TCS ONTERNSHIP PROJECT

Table of Contents

Project accessible at:

https://drive.google.com/open?id=1LSUtlqSZ6G-z-NNAUvY1YbYqx9B4ZG6d

Problem Statement

Create a ***chatbot*** to handle banking issues raised by customer with the help of intent classification and NLP.

Use Case Details

In this problem we have to create a chatbot which can handle incidents raised by users by reporting issues faced while accessing their bank account. The user will report the issue to a customer care executive who will then initiate chat with the bot mentioning an incident number and resolve the issue raised.

Approach

The above use-case is implemented in three steps:

1. First step is to categorise the problem raised by the user by applying some machine learning algorithms.
2. Second step is to find the root cause of the issue raised by user by asking more relevant details of the issue through chatbased on the knowledge repository.
3. After understanding the root cause of the issue third step is to resolve the issue with the help of chatbot.

Review Result of Tools

First stage of the problem –classify the complaints into different categories:

- We categorised the problem into three different categories:
    1. User Information related issues
    2. Debit/Credit card related issues
    3. Transaction related issues
- The problem was solved in two different ways:
    1. Using a single layer Neural Network.
       Accuracy on test set :0.714
    2. Using Recurrent Neural Network(Deep Learning).
       Accuracy on test set :0.810
- Tools used: python, jupyter notebook, keras, tensorflow, numpy, matplotlib, Global Vectors for word representation (GloVe) etc.

Papers Referred

- Long Short-term Memory – SeppHochreiter & JurgenSchmidhuber, 1997
- Dropout: A Simple Way to Prevent Neural Networks from Overfitting - Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, IlyaSutskever, Ruslan Salakhutdinov; 15(Jun):1929−1958, 2014.
- Adam: A Method for Stochastic Optimization - Diederik P. Kingma& Jimmy Ba, 2015.
- Recurrent Convolutional Neural Networks for Text Classification- Siwei Lai, LihengXu, Kang Liu, Jun Zhao.

Current Work

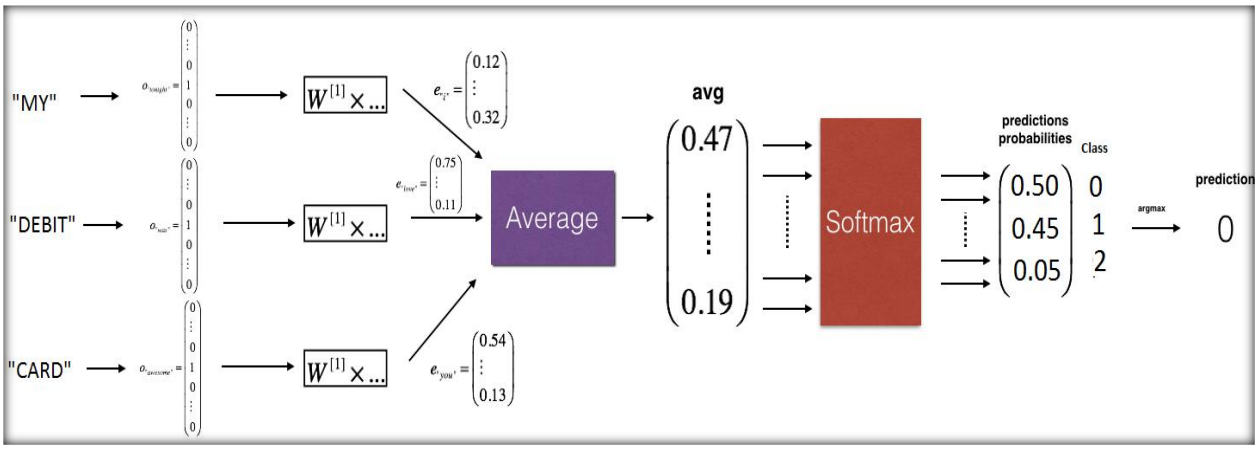
1. Using a single layer Neural Network:

Key specifications:

- Layers: 1 (Softmax layer)
- Iterations: 150
- Learning rate: 0.01
- Dataset: 100 (Training : Test = 80:20)

Architecture:

A very basic architecture was used consisting of only a softmax layer. Word-embedding from GloVe is used. The following figure shows the entire architecture of the model used.



Results:

CONFUSION MATRIX

Predicted	0.0	1.0	2.0	All
Actual				
0	8	1	0	9
1	0	5	0	5
2	4	1	2	7
All	12	7	2	21

2. Using Recurrent Neural Network (Deep Learning):

Model Architecture:

Relatively deeper model was used consisting of LSTM layers, Dropout layers and softmax layer. Word-embedding from GloVe is used. The following figure shows the entire architecture of the model used.

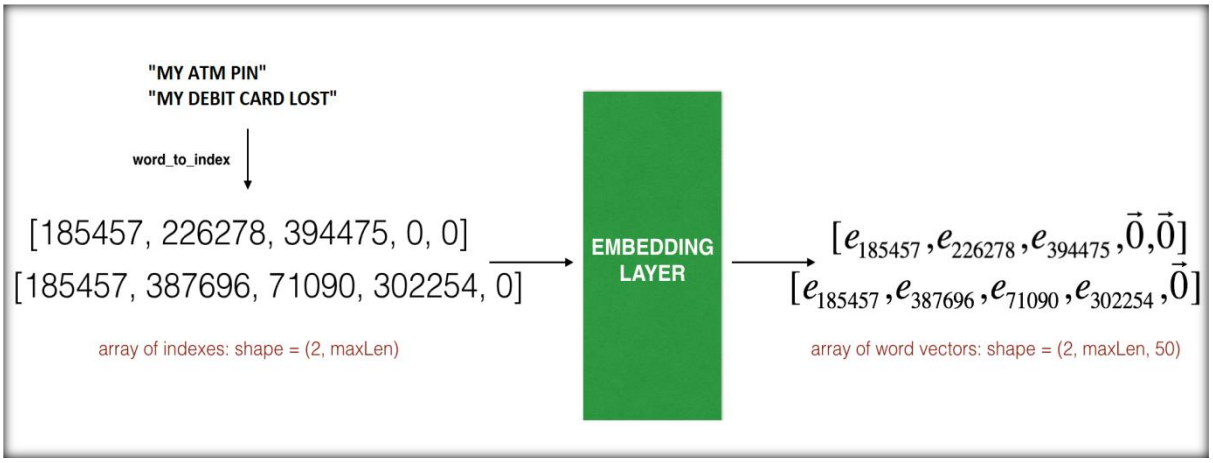
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 48)	0
embedding_1 (Embedding)	(None, 48, 50)	20000050
lstm_1 (LSTM)	(None, 48, 128)	91648
dropout_1(Dropout)	(None, 48, 128)	0
lstm_2 (LSTM)	(None, 128)	131584
dropout_2 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 3)	387
activation_1 (Activation)	(None, 3)	0

Total params: 20,223,669

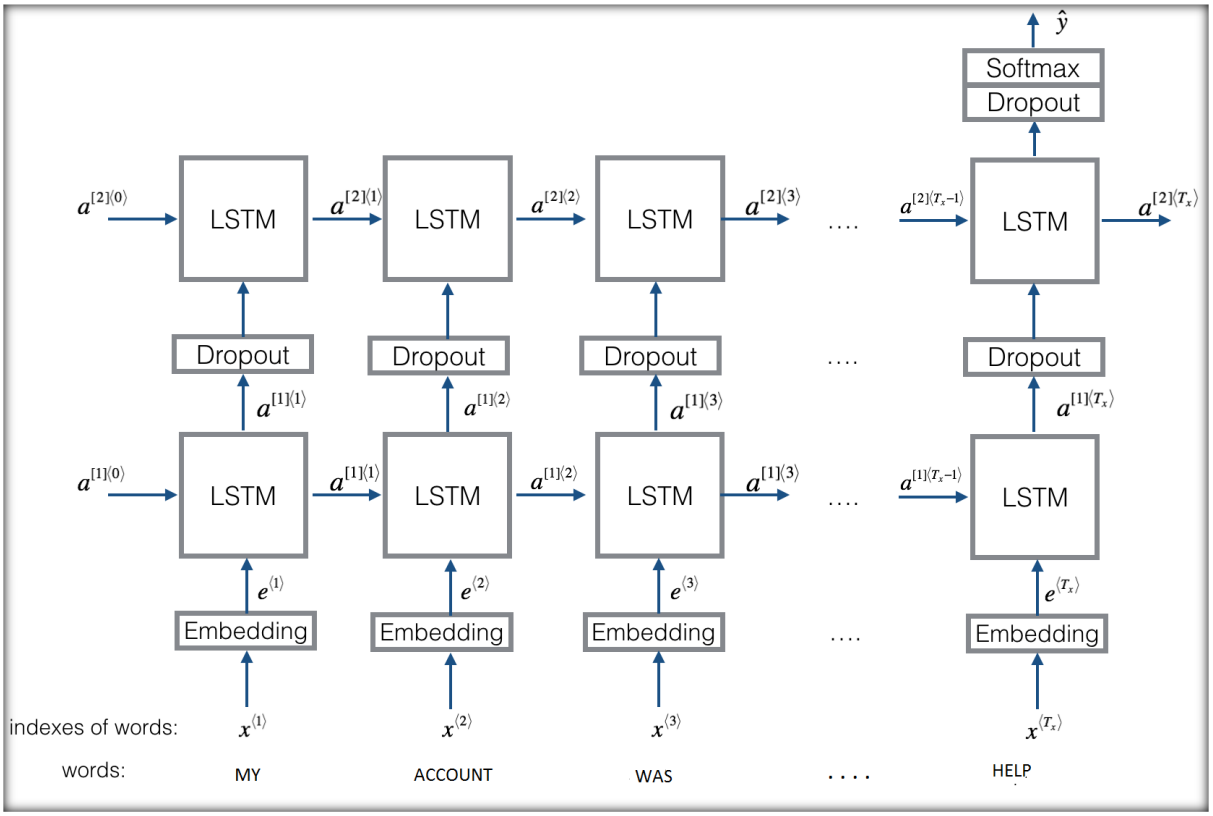
Trainable params: **223,619**

Non-trainable params: 20,000,050

- Sentences to Embedding:



- Embedding to classes:



Key Specifications:

- Batch Size: 10
- Epochs: 30
- Optimiser: Adam(lr=0.001, beta_1=0.9, beta_2=0.999)
- Loss function: Categorical crossentropy

Results:

CONFUSION MATRIX

	0	1	2	
Predicted	0.0	1.0	2.0	All
Actual				
0	7	1	1	9
1	0	5	0	5
2	2	0	5	7
All	12	7	2	21

Scope of Improvement

1. Enriching the knowledge repository with more distinct words from the user.

2. The cases of conflict among various intents have to be resolved further.

3. We could improve the accuracy further, avoiding overfitting condition.

4. Various independent parameters used could be varied with the size of dataset to produce more accurate results.

5. Words like “Adhaar”,”pan”, etc are not included in the word-embedding used. These can be included requiring retraining the word-embedding.

6. Increasing the dataset to over thousands of entries and using a GPU for training can give higher accuracies.