más importante en una red neuronal es su capacidad de aprendizaje. Una vez elegido un volumen de datos considerable para su entrenamiento y para la validación de este, la red neuronal procede a hacer un ajuste de los pesos de las conexiones entre sus neuronas a medida que recorre los datos de entrada.

Paralelismo

El paralelismo [3] es un modelo de computación en el cual se divide problemas grandes en varios problemas pequeños, como se muestra en la Figura 3, que se computan de forma simultánea (en paralelo). El objetivo es utilizar múltiples procesadores permitiendo así que un problema mayor sea resuelto en un período de tiempo aceptable al aumentar la velocidad de cómputo. Además, el paralelismo permite evaluar datos varias veces con diferentes valores de entrada, ya que se pueden ejecutar múltiples instancias del mismo programa en diferentes procesadores/computadoras simultáneamente.

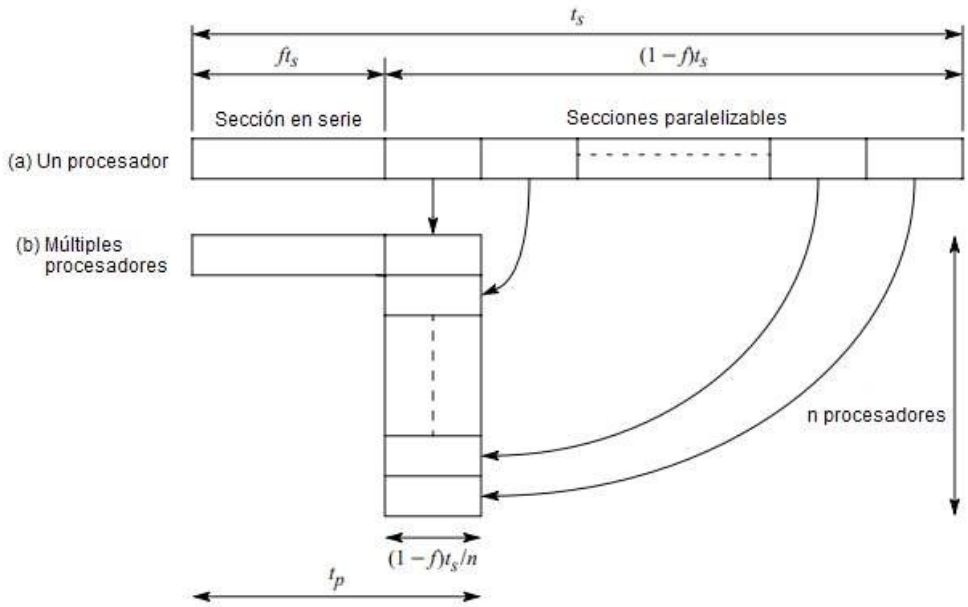

Fig.3. Paralelización de un problema

Para medir la eficiencia de un algoritmo paralelo, se hace uso del Factor de Speedup[3], el cual es una medida relativa del tiempo de ejecución del mejor algoritmo secuencial t_S ejecutándose en un solo procesador y el tiempo de ejecución para resolver el mismo problema en un multiprocesador t_p.

$$S(p)=t_S/t_p \tag{3}$$

Siendo p la cantidad de procesadores utilizados, la Eficiencia permite saber cuánto tiempo son utilizados los procesadores en el cálculo,

$$E=S(p)/p\times 100\,\% \tag{4}$$

Avances y objetivos

Actualmente, el proyecto se encuentra en estado inicial donde hasta el momento se ha estudiado las bases sobre las redes neuronales. Se espera aplicar el tema de estudio en el desarrollo de una red neuronal orientada al reconocimiento y clasificación de objetos en una imagen. Además, brindar una mejora en la velocidad de aprendizaje de las redes neuronales multicapa al utilizar paralelismo y expresar estadísticamente los beneficios del cómputo paralelo en éstas. Para contabilizar estas estadísticas, se busca realizar una comparativa del tiempo de ejecución requerido para conseguir un aprendizaje adecuado de distintas redes neuronales multicapa. Éstas se diferenciarán según el tamaño, la cantidad de ejemplos que requieren aprender para funcionar correctamente, y la cantidad y las características de los recursos utilizados.

El desarrollo detallado por Valero Gómez en [7], constituye un ejemplo de la aplicación efectiva del cómputo paralelo y un modelo a mejorar respecto a la paralelización de redes neuronales multicapa. En base a este, proponemos la aplicación del cómputo paralelo a nivel de capas de una red neuronal. En el laboratorio donde se estudia este proyecto, se dispone de un clúster de computadoras con 7 nodos para realizar las pruebas del algoritmo. La primer idea es distribuir el cómputo en los 7 nodos de modo que, en un instante de tiempo, un nodo estaría realizando los cálculos de 1 o más neuronas de una capa. Luego, se espera sumarle la paralelización a nivel de caso de aprendizaje, es decir, en un instante de tiempo un nodo paralelizaría 1 o más ejemplos para agilizar el aprendizaje de la red neuronal, sin afectar idealmente la precisión del mismo. Con esta primera aproximación al problema, se pretende encarar el mismo de forma que ofrezca una mejoría tanto en el tiempo que demanda el procesamiento de las muestras a aprender como en la

precisión con la que la red neuronal resuelve el reconocimiento de imágenes. El primer obstáculo que se presenta se sitúa en el lenguaje que se utiliza con mayor frecuencia para la inteligencia artificial. Dado que la mayoría de las librerías de inteligencia artificial como TensorFlow [7] están programadas en Python, estamos tomando las decisiones de diseño e implementación necesarias para lograr la compatibilidad con Python y C, siendo C el lenguaje apto para utilizar en nuestro clúster. La principal ventaja de la utilización de un clúster, es que se puede construir con computadoras de cualquier estructura, por lo que el costo es menor si se compara con la compra de computadoras de iguales especificaciones. Esto se debe a que se puede reutilizar recursos o ir mejorando la calidad del clúster de forma gradual. Sin embargo, la diferencia en su estructura puede producir que la sincronización se dificulte y este sería un problema que también se espera encontrar una solución, el equilibrar el costo – resultado esperado.

A partir de los avances que se sigan obteniendo en torno a la visión computacional, se encarará a futuro un proyecto que tendrá como objetivo detectar delitos o accidentes viales en tiempo real, con el fin de alertar de forma inmediata a los organismos correspondientes.

La línea de trabajo presentada y sus futuros avances se enmarca en un proyecto realizado en el Laboratorio de Investigación en Cómputo Paralelo [6], el cual propone la iniciación de docentes y estudiantes en el proceso de investigación científica. Se espera mejorar la experiencia en torno al proceso de investigación en temas de interés. En el caso particular de este artículo, la investigación se dirige a la aplicación del cómputo paralelo en el ámbito de la visión computacional.

Conclusiones

La visión computacional utiliza ampliamente redes neuronales para diferentes funciones, como clasificar objetos, reconocer textos u objetos, entre otras. En el caso particular de este trabajo, se busca aplicar las redes neuronales al reconocimiento y clasificación de objetos en una imagen, para a futuro reconocer objetos y personas para determinar acciones de delitos u accidentes viales. Sin embargo, entrenar una red neuronal puede demorar una cantidad de tiempo inaceptable si su estructura es de un tamaño apreciable. Por esto mismo, el cómputo paralelo es la herramienta computacional necesaria para entrenar una red neuronal de gran tamaño, ya que el mismo es de gran utilidad en problemas donde se suele necesitar enormes cantidades de cálculos repetitivos en grandes volúmenes de datos, para dar resultados válidos dentro de un período de tiempo razonable.

Referencias

[1] García I., Caranqui V. (2015) La Visión Artificial y los Campos de Aplicación. Artículo Revista Digital “Tierra Infinita”. URL: https://revistasdigitales.upec.edu.ec/index.php/tierrainfinita/article/view/76 (fecha de consulta: abril 2021).

[2] Davies E.R. (2012) Computer and Machine Vision: Theory, Algorithms, Practicalities. Elsevier.

[3] Wilkinson B., Allen M. (2005) Parallel Programming – Techniques and Applications Using Networked Workstations and Parallel Computers. Pearson.

[4] Fausett L. (1994) Fundamentals of Neural Networks – Architectures, Algorithms, and Applications. Pearson.

[5] Berzal F. (2018) Redes Neuronales y Deep Learning. Autoedición.

[6] Proyecto PID 'Formación de docentes y alumnos de grado como Investigadores Científicos Iniciales en las áreas de Informática y Ciencias de la Computación' Disposición SCTyP°222/2019. https://www.utn.edu.ar/images/Secretarias/SCTYP/DISP-0222-04-12-2019_CI.pdf (fecha de consulta: abril de 2021)

[7] Valero Gómez J. (2019) ‘Análisis de modelos predictivos basados en visión computacional aplicados al paralelismo’. Tesis Universidad Nacional de Moquegua.