

Informe MLP - Página 1

Proyecto: MLP desde cero + Intérprete -> Keras

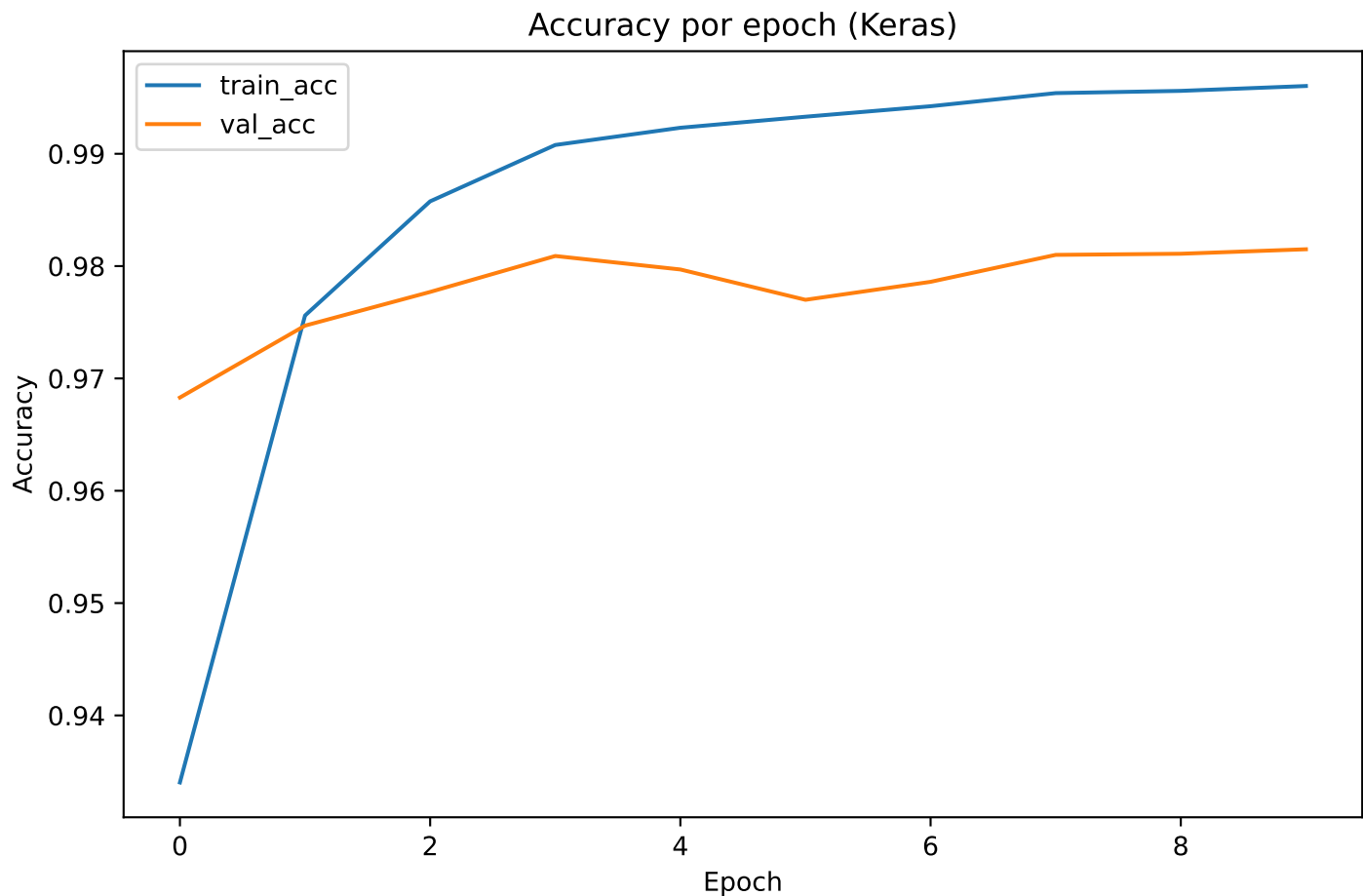
Arquitectura usada (Keras):

Dense(784,relu) -> Dense(128,relu) -> Dense(10,softmax)

Resultados (Keras): test_accuracy = 0.9815, test_loss = 0.0834

Notas:

- Parte 1: Implementación MLP con NumPy (forward y backprop simple).
- Parte 2: Intérprete `compile_model(architecture_string)` que construye un `tf.keras.Sequential`.
- Parte 3: Entrenamiento y evaluación en MNIST con Keras.



Análisis y conclusiones

1) Desafíos de implementar backprop a mano para una red grande:

- Complejidad algorítmica y bugs numéricos (overflow/underflow).
- Coste computacional: muchas operaciones de matriz y uso intensivo de memoria.
- Debugging y optimizaciones (initialization, learning rate schedules, regularización).

2) Ventajas de un 'compilador' (compile model):

- Abstracción: permite definir arquitecturas de forma legible y reproducible.
- Reutilización: cambia la arquitectura sin tocar el código de entrenamiento.
- Integración con frameworks: traduce a objetos optimizados y acelerados (e.g., Keras/TensorFlow).

3) Recomendaciones:

- Para mejor rendimiento usa CNNs (convolucional) en MNIST.
- Añadir regularización (Dropout, BatchNorm) y aumentar epochs.

