Market Microstructure Signals for Microsoft 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

Strategy for solving the problem

I plan on developing a multi class classifier to solve the problem of price movement prediction. The aim is to develop a deep learning based multi class classifiers .I plan to use deep learning though there are classical techniques for multi class classification like LQA,SVM and RandonForest so that the deep learning solution can be extended further for online real time learning using computational power of GPU where the model can adapt to changing market microstructure.

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

Descriptive statistic of Microsoft Data

	OpenPrice	HighPrice	LowPrice	TotalVolume	priceChange	movingAverage	upperBB	lowerBB	movingAverage5	upperBB5	lowerBB5	Chaikin
count	356.000000	356.000000	356.000000	3.560000e+02	356.000000	356.000000	356.000000	356.000000	356.000000	356.000000	356.000000	3.560000e+02
mean	53.212107	53.232051	53.191573	2.886327e+06	-0.001517	53.224635	53.279744	53.169584	53.213584	53.240654	53.186514	-1.274745e+04
std	0.151454	0.148732	0.153351	1.697601e+06	0.030060	0.135980	0.135353	0.141106	0.148928	0.146762	0.152321	8.609910e+05
min	52.850000	52.900000	52.850000	1.508576e+05	-0.090000	52.964000	53.011000	52.911000	52.898000	52.927000	52.865000	-2.680647e+06
25%	53.100000	53.120000	53.067500	1.673095e+06	-0.020000	53.166250	53.207000	53.117000	53.087500	53.129000	53.051750	-4.751808e+05
50%	53.225000	53.240000	53.210000	2.470828e+06	0.000000	53.240000	53.292000	53.196000	53.225000	53.250000	53.200000	2.406040e+04
75%	53.340000	53.360000	53.320000	3.684638e+06	0.020000	53.349250	53.400000	53.309000	53.344000	53.369000	53.319250	5.274113e+05
max	53.520000	53.520000	53.470000	8.938364e+06	0.090000	53.409000	53.458000	53.377000	53.466000	53.506000	53.430000	2.327377e+06

Covariance Matrix

	OpenPrice	HighPrice	LowPrice	TotalVolume	priceChange	movingAverage	upperBB	lowerBB	movingAverage5	upperBB5	lowerBB5	Chaikin
OpenPrice	1.000000	0.991073	0.992109	-0.354858	-0.077751	0.881564	0.871194	0.863522	0.987870	0.982638	0.984945	-0.080823
HighPrice	0.991073	1.000000	0.992661	-0.335710	0.024640	0.874924	0.868164	0.853638	0.983439	0.980587	0.978256	-0.030580
LowPrice	0.992109	0.992661	1.000000	-0.391638	0.020196	0.881548	0.869467	0.865125	0.984828	0.977880	0.983581	-0.027562
TotalVolume	-0.354858	-0.335710	-0.391638	1.000000	-0.067439	-0.349466	-0.288162	-0.396973	-0.365440	-0.334954	-0.416480	-0.084983
priceChange	-0.077751	0.024640	0.020196	-0.067439	1.000000	-0.031126	-0.027809	-0.033148	-0.042174	-0.041648	-0.042340	0.664222
movingAverage	0.881564	0.874924	0.881548	-0.349466	-0.031126	1.000000	0.983117	0.984489	0.905205	0.896377	0.906411	-0.008353
upperBB	0.871194	0.868164	0.869467	-0.288162	-0.027809	0.983117	1.000000	0.935777	0.895236	0.892003	0.891132	-0.010032
lowerBB	0.863522	0.853638	0.865125	-0.396973	-0.033148	0.984489	0.935777	1.000000	0.886047	0.872163	0.892279	-0.006390
movingAverage5	0.987870	0.983439	0.984828	-0.365440	-0.042174	0.905205	0.895236	0.886047	1.000000	0.995737	0.996043	-0.093875
upperBB5	0.982638	0.980587	0.977880	-0.334954	-0.041648	0.896377	0.892003	0.872163	0.995737	1.000000	0.983600	-0.101964

lowerBB5	**0.984945**	**0.978256**	**0.983581**	**-0.391867**	**-0.042340**	**0.906411**	**0.891132**	**0.892279**	**0.996043**	**0.983600**	**1.000000**	**-0.085324**
Chaikin	**-0.080823**	**-0.030580**	**-0.027562**	**-0.084983**	**0.664222**	**-0.008353**	**-0.010032**	**-0.006390**	**-0.093875**	**-0.101964**	**-0.085324**	**1.000000**

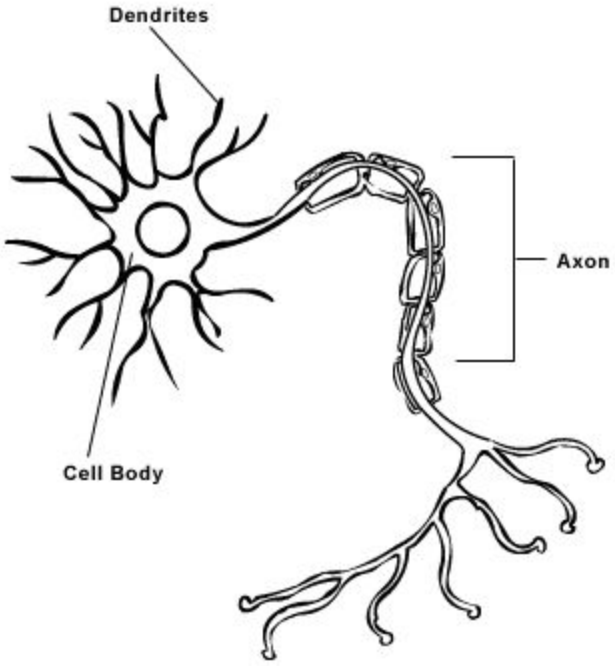
Highlights of Data Statistics

- There is no outliers for outliers with z score greater than 3.The sample data was from reliable data source
- Chaikin oscillator(reflects the traded volume) and price change are correlated which is in line with supply and demand theory

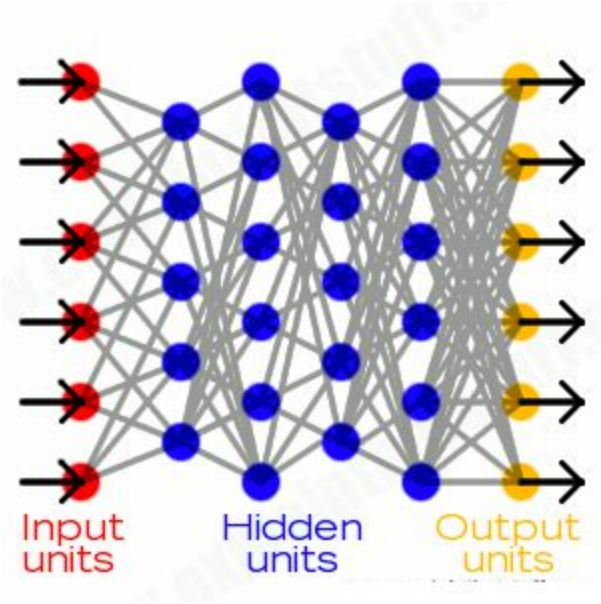
Brief description of the Deep learning algorithms

Simple Neural Networks

The basic idea of neural network is to simulate densely connected brain cells so that it can lean, recognize and make decisions like human. The neuron fires only if the signals received from various other neurons at the dendrites is above a certain threshold or activation level . The signal transmission is dependent on the capacity of dendrites to carry the signal and it is affected by a process called synapses which is analogous to learning process of ANN



Software representation of Network



Learning Process
The network learns as the input data is fed through the network and output observed . This is called feed forward . The output is compared to expected out expected output and the weights are readjusted based on the error at every node . This is called backward propagation .Simple neural networks are stateless and does not carry any state information

Advantages of Neural Network

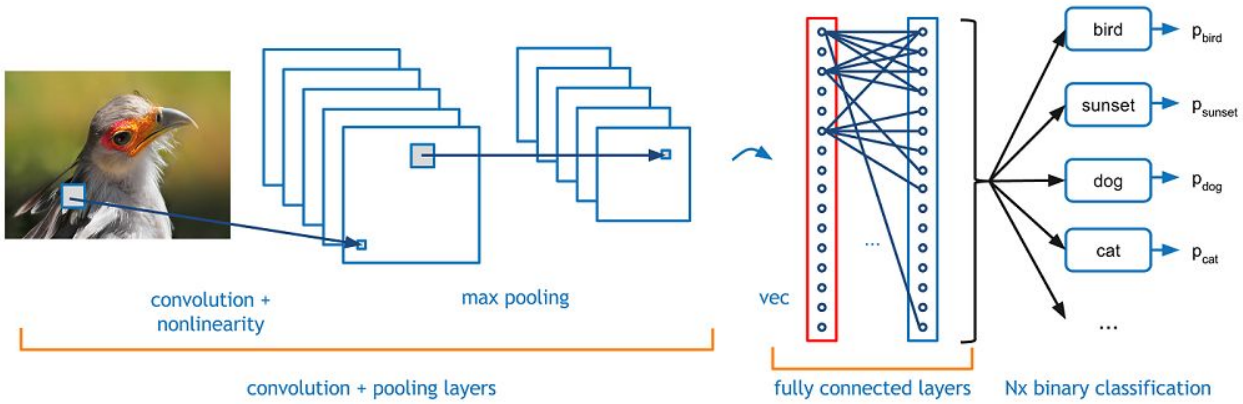
NN applies to problems that are nonlinear and difficult to model using conventional techniques.

Limitations

Training is computationally intensive and backward propagation behave like black boxes and require multiple epochs to train

Convoluted Neural Networks

CNN tries to mimic the visual recognition capacity of the human brain. Example would be when recognizing a cat , where low level features such as edges and curves such as paws and number of legs are identified and building more abstract concepts through a series of steps .
Intuitively, though not exactly CNN can be visualized as PCA layer before Simple Neural Network layer.There is sufficient research to prove that CNN is better than PCA
Example Image (Ref provided below)



Recurrent Neural Networks

In simple neural network , the inputs are independent of outputs . In RNN, the idea is to make use of sequential information . The output of first sequence is fed as input to the next sequence and hence used in time series predictions .LSTM and Echo networks are special cases of RNN

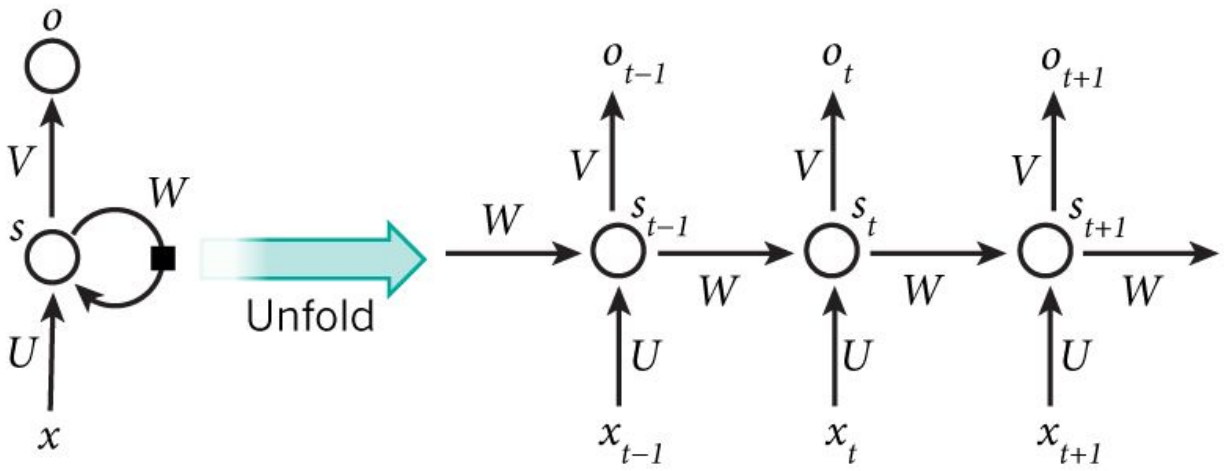
Advantages

They can be used in time series or character sequence prediction

Disadvantages

In RNN , the outputs are inputs to the next sequence and hence leads to noise amplification and leads to difficulty in training the network

Picture of RNN(Ref below)



Benchmark

Target accuracy based on similar work done and also considering the limitation of free data was set to 65%
References
http://jonathankinlay.com/2016/08/machine-learning-model-spy/
https://www.hindawi.com/journals/ads/2009/125308/

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

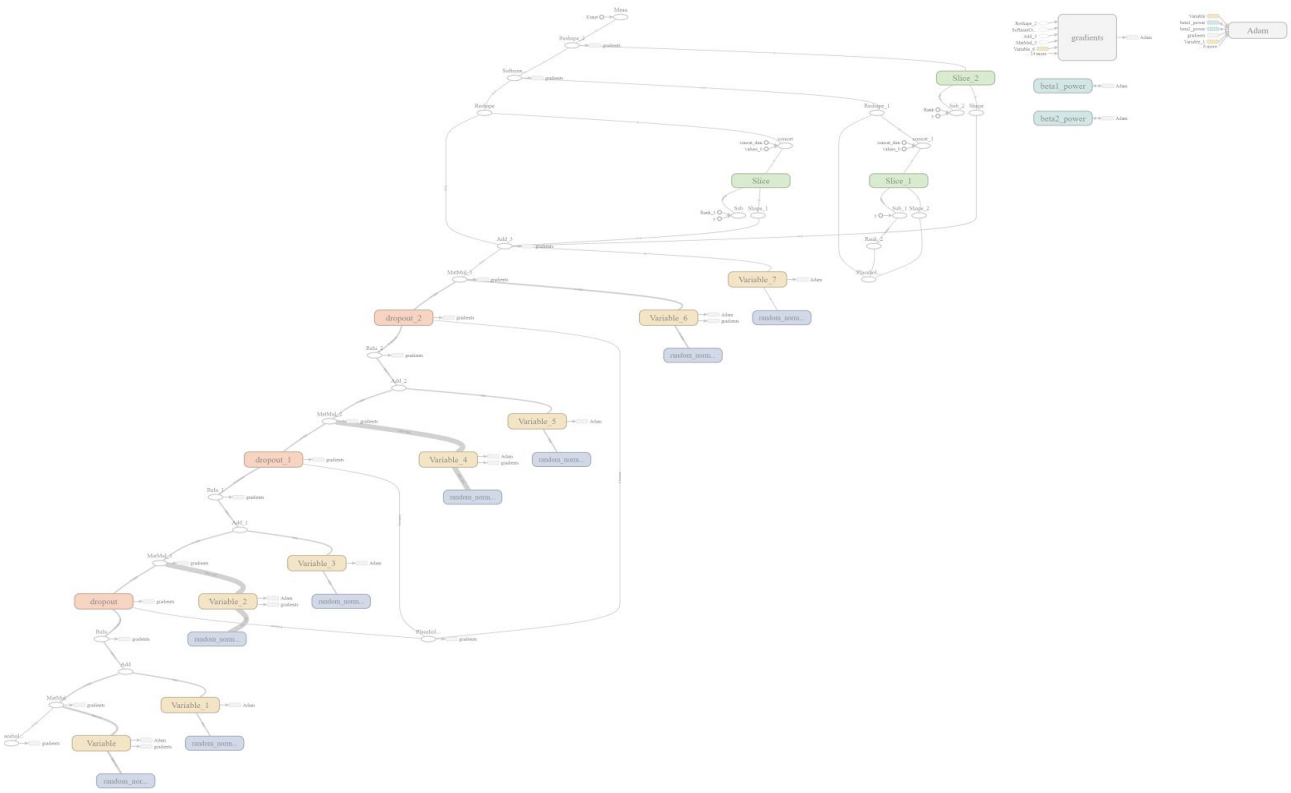
Description of the Simple Neural network constructed using Tensor flow

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%

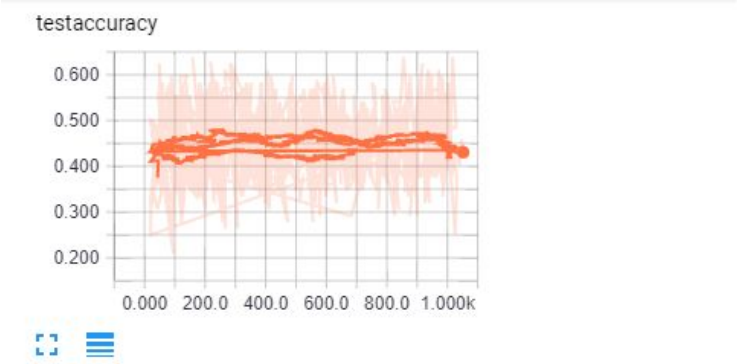
SNN Graph from tensor board



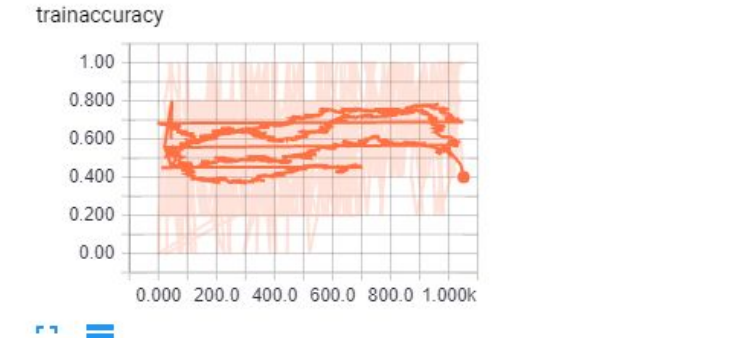
Parameters Used

DropOut	BacthSize per epoch	NumberOfEpoch
0.9	5	200

SNN Training and Testing Accuracy from Tensorboard



trainaccuracy



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

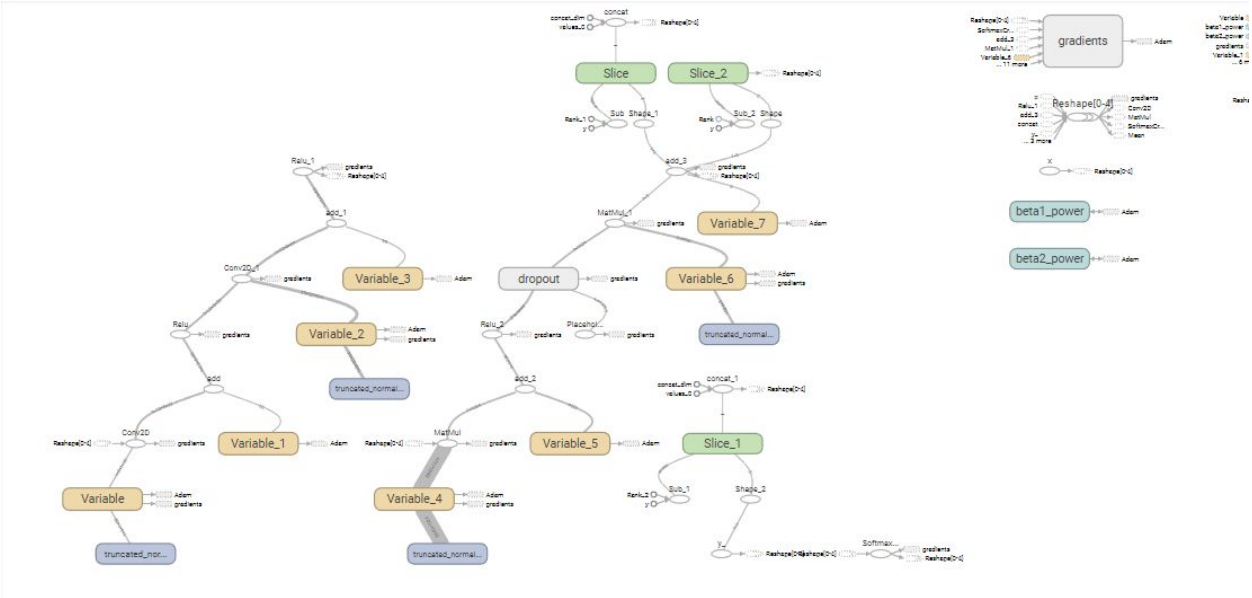
Description of the CNN constructed

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%

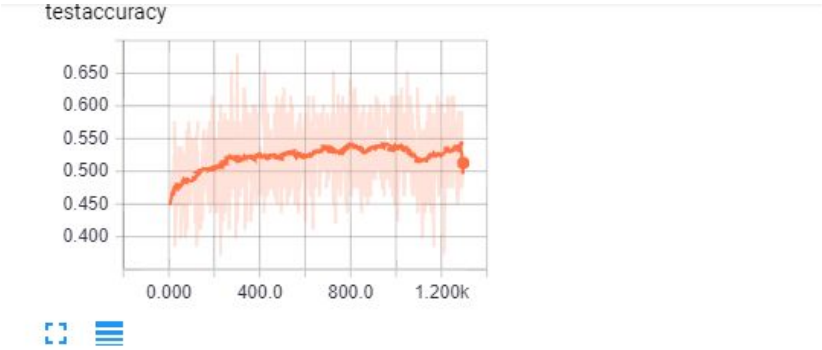
Parameters Used

DropOut	BacthSize per epoch	NumberOfEpoch
0.9	5	200

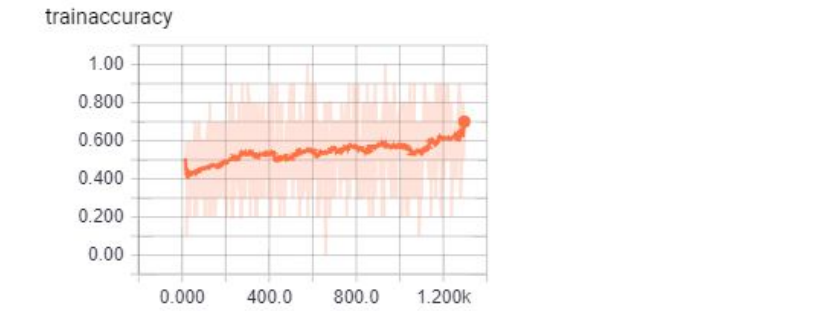
Graph from tensor board



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 .

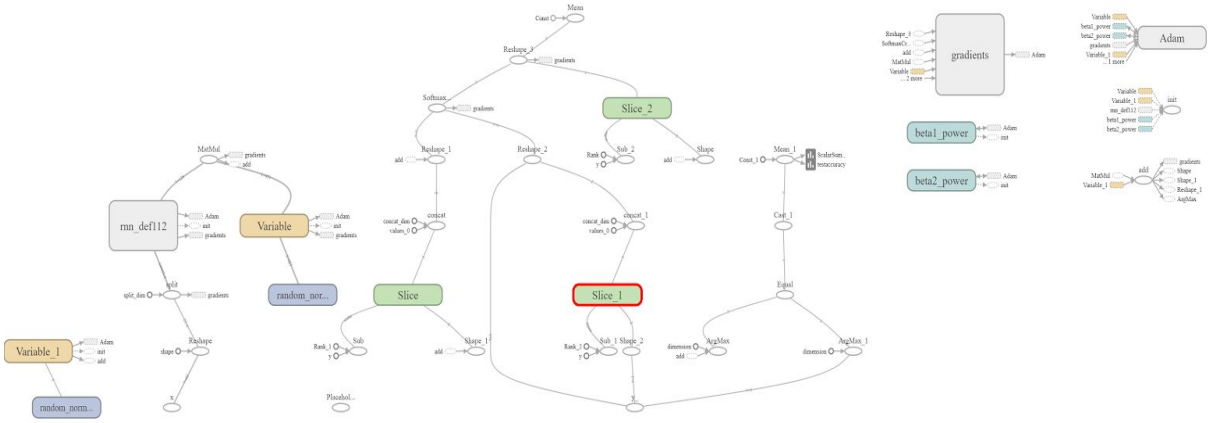
Description of Recurrent neural network implemented

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

Comparison to the Benchmark

SNN and CNN performed better than the targeted benchmark .However , quite contrary to the initial expectation, RNN's performance was suboptimal.

Model Robustness

The performance of all the models developed are repeatable. The repeatability of the performance of the models gives enough confidence that the models developed can be trusted . However, one of the disadvantages of deep learning models is that deep learning model are in a sense black boxes .

Challenges/Learning experience from the project

A significant amount of time was spent understanding tensor flow API 's than core deep learning algorithms. Tensordlow API's is not intuitive like other machine learning packages . However, tensor flow does offer very good performance and support for parallel processing.

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-preferd-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/

https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/
http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/