FAST-NUCES

BSCS (Computer Science)

Machine Learning

Sales Prediction using LSTM (Long Short Term Memory)

Author:
Ibtihaj Tahir 13-6064

Supervisor:
Dr. Mohammad Nauman

May 21, 2018

Contents

1 Introduction **2**

2 LSTM **3**

2.1 Squishing Function 3

2.1.1 tanh: Tan hyperbola 4

2.1.2 Sigmoid 4

2.2 Gating 5

2.3 Possibilities Filter 5

2.4 Ignoring Filter 5

2.5 Forgetting Filter 5

2.6 Selection Filter 6

2.7 The Predictions 6

3 Results **6**

3.1 Tools 6

3.2 Dataset 6

3.3 LSTM Model(Keras) 7

3.4 RMSE (Root Mean Squared Error) 7

3.5 Predictions 9

4 Conclusion **9**

Abstract

Machine Learning and Deep Learning is growing vastly these days. Models can learn a lot faster than a new born baby. Metaphorically they can go from new born baby to a PHD level Doctor in just a matter of weeks. They can classify things in categories as well as predict the future via regression. When it comes to some data including time element(time series data), these models don't perform that well. The main reason is because these models can't keep track of the time based data or it is hard for them to keep track of that time based variation. **LSTM (Long Short Term Memory)** model is a way to keep track of that recurrence.

1 Introduction

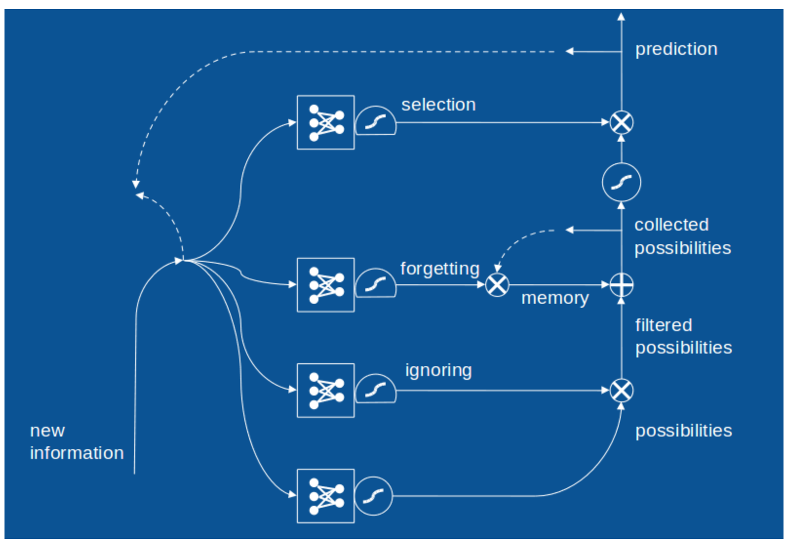
Outline Given you have a room mate who makes you dinner every night. He has a unique pattern of three things that he makes. Pizza, Sushi and Waffles. On day one it would be pizza and so on respectively. One day you were in a late meeting and missed the dinner. Next day you will have to predict the dinner for tonight. As you missed the last night's dinner, you will have to predict the previous to last night's dinner. This gives us an idea what we mean when it comes to time series data.

2 LSTM

Model Description LSTM is a deep learning model that performs well on time series data. This is an application of the deep neural networks. LSTM uses multiple DNNs to perform different functionalities to generate a time series function close to ground truth.

Input The core idea behind the LSTM's input is to input the previous predictions concatenated with the current inputs.

LSTM's Architecture The architecture of the LSTM involves 4 deep neural networks performing differently to contribute. These 4 deep neural networks are trained with the same input being previous predictions and current input. Every DNN will accept these inputs and train itself for the functionality they are providing. These 4 deep neural networks are filters that filter the data in a meaningful way.



2.1 Squishing Function

When it comes to datasets, any range of values can be there. As we are manipulating the data to convert it into a meaningful form, the model variables can interfere and may form some huge values. What we want to do is

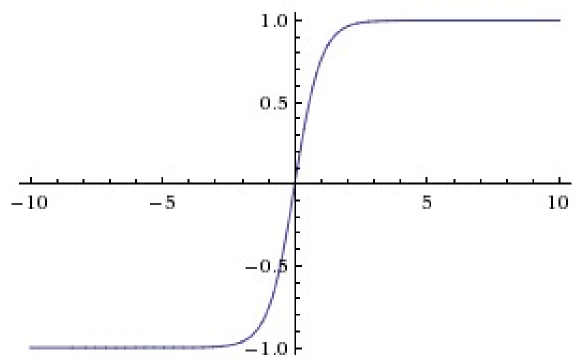
to bring those values in a way that they can be squished to a specific range without losing there contribution. In other words we want to normalize the data.
2 types of Squishing Functions are used in LSTM.

1 tanh

2 Sigmoid

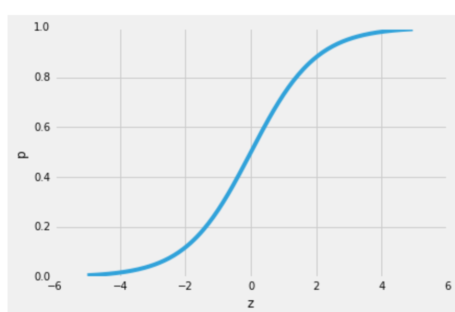
2.1.1 tanh: Tan hyperbola

This function squishes the ouput in the range of (-1, 1)

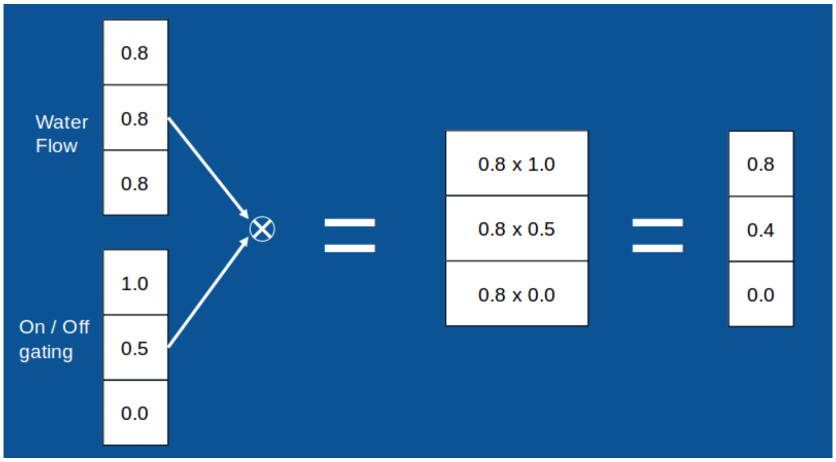


2.1.2 Sigmoid

This function squishes the ouput in the range of (0, 1)



2.2 Gating



Given we have a vector controlling water flow from 3 channels giving same pressure of the water. Suppose we have 3 gates from which 1 is full open, 1 is half open and the last one is not open at all, Gating procedure will allow the flow of water to pass. 1st gate will let the full pressure pass and vise versa.

2.3 Possibilities Filter

This is the main prediction generator whose predictions get filtered by the other DNNs.

2.4 Ignoring Filter

The predictions coming from **Possibilities Filter** is gated with the predictions coming from **Ignoring Filter**. This gating procedure helps us to ignore the recurring input data so that the things don't get repeated. In our case if yesterday was pizza, today can't be pizza again. So the predictions containing pizza will be ignored.

2.5 Forgetting Filter

This is the place where the idea of Long Short Term Memory comes in. After filtering the predictions, those filtered predictions are added up in a memory section so they can be accessed later. The forgetting DNN learns through time, which of these memories should be forgotten. Here again comes the idea of gating whose threshold is adjusted in such a way that memories important should stay in memory where as the others can be forgotten. Some of these memories are added up in the coming predictions.

2.6 Selection Filter

There is a chance that too much memories can add up when data passes from the **Forgetting filter**. So there is a DNN trained to select some of the predictions so that there are less memories contributing in the next epoch.

2.7 The Predictions

After all the filter applied, we will have our predictions and these predictions will contribute with the new input and will be feed to all the DNNs in the LSTM's Architecture.

3 Results

3.1 Tools

1 Python

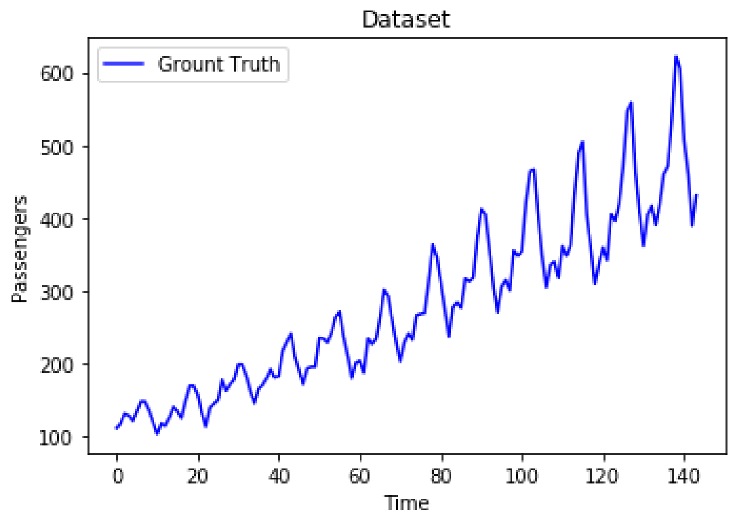
2 Keras

3 Tensorflow (GPU Based)

3.2 Dataset

Air Passengers Is a dataset that holds the values of passengers per month. As the sales variate every month, we need to keep track of the sales every month. There is long term random function between the time and the sales that we want to predict.

The data is plotted with respect to time and number of passengers. The label is set to ground truth as this is the actual data which will be used to calculate the accuracy of the model trained.



3.3 LSTM Model(Keras)

Look Back Is a threshold of how far can we look back into the memory. The procedure of looking back into the memory is known as **Unrolling**

The model built in python is Sequential. It has 4 filters and the inputshape of the size (1, lookback). 1 Dense layer is added having 1 neuron.

Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 4)	112
dense_2 (Dense)	(None, 1)	5

```
Total params: 117
Trainable params: 117
Non-trainable params: 0
```

3.4 RMSE (Root Mean Squared Error)

The RMSE shows us the cost of the network. The goal is to minimize this. The lower the RMSE, the better the model. After training the RMSE tends to,

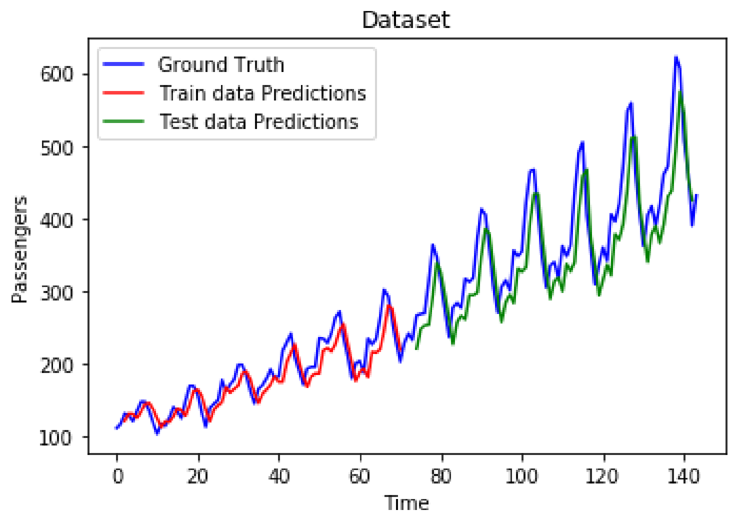
$$TrainScore : 20.85$$

$$TestScore : 49.67$$

We can see that there is a considerable amount of variation between both the test error and train error so we can say that the model is not over fitted.

3.5 Predictions

After 100 epochs the network shows us the above mentioned RMSE. Let's Plot the predictions with the orignal dataset to see the visual difference between the actual random function and model's predicted function.



Where the Red line shows the predictions based on training data and green shows us the predictions based on testing data.

4 Conclusion

We can see that LSTM have a very high accuracy on the time series data. We can predict the values upto a total of 98.4% of accuracy. This is more then good as now our model is able to predict sales based on the time of the year or month of the year.