Clase 140 — Análisis de sentimiento

Parte: 2 — Deep Learning · Fuente: Géron, cap. 16 § Sentiment Analysis. Duración estimada: 65 min.

Clase 140 — Análisis de sentimiento

> *Parte: 2 — Deep Learning · Fuente: Géron, cap. 16 § Sentiment Analysis. Duración estimada: 65 min.*

Objetivo

Aplicar un modelo de clasificación de texto sobre IMDB reviews — la tarea NLP más clásica para benchmarks. Pipeline completo: TextVectorization → Embedding → arquitectura (Dense / CNN / RNN / Transformer) → Dense(1, sigmoid). Comparar el zoo de approaches y reconocer que con Hugging Face hoy se hace en 3 líneas (clase 127).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Tokenizar y vectorizar texto con TextVectorization(max_tokens=20_000, output_sequence_length=200).
- Aplicar Embedding(vocab_size, dim) y entender que es una lookup table aprendible.
- Construir 4 arquitecturas: bag-of-embeddings (sin orden), Conv1D, LSTM, Bidirectional LSTM.
- Comparar accuracy de las 4 vs un baseline TfidfVectorizer + LogisticRegression.
- Usar Embedding(..., mask_zero=True) para manejar padding correctamente.

Temas

- TextVectorization moderno (Keras 3+).
- Embedding: lookup table inicializada random y entrenable.
- BagOfEmbeddings (mean pooling) como baseline DL.
- Conv1D para texto: capta n-gramas.
- LSTM + Bidirectional: captura contexto largo y bidireccional.
- Pre-trained embeddings (GloVe, Word2Vec) — históricamente importantes; hoy reemplazados por embeddings de transformers.

Definiciones y características

- Embedding: matriz (vocab_size, embed_dim). La fila i es la representación aprendida del token i.
- Pooling: tras Embedding tenés (batch, T, dim); pooling reduce a (batch, dim). Mean, max, attention.
- Masking: ignorar posiciones con padding (0 por convención).
- Bidirectional: corre LSTM forward + backward y concatena.

Dataset / recursos

- keras.datasets.imdb.load_data() o tfds.load('imdb_reviews').
- Librerías: tensorflow, keras.

Ejercicios

1. Baseline ML clásico: TfidfVectorizer + LogisticRegression. Accuracy de referencia (~0.88).

2. Bag-of-embeddings: Embedding → GlobalAveragePooling1D → Dense(1, sigmoid). Reportar accuracy.
3. Conv1D: Embedding → Conv1D(64, 5) → GlobalMaxPool1D → Dense(1, sigmoid).
4. Bidirectional LSTM: Embedding → Bidirectional(LSTM(64)) → Dense(1, sigmoid).
5. Pre-trained: cargar GloVe 100d y inicializar la matriz de Embedding con ellos. Comparar accuracy contra inicialización random.

Homework verificable

IMDB sentiment classifier con 3 arquitecturas:

1. Pipeline TextVectorization(20_000, 200).
2. Tres modelos: Conv1D, Bidirectional LSTM, BagOfEmbeddings.
3. Reportar accuracy en test para cada uno.
4. Comparar contra TFIDF + LogReg.

Criterio de aceptación: al menos uno de los modelos DL debe ≥ 0.88. Bidirectional LSTM suele ganar pero por margen pequeño vs Conv1D.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
output_sequence_length muy chico	Trunca reviews largos. Fix: 200-500 para r
Olvido mask_zero=True en Embedding	LSTM/GRU aún ven el padding como token vál
Sin pooling antes del Dense → shapes incom	Fix: GlobalAveragePooling1D().
GloVe sin escalado de dimensiones	Embeddings vienen pre-trained con norma es
Reviews en otros idiomas con tokenizer ent	Mal performance. Fix: usar un tokenizer mu

Preguntas frecuentes

¿Sentimiento en 2026: TextVectorization o HF?

Para producción serio: HuggingFace + distilbert-base-uncased-finetuned-sst-2-english (94 % accuracy, 5 líneas de código). Esta clase es pedagógica.

¿Embeddings pre-trained todavía importan?

Para tokens "raw" no — todo modelo moderno tiene su propio embedding integrado. Para visualizar relaciones semánticas (analogías), sí.

¿Cuántos tokens y cuánta longitud?

IMDB: max_tokens=20_000 cubre vocab; output_sequence_length=200 cubre el 90 % de reviews.

¿Bidirectional siempre mejor?

Para clasificación (non-causal), sí. Para generación o real-time, NO (necesitarías ver el futuro).

¿Sentimiento multi-clase (estrellas 1-5)?

Cambiar última capa a Dense(5, softmax) + sparse_categorical_crossentropy. Para sentiment ordinal, considerar ordinal regression.

Referencias

- Géron, cap. 16 — Sentiment Analysis.
- Maas et al. (2011), Learning Word Vectors for Sentiment Analysis — IMDB dataset.
- Pennington et al. (2014), GloVe, EMNLP.

Siguiente clase

Clase 141 — Encoder-Decoder para traducción

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb