

EXAMEN FINAL

TALLER DE APLICACIONES CON MACHINE LEARNING

DÍA 1: Construcción, entrenamiento y visualización de redes neuronales con Keras y Scikit-Learn

Nota: Asegurarse que se tenga Tensorflow, Keras y Scikit-Learn instalados en sus PC locales.

Descargar el dataset fashion_mnist

```
from tensorflow.keras.datasets import fashion_mnist
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
```

Implementar una red Perceptrón multicapa (MLP) usando Scikit-learn y una red convolucional (CNN) usando Keras, entrenarlas y graficar los resultados de rendimiento. Se evaluará la capacidad de codificación, análisis de resultados y visualización.

Instrucciones:

1. Red Perceptrón con Scikit-learn

- Implementa una red MLP usando el dataset digits de sklearn.datasets.
- La red debe tener:
 - Una capa oculta con 64 neuronas
 - Función de activación ReLU
 - Optimizador adam
 - Entrenar durante 05 épocas
- Evalúa el modelo con una matriz de confusión y accuracy_score.

2. Red Convolucional con Keras

- Usa el dataset Fashion MNIST (puedes cargarlo directamente desde Keras).
- Implementa la siguiente arquitectura:
 - Conv2D(32, kernel_size=(3,3), activation='relu')
 - MaxPooling2D(pool_size=(2,2))
 - Conv2D(64, kernel_size=(3,3), activation='relu')
 - MaxPooling2D(pool_size=(2,2))
 - Flatten()
 - Dense(128, activation='relu')
 - Dense(10, activation='softmax')
- Entrena durante 5 épocas.

3. Gráfico y Explicación

- Grafica:
 - Curvas de precisión y pérdida del modelo CNN.
 - Matriz de confusión para MLP.
 - Explica cada gráfico: para qué sirven y su interpretación.
 - Explica qué es el puntaje de precisión (accuracy score) y su interpretación en el modelo.
 - Haz que la red MLP entrene por 100 épocas, ¿existen cambios? Por qué?