E2E Systems : Project Report

EE17B035(V Sai Krishna) - EE17B047(Kommineni Aditya)

July 2020

1 Introduction

This project involves the use of End to End Encoder-Decoder models to facilitate the task of classifying speech files of isolated digits into the corresponding classes. We employed the used of PyTorch Deep Learning framework for purposes of coding.

The models which have been trained have been dicussed in the sections as mentioned below :

- Encoder-Decoder without Attention Mechanism (Section 2)
- Encoder-Decoder with Attention Mechanism (Section 3)

The Speech files were converted into MFCC features with each feature having a dimensionality of 38.

2 Encoder-Decoder without Attention

2.1 Structure

In this architecture type, we used LSTM cells as the components for both the encoder and decoder. The size of the hidden layer of the LSTM cells is 512. The hidden state and the cell state of the encoder layer has been fed as the first hidden state input and first cell state input to the decoder respectively.

Now, during the training phase of the decoder, force learning was implemented (i.e. the inputs provided to the decoder are the ground truths in contrast to the decoder output at the previous time step) with a probability of 0.5. This enables faster convergence and does not have any impact on the final accuracy of the model. The produced outputs are then compared to the input mfcc features.

- Hidden Layer Units : 512
- Optimizer : Adam
- Loss Function : Mean Squared Error
- Epochs : 15

Once the above model has been sufficiently trained, we could take the last hidden state and cell state of the LSTM encoder cell and use it as a representation for that specific speech file.

Then, we built an ANN model with appropriate complexity which would act as the discriminator. The configuration of the neural network model used is as follows :

- Loss Function : Cross Entropy Loss
- Optimizer : Adam
- Epochs : 100
- NN Layers : 1,2 and 4 hidden layers

2.2 Loss Curve for Encoder-Decoder Configuration

The Training Loss curve for the above mentioned encoder decoder configuration is as follows :

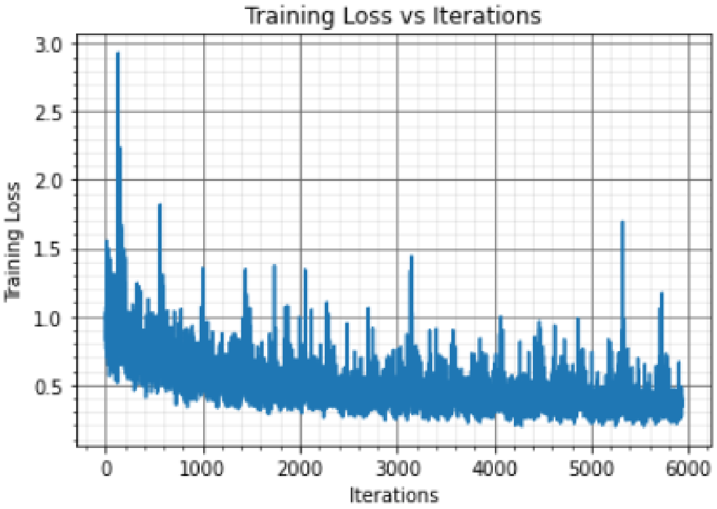

Figure 1: Training Loss vs Iterations

2.3 Loss , Accuracy & t-SNE plots for ANN Classifier

The training loss curves for the neural network configuration is as follows :

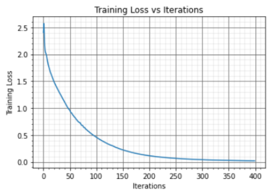

Figure 2: with 1 Hidden Layer

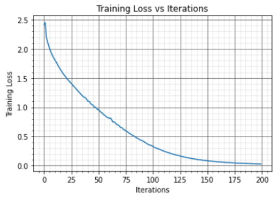

Figure 3: with 2 Hidden Layers

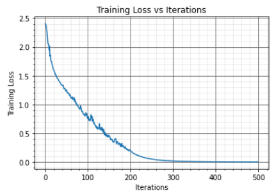

Figure 4: with 4 Hidden Layers

The training accuracy curve for the neural network configuration is as follows :

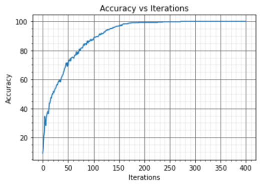

Figure 5: with 1 Hidden Layer

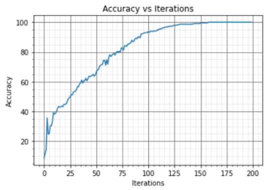

Figure 6: with 2 Hidden Layers

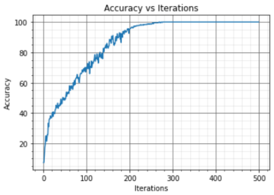

Figure 7: with 4 Hidden Layers

The test accuracy of the ANN classifiers are :

- with 1 Hidden layer: 47.72 %
- with 2 Hidden layers: 49.24 %
- with 4 Hidden layers: 46.63 %

The t-Sne plots for the ANN layers are as follows :

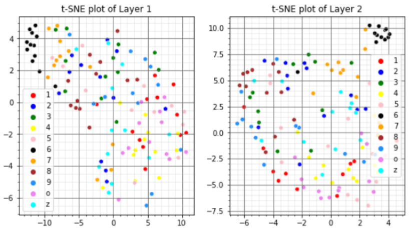

Figure 8: t-SNE of ANN with 1 Hidden layer

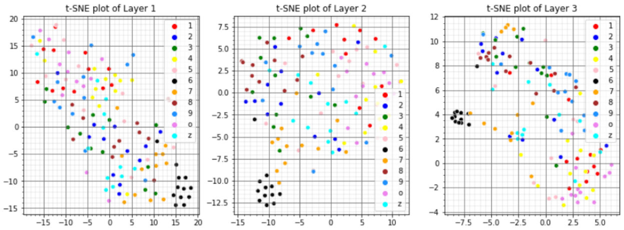

Figure 9: t-SNE of ANN with 2 Hidden layers

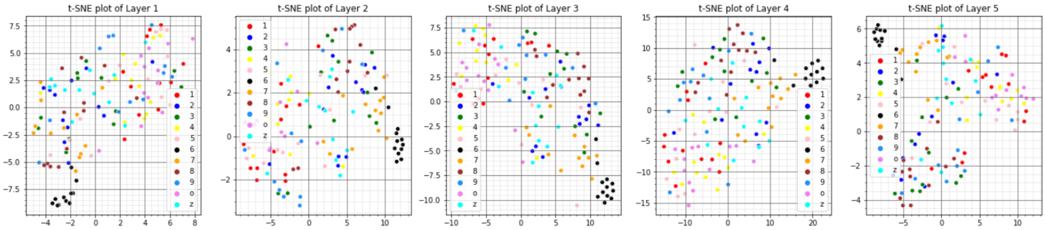

Figure 10: t-SNE of ANN with 4 Hidden layers

2.4 Observations

Even though the training loss of the Encoder-Decoder network doesn't converge to zero, our ANN classifiers managed to achieve 100% training accuracy. But the test accuracy is not as good as expected. This tells us that our Encoder performance to an untrained input is sub-optimal as we force it to put together all the important content from the MFCC file into the final hidden state and cell state.

Also, too many hidden layers in the ANN leads to over-fitting. That's why the test accuracy of ANN with 2 hidden layers is better than the test accuracy of ANN with 4 hidden layers.

3 Encoder-Decoder with Attention

3.1 Structure

The architecture is similar to the one mentioned in the previous section. The only difference is that we will incorporate attention by taking the weighted average of the encoder hidden states and cell states over all time steps and passing those as the inputs to the decoder unit. We used the dot-product algorithm to calculate the weights for hidden state and cell state.

3.2 Loss Curve for Encoder-Decoder Configuration

The Training Loss curve for the above mentioned encoder decoder configuration is as follows :

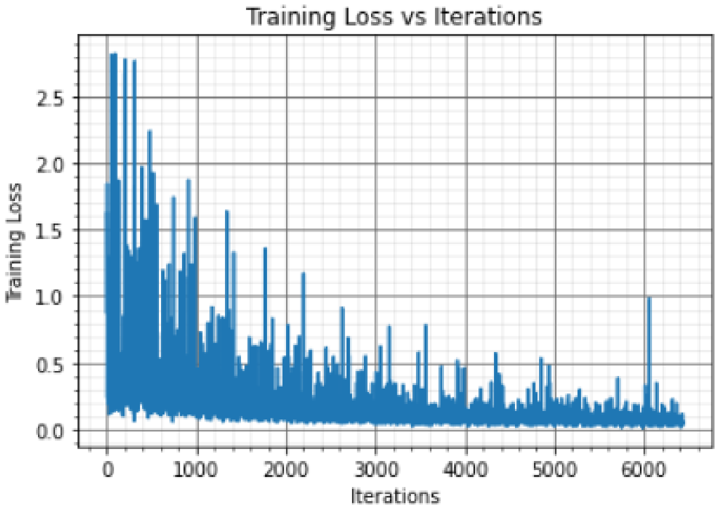

Figure 11: Training Loss vs Iterations

3.3 Loss , Accuracy & t-SNE plots for ANN Classifier

The training loss curves for the neural network configuration is as follows :

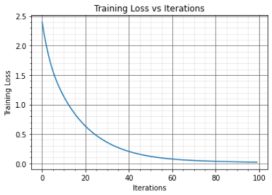

Figure 12: with 1 Hidden Layer

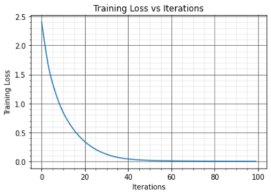

Figure 13: with 2 Hidden Layers

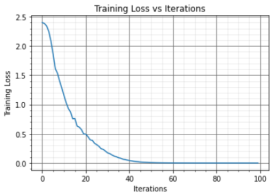

Figure 14: with 4 Hidden Layers

The training accuracy curve for the neural network configuration is as follows :

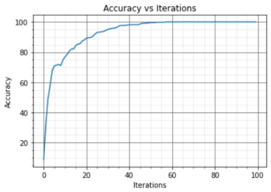

Figure 15: with 1 Hidden Layer

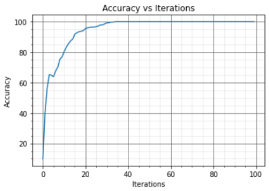

Figure 16: with 2 Hidden Layers

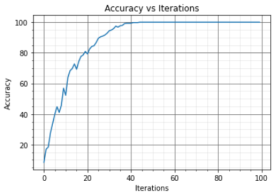

Figure 17: with 4 Hidden Layers

The test accuracy of the ANN classifiers are :

- with 1 Hidden layer: 88.63 %
- with 2 Hidden layers: 91.90 %
- with 4 Hidden layers: 83.33 %

The t-SNE plots for the ANN layers are as follows :

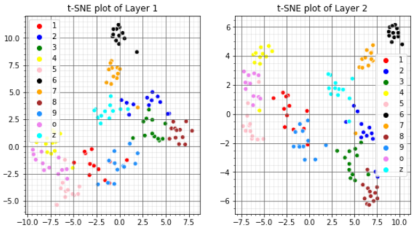

Figure 18: t-SNE of ANN with 1 Hidden layer

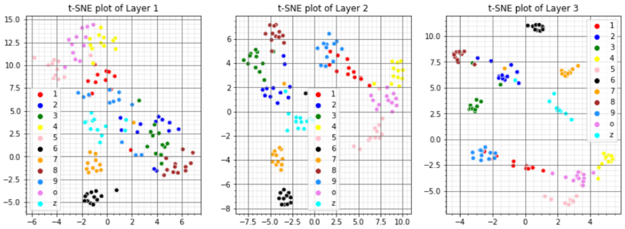

Figure 19: t-SNE of ANN with 2 Hidden layers

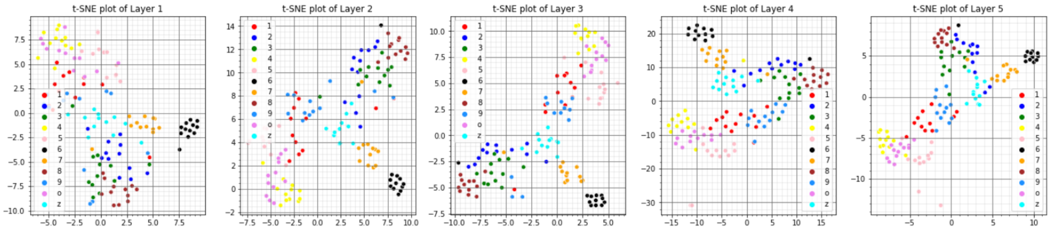

Figure 20: t-SNE of ANN with 4 Hidden layers

3.4 Observations

When attention mechanisms are used, the training loss of the Encoder-Decoder network converges to a value very close to zero. Also, our ANN classifiers managed to achieve 100% training accuracy and really good test accuracy (88-92%). This tells us that our Encoder-Decoder network with attention is better at generalizing, making it optimal for untrained/unseen input.

Also, just like in the previous section, using more than adequate number of hidden layers in the ANN leads to over-fitting. That's the reason why the test accuracy of ANN with 2 hidden layers is better than the test accuracy of ANN with 4 hidden layers.

4 Conclusion

- Firstly, the most profound observation is the fact that the application of attention greatly improves the performance of the model.
- From the various possible methods of attention, we chose to use the **Content Based Dot-Product** attention method. In this manner, we have two benefits of taking into consideration all the encoder outputs as well as avoiding the introduction of new learnable parameters into the network.
- From the t-SNE plots, we can clearly observe that when attention is absent there aren't clear distinctions between the various number clusters owing to which we theorize that the accuracy is lower. However, when attention is employed we can clearly see the number clusters which will enable a clear separation of the numbers.

5 References

- https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html
- https://medium.com/@adam.wearne/seq2seq-with-pytorch-46dc00ff5164

- https://arxiv.org/pdf/1409.0473.pdf
- Link to the code is: Link