

# Image2 Circuit - Clasificador de elementos electrónicos usando una CNN mediante Tensorflow

Proyecto Final. Introducción a la programación en paralelo con MPI, OpenMP y CUDA.

Oscar Andres Rosas Hernandez  
SoyOscarRH@ciencias.unam.mx  
Facultad de Ciencias, UNAM

## ABSTRACT

Las redes neuronales convoluciones (CNN) son la herramienta por excelencia en la actualidad para crear un sistema automático que analiza imágenes. En este proyecto buscaré investigar la fiabilidad y los resultados de usar una CNN para poder clasificar imágenes tanto fotografías como imágenes esquemáticas creadas por computadora de diversos elementos eléctricos (se piensa trabajar como mínimo con los 4 elementos básicos: resistores, capacitores e inductores y fuentes de voltaje) usando la librería TensorFlow y haremos una comparativa de la diferencia que causa usar GPU's en el desarrollo de la misma.

## I. OBJETIVOS

Se dará una breve introducción a las redes neuronales convoluciones y términos como data augmentation y drop out. Una vez entendida la teoría, se procederá a generar la implementación de un modelo en TensorFlow para Python, medir su rendimiento así como la influencia que tiene usar una GPU en los tiempos de entrenamiento del mismo. Finalmente se mostrarán los resultados de evaluación del modelo propuesto aquí.

## II. MARCO TEÓRICO

### II-A. Machine learning

Definimos al machine Learning como el área que estudia como hacer que las computadoras puedan aprender de manera automática usando experiencias (data) del pasado para predecir el futuro.

La mayoría del aprendizaje automático práctico utiliza aprendizaje supervisado. El aprendizaje supervisado es donde se tiene variable(s) de entrada (x) y una variable de salida (Y) y se utiliza un algoritmo para “aprender” la función que mapea la entrada a la salida.

$$Y = f(X) \quad (1)$$

El objetivo es aproximar la función tan bien que cuando tenga nuevos datos de entrada (X) pueda predecir las variables de salida (Y) para esos datos.

Se llama aprendizaje supervisado porque el proceso de un algoritmo que aprende del conjunto de datos de capacitación puede verse como un profesor que supervisa el proceso de aprendizaje. Conocemos las respuestas correctas, el modelo realiza predicciones de forma iterativa sobre los datos de entrenamiento y es corregido por el profesor.

Durante el entrenamiento, un algoritmo de clasificación recibirá una serie de datos con una categoría asignada. El trabajo de un algoritmo de clasificación es tomar un valor de entrada y asignarle

una clase o categoría que se ajuste según los datos de entrenamiento proporcionados. [1]

*II-A1. Accuracy:* Esta dada por “de todos nuestros datos, que tanto % clasificamos correctamente” En otras palabras:

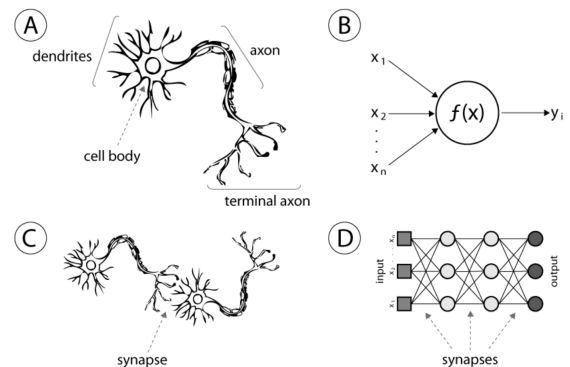
$$accuracy := \frac{true\_negative + true\_positive}{all} \quad (2)$$

Ahora, el “accuracy” no es siempre la mejor métrica, sobre todo cuando nuestros datos no están balanceados, es decir cuando en cada una de nuestras clases tenemos una cantidad diferente de datos.

### II-B. Redes Neuronales

Es una práctica común conceptualizar una red neuronal artificial como una neurona biológica. Técnicamente hablando, una red neuronal artificial, como lo expresó Ada Lovelace, es el “cálculo del sistema nervioso”. Una versión muy, muy simplificada del sistema nervioso. Tanto es así que algunos neurocientíficos probablemente han perdido las esperanzas tan pronto como vieron lo sencilla y alejada de la realidad que es.

Las redes neuronales consisten en una capa de entrada, una o varias capas ocultas y una capa de salida. Hay que tener en cuenta que las capas se cuentan desde la primera capa oculta. Cuantas más capas ocultas haya, más compleja es la red.



[3]

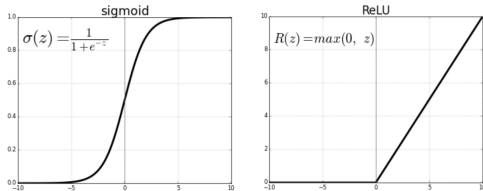
Las sinapsis toman el valor de la entrada almacenada de la capa anterior y la multiplican por un peso (w). Entonces podemos decir que la tarea principal dentro del aprendizaje profundo es calibrar (o encontrar) los pesos a un valor específico para obtener un resultado preciso.

“Es como cuando te estás bañando y hay que mover las dos manijas para encontrar el punto perfecto donde esta la temperatura ideal.”

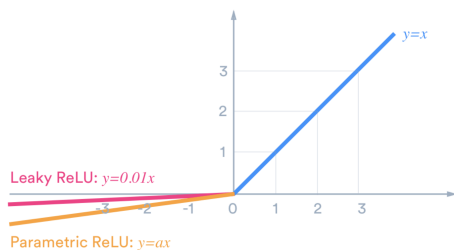
Posteriormente, las sinapsis propagán hacia adelante sus resultados a las capas siguientes. En general se pueden ver como:

$$Y = b + \sum w_i x_i \quad (3)$$

Finalmente tras realizar este cálculo y para lograr la no linealidad que generalmente buscamos se aplican funciones de activación, entre las más famosas están la ReLU (y su prima la Leaky ReLU) y la sigmoide.



[3]

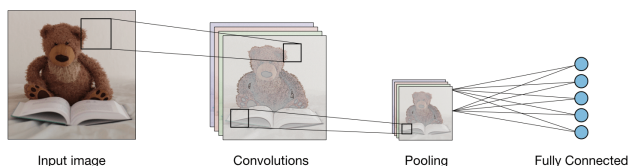


[3]

### II-C. Redes Neuronales Convolucionales

Cada parte de una red neuronal convolucional está entrenada para realizar una tarea, por lo que el entrenamiento de cada una de las partes se desarrolla de manera individual y se efectúa más rápido. Estas redes se utilizan en especial para el análisis de imágenes.

Las CNN están formadas por 3 partes generalmente:



- Las capas convolucionales usan filtros que realizan operaciones de convolución mientras se escanea / pasa la “imagen” entrada, sus hiperparámetros incluyen el tamaño del filtro y el salto que se da, así como que es lo que pasa en las orillas de la imagen de entrada, la salida se llama mapa de características o mapa de activación.
- Agrupación (pooling): Es una operación que busca la disminución de la resolución, generalmente se aplica después de una capa de convolución; produce cierta invariancia espacial. En particular, el max y mean pooling son tipos especiales de pooling donde se toman el valor máximo y promedio, respectivamente.
- Lineales: Totalmente conectada: Funciona en una entrada aplanada donde cada entrada está conectada a todas las neuronas, para esto tenemos que “espagetificar” nuestra entrada, estas si

están presentes, se encuentran hacia el final de la arquitectura y se pueden usar para optimizar objetivos como las probabilidades de una clase.

### II-D. Dropout

En pocas palabras, el dropout se refiere a ignorar unidades (es decir, neuronas) durante la fase de entrenamiento de cierto conjunto de neuronas que se elige al azar.

Por “ignorar”, quiero decir que estas unidades no se consideran durante un pase particular hacia adelante o hacia atrás.

Más técnicamente, en cada etapa de entrenamiento, los nodos individuales se eliminan de la red con probabilidad  $1-p$  o se mantienen con probabilidad  $p$ , de modo que queda una red reducida; Los bordes entrantes y salientes a un nodo abandonado también se eliminan.

**II-D1. ¿Por qué lo necesitamos?:** ¿Por qué necesitamos literalmente apagar partes de una red neuronal? La respuesta a estas preguntas es para evitar el overfitting.

Una capa totalmente conectada ocupa la mayoría de los parámetros y, por lo tanto, las neuronas desarrollan una codependencia entre ellas durante el entrenamiento, lo que frena el poder individual de cada neurona, lo que lleva a un ajuste excesivo de los datos de entrenamiento.

Para esto es que el dropout cae como anillo al dedo.

## III. EL PROBLEMA

### III-A. El dataset

### III-B. El modelo que se usó para resolverlo

### III-C. Resultados

## REFERENCIAS

- [1] Aidan Wilson, Sep 29, 2019  
<https://towardsdatascience.com/a-brief-introduction-to-supervised-learning>
- [2] June 1, 2020  
<https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks>
- [3] June 1, 2020  
<https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>