

Plan Formativo: Ciencia de Datos	Nivel de Dificultad
Módulo: Fundamentos de Deep learning	Medio
Tema: CNN	
Intención del aprendizaje o aprendizaje esperado:	
<ul style="list-style-type: none"> <li>● Implementar un modelo predictivo utilizando redes neuronales convolutivas para el reconocimiento de imágenes en Python</li> </ul>	
Ejercicios planteados	
Caso	
<p>El conjunto de datos Fashion-MNIST es un conjunto de datos de imágenes de artículos de Zalando, con imágenes en escala de grises de 28x28 de 70 000 productos de moda de 10 categorías y 7000 imágenes por categoría. El conjunto de entrenamiento tiene 60.000 imágenes y el conjunto de prueba tiene 10.000 imágenes. ¡Puede verificar esto más tarde cuando haya cargado sus datos! ;)</p> <p><b>Cargar los datos</b></p>	

Keras viene con una biblioteca llamada datasets, que puede usar para cargar conjuntos de datos listos para usar: descarga los datos del servidor y acelera el proceso, ya que ya no tiene que descargar los datos a su computadora. Las imágenes del tren y de la prueba, junto con las etiquetas, se cargan y almacenan en las variables train\_X, train\_Y, test\_X, test\_Y, respectivamente.

```
pip install --upgrade tensorflow

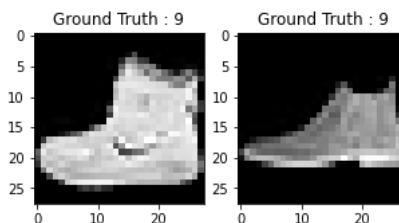
from keras.datasets import fashion_mnist
(train_X, train_Y), (test_X, test_Y) = fashion_mnist.load_data()
import numpy as np
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt
%matplotlib inline

print('Training data shape : ', train_X.shape, train_Y.shape)

print('Testing data shape : ', test_X.shape, test_Y.shape)
```

Realice lo siguiente:

1. Obtenga el tamaño de los datos de test y testeo, además de las dimensiones de las imágenes.
2. Obtenga las categorías de la base
3. Visualice sus datos.



4. Convierta cada imagen de 28 x 28 del tren y del equipo de prueba en una matriz de tamaño 28 x 28 x 1 que se alimenta a la red. Además, debe convertir su tipo a float32, y también debe volver a escalar los valores de píxel en el rango 0 - 1 inclusive.
5. Convierta las etiquetas de entrenamiento y prueba en vectores de codificación one-hot.
6. Separar datos de entrenamiento y testeo

7. Para modelar sus datos, utilice los siguientes paquetes:

```
import keras
from keras.models import Sequential, Input, Model
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras.layers.normalization import BatchNormalization
from keras.layers.advanced_activations import LeakyReLU
```

Usará un tamaño de lote de 64 y también es preferible usar un tamaño de lote más alto de 128 o 256, todo depende de la memoria. Contribuye enormemente a determinar los parámetros de aprendizaje y afecta la precisión de la predicción. Entrenarás la red durante 20 épocas.

```
batch_size = 64
epochs = 20
num_classes = 10
```

8. Agregue la capa de agrupación máxima con MaxPooling2D() y así sucesivamente.

9. Compilación y visualización del modelo

## Preguntas guía

- Qué es Keras
- Redes neuronales convolucionales (CNN)
- Capas

## Recursos Bibliográficos:

## Referencias

[1] Red neruonal convolucional



[https://es.wikipedia.org/wiki/Red\\_neuronal\\_convolutacional](https://es.wikipedia.org/wiki/Red_neuronal_convolutacional)

[2] Introducción a Redes neuronales convolucionales

<https://bootcampai.medium.com/redes-neuronales-convolucionales-5e0ce960caf8>

[3] Creacion de redes en Keras

<http://blog.hadsonpar.com/2021/08/crear-una-red-neuronal-convolutacional-en.html>