

DEPARTAMENTO DE PROCESOS Y SISTEMAS

MATERIA: COMPUTACIÓN EMERGENTE

PROFESOR: NICOLAS ARAQUE VOLK

**Proyecto N° 3: Computación Emergente. Informe**

Integrantes:

Catapano, Mario. N° Carnet: 20171110517

Di Gennaro, Massimo. N° Carnet: 20171110123

Caracas, Noviembre 2019

1. Preprocesamiento de datos: Aquí se deben explicar todas las herramientas y tácticas que se utilizaron para preprocesar los datos. Variables de entrada al modelo, normalización, estandarización, etc.

Se estudiaron los siguientes dos embeddings: glove.840B.300d y wiki-news-300d-1M, y se descubrió que (antes de hacer algún tipo de preprocesamiento al texto), el porcentaje de palabras del set de entrenamiento que se encontraban en el embedding glove.840B.300d (33,024%), era mayor a la cantidad del embedding wiki-news-300d-1M (30.05%). Por lo que se seleccionó el embedding glove.840B.300d. En el preprocesamiento de datos, se procede a calcular el porcentaje de palabras que se encuentran en el embedding, y luego se muestran las palabras con mayor frecuencia, que no aparecen en el embedding. Se puede observar que la mayoría de las palabras poseen signos y por ello no las encuentra. Entonces se procede a quitar los signos de las preguntas de Quora suministradas, y se vuelve a calcular el porcentaje anterior, el cual aumenta a 70,9% luego de este proceso. Posteriormente, se cambiaron a minúsculas todas las palabras de las preguntas de Quora que se encontraban en mayúsculas, y se recalculó el porcentaje, con lo cual aumentó ligeramente a 71,6%.

2. Arquitectura del modelo: Aquí se debe explicar la arquitectura utilizada del modelo de Deep Learning. Se deben especificar todos los elementos que hemos visto en clase y algún otro que sea relevante (número de capas, número de neuronas, funciones de activación, regularización, etc).

Se utilizaron dos capas de LSTM bidireccional. El modelo se realizó con vectores de activación de dimensión 64 en el vector hacia adelante, y 64 en el vector hacia atrás (128 en total). El modelo posee un input de máximo 30 palabras, representadas por vectores de dimensión 300. Luego de que el input pasa por las capas de LSTM bidireccional, se pasa a una neurona con activación sigmoide, la cual realiza la predicción.

3. Entrenamiento del modelo: Aquí se debe explicar los elementos de entrenamiento del modelo (learning rate, epochs, algoritmo de optimización, etc).

El learning rate del modelo fue de 0,001. El total de epochs para entrenar el modelo son 10 epochs. El algoritmo de optimización utilizado fue Adam. Para calcular la pérdida se utilizó la función binary\_crossentropy.

4. Resultado de modelo: Aquí se debe detallar el resultado del modelo específicamente en: accuracy en entrenamiento y validación, resultado en leaderboard de Kaggle, tiempo de ejecución del notebook y cualquier otro indicador que sea relevante.

El accuracy de entrenamiento fue de 0,9592 y el de validación fue de 0,9620. El tiempo de ejecución total del notebook fue de 3737,9 segundos. El resultado en el submission de Kaggle es de 0,63077 en el score privado y 0,62166 en el score público.

5. Link al Kernel de Kaggle, este debe hacerse público.

Proyecto en Kaggle:

<https://www.kaggle.com/mariocatapano/proyecto-3-quora-rnn>

Versión con mejor resultado: <https://www.kaggle.com/mariocatapano/proyecto-3-quora-rnn?scriptVersionId=24222721>

Se debe de explicar en detalle todas las técnicas distintas a las vistas en clase utilizando referencias bibliográficas de fuentes primarias. Solo se podrán utilizar arquitecturas de tipo Red Neuronal Artificial Recurrentes.

**Kernels de Kaggle consultados**

<https://www.kaggle.com/christofhenkel/how-to-preprocessing-when-using-embeddings>

<https://www.kaggle.com/theoviel/improve-your-score-with-text-preprocessing-v2>

<https://www.kaggle.com/mihaskalic/lstm-is-all-you-need-well-maybe-embeddings-also>