

Tarea 8

Javier Valencia Goujon 123227

Alfie Sergio González Salcedo 181574

== Q1 ==

A pesar de tener muy buena exactitud, la pérdida no bajó de 0.13, y más aún, la salida de la celda anterior muestra que se equivocó al pronosticar las clases positivas, incluso en el set de entrenamiento. ¿Por qué crees que eso sucede?

Pensábamos que podía ser por el número de palabras del vocabulario. Sin embargo, probamos con un vocabulario de 6,000 palabras y con otro de 9,000 y no cambiaron mucho los resultados.

== Q2 ==

¿Por qué crees que este modelo toma mucho más tiempo para entrenarse, en comparación con los modelos que habíamos entrenado anteriormente?

== Q3 ==

¿Recuerdas cómo funciona el layer Embedding? ¿Qué significan los parámetros de entrada y salida?

Toma vectores en los que las palabras se representan como enteros y los transforma a vectores de una dimensión dada.

`model.add(Embedding(input_dim=2000, output_dim=16))` significa que el vocabulario es de tamaño 2000 y el vector de salida es de tamaño 16.

== Q4 ==

Diseña y entrena un modelo que minimice la pérdida hasta 0.05 o menos.

Además de jugar con los parámetros que ya conocemos, como el número de layers, el número de parámetros en cada layer, loss, optimizer, etc., si quieres también explora otros valores para el número de palabras totales en el diccionario (definidas en el Tokenizer) y los argumentos para el padding.

Se probaron varios modelos, pero ninguno mejora mucho los resultados del modelo original.

Si se usa un vocabulario de 9,000 palabras, se alcanza una pérdida del orden de 12 punto algo, pero luego empieza a haber un ligero sobreentrenamiento.

== Q5 ==

Investiga como instanciar GRU

Compara RNN vs LSTM vs GRU. Reporta comparación en términos de número de parámetros, tiempo requerido para entrenamiento, y desempeño obtenido.

Con GRU se obtienen resultados similares.