

## Trabajo practico N° 3 – Ejercicio 2

Masi Juan, Perez Damián, Vildozola Mariano  
37981647, 35375255, 32254403  
Martes 19 a 23 hs, 6

<sup>1</sup>Universidad Nacional de La Matanza,  
Departamento de Ingeniería e Investigaciones Tecnológicas,  
Florencio Varela 1903 - San Justo, Argentina

**Resumen.** En este notebook se usara Keras, MNIST y CatBoost para clasificación de dígitos del 0 al 9 escritos a mano para crear una Convolutional Neural Network (CNN) para crear un modelo que identifica y predice una imagen de un dígito que nunca ha visto con un 99% de exactitud a mano.

**Palabras claves:** Python, GPU, CatBoost, Cuda, Colab, Kera, Machine learning, Gradient Boosted Decision Trees.

### 1 Introducción

Google Colaboratory, también llamado "Colab", permite ejecutar y programar en Python directamente en el navegador. Posee ciertas ventajas: no requiere configuración, se pueden correr en la nube, es posible elegir correr en una CPU o GPU de forma gratuita, y permite compartir contenido fácilmente.

CUDA es una plataforma de computación paralela general y un modelo de programación. Puede administrar GPU, recursos de hardware y utilizar GPU para realizar computación compleja.

El programa CUDA consta de dos partes: una es el código de host que se ejecuta en la CPU y otra es el código del dispositivo ejecutándose en GPU. El programa paralelo que se ejecuta en la GPU también se conoce como kernel.

Utilizando las herramientas de Keras y data MNIST(Modified National Institute of Standards and Technology database), que contiene un conjunto de entrenamiento de 60000 imágenes de dígitos manuscritos, se va a predecir una imagen de un dígito, que nunca ha visto con un 99% de exactitud.

## 2 Desarrollo

- En este notebook estamos utilizando redes neuronales convulsionales, es un tipo de red neuronal artificial con aprendizaje supervisado que procesa sus capas imitando la corteza visual del ojo humano para identificar distintas características en las entradas que en definitiva hacen que puedan identificar objetos y producir inferencias a partir de los datos disponibles, gracias a su capacidad de aprendizaje.
- Inputs: Se tienen las 60000 imágenes provenientes de la base de datos de MNIST.
- Output: Imagen procesada inferida, resultado de haber ejecutado esta red neuronal.
- Manual de usuario: Primero se debe ejecutar el armado del ambiente, donde se almacenan las imágenes, recibidas del dataSet de MNIST. Como entorno de ejecución se tiene que cambiar el tipo de entrono de ejecución y como acelerador de hardware, elegir GPU. Luego se define el armado y ejecución del modelo, donde se definen los parámetros del modelo, en el cual hay un parámetro *task\_type*, el cual sirve para seleccionar el entorno de ejecución, es decir CPU o GPU. Después de realizar esto se ejecuta visualizando la imagen del dígito predecido.
- [https://github.com/juanagustinmasi/VacunarTech/blob/main/TP3/Cuaderno\\_2\\_martes\\_grupo6\\_2021.ipynb](https://github.com/juanagustinmasi/VacunarTech/blob/main/TP3/Cuaderno_2_martes_grupo6_2021.ipynb)

## 3 Conclusiones

Uno de los problemas surgidos en el desarrollo del ejercicio, fue utilizar el procesamiento con CPU ya que tardaba demasiado en ejecutar para obtener los resultados, lo que demoraba las pruebas para comparar la eficiencia entre GPU y CPU.

Adicionalmente, tuvimos que descargar la biblioteca CatBoost porque no está incluida en la API de TensorFlow.

La conclusión de mayor importancia que encontramos, realizando este notebook, es la gran diferencia de tiempo que hay utilizando el procesamiento con GPU contra el procesamiento con CPU.

## 4 Referencias

- 1- <https://keras.io/>

- 2- <https://catboost.ai/>
- 3- <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119564843.ch5>
- 4- <https://www.sciencedirect.com/science/article/abs/pii/S0040162521000901>