Name:- Swapna Vaidya ***Class:-*** MSDS – 422

COLAB NOTEBOOK LINK :- https://colab.research.google.com/drive/1ZvEjZ13PVEMWii61rSnLbZcaZvl0Ymt7

Recurrent Neural Networks Models

(1) ***A summary and problem definition for management***

This dataset is on movie reviews from Internet Movie Database. This dataset is used to experiment with the Recurrent Neural Network models. The findings of this modeling exercise will be used to make a recommendation to a management thinking about using a language model to classify written customer reviews and call and complaint logs. If the most critical customer messages can be identified, then customer support personnel can be assigned to contact those customers. The recommendation would be on what kinds of systems and methods would be most relevant to the customer services function? What is needed to make an automated customer support system that is capable of identifying negative customer feelings? What can data scientists do to make language models more useful in a customer service function?

(2) ***Discussion of the research design, measurement and statistical methods, traditional and machine learning methods employed***

The IMDB data set used here is a movie reviews dataset that consists of 50K records from the internal movie database. The records are split into 25,000 reviews for training and 25,000 reviews for testing. The training and testing sets are *balanced*, meaning they contain an equal number of positive and negative reviews.

(3) ***Overview of programming work***

This assignment was done in Google Colab, using 4 main libraries (Pandas, numpy, Skylearn and Keras).

- ***Ingestion:-*** The IMDB data set is available on the TensorFlow datasets. Via Colab using the tensorflow load method() the data is downloaded and used.
- ***Data Cleaning and Transformation:-*** The IMDB data has 50:50 in training and test data. Using the split method of the tensorflow, the training data is split into 60:40 ratio into training and validation dataset.
- ***Modeling:-*** The base model used was a Sequential. A pre-trained text embedded model from tensorflow hub called "google/tf2-preview/gnews-swivel-20dim/1." was used an input layer to map a sentence into its embedding vector. This layer splits the sentence into tokens, embeds each token and then combines the embedding. One additional fully connected dense layer with 8,16,32 combination of hidden tokens were used with the activation method "Relu". The last layer is densely connected with a single output node. Using the sigmoid activation function, this value is a float between 0 and 1, representing a probability, or confidence level.
  The model was then compiled with a binary_crossentropy loss function, with an 'RMSprop' optimizer and metrics as accuracy. The model was then fit on the train data and validated on the validation data set. The train data set was shuffled and combination of epochs 10,20 and 100 were used. Finally, the model was evaluated on the test data set. The results are the following:-

Hidden Tokens	Epochs	Loss	Accuracy
8	20	0.351	0.864
16	20	0.3181	0.8649
32	20	0.343	0.866
8	100	0.956	0.837
16	100	0.987	0.841
32	100	1.077	0.838
8	10	0.35	0.849
16	10	0.4452	0.8522
32	10	0.406	0.8515

(4) ***Review of results with recommendations for management***

Based on the above modeling and loss score my recommendation to the management would be the following:-

- To go with Recurrent Neural Network instead of other machine learning techniques to identify most critical customer messages.
- While using RNN networks, definitely use runtime as GPU instead of CPU or even TPU. With CPU depending on the size of the training dataset, it either crashes or runs very slowly. With GPU it atleast runs the model and is faster than CPU.
- Keep the RNN model simple, using 3 layers
- Use pre-trained text embedded model as an input layer to map a sentence into its embedding vector
- Based on the above results it seems like keeping the hidden tokens to 16 with 20 epochs gives the lowest loss and highest accuracy