

Máster Universitario en Ciencia de Datos, Máster Universitario en Ingeniería Informática, y
Máster Universitario en Sistemas Interactivos Inteligentes

Procesamiento del Lenguaje Natural, y Aprendizaje Profundo para el Procesamiento del Lenguaje Natural 2024-25

Práctica de laboratorio 3: Análisis de sentimientos mediante redes neuronales

Entrega: 7 de marzo de 2025

Modelo de trabajo: la práctica se realizará y entregará por parejas

En esta práctica haremos una serie de ejercicios de programación con la librería TensorFlow para construir y evaluar varios modelos neuronales de aprendizaje profundo en una tarea de análisis de sentimientos en textos. En particular, exploraremos diferentes arquitecturas de redes neuronales feedforward (FFN) y recurrentes con unidades long short-term memory (LSTM), sin usar y usando word embeddings pre-entrenados¹ (Glove), sobre el corpus Stanford Sentiment Treebank² (SST).

El código Python solicitado ha de completar el proporcionado en el cuaderno Jupyter `nlp2425-lab3.ipynb`.

Debido a los requisitos de cómputo exigentes de alguno de los ejercicios de la práctica, se sugiere implementar y ejecutar el código en la plataforma Google Colab³, en cuyo entorno de ejecución se puede solicitar el uso de GPU y TPU en la nube.

Recursos:

- Ficheros con los conjuntos de entrenamiento, validación y test del corpus SST: `sst_training.txt`, `sst_validation.txt` y `sst_test.txt` (2.86 MB).
- Fichero comprimido con los word embeddings Glove: `glove.6B.50d.txt.bz2` (53.1 MB).

Tareas:

1. **Redes neuronales feedforward. [3 puntos]** En este ejercicio, construiremos y evaluaremos varios modelos FFN para la clasificación de sentimientos de los textos del corpus SST. Específicamente, se pide:
 - a) Con la función `plot` de la librería `matplotlib.pyplot`, visualizar la evolución del loss y el accuracy de entrenamiento y validación de la FFN de una capa oculta creada y entrenada en `nlp_lab2.ipynb`. Los valores de estas métricas para cada época de entrenamiento están disponibles en la variable `history`. Explicar brevemente las gráficas generadas. **[0,5 puntos]**
 - b) Añadir a la FFN del ejercicio 1a) una capa oculta adicional con 16 unidades. Entrenar la red durante 100 épocas y visualizar la evolución del loss y el accuracy de la nueva FFN creada. Para ello, emplear una variable `history_2`. Además, generar otra gráfica en la que se comparen las curvas de loss de entrenamiento y validación para las dos redes: la de una capa y la de dos capas. Para ello, emplear las variables `history` y `history_2`. Explicar brevemente esta última gráfica. **[0,5 puntos]**

¹ <https://nlp.stanford.edu/projects/glove>

² <https://nlp.stanford.edu/sentiment>

³ <https://colab.research.google.com>

- c) Explorar estrategias para evitar el sobreajuste de la red de 2 capas ocultas: early stopping, regularización de pesos (L2) y dropout (0.5). Análogamente a los apartados anteriores, visualizar la evolución del loss de entrenamiento y validación para la construcción de la red con y sin el uso de cada una de las estrategias anteriores. Para ello, elaborar 3 gráficas comparativas a través de variables `history` y `history_2`. Explicar brevemente las gráficas obtenidas. **[0,5 puntos]**
- d) Realizar una evaluación de una red neuronal profunda con tres capas ocultas de 512, 256 y 128 unidades, mediante grid search de parámetros: learning rate (p.ej., 10^{-4} , 10^{-3} , 10^{-2}), grados de dropout (p.ej., 0, 0.2, 0.5) y factores de regularización L2 (p.ej., 10^{-4}). Obtener una tabla que muestre el mejor valor de accuracy en test alcanzado por cada configuración de parámetros evaluada. En la tabla, indicar además la época de entrenamiento en la que se alcanzó el valor óptimo de accuracy en test para cada configuración de parámetros. Se recomienda la implementación de una función que genere un modelo neuronal a partir de unos parámetros pasados como argumentos de entrada **[1,5 puntos]**
2. **Redes neuronales recurrentes. [2 puntos]** En este ejercicio, consideraremos el uso LSTM para abordar la tarea de clasificación de sentimientos de la práctica, con el fin de comprobar si hay mejora en los resultados alcanzados en comparación con FFN. Específicamente, se pide:
- a) Visualizar la evolución del loss y el accuracy de entrenamiento y validación de la LSTM de una capa creada y entrenada en `nlp_lab2.ipynb`. Los valores de esas métricas para cada época de entrenamiento están disponibles en la variable `history_lstm`. Explicar brevemente las gráficas generadas, y comparar los resultados y tiempos de ejecución con los del ejercicio 1. **[0,5 puntos]**
- b) Comprobar el efecto de early stopping al entrenar la LSTM, visualizando resultados de una variable `history_lstm_stop`. Explicar brevemente lo observado. **[0,5 puntos]**
- c) Analizar el efecto de variar el tamaño de los embeddings y el número de unidades LSTM, tanto mayores como menores a los usados en el apartado 2a). Explicar brevemente lo observado. **[1 punto]**
3. **Redes neuronales recurrentes complejas. [2 puntos]** En este ejercicio, evaluaremos el uso de LSTM de complejidad mayor a las del ejercicio 2. Las ejecuciones a realizar conllevan un tiempo relativamente alto. Se sugiere empezar usando batches de mayor tamaño (p.ej., 128) y un número de épocas de entrenamiento pequeño (p.ej., 10). Específicamente, se pide:
- a) Obtener resultados de Stacked LSTMs; al menos de una red de 2 capas. Se sugiere usar variable(s) `history_lstm_stacked`. Explicar brevemente lo observado. **[1 punto]**
- b) Obtener resultados de Bidirectional LSTMs; al menos de una red. Se sugiere usar variable(s) `history_lstm_bidirectional`. Explicar brevemente lo observado. **[1 punto]**
4. **Embeddings pre-entrenados. [3 puntos]** En este ejercicio, consideraremos el uso de word embeddings pre-entrenados (Glove) para la construcción de LSTMs. Específicamente, se pide:
- a) Cargar los embeddings del fichero `glove.6B.50d.txt.bz2` proporcionado en la práctica. **[1 punto]**
- b) Construir y evaluar LSTMs usando los embeddings cargados. Para ello, habrá que inicializar la primera capa de una LSTM con los embeddings y congelar los pesos asociados para el entrenamiento de la red. Explicar brevemente el estudio realizado y discutir los resultados obtenidos. **[1,5 puntos]**
- c) Explorar alguna estrategia (p.ej., dropout) para evitar el sobreajuste de la red. Explicar brevemente lo observado. **[0,5 puntos]**

Entregable:

El entregable de las tareas de laboratorio consiste en los siguientes archivos:

- `nlp2425-lab3-XX.ipynb`: un cuaderno Jupyter con el código Python desarrollado.
- `nlp2425-lab3-XX.pdf`: un documento que recoja análisis y discusiones de los ejercicios realizados.

Enviar estos archivos a través de Moodle en un archivo zip llamado `nlp2425-lab3-XX.zip`, donde XX debe sustituirse por el identificador del equipo, por ejemplo, 01, 02, ...