

**Instituto Tecnológico y de Estudios Superiores de Monterrey  
GDL Campus**



**Tecnológico  
de Monterrey**

**Inteligencia artificial avanzada para la ciencia de datos II (Gpo 501)**

Clave: TC3007C

Arah Rojas Blanco (A00834299)

**Profesor:**

Prof. Obed Noé Sámano Abonce

**Campus:** Guadalajara

16 nov 2025

## **Modelo**

El modelo que se empleo es una LSTM (Long Short-Term Memory), un tipo de red neuronal recurrente especializada en secuencias de datos, una LSTM analiza los frames uno a uno recordando los pasos anteriores, lo que le permite aprender dependencias temporales en la secuencia de movimientos.

En cuanto a la arquitectura, se diseñó una red con dos capas LSTM en secuencia más una capa densa final.

Para ayudar al modelo a generalizar mejor, se incorporó también capas de Dropout entre las LSTM y antes de la capa de salida. El dropout consiste en desactivar aleatoriamente una fracción de neuronas en cada paso de entrenamiento, lo que previene que la red se sobreajuste (memorice en exceso) al obligarla a no depender de neuronas específicas. En este caso se utilizó un dropout del 50% después de cada LSTM, es decir, la mitad de las salidas de cada LSTM se anulan aleatoriamente en cada iteración de entrenamiento.

## **Entrenamiento y mejoras**

Se entrenó el modelo en Google Colab usando TensorFlow/Keras. Para la compilación, se utilizó una función de pérdida apropiada de clasificación (cross-entropy categórica) y el optimizador Adam con una tasa de aprendizaje inicial de 0.001. Configuramos un número máximo de 50 épocas de entrenamiento, pero con una técnica de early stopping para no sobre-entrenar innecesariamente. El early stopping (detección temprana) monitorea la pérdida o precisión en el conjunto de validación durante el entrenamiento, y si el modelo deja de mejorar después de cierto número de épocas consecutivas, detiene el entrenamiento automáticamente. En este caso se definió patience = 5, es decir, si en 5 épocas seguidas la pérdida de validación no mejora, se para el entrenamiento y se restauran los pesos de la mejor época. Gracias a esto, evitamos el sobreajuste y también se ahorra tiempo de cómputo: aunque habíamos fijado 50 épocas, el entrenamiento se detuvo antes (alrededor de la época 15) porque el desempeño en validación dejó de aumentar significativamente.

A lo largo del entrenamiento se fue observando que la precisión en entrenamiento subía gradualmente y la precisión en validación alcanzó un máximo de ~62%.

## **Resultados**

Después de entrenar el modelo LSTM, evaluamos su desempeño tanto en el conjunto de validación como en el conjunto de prueba (test). En validación el modelo logró alrededor de 62% de precisión, y luego al evaluar en el conjunto de prueba alcanzó una precisión final de 63.37%. Esto confirma que el modelo generaliza bien a datos no vistos, ya que obtiene resultados similares en validación y prueba.

