

Hipótesis de Solución para el Análisis de Sentimientos en Comentarios de Películas

Modulo 2 – Unidad 2: Redes Neuronales Recurrentes (RNN)

Grupo 5

Javier Pinto Losada

Hector Tovar Jaimes

Juan José Restrepo

1. Introducción:

El análisis automático de sentimientos en comentarios cinematográficos plantea el desafío de interpretar lenguaje natural—con variaciones de estilo, ironía y ambigüedad—para clasificar cada texto como positivo o negativo. A lo largo de este curso hemos estudiado las redes neuronales recurrentes (RNN) y sus variantes (LSTM, GRU), que resultan idóneas para modelar dependencias secuenciales en datos textuales.

2. Objetivo:

Formular una solución basada en modelos secuenciales aprendidos durante la unidad, para desarrollar un clasificador de sentimientos que:

1. Aproveche las capacidades de RNN, LSTM y GRU para capturar contexto y dependencias a largo plazo.
2. Sea compatible con el dataset de IMDB (comentarios de películas).
3. Genere una salida binaria clara (positivo vs. negativo).

3. Requerimientos de los Datos

1. Texto de Comentario:

- Cada muestra es un párrafo u oración, codificado en UTF-8.

2. Etiqueta Binaria:

- **0:** comentario negativo,
- **1:** comentario positivo.

3. Preprocesamiento:

- Tokenización a nivel de palabra.
- Padding/Truncation a longitud fija (p. ej. 200 tokens).

- Vocabulario limitado (p. ej. 20 000 palabras más frecuentes).

4. Balance de Clases:

- Idealmente un número similar de ejemplos por categoría para evitar sesgos.

4. Definición de Entradas y Salidas

Componente	Descripción
Entrada	Secuencia de enteros (tokens), longitud fija tras padding.
Salida	Valor escalar en [0, 1] (sigmoid) o vector de dos probabilidades (softmax): sentimiento final.

5. Arquitectura Propuesta

Se adoptará un modelo secuencial en Keras con la siguiente estructura:

1. Capa de Embedding

- Tamaño de vocabulario: 20.000
- Dimensión de embedding: 128

2. Capa Recurrente Bidireccional

- Variante: RNN / LSTM / GRU (64 unidades)
- Permite capturar contexto anterior y posterior.

3. Capa de Dropout

- Tasa: 0.3 (regularización).

4. Capa Dense de Salida

- Activación sigmoid (1 neurona) o softmax (2 neuronas).

6. Justificación Técnica

- **RNN Vanilla:** modelo base, captura secuencia pero sufre con dependencias largas.
- **LSTM:** supera el desvanecimiento de gradientes gracias a puertas de entrada, olvido y salida.
- **GRU:** similar a LSTM, con menor complejidad computacional (2 puertas).
- **Bidireccionalidad:** contextualiza cada token atendiendo al pasado y al futuro inmediato, crucial en lenguaje natural.
- **Embedding:** vectoriza palabras en un espacio semántico, favoreciendo la capacidad de generalización.
- **Dropout:** mitiga sobreajuste al apagar aleatoriamente neuronas durante el entrenamiento.
- **Adam:** combina momentum y adaptación de tasa de aprendizaje, ideal para tareas secuenciales con ruido en gradientes.

7. Conclusión

Mediante el uso de redes neuronales recurrentes y sus variantes LSTM/GRU, junto con técnicas de embedding y regularización, es factible construir un clasificador de sentimientos robusto y eficiente. Este enfoque aprovecha la teoría y prácticas vistas en el curso, garantizando una solución alineada con los contenidos académicos y con sólida fundamentación técnica.

Referencias

1. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2ª ed., O'Reilly, 2022.
2. S. Hochreiter & J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, 1997.
3. K. Cho et al., "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," *EMNLP*, 2014.
4. D. P. Kingma & J. Ba, "Adam: A Method for Stochastic Optimization," *ICLR*, 2015.
5. TensorFlow Datasets, *IMDB Reviews*,
https://www.tensorflow.org/datasets/catalog/imdb_reviews.

