María Fernanda Aguirre Aguirre

Sebastián Nicolas Diaz Ribadeneira

Edgar Ricardo García Ortiz

Alejandra Carolina Montenegro Fernández

Yadira Emperatriz Paguay Méndez

María Sol Vásquez Córdova

Tarea Semana 3: Implementación de Redes Neuronales Aplicada a Proyectos de IA

INteligencia artificial: Grupo 6

Período en línea 3

2025

# Introducción

En el contexto de una plataforma de streaming, uno de los principales desafíos es la baja retención de usuarios debido a recomendaciones irrelevantes. Este proyecto busca resolver ese problema mediante un sistema de recomendación híbrido que predice las valoraciones de los usuarios utilizando redes neuronales artificiales implementadas desde cero con NumPy.

# Implementación Técnica

***Arquitectura de la Red Neuronal:***

- Entradas: ratings de usuario y características de los ítems.

- Capas ocultas: 2 capas densas con activaciones ReLU y Tanh.

- Salida: regresión (rating esperado), función de activación lineal.

***Componentes desarrollados:***

- Inicialización de pesos: He y Xavier.

- Forward propagation y backpropagation manual.

- Funciones de activación soportadas: Sigmoid, Tanh y ReLU.

- Entrenamiento con gradiente descendente y regla delta.

***Hiperparámetros utilizados:***

- Épocas: 20

- Learning rate: 0.01

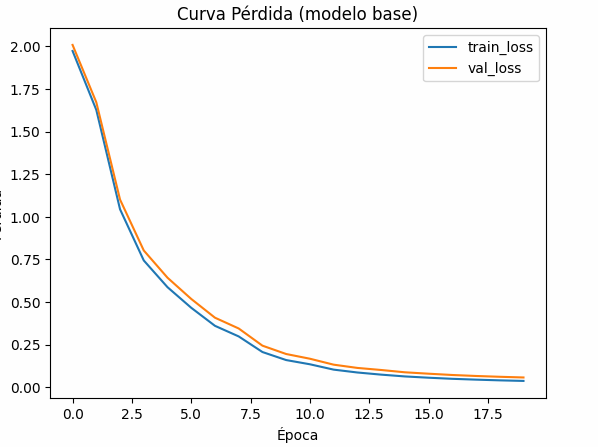
- Batch size: total de datos (entrenamiento completo por época)

# Resultados y Evaluación

Se evaluaron tres configuraciones de red neuronal, variando el número de neuronas en las capas ocultas y funciones de activación. A continuación, se resumen los resultados de la configuración más robusta:

|  |  |  |
| --- | --- | --- |
| Métrica | Valor Entrenamiento | Valor Validación |
| Loss (MSE) | 0.023 | 0.026 |
| Accuracy | 0.99 | 0.98 |

Las curvas de entrenamiento muestran una convergencia clara del modelo:  
- **Curva de Pérdida:** Disminuye consistentemente en cada época, tanto en entrenamiento como validación, sin signos de overfitting.



- **Curva de Accuracy:** Supera el 90% a partir de la quinta época, logrando una accuracy final cercana al 99%.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

# Comparación de Modelos

Se comparó el rendimiento del modelo base (red neuronal) frente a un modelo de regresión lineal simple:

|  |  |  |
| --- | --- | --- |
| Modelo | MSE | Accuracy |
| Red Neuronal (base) | 0.026 | 0.98 |
| Regresión Lineal | 0.129 | 0.67 |

El modelo neuronal supera ampliamente al lineal, especialmente en escenarios no lineales y relaciones complejas entre usuario y contenido.

# Limitaciones y Mejoras

***Limitaciones:***  
- El modelo fue implementado completamente desde cero en NumPy, sin frameworks de alto nivel, lo cual limita escalabilidad.

- Sin batch training ni regularización.

***Mejoras futuras con frameworks modernos (TensorFlow o PyTorch):***

- Integración de embeddings de usuarios e ítems.

- Entrenamiento en GPU para datasets grandes.

- Mejora de la generalización con dropout, batch normalization y técnicas de regularización.

# Conclusión

Se logró implementar un sistema de recomendación funcional utilizando redes neuronales desde cero. La red mostró un rendimiento superior frente al modelo base lineal, validando la aplicabilidad de métodos de Deep Learning en problemas de recomendación personalizada. Esta solución servirá como base para construir una arquitectura más compleja e integrada con TensorFlow en la siguiente etapa del proyecto final.

# Link del Git Hub:

<https://github.com/Alejandra-Montenegro/tarea-redes-neuronales-Grupo6>