

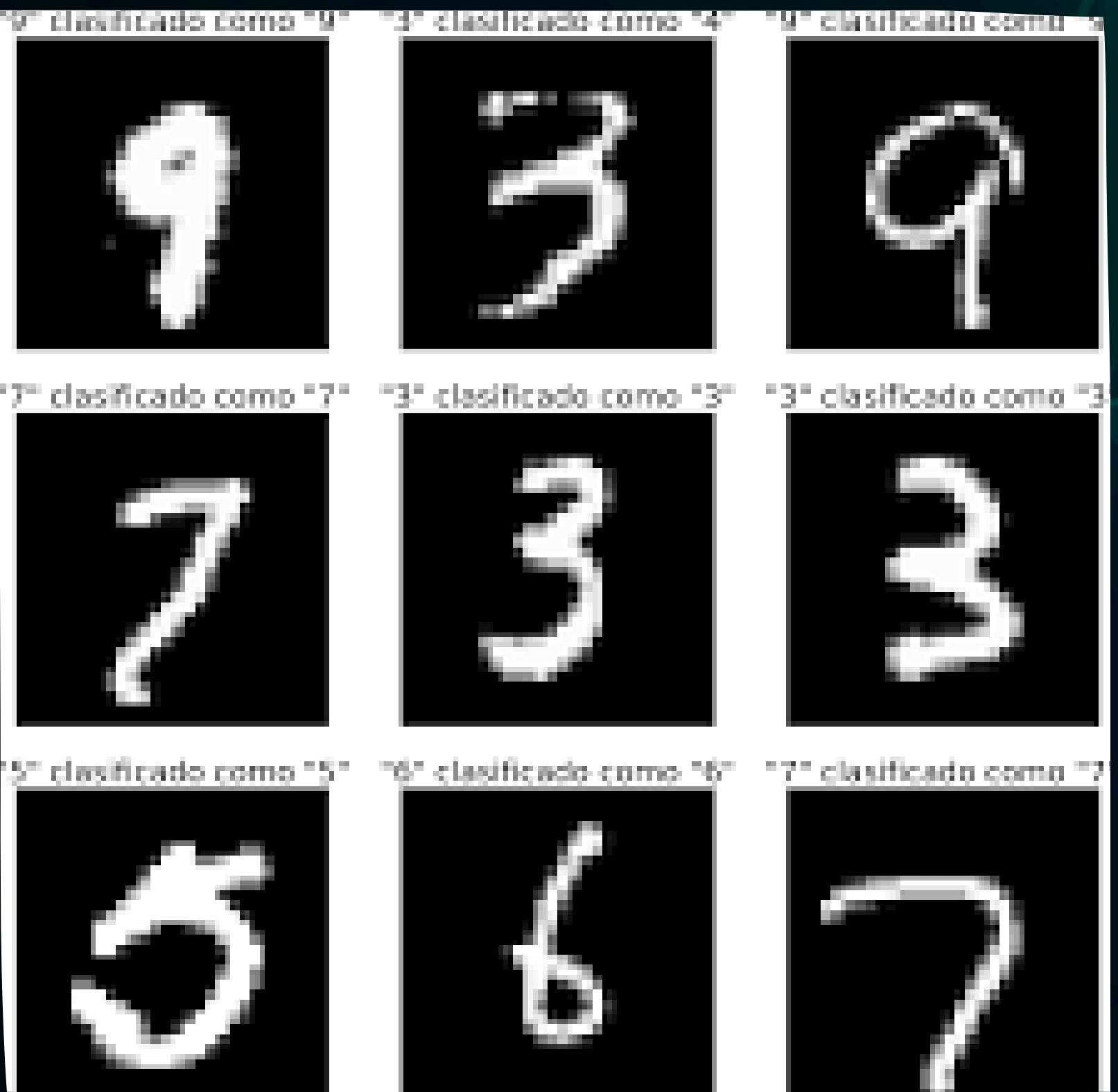
PAM - IA

REDES NEURONALES MULTICAPA Y CONVOLUCIONALES APLICADAS AL RECONOCIMIENTO DE DIGITOS MANUSCRITOS

Por Yamil Aucca

RESUMEN

El proyecto consiste en la creación y el entrenamiento de una MLP y una CNN para el reconocimiento de dígitos manuscritos utilizando el dataset MNIST, para luego implementarlos en una pequeña aplicación web donde poder probar las predicciones que realizan los modelos entrenados.

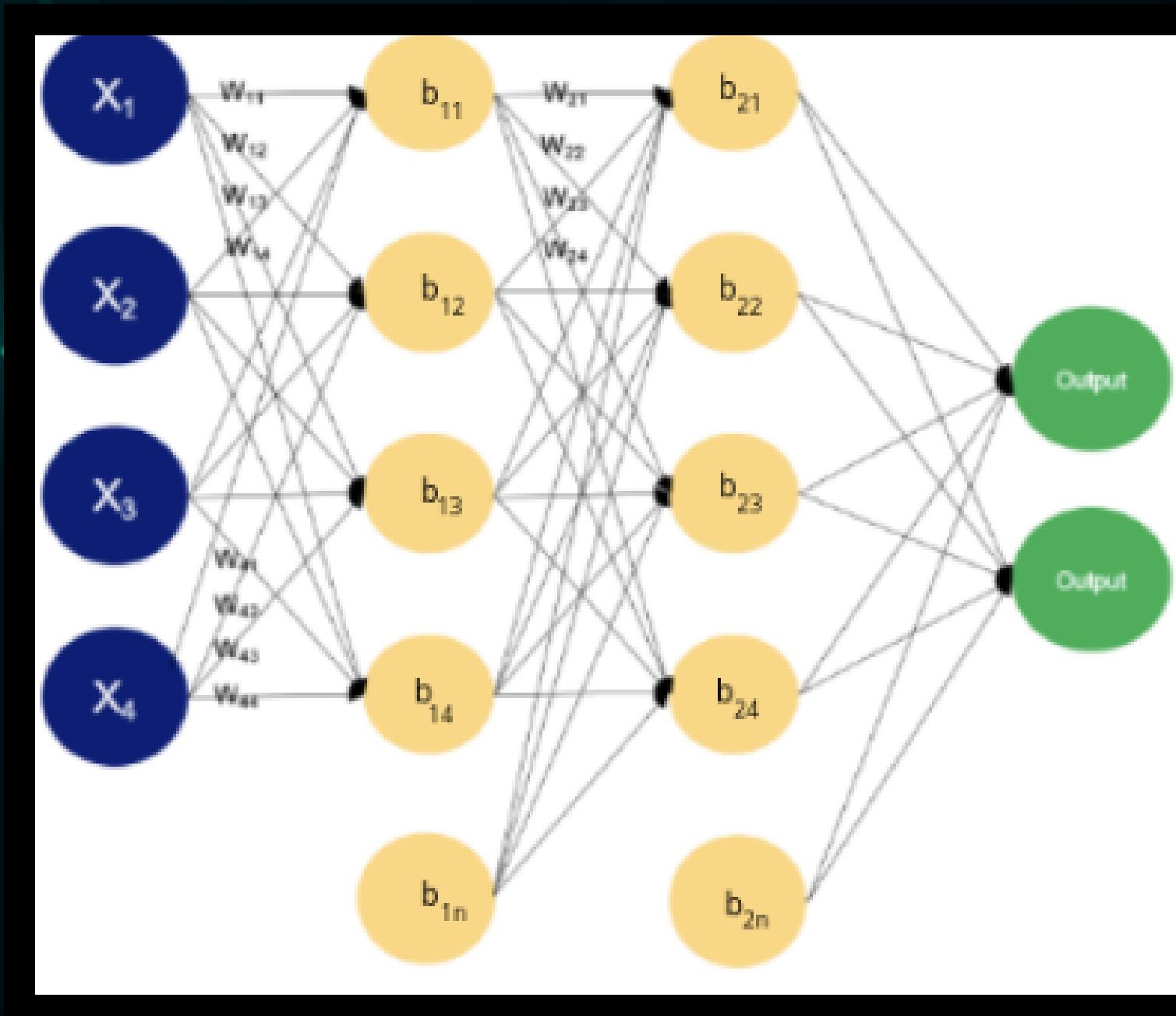


OBJETIVO PRINCIPAL

Determinar cuál es la red neuronal más eficiente para la tarea de reconocimiento de dígitos manuscritos.

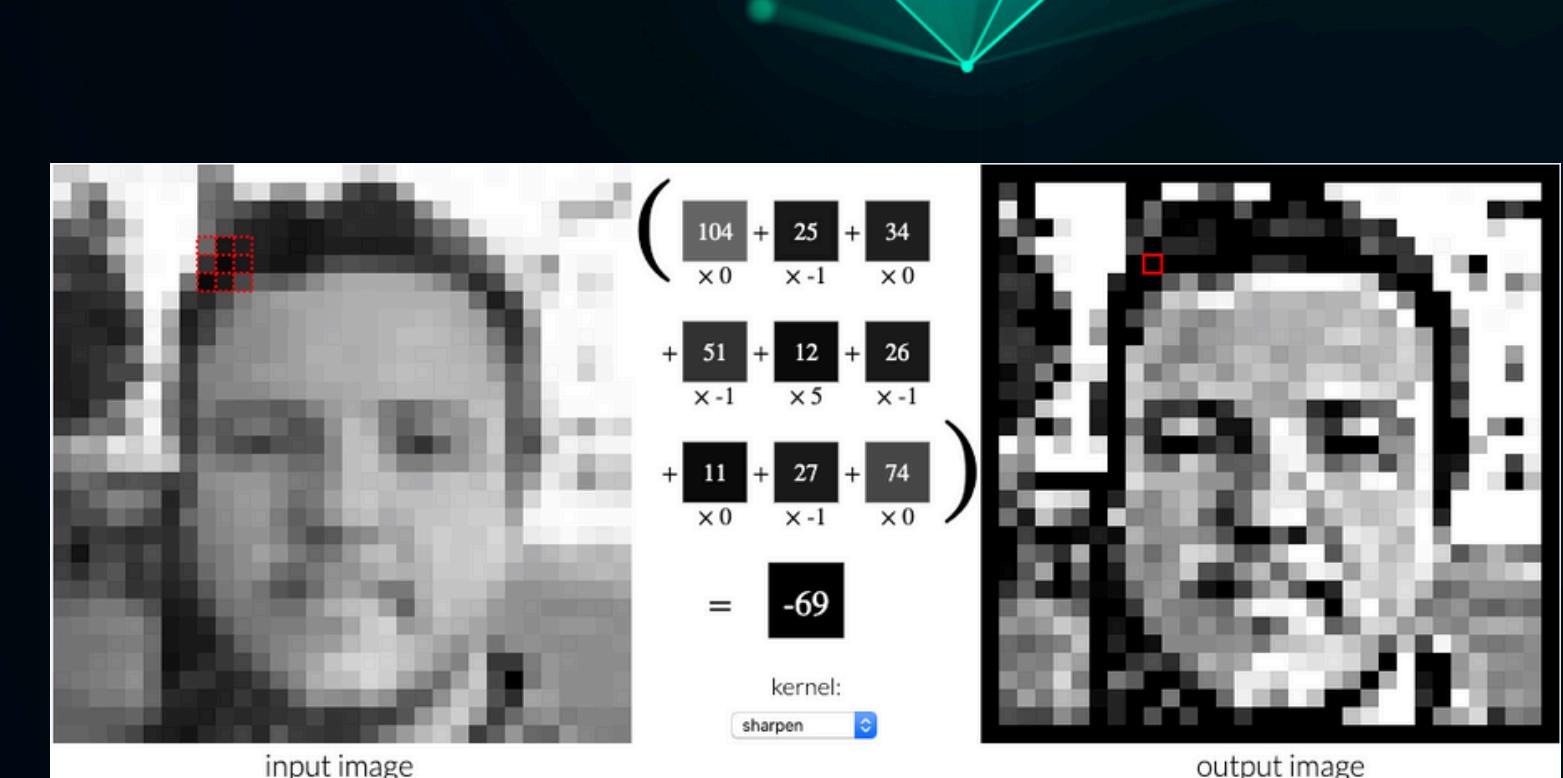
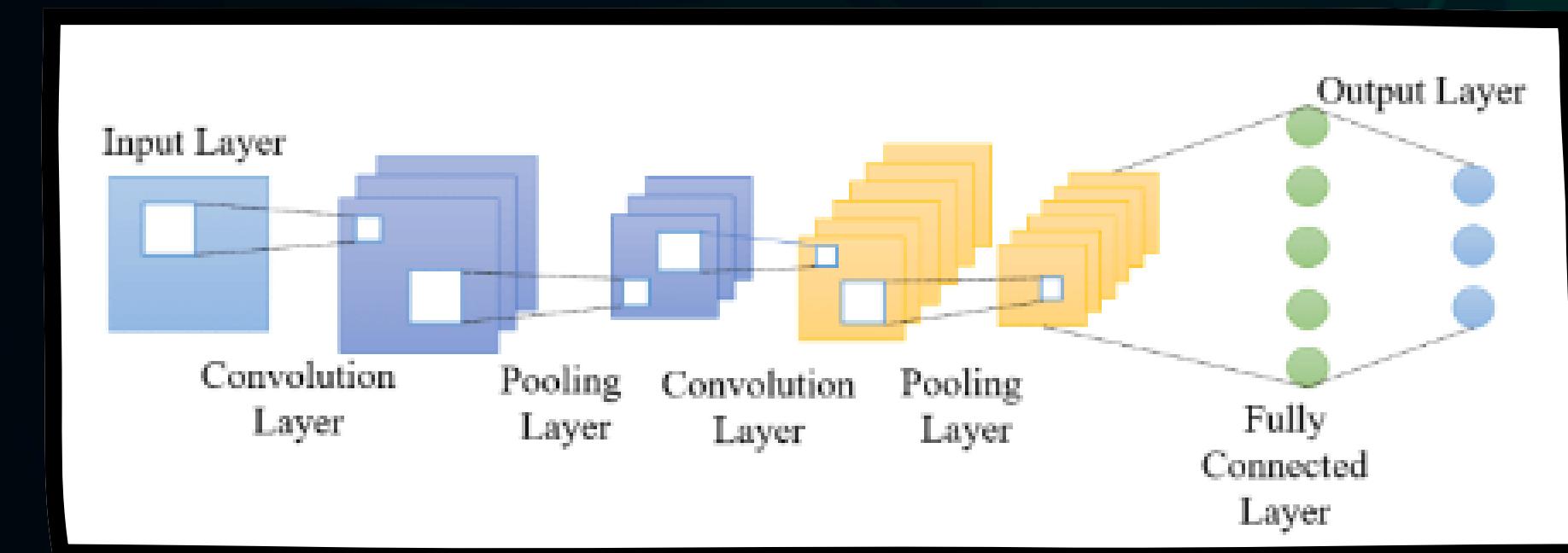
RED NEURONAL MULTICAPA (MLP)

Formada por tres tipos de capas: Input layer, hidden layer y output layer. Todas estas tienen forma de vector, donde cada neurona posee un bias (sesgo) y se encuentra conectada a todas las neuronas de la capa que le precede, cada una de estas conexiones tiene asignado un weight (peso).



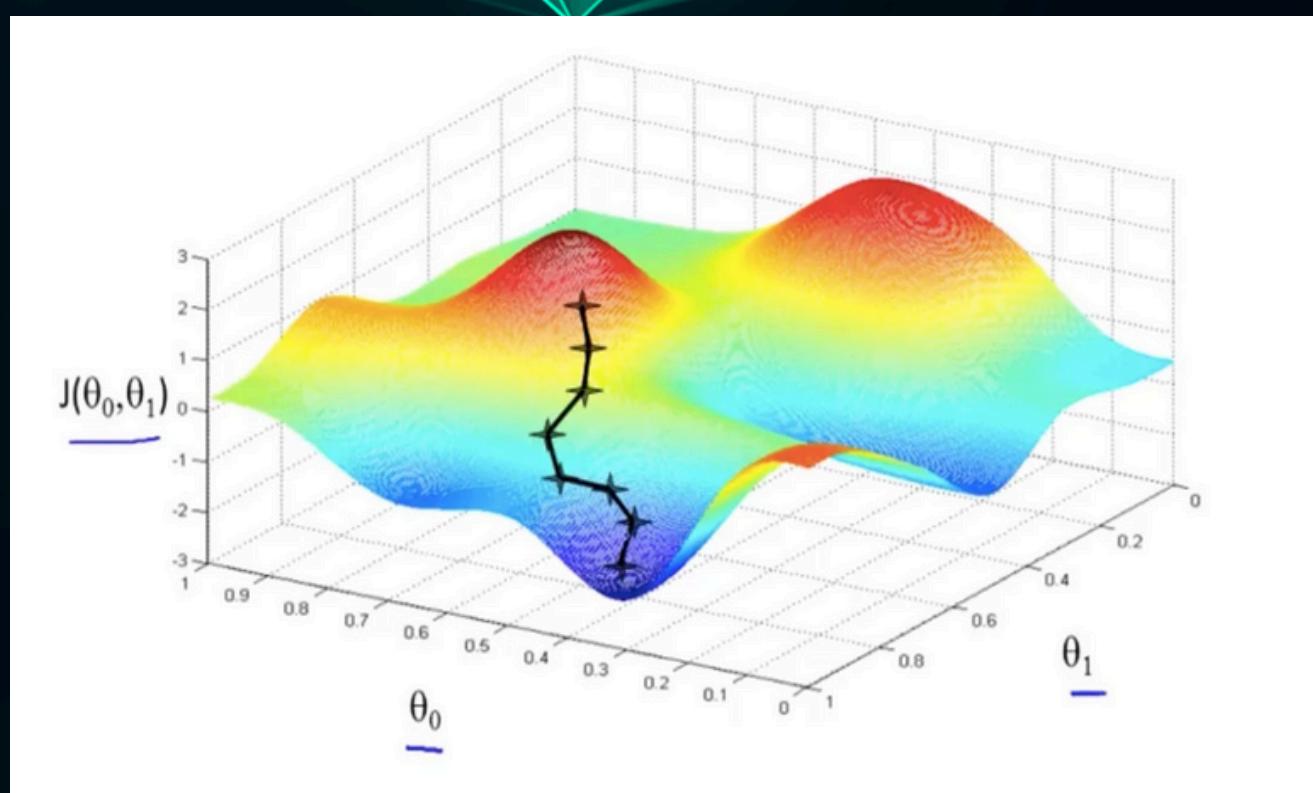
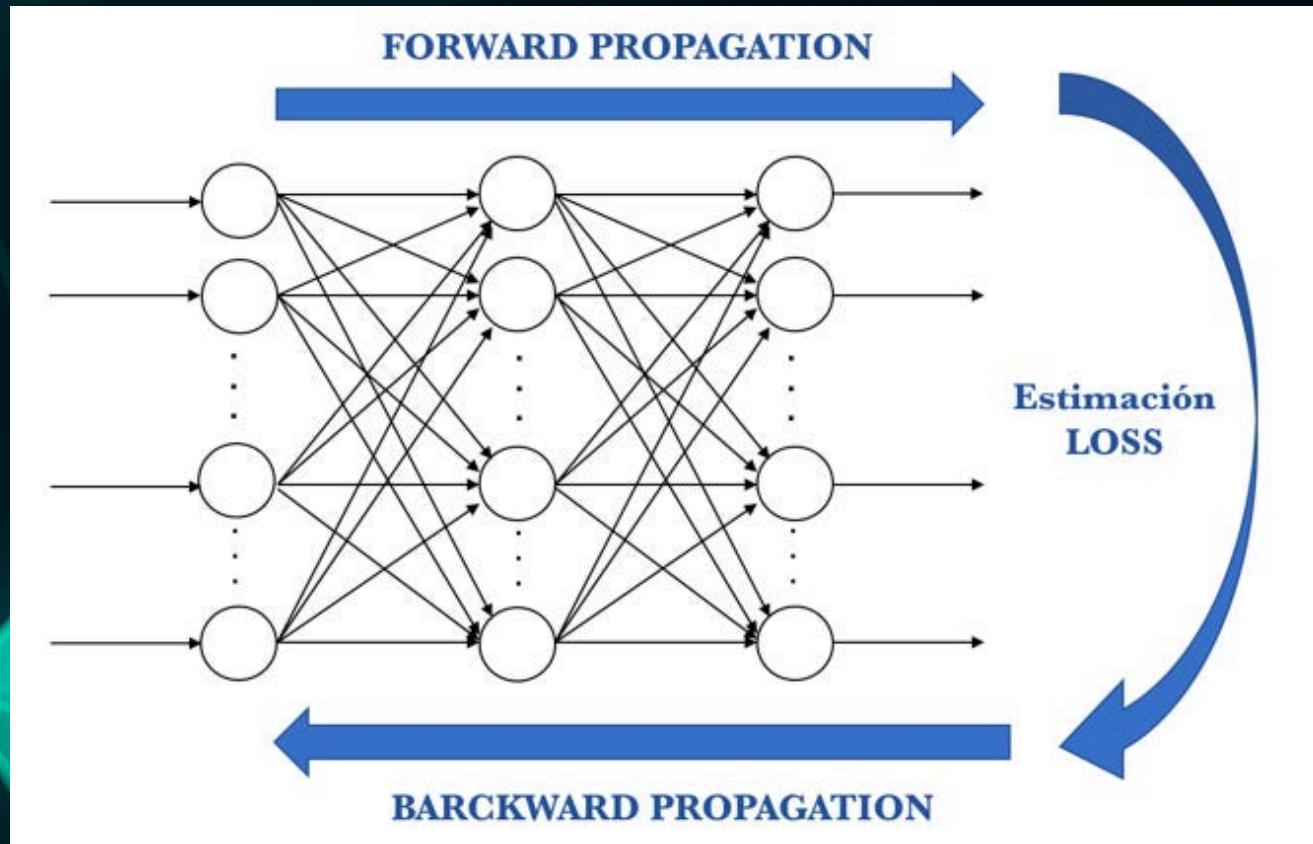
RED NEURONAL CONVOLUCIONAL (CNN)

Formada por tres tipos de capas: Input layer, hidden layer y output layer. Todas estas capas tienen forma de matriz (aunque exactamente no es así). Las capas de convolución y pooling realizan dichas operaciones a través de kernels (filtros). Finalmente, la entrada procesada se aplana (forma de vector) para pasar por capas unidimensionales y, con ello, generar una salida.



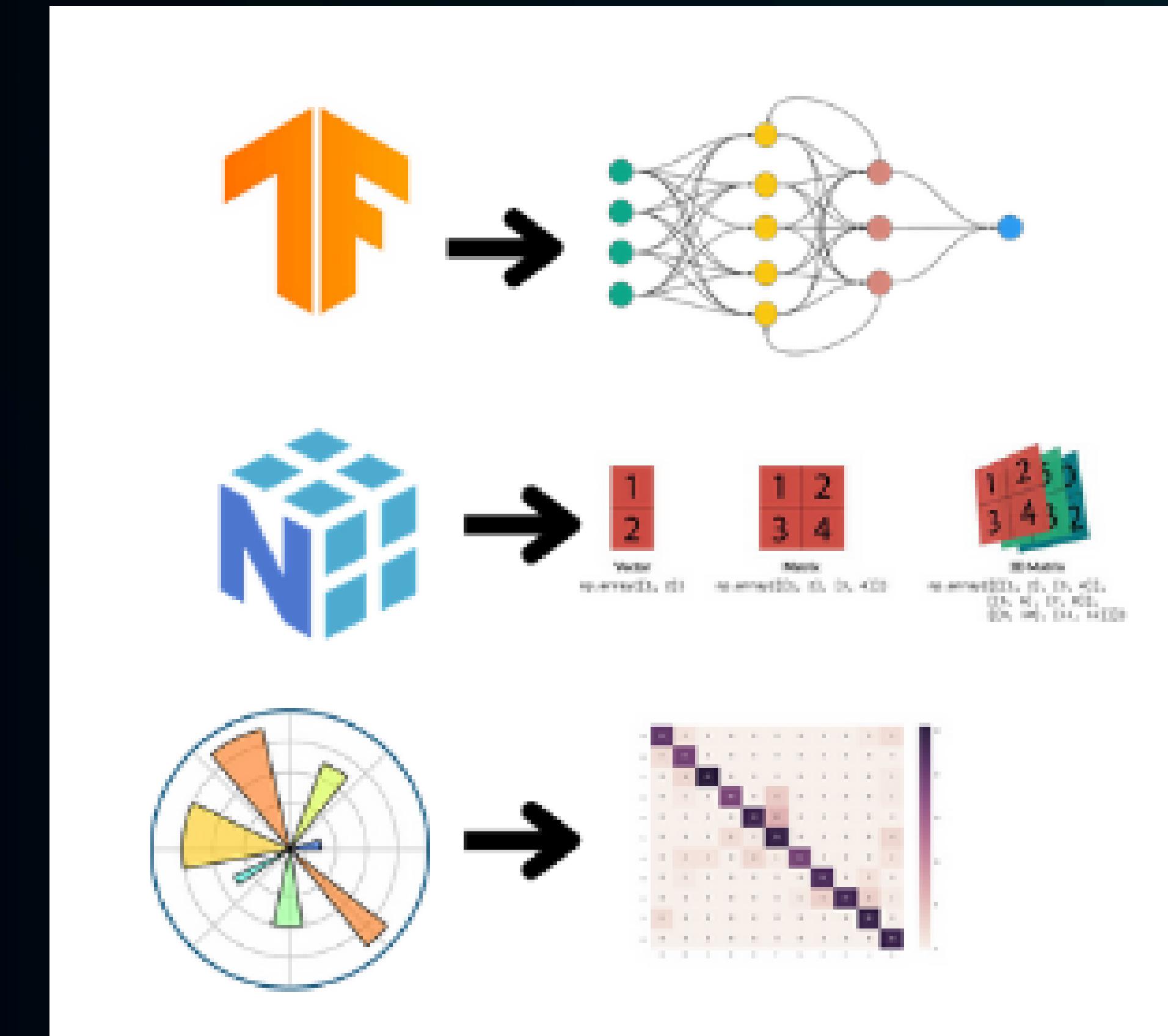
ENTRENAMIENTO DE REDES NEURONALES

1. Inicialización de parámetros (weight y bias)
2. Propagación hacia adelante (Forward Propagation)
3. Cálculo de la función de costo
4. Propagación hacia atrás (Backpropagation, cálculo de gradientes)
5. Actualización de parámetros (con un optimizador, gradientes y learning rate)
6. Repetir
7. Evaluación



MÓDULOS PRINCIPALES

- TensorFlow
- Numpy
- Matplotlib





MNIST

El dataset utilizado para el proyecto fue MNIST. Este contiene 70 mil imágenes (60 mil para el entrenamiento y 10 mil para el testeо). Cada una de 28x28 pixeles y respectivamente etiquetadas con el dígito al que corresponden
(Aprendizaje supervisado)

ARQUITECTURA DE LA MLP

```
mlp_model = Sequential([
    Flatten(input_shape=(784, )),
    Dense(256, activation=relu),
    Dense(128, activation=relu),
    Dense(64, activation=relu),
    Dense(10, activation=softmax)
])
```

ARQUITECTURA DE LA CNN

```
● ● ●  
cnn_model = Sequential([  
    Conv2D(32, (3, 3), activation=relu, input_shape=(28, 28, 1)),  
    MaxPooling2D((2, 2)),  
    Conv2D(64, (3, 3), activation=relu),  
    MaxPooling2D((2, 2)),  
    Flatten(),  
    Dense(128, activation=relu),  
    Dense(10, activation=softmax)  
])
```

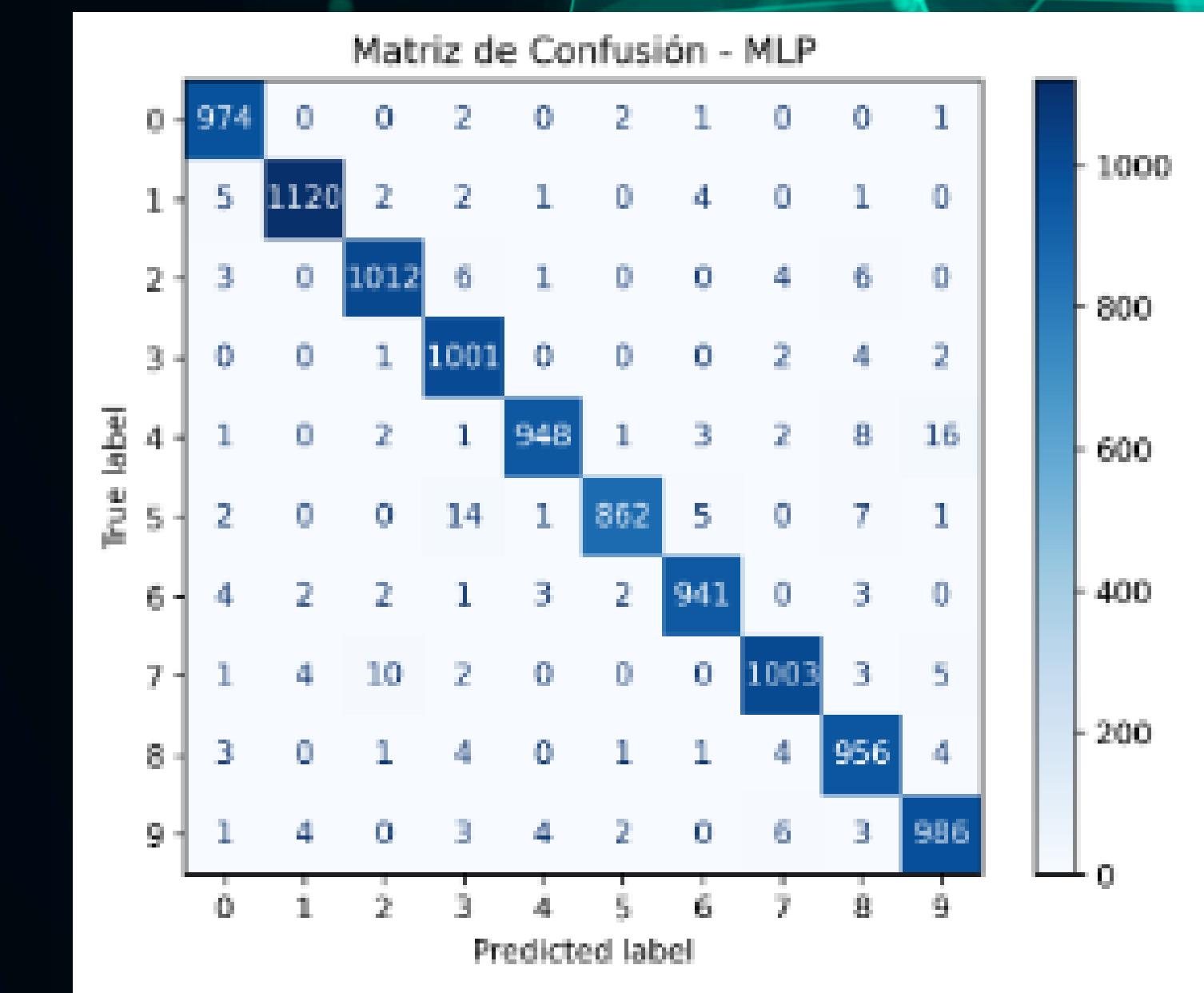
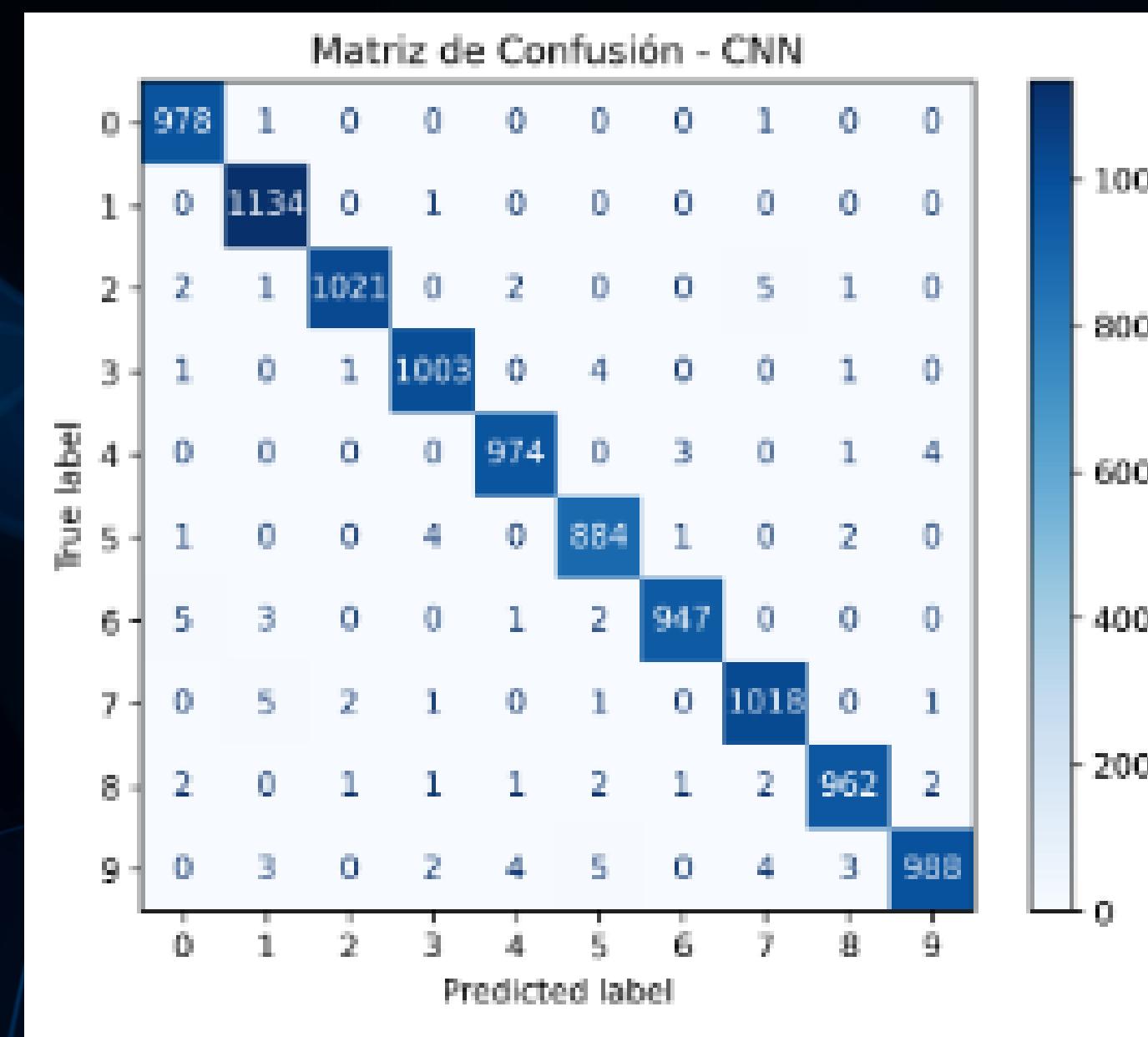
PARÁMETROS DEL ENTRENAMIENTO

```
# Configuracion del entrenamiento de los modelos
training:
    epochs: 10
    batch_size: 32
    learning_rate: 0.001
    optimizer: 'adam'
    loss: 'categorical_crossentropy'
    metrics: ['accuracy']

# Funciones de activacion
activation_functions:
    - relu: 'relu'
    - softmax: 'softmax'
```

RESULTADOS

La precisión alcanzada por la MLP fue de 99.21%, mientras que la de la CNN fue de 99.85%



IMPLEMENTACIÓN DE LA PÁGINA WEB

Predicción de Dígitos MNIST



Predecir

Limpiar

Predicción CNN: 3, Predicción MLP: 3

CONCLUSIONES

- La red neuronal convolucional resulta ser más precisa que la red neuronal multicapa para el reconocimiento de dígitos manuscritos.
- El preprocessamiento de los datos de entrada resulta muy importante para poder realizar un buen entrenamiento y una predicción adecuada.

PAM - IA

GRACIAS

