Medical Abstracts Machine Learning Project

Dallas Strandell

November 1, 2022

1 Introduction

This document describes the machine learning project to fit sentence in medical abstracts into different classifications such as background, methods, results, etc. The source of the data is:
kaggle.com/datasets/anshulmehtakaggl/200000-abstracts-for-seq-sentence-classification

The data was modeled in the notebook NLP.ipynb which can be found here:
github.com/dstrande/abstracts_ML.

The repository contains one notebook for initial data processing and a Python script for creating and comparing multiple models. The comparison was performed using tensorboard. In logs.zip you can find the tensorboard logs files.

2 Initial Data Processing

The data available from the source above is stored in a txt file separated by abstract with the label at the beginning of each sentence. For example:

"METHODS\tA total of 100 patients with primary knee OA were randomized 2:3 ; 2 received 10 mg/day of prednisolone and 3 received placebo for 10 weeks .\t"

is an example of a method sentence. The first step was to separate the labels and features. This was using the Pandas function split(). The labels were mapped into integers ranging from 0 to 4 following: {'BACKGROUND': 0, 'OBJECTIVE': 1, 'METHODS': 2, 'RESULTS': 3, 'CONCLUSIONS': 4}. Rows with NaN values were removed; most of these rows were comments rows in the original txt files.

The features were converted into sentence embeddings with SentenceTransformers (or SBERT). The pre-trained model "all-mpnet-base-v2" was used to create the embeddings. The max sequence length was

set to 300 as this will include the maximum feature length. The processed data was then saved into a hdf5 file for later importing. Note that processed 200k abstracts file results in a large ∼7GB file. When loading the data for training, memory restrictions required loading only half the rows on my computer (32GB RAM, 6GB GPU memory). When loading the data I also convert the labels into categorical data with tf.keras.utils.to_categorical.

3 Model Results

As mentioned, half the data from the hdf5 file was imported and the labels were converted into categorical data. An example model used is seen in Figure 1. Many hyperparameters were varied and tracked using tensorboard and hparams. For example, the command "tensorboard –logdir logs/sept4-v6" can be used in the terminal to view the logs and compare the models from the sixth run on September 4th.

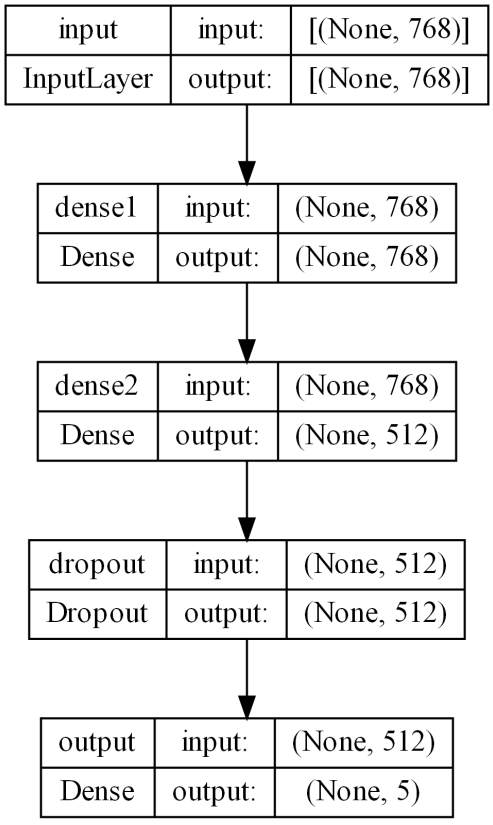

Figure 1: Model schematic. The second dense layer (and therefore the dropout layer) had a variable size from 5 to 768. The dropout fraction was varied from 0 to 0.5.

Learning rate, batch size, dropout rate, layer size (for the 2nd layer) were varied during multiple parameter

searches. Additionally, multiple Keras optimizers and activation functions were tested. See Figure 2 for an example trend. This specific hyperparameter search included varying the dropout rate after dense2 and the number of units in dense2.

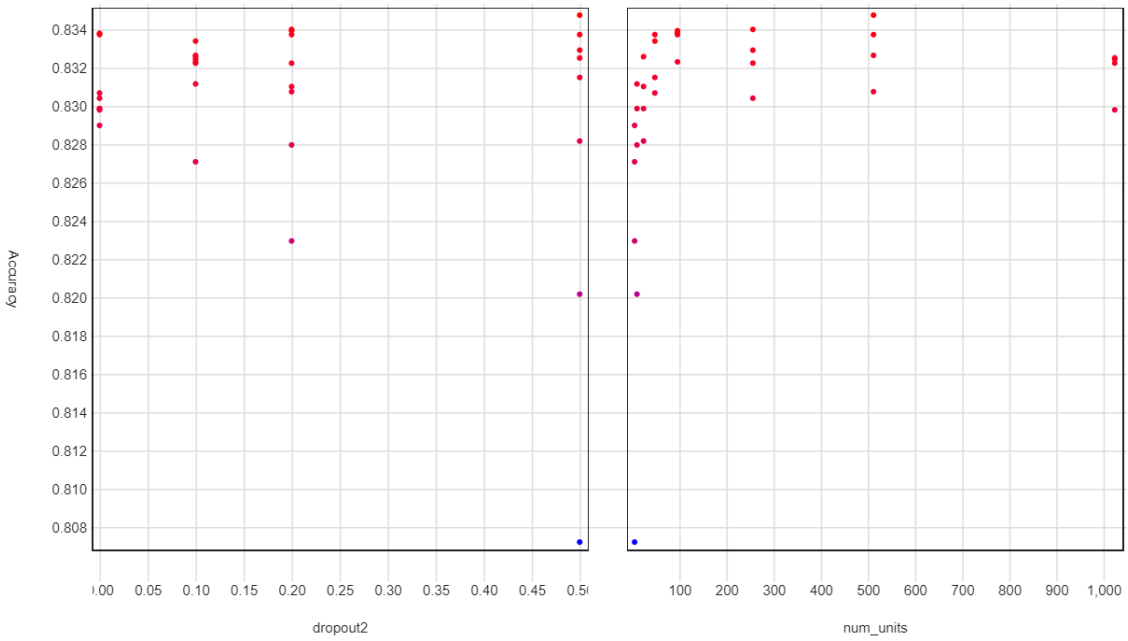

Figure 2: Plot showing how the model accuracy varied with dropout rate and layer size. The best layer size was 512 and the accuracy dropped off after this.

4 Conclusion

Sentences from medical journal abstracts were sorted into 5 categories using a neural network. A hyperparameter search was performed. The maximum accuracy observed was 0.834. Future work can include comparing other pre-trained models and using other methods of embedding sentences.