



## Aprendizaje profundo Taller 2

Profesor:

Ing. Julio Omar Palacio Niño, M.Sc.  
[palacio\\_julio@javeriana.edu.co](mailto:palacio_julio@javeriana.edu.co)

El objetivo del siguiente taller es hacer un análisis de clasificación empleando redes neuronales convolucionales (CNN)

### Consideraciones de presentación del trabajo

Para realizar el análisis del problema se deberá descargar el dataset indicado en el taller y disponer de un ambiente de trabajo que incluya un intérprete de Python3 (Anaconda, Colab, etc.).

Puede emplear las librerías para el tratamiento, manipulación y visualización de resultados de datos como pandas, matplotlib, etc.

Para la programación de las redes puede programar sus propios algoritmos en Python, o emplear las librerías de Scikit-learn, TensorFlow y Keras.

La entrega del taller debe ser en un documento PDF tipo informe con el desarrollo de los puntos y preguntas del trabajo, así como también el código fuente.

Todo se entregará por medio del campus virtual.

### Taller

#### 1) Comprensión del dataset

Se trabajará con el dataset de la página Kaggle, por lo cual deberá ser descargado del siguiente enlace

<https://www.kaggle.com/datasets/wjybuqi/traffic-sign-classification-and-recognition>

#### 2) Comprensión del dataset

- ¿Qué información presenta el dataset?,
- Describir las características de las imágenes
- Indicar conjuntos de entrenamiento y pruebas
- Indicar las clases de clasificación

### 3) Construcción del dataset

Para la construcción del dataset realizar un particionamiento de 80% como conjunto de entrenamiento y 20% como conjunto de pruebas

### 4) Elaboración del modelo

Para la construcción del clasificador la red neuronal deberá contar con las siguientes especificaciones

- Capa convolucional con 32 detectores de características, tamaño de kernel de 5x5, función de activación lineal rectificada (ReLU) de paso 1 y con padding.
- Capa de max pooling de tamaño 5x5
- Capa de Flattening
- Capa full conectada con 100 neuronas y función de activación lineal rectificada (ReLU)
- Capa de salida del mismo número de las clases a clasificar y función de activación softmax.

### 5) Entrenamiento

Para el proceso de entrenamiento deberá tener los siguientes parámetros

- Epochs: 1-50
- Batch: 1-64 (1-2-4-16-32-64)

Escoger uno de ellos y justificar su respuesta

### 6) Elaboración del modelo 2

Para la construcción del clasificador la red neuronal en un segundo parámetro de configuración deberá contar con las siguientes especificaciones

- Capa convolucional con 48 detectores de características, tamaño de kernel de 3x3, función de activación lineal rectificada (ReLU) de paso 1 y con padding.
- Capa de max pooling de tamaño 2x2
- Capa convolucional con 96 detectores de características, tamaño de kernel de 3x3, función de activación lineal rectificada (ReLU) de paso 1 y con padding.
- Capa de max pooling de tamaño 2x2
- Capa de Flattening
- Capa full conectada con 100 neuronas y función de activación lineal rectificada (ReLU)

- Capa full conectada con 100 neuronas y función de activación lineal rectificada (ReLU)
- Capa de salida del mismo número de las clases a clasificar y función de activación softmax.

## 7) Entrenamiento 2

Para el proceso de entrenamiento deberá tener los mismos parámetros de entrenamiento anteriores.

¿existe alguna dificultad?

## 8) Ajustes

- ¿existe alguna diferencia con el entrenamiento anterior?
- ¿Qué ajustes realizaría al modelo de entrenamiento para lograr obtener un mejor resultado?
- Explique que otros ajustes puede hacerse al dataset, modelo o técnica para construir un nuevo modelo con el cual posiblemente se pueda mejorar la predicción y modelamiento de los datos.
- Considere cambios de la función de activación, aumento o disminución de neuronas y/o capas, función de aprendizaje
- Tratamiento adicional de los datos
- Rendimiento, ¿es necesario agregar o eliminar neuronas o capas?, métodos de entrenamiento (secuencial, batch)
- En qué medida influye agregar una capa de Dropout, y qué nivel

El plantear la hipótesis no necesariamente considera su desarrollo, simplemente justificar las modificaciones que realizaría para mejorar los resultados y obtener un mejor modelo.

En esta etapa del desarrollo queda descartado el uso de *Transfer Learning*, como la implementación de modelos preentrenados que apoyen el proceso de detección de imágenes, dado que el objetivo del ejercicio es determinar las características propias del proceso de convolución.

No se desilusione si los resultados no son los esperados.

## 9) Análisis de Resultados

Construya la matriz de confusión, realice un análisis grafico así como una evaluación de las metricas de evaluación (acuraccy, precisión, recall, f1 score)

Realice un análisis comparativo de cada uno de los resultados de cada modelo.

## 10) Bono

- En base a las hipótesis planteadas en el punto anterior realizar la construcción de uno de los modelos.
- ¿mejoro o empeoró el modelo?, ¿la hipótesis fue correcta?, justifique su respuesta
- Existieron dificultades para la puesta en marcha del modelo

Para realizar el análisis del problema se deberá descargar el siguiente dataset  
Para el desarrollo del presente taller es necesario disponer de un ambiente de trabajo que incluya un intérprete de Python3.

Emplee las librerías para el tratamiento, manipulación y visualización de resultados de datos como pandas, matplotlib, etc.

Para la programación de las redes puede emplear algoritmos puros en Python, o emplear las librerías de Scikit-learn, TensorFlow y Keras

La entrega del taller debe ser en un documento PDF con el desarrollo de las preguntas y los archivos que incluyan las implementaciones.