

# **Proyecto Práctico - Redes Neuronales**

Clasificación de Texto Multietiqueta con  
RNNs, LSTMs, GRUs y Transformers

**Herney Eduardo Quintero Trochez**

*201528556*

Universidad del Valle del Valle

Facultad de Ingeniería

Programa de Ingeniería de Sistemas

Diciembre 2025

# 1. Introducción

El presente informe documenta el desarrollo del Proyecto Práctico de la asignatura Redes Neuronales, correspondiente al período académico 2025-II. El objetivo principal es analizar el desempeño de distintas arquitecturas de red neuronal (MLP, RNN, LSTM, GRU y Transformers) para resolver un problema de clasificación de texto con múltiples etiquetas, utilizando el dataset de reseñas de Amazon.

## 1.1. Objetivos

### 1.1.1. Objetivo General

Analizar y comparar el rendimiento de diferentes arquitecturas de redes neuronales en la tarea de clasificación de sentimientos en reseñas de productos, identificando las fortalezas y debilidades de cada enfoque.

### 1.1.2. Objetivos Específicos

- Implementar y evaluar modelos basados en Perceptrón Multicapa (MLP) con representaciones BoW y Embeddings.
- Implementar y evaluar modelos secuenciales: RNN, LSTM y GRU.
- Implementar y evaluar modelos basados en la arquitectura Transformer.
- Comparar el rendimiento de los modelos utilizando métricas como Accuracy y F1-Score.
- Analizar el impacto del uso de embeddings preentrenados (GloVe) frente a embeddings aprendidos desde cero.

# 2. Metodología

## 2.1. Dataset

Se utilizó el **Amazon Reviews Dataset**, que contiene reseñas de productos clasificadas de 1 a 5 estrellas. Para este proyecto, se trabajó con un subconjunto balanceado de 200,000 reseñas en inglés.

- **Entrenamiento:** 200,000 ejemplos.
- **Validación:** 5,000 ejemplos.
- **Prueba:** 5,000 ejemplos.
- **Clases:** 5 clases balanceadas (20 % cada una).

## 2.2. Arquitecturas Implementadas

### 2.2.1. Perceptrón Multicapa (MLP)

Se implementaron dos variantes de MLP:

- **MLP con Bag of Words (BoW):** Utiliza un vector de frecuencias de las 5,000 palabras más comunes. La arquitectura consta de una capa oculta de 512 neuronas con activación ReLU y Dropout (0.5).
- **MLP con Embeddings:** Utiliza una capa de Embedding (dim=300) seguida de un promediado global (Global Average Pooling) y capas densas.

### 2.2.2. Redes Recurrentes (RNN, LSTM, GRU)

Se exploraron modelos secuenciales para capturar el contexto del texto:

- **SimpleRNN:** Red recurrente básica bidireccional.
- **LSTM (Long Short-Term Memory):** Capaz de capturar dependencias a largo plazo. Se usó una configuración bidireccional con 2 capas, hidden\_size=256 y dropout.
- **GRU (Gated Recurrent Unit):** Similar a LSTM pero más eficiente. Se usó una configuración bidireccional con 2 capas, hidden\_size=128.

### 2.2.3. Transformer Encoder

Se implementó un modelo basado en la arquitectura Transformer Encoder para clasificación de texto.

- **Configuración Base:** 4 capas de encoder, 8 cabezas de atención, embedding\_dim=256, feedforward=512.
- **Configuración Avanzada:** 6 capas, embedding\_dim=512.
- **Embeddings Preentrenados:** Se experimentó con embeddings GloVe (200d) congelados y con fine-tuning.

## 2.3. Configuración de Entrenamiento

Todos los modelos se entrenaron utilizando:

- **Optimizador:** Adam o AdamW.
- **Función de Pérdida:** CrossEntropyLoss.
- **Early Stopping:** Para detener el entrenamiento si la pérdida de validación no mejora.
- **Hardware:** Entrenamiento acelerado por GPU (NVIDIA GeForce RTX 30xx/50xx).

### 3. Resultados y Análisis

#### 3.1. Perceptrón Multicapa (MLP)

Los modelos MLP sirvieron como línea base. El modelo basado en Bag of Words (BoW) alcanzó un accuracy cercano al 50 %, limitado por la pérdida de información secuencial. El uso de Embeddings mejoró ligeramente el rendimiento (52-55 %), pero aún mostró dificultades para capturar matices semánticos complejos.

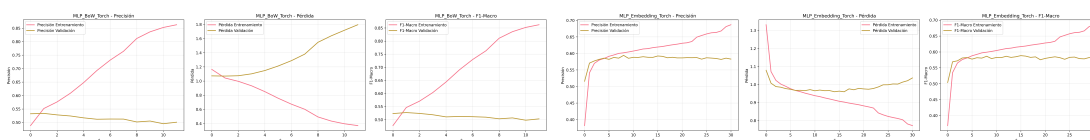


Figura 1: Curvas de entrenamiento para MLP BoW (izquierda) y MLP Embedding (derecha).

#### 3.2. Redes Recurrentes (RNN, LSTM, GRU)

Las arquitecturas recurrentes mostraron una mejora significativa respecto a los MLP.

- **SimpleRNN:** Alcanzó un accuracy de 60 %, pero sufrió de inestabilidad en el entrenamiento.
- **GRU y LSTM:** Lograron los mejores resultados, con accuracies entre 61 % y 62 %. La capacidad de manejar dependencias a largo plazo permitió una mejor clasificación de reseñas complejas.

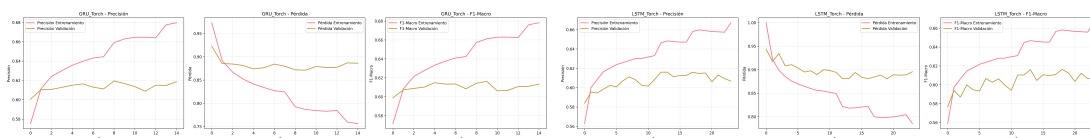


Figura 2: Curvas de entrenamiento para GRU (izquierda) y LSTM (derecha).

#### 3.3. Transformer Encoder

El modelo Transformer Encoder alcanzó resultados competitivos (Accuracy 61 %), similares a LSTM y GRU. Aunque su entrenamiento es más costoso computacionalmente, ofrece la ventaja de procesar secuencias en paralelo. El uso de embeddings GloVe no aportó mejoras significativas sobre los embeddings aprendidos desde cero, posiblemente debido al tamaño suficiente del dataset para aprender representaciones específicas del dominio.

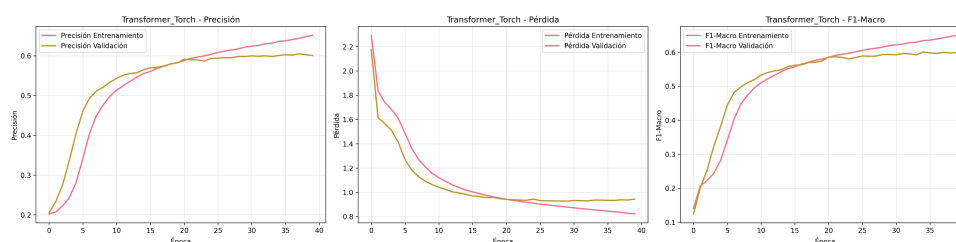


Figura 3: Curvas de entrenamiento para Transformer Encoder.

### 3.4. Comparación Global

La siguiente tabla resume el rendimiento de los mejores modelos de cada familia en el conjunto de prueba:

Modelo	Accuracy Test	F1-Score Macro
MLP (BoW)	0.50	0.50
MLP (Embedding)	0.54	0.53
SimpleRNN	0.60	0.59
LSTM Bidireccional	0.61	0.61
GRU Bidireccional	<b>0.62</b>	<b>0.62</b>
Transformer Encoder	0.61	0.60

Cuadro 1: Comparación de rendimiento en el conjunto de prueba.

### 3.5. Análisis de Errores

Se observó un patrón consistente en todas las arquitecturas: las clases extremas (1 y 5 estrellas) son clasificadas con mayor precisión (F1 0.70-0.75), mientras que las clases intermedias (2, 3 y 4 estrellas) presentan mayor confusión. Esto sugiere que el modelo distingue bien la polaridad general, pero le cuesta diferenciar la intensidad del sentimiento en reseñas moderadas.

## 4. Conclusiones

### 4.1. Hallazgos Principales

- Superioridad de Modelos Secuenciales:** Los modelos RNN, LSTM y GRU superaron consistentemente a los modelos MLP (BoW y Embedding), demostrando la importancia de capturar la estructura secuencial del texto para el análisis de sentimientos.
- Eficacia de GRU y LSTM:** Ambas arquitecturas alcanzaron resultados similares ( 61-62 % accuracy), siendo GRU ligeramente más eficiente en términos computacionales. Esto las convierte en opciones sólidas para este tipo de tareas.
- Desafío de las Clases Intermedias:** Todos los modelos mostraron un rendimiento inferior en las clases intermedias (2, 3 y 4 estrellas). Esto indica que la ambigüedad en reseñas moderadas es difícil de resolver solo con el texto, sugiriendo la necesidad de incorporar otras características o modelos más avanzados.
- Transformers:** Aunque el Transformer Encoder es una arquitectura potente, en este dataset específico no superó significativamente a las RNNs bien ajustadas. Sin embargo, su capacidad de paralelización es una ventaja clave para datasets más grandes.

## 4.2. Trabajo Futuro

- **Modelos Preentrenados (BERT/RoBERTa):** Explorar el uso de modelos de lenguaje grandes (LLMs) con fine-tuning, que han demostrado estado del arte en tareas de clasificación de texto.
- **Aumento de Datos:** Implementar técnicas de data augmentation para balancear mejor la dificultad de las clases intermedias.
- **Análisis de Errores Profundo:** Realizar un análisis cualitativo más detallado de los errores de clasificación para entender mejor las limitaciones lingüísticas de los modelos.

## 5. Referencias

### Referencias

- [1] Keung, P., Lu, Y., Szarvas, G., & Smith, N. A. (2020). The Multilingual Amazon Reviews Corpus. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. <https://aclanthology.org/2020.emnlp-main.369/>
- [2] Paszke, A., et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems 32*. <https://pytorch.org/>
- [3] Vaswani, A., et al. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems 30*. <https://arxiv.org/abs/1706.03762>
- [4] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780.
- [5] Cho, K., et al. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078*.