Market Microstructure Signals for Apple Stock Using Deep Learning

Project Overview

Market microstructure is a branch of finance concerned with the details of how exchange occurs in markets. While the theory of market microstructure applies to the exchange of real or financial assets, more evidence is available on the microstructure of financial markets due to the availability of transactions data from them.
https://en.wikipedia.org/wiki/Market_microstructure
In the context of Stock trading, real time market data is used to study the market microstructure and models can be developed to predict prices. In the project , I created models that can be used to predict price movements of a particular stock

Problem Statement

The goal of this project to develop deep learning models to predict price movements of a particular stock . The following tasks are involved

1) Downloading one day of market data for all S&P stocks from quandl(free data for one day available)
2) Develop a multi classifier using various deep learning algorithms in tensor flow
3) Compare the various algorithms

Reason for Using Intra Market Data

The intraday market data was used for the following reasons

1) The data is transactional based on supply and demand and impact of value investing is minimized. Inter day Price movement study will require additional input attributes like P/E ratio of the stock
2) Impact of market events on interday price movement study require additional input parameters like fed rates
3) The data needed for interday price movement study is made available only for subscription fee

Metrics

This a multi classifier problem with three possible outcomes . The model will predict whether the price of stock is going to increase , decrease or remain the same .
Price Increase : A price increase above certain band in any of the following five consecutive ticks is considered as price increase
Price Decrease : A price decrease below certain band in any of the following five consecutive ticks is considered as price increase
No Change : If the price stays with a certain band for the next five consecutive ticks
The accuracy is calculated as follows

Accuracy = True Predicted Outcomes of given test set /Total Test Set

Description of Data

Intraday market data for S&P stocks is made by various data provider by paying a subscription fee. A sample one day data was available from one of the market data providers. The following are the information provided in the intra day market data file DailyTickSandPNov112017.csv. Per minute information is made available in the sample data

Date	**Timestamp**	**Ticker**	**OpenPrice**	**HighPrice**	**LowPrice**	**Close Price**	**TotalVolume**	**TotalQuantity**	**TotalTradeCount**
Date	Time of the day	Stock Symbol	Open price in time window	Highest price in one minute	Lowest price in one minutes	Close price at the end of the minute	Total volume traded in dollar	Total number of stocks traded	Total number of trades in one min window

Sample Data

Date	Timestamp	Ticker	OpenPrice	HighPrice	LowPrice	ClosePrice	TotalVolume	TotalQuantity	TotalTradeCount
20151117	9:31:00	TSS	52.86	53	52.83	53	42334	800	8
20151117	9:33:00	TSS	52.98	53.05	52.96	53.05	54099.58	1021	10
20151117	9:34:00	TSS	53.05	53.05	52.97	53.04	42424	800	5

Data Preprocessing

MinMax Scalar was employed for the all input attributes . The market data was from a reliable vendor to have errors . However, the data was analyzed for outliers for all the input variables with Z score of the input attributes to be greater than 3

Algorithms and Classifiers

Generating Market signals with limited data (free data) does have its limitation . However , i plan to develop the following models using various deep learning techniques and evaluate the performance of each model

The following deep model methods were developed and the performance evaluated

Simple Neural Network

A simple neural network was used to represent a nonlinear classification model .Since simple neural network does not represent state information of the past events , it was compensated by adding derived attributes to the input label which carry state information of the past . The derived attributes were added in addition to the above mentioned labels

priceChange	**movingAverage**	**upperBB**	**lowerBB**	**chaikinOscillator**
priceChange between successive ticks	Moving Price average of last 5 and 20 ticks were used	Upper bollinger band is average price plus standard deviation of last 30 ticks	Lower bollinger band is average price minus one standard deviation of last 30 ticks	http://www.investopedia.com/terms/c/chaikinoscillator.asp?lgl=rira-baseline-vertical

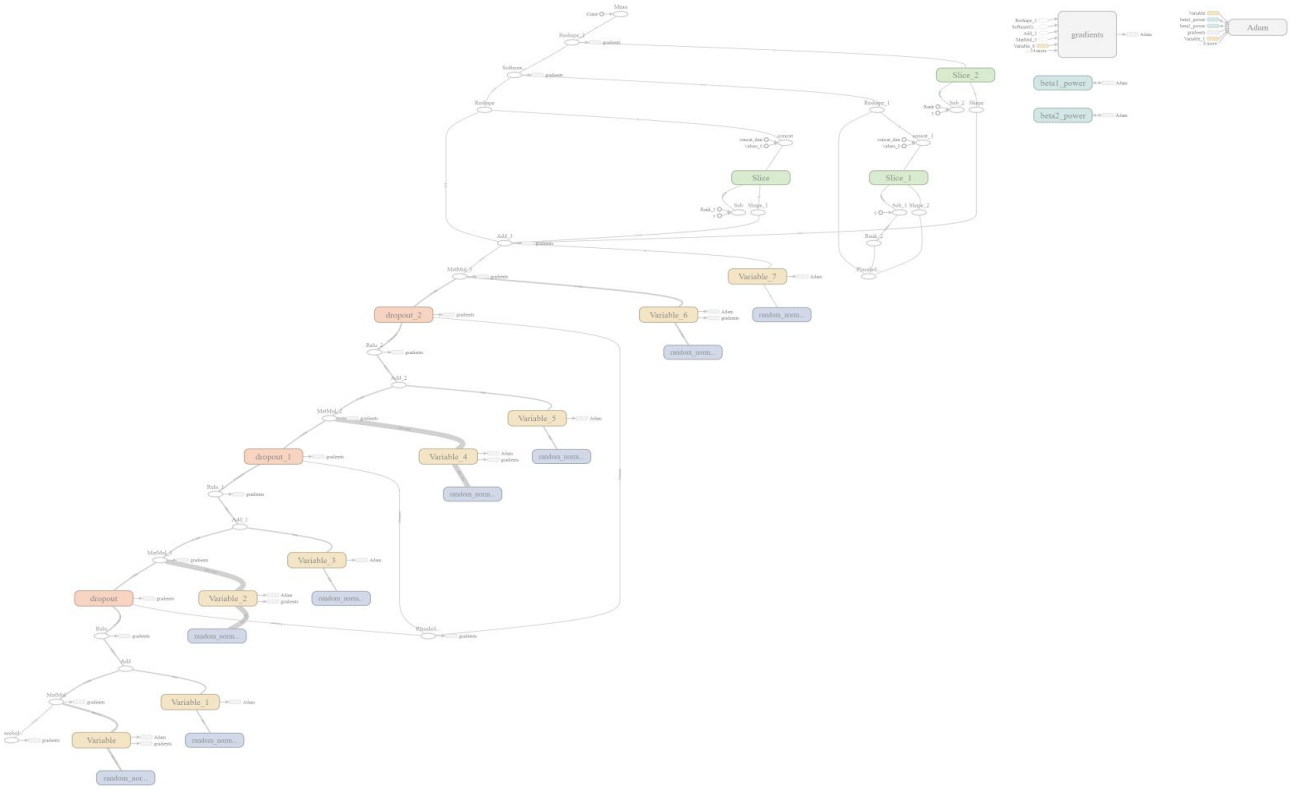
Description of the Simple Neural network constructed using Tensor flow

A five layer network was constructed with one input layer , three hidden layers with 500 nodes each and output layer of 3 nodes representing the classifier of posChange, negCHange and noChange

DropOut layer was added to each hidden layer

The Neural network was trained for 500 epoch and the maximum accuracy obtained is 69.44%

SNN Graph from tensor board



Parameters Used

DropOut	BacthSize per epoch	NumberOfEpoch
0.9	5	200

SNN Training and Testing Accuracy from Tensorboard

Classifier using Convoluted neural network

Convoluted neural network is used commonly for image recognition . A CNN was developed to identify trends/patterns and predict market movements accordingly

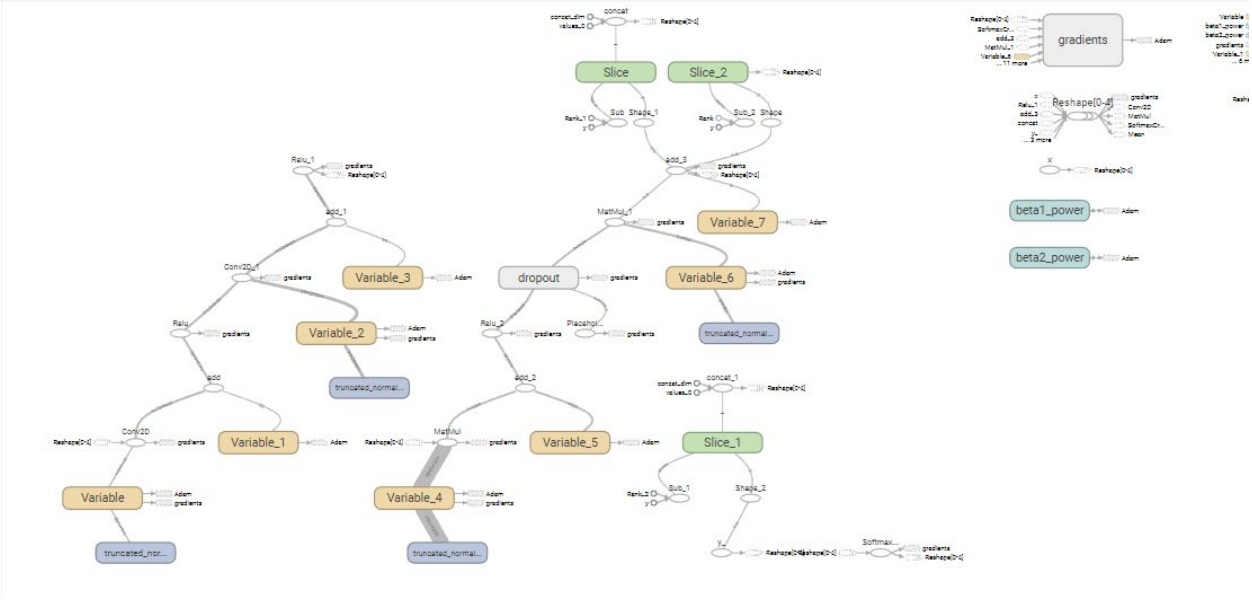
Description of the CNN constructed

Two convoluted layers were employed with each convoluted layer extracting 32 attributes and 64 attributes respectively of last five market data records .Maxpooling was not used because the problem is slightly different from image recognition . The output of the second convoluted layer was connected to dense layer(with dropout support) of 1024 nodes which was then connected to output layer of 3 Maximum accuracy obtained was 69.23%

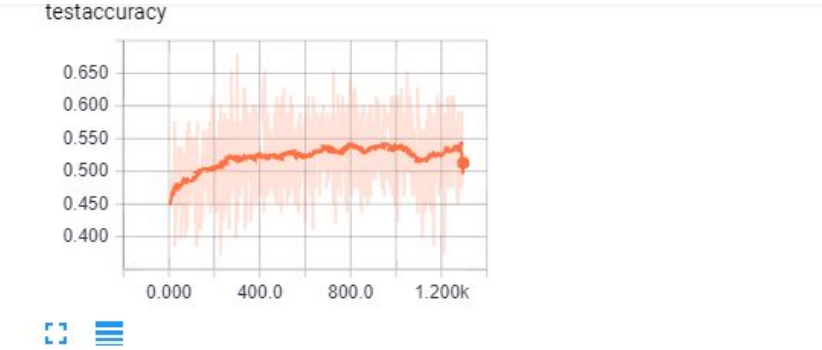
Parameters Used

DropOut	BacthSize per epoch	NumberOfEpoch
0.9	5	200

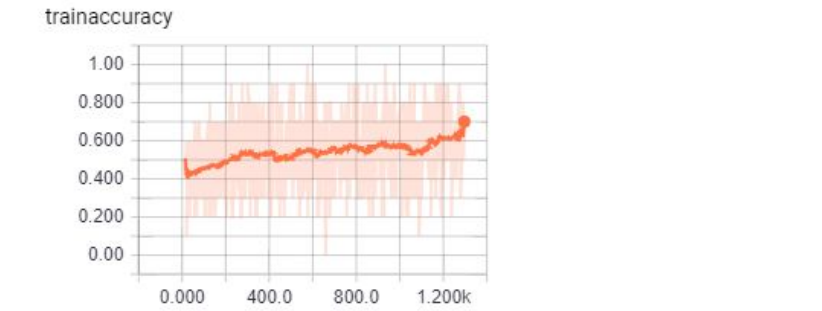
Graph from tensor board



CNN Training and Testing Accuracy from Tensorboard



trainaccuracy



Classifier using Recurrent Neural Network

Recurrent neural Network
A **recurrent neural network** (**RNN**) is a class of artificial neural network where connections between units form a directed cycle. This allows it to exhibit dynamic temporal behavior. Unlike feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs.
https://en.wikipedia.org/wiki/Recurrent_neural_network

A recurrent neural network can be used for time series prediction by retaining the memory of past events .

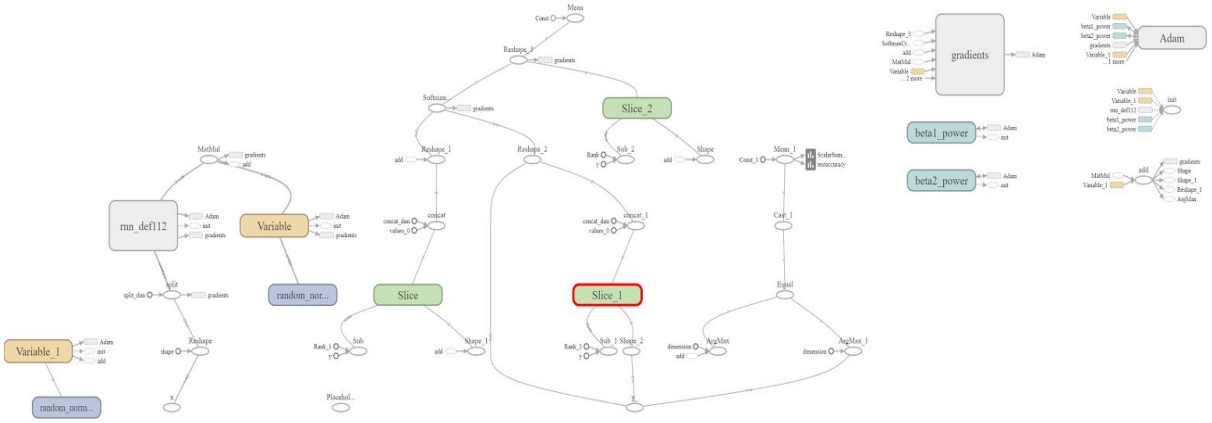
Description of Recurrent neural network implemented

A recurrent neural network with 128 LSTM cells was constructed and the output was fed to output layer with three nodes . The max efficiency achieved was 57%

Parameters Used

DropOut	BacthSize per epoch	NumberOfEpoch
0.9	5	200

RNN Graph from tensor board



RNN Training and Test Accuracy

Conclusion

The performance of various deep learning techniques for price movement prediction are as follows

Model	Accuracy

Simple Neural Network	**69.44%**
CNN	**69.23%**
RNN	**57%**

I was expecting RNN to have a better performance considering that it is a time series prediction. However, RNN as expected from theoretical study has proven to be unstable because of error amplification because outputs are inputs for the next tick and difficult to converge .Simple neural network and CNN have performed similar

Further Study

1) The networks will have to be trained for more data set which has to be obtained by paying a subscription fee
2) The training time can be optimized further using GPU capabilities of tensorflow
3) A real time market signal can be set up with high performance computation power(GPU) to train online and provide real time signals with minimal latency

References

https://en.wikipedia.org/wiki/Market_microstructure
http://www.sciencedirect.com/science/article/pii/S2405918815300179
http://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/
https://medium.com/towards-data-science/understanding-recurrent-neural-networks-the-preferred-neural-network-for-time-series-data-7d856c21b759
https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/
http://cs231n.github.io/understanding-cnn/
https://www.tensorflow.org/tutorials/