

Aaron Shabanian

Machine Learning

Assignment 4

## **Introduction**

In this assignment there were two main parts. First, there were two tutorials with one being for a multiclass text classification model and one being a time series prediction model. Second, I built a model that generates nursery rhymes based off of reading previous ones. This paper is separated into two parts. The first part will go over the tutorials and the second part will go over the nursery rhyme generation.

## **Part 1: Tutorials**

### **Background**

Text classification is one of the biggest things that is noticed in everyday life. For example, on most news sites, the news is categorized into different categories like Politics, Business, Weather, etc. Time series predictions are also used everyday on things like stock predictions and economics metrics. Both of these models use sequential neural networks to try and “predict” either the category of news or the number of airline predictions. Sequential models are models where the input and output are sequences of data. This can be streams of text, numbers, etc.

### **Findings**

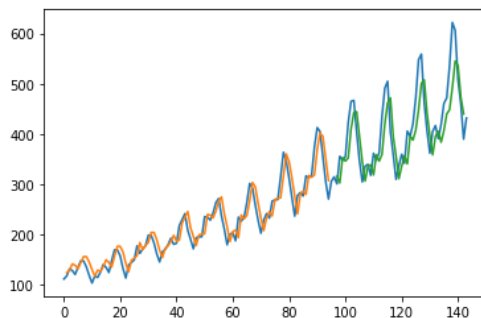
#### **Multiclass Text classification**

When classifying the various news articles, the first step was to get rid of common words. These include words like and, the and it. I got rid of those words with the nltk python package

which is used for natural language processing. Getting rid of these words is important as they will be in every article and shouldn't be used as factors in deciding what type of article it is. The next step before feeding it into the model is to tokenize the words and make it into a series of integers that can be fed into the model. Once the model is complete and fit, I got a perfect training accuracy of 1.0 and a validation accuracy of .94. This is an extremely high accuracy that is only slightly overfit since the training accuracy is only slightly better than the validation accuracy. I then ran two different examples. It correctly predicted both of them as politics and business.

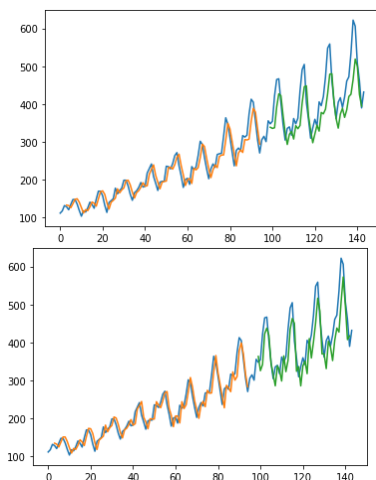
## Time Series Prediction

Based on an initial time series of airline passengers over time, I built a model that predicts the number of airline passengers in the future. With 100 epochs I was able to get to a small amount of loss of only 0.002. In the initial model the test RMSE was 49. This is not that bad of an RMSE. The graph of prediction vs reality is shown below:



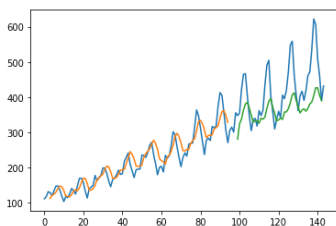
As can be seen, the prediction is pretty close to what actually happened but can still be improved. Second, I increased the window size and saw if that would improve or make the model worse. The loss is around

the same with 0.002. The test RMSE is a little higher in this one at 57 so this didn't help the model much but rather hurt it.

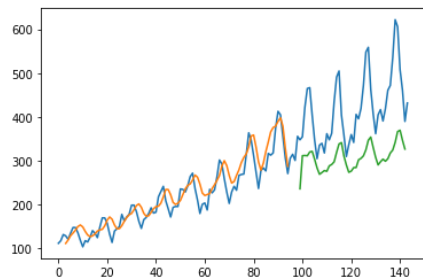


I can see a little more gaps between what actually happened and the prediction in this graph versus the original model. I then tried to increase the add time steps and re- run the model.

The test RMSE is in between both models at 50. Next I tried to make an LSTM with memory between batches and ran the model.



This model was much worse with a test RMSE of 80. This shows that it might not be effective to use an LSTM between batches. Lastly, I ran a stacked LSTM with memory between batches.



This model is by far the worst of all of them as can be seen in the gap between the prediction and what actually happened. The test RMSE of 125.62 is the worst one of all the models.

Overall, I can see that the models continuously got worse the more I did to them. The based model ended up doing the best

## Part 2: Nursery Rhyme Generation

## Background

Recurrent Neural networks solve the issue of information getting lost in between multiple layers that's important to make the prediction. It does this by feeding in old information that was outputted by a previous layer and adds it as an extra input in a future layer. It takes in sequential or time series data too. Some applications of recurrent neural networks are voice search, language translation and virtual assistants.

## Findings

Like the tutorials, I had to first tokenize all the data and make them all into integers that can be fed into the model. Using 300 epochs, I was able to get to a 99.5% accuracy score. Although this doesn't mean much since there isn't an actual correct answer to predict when choosing a word, it still shows its good at guessing the next word. I created a random seed generator that chooses a random seed of 10 words that were in previous rhymes. The rhyme that was created by the model below:

woman went fortunetelling by cherrystones one i love two i love three i love i say four i love with  
all my heart five i cast away six he loves seven she loves eight both love nine he comes ten  
he tarries eleven he courts and twelve he marries little bopeep little bopeep has lost her sheep and cant tell  
where to find them let them alone and theyll come home and bring their tails behind them little bopeep fell  
fast asleep and dreamt she heard them bleating and when she awoke she found it a joke for still they  
were all fleeing then up she took her little crook determined for to find them she found them indeed but  
it made her heart bleed for theyd left all their tails behind them it happened one day as bopeep did  
stray into a meadow hard by there she espied their tails side by side all hung on a tree to  
dry she heaved a sigh and wiped her eye and went over hill and dale oh and tried what she  
could as a shepherdess should to tack to each sheep its tail oh to bed come lets to bed says  
sleepyhead sit up a while says slow put on the pan says greedy nan lets sup before we go of  
going to bed go to bed first a golden purse go to bed second a golden pheasant go to bed  
third a golden bird grace before meat here a little child i stand heaving up my either hand cold as  
paddocks though they be here i lift them up to thee for a benison to fall on our meat and  
on us all there was a butcher there was a butcher cut his thumb when it did bleed then blood  
did come there was a chandler making candle when he them stript he did them handle there was a cobbler  
clouting shoon when they were mended they were done there was a crow sat on a stone when he was  
gone then there was none there was a horse going to the mill when he went on he stood not  
still there was a lackey ran a race when he ran fast he ran apace there was a monkey climbed  
a tree when he fell down then down fell he there was a navy went into spain when it returnd  
it came again there was an old woman lived under a hill and if shes not gone she lives there  
still winter has come cold and raw the north wind doth blow bleak in a morning early all the hills  
are covered with snow and winters now come fairly mondays child mondays child is fair of face tuesdays child is  
full of grace wednesdays child is full of woe thursdays child has far to go fridays child is loving and  
giving saturdays child works hard for its living but the child that is born on the sabbath day is bonny

and blithe and good and gay jack and jill jack and jill went up the hill to fetch a pail  
of water jack fell down and broke his crown and jill came tumbling after then up jack got up and  
off did trot as fast as he could caper to old dame dob who patched his nob with vinegar and  
brown paper charley charley charley charley stole the barley out of the bakers shop the baker came out and gave  
him a clout which made poor charley hop the pipers cow there was a piper had a cow and he

I can see that it created an interesting nursery rhyme. While it does sound weird in some parts, I  
can still see some rhyming in it.

## **Overall**

### **Conclusion**

Looking at both the parts, the tutorials and the rhyme, I can see the benefits of the  
recurrent neural networks and how they can be useful for sequential data. This worked in reading  
a news article, predicting a time sequence and writing its own words based on an example.