Stock Market Prediction Using Deep Learning

Prateek Yadav
Department of Mathematics
Indian Institute of Science, Bangalore
prateek@ug.iisc.in

Abstract

In this report we wanted to give our reader a general overview of what a stock market is, How it works, Various modelling techniques and the kind of thing that are currently being used in the real world by professional traders and big trading firms to Predict stock prices. This Report is pretty much self contained and will introduce the reader to basic terms and mathematical formulations of the model.

1 Introduction

In this Report We have done a literature survey of the General technique and principles of Stock Market. As we know that stock market is highly unpredictable and modelled using Stochastic processes mainly Brownian motion. Even though the process is random we want to predict future prices, for this purpose we will introduce various technique of time series analysis and text mining to create appropriate models to reduce the random noise and predict more effectively. Some researcher believe that the stock market can be modelled up to a certain extent and the process is not completely random, so we continue with this belief and model time series, public opinions, News information, Social Network sentiments. Deep learning is gaining a lot of popularity in the machine learning community. It consists a set of learning algorithms designed to train so called artificial neural networks an area of research in machine learning and artificial intelligence (AI). Neural networks are hard to train if they become too complex. We will also see some Technique which use deep architectures to model social sentiment and to extract out event embedding from News Corpus. We will begin by introducing the reader to the basics of stock market.

2 Stock Trading: Basics

This section is to introduce the reader with the fundamentals of Stock Market, General principals, will introduce with various resources available for Stock Market analysis and Prediction.

2.1 Definition of Stock Price

In finance, a stock represents a share in the ownership of an incorporated company. Stocks are evidences of ownership, or equity. Investors buy stocks in the hope that it will yield income from dividends and appreciate, or grow, in value. Shares of widely held companies are traded on stocks markets. Stock holding is popular because stocks represent ownership of capital that can be easily transferred by means of organized trading in the stock markets.

2.2 Efficient Market Hypothesis(EMH)

In financial markets the dynamics of stock prices are reflected by uncertain movements of their value over time. One possible reason for the random behavior of the asset price is the efficient market hypothesis (EMH).The EMH basically states two things:

1. The past history of a stock price is fully reflected in present prices.
2. The markets respond immediately to any new information about the stock.

These two assumptions imply that changes in the stock price are a Markov process. This means that the expected future value of a stock depends only on its current price. Predictions remain uncertain and may be only expressed in term of probability distribution.
In this context, modelling the stock price is concerned with modelling the arrival of new information, which affects the price. Two important things to retain are:

1. Probability distribution
2. Information

These play a major role in the modelling of future stock prices. In other words, the future price of a stock can be predicted within a certain level of exactitude, if one can anticipate new information about the stock.

2.3 Why Stock Prices Fluctuates

Stock Price of any company inherently depends on several factors, including Interest Rates, Demographics Changes, Economy of the country/world and Fundamentals of the company, but like any other market stock prices also primarily depends on the principle of **"Supply and Demand"**. When there is high demand and less supply of some stock then the price of that stock increases, while if the demand is less and supply is high, then the stock price decreases. In simple word if more people tend to sell the price falls and vice-versa.
The fluctuations in the stock price shows the same behavior as a stochastic process called **"Brownian Motion"**, thus we can import the properties of Brownian motions for the analysis of stock price movements.

2.4 Forecasting and Stock Market Analysis Methods

1. *Fundamental Analysis* is based on the study of the economy, field and society state, with a view to determining the value of the share of a specific company. Fundamental analysis monitors the profits of the company and the dividends that the company offers, takes into account the expectations about interest rates, and evaluates the risk associated with the company. It uses statistical, mathematical and financial algorithms, applied to the official periodic financial statements of the company, in order to evaluate, as correctly as possible the shares price.

2. *Technical Analysis* is based exclusively on the study of the internal data of the stock market, considering that all the economic, financial, political and psychological factors are incorporated into a single element: the share quotation. The technical analysts study the short-term changes of the shares price, starting with a study of the history of the quotations, within an interval of at least 6 months, and assume that the past behavior will extend into the future. The technical analysis offers information about the possible future evolution of the stock market.

2.5 Other Useful Things

1. **Data:** The time series data of all the companies is available for over past 100 years and all other information like interest rate, Inflation, Economic growth is available on several website including Google Finance, Yahoo Finance, Bloomberg and Reuters which can be easily imported.
2. **Standard Poor's 500 index:** It contains 500 USA companies which are listed on NYSE, from various sectors which are selected on the basis of market capitalization, liquidity, domicile, public float, sector classification. It is an index which show that how an economy is doing as a whole, If the weighted average stock price of these company go up then we can say that economy is doing good as these 500 companies are representative of all the different sectors and are very diversified. Basically Standard Poor's intention is to have a

price that provides a quick look at the stock market and economy. There is another index known as Dow Jones Industrial Average(DJIA) which is also for similar purposes but SP 500 is more popular and useful in most cases.

3 Mathematical Formulation of Time Series

The time series model takes into consideration the past behavior of a given variable and uses this information to predict its future behavior. It considers that there is no information on the causality relationship, which affects the variable that has to be forecast. Generally speaking, a time series model will be preferred when:

1. We have little information on the factors that affect the behavior of the variable.
2. The main aim is short-term prediction.
3. We have lots of data.

The time series analysis starts by building a data model whose properties are similar to the generation of the analyzed process. If we suppose that the properties of the analyzed process, which in Stock data case turns out to be Brownian motion, is included in the model and will continue into the future then the model can be used for prediction.
The extrapolation with time series differs from simple extrapolation. The difference occurs because the time series analysis presupposes that the series whose behavior has to be forecast was generated by a stochastic process, whose structure can be characterized and described. In other words, a time series model gives a description of the nature of the process that generated the time series. The description is not carried out in terms of cause-effect, as in the case of the regression model, but in terms of the form in which the event is incorporated into the process.
The traditional methods of time series analysis suppose that the series are composed of four elements: *the tendency*, *the cyclic component*, *the seasonal component*, and *random changes*. The first three of the above-mentioned components are deterministic, systematic, whereas the last one is a residual component that provides the analyzed phenomena with the feature of a stochastic process.
The roles of the components in the forecast process are different, depending on the length of the time interval for which the forecast is carried out. In short-term forecasts, the residual component has a major importance. In long-term forecasts, the most important one is the trend component.
Now we are going to Mathematically model stock price data using stochastic processes.

3.1 Random Walks

The **Random Walk Hypothesis** states that the stock market prices changes according to Random Walks.
A random walk, sometimes also called a *drunkards walk*, is the first step in understanding the Brownian motion. A random walk is a formalization of the intuitive idea of taking successive steps. The simplest random walk is a path constructed according to the following rules:
For an integer $n, n > 0$, we define the Random Walk process at the time $t\ W_n(t), t > 0$as follows:

1. The initial value of the process is: $Wn(0) = 0$
2. The layer spacing between tow successive jumps is equal to $1/n$
3. The *Up* and *down* jumps are equal and of size $1/n$, with equal probability.

In other words, if we consider a sequence of independent binomial variable X_i taking values $+1$ or 1 with equal probability , then the value of the random walk at the i^th step is defined recursively as follows:
$W_n(i/n) = W_n(i - 1/n) + X_t/\sqrt{n}$ for all $i >= 1$

3.2 Brownian Motion

The Brownian motion is a fundamental process that serves as a part of many different processes. This mathematical models is used to describe random movements. It is the scaling limit of random walk in one dimension as the time steps go to zero i.e. the number of steps becomes large.

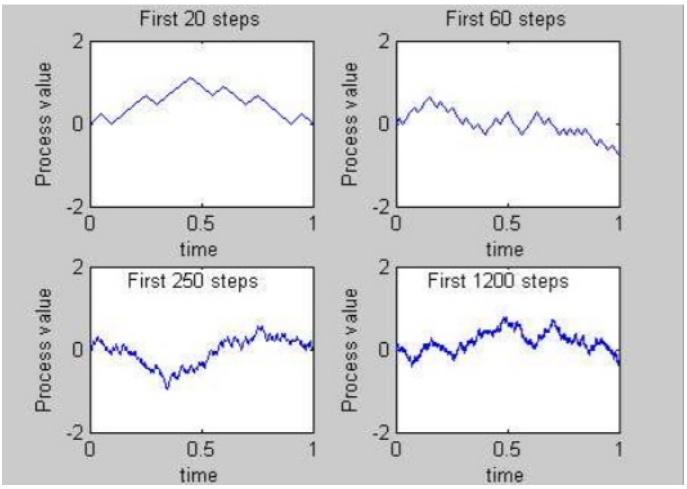


Figure 1: A demonstration how increasing steps in random walk results in a Brownian motion sort of figure.

3.2.1 Properties

We refer to the Brownian motion as a process B_t. The Brownian motion has the following properties:

1. *Continuity:* B_t has a continuous path and $B(t=0) = 0$
2. *Normality:* The increment of the Brownian process in the interval of time of length t between the two moments s and $s+t$ is $B_{s+t} - B_s$. This increment is normally distributed with mean zero and variance equal to the time increment t.
   $B_{s+t} - B_s \sim N(0, t)$.
3. *Markov property:* The conditional distribution $B(t)$ given information up to time $s < t$ depends only on the $B(s)$.

Now we can model stock price data using something called *Brownian motion with Drift*, which is a stochastic process B_t. For given constants μ and σ the process has the following form$B_t = \mu t + \sigma W_t$ Where t represents time and W_t is a random walk process as described in the previously.

4 Stock Prediction: Techniques

4.1 Time Series Analysis

4.1.1 Stationary Process

A time series is said to be *Strictly stationary* if the joint distribution of $X(t_1), X(t_2), \ldots, X(t_k)$ is the same as the joint distribution of $X(t_1+\tau), X(t_2+\tau), \ldots, X(t_k+\tau)$ for all $t_1, t_2, \ldots, t_k, \tau$. In other words, shifting the time origin by an amount τ has no effect on the joint distribution, which must therefore depend only on the intervals between $t_1, t_2, \ldots, t_k$.
The time difference $(t_i - t_{i-1}) = \tau$, is called the ***lag***.
Note that a purely random process is sometimes called ***White noise***.

4.1.2 Relative Strength Index (RSI)

The *relative strength index (RSI)* is a momentum indicator, that compares the magnitude of recent gains and losses over a specified time period to measure speed and change of price movements of a security. It is primarily used to attempt to identify overbought or oversold conditions in the trading of an asset.

4.1.3 Fibonacci Retracement Method

Fibonacci retracement is created by taking two extreme points (usually a major peak and trough) on a stock chart and dividing the vertical distance by the key Fibonacci ratios of 23.6%, 38.2%, 50%, 61.8% and 100%. Once these levels are identified, horizontal lines are drawn and used to identify possible support and resistance levels.
Support Level: A support level is a level where the price tends to find support as it falls. This means the price is more likely to "bounce" off this level rather than break through it. However, once the price has breached this level, by an amount exceeding some noise, it is likely to continue falling until meeting another support level.
Resistance Level: A resistance level is the opposite of a support level. It is where the price tends to find resistance as it rises. This means the price is more likely to "bounce" off this level rather than break through it. However, once the price has breached this level, by an amount exceeding some noise, it is likely to continue rising until meeting another resistance level.

4.1.4 Moving Averages

A widely used indicator in technical analysis that helps smooth out price action by filtering out the *noise* from random price fluctuations. A **moving average** (MA) is a trend-following or lagging indicator because it is based on past prices. The two basic and commonly used MA's are the *simple moving average* (SMA), which is the simple average of a security over a defined number of time periods, and the *exponential moving average* (EMA), which gives bigger weight to more recent prices. The most common applications of MA's are to identify the trend direction and to determine support and resistance levels.
There are various versions of moving averages that are used: arithmetic moving averages, geometric moving averages, harmonic moving averages, quadratic moving averages, weighted moving averages, and double moving averages. The moving average models are certainly useful, but they offer no information about the trust in forecast, and do not explain the random behavior of the time series. The stochastic models are therefore necessary, because the random component brings information about forecast errors.

Mathematical Formulation:
Suppose that Z_t is a purely random process with mean zero and variance σ_Z^2. Then a process X_t is said to be an moving averages process of order q (abbreviated to an $MA(q)$) if,

$$X_t = \beta_0 Z_t + \beta_1 Z_{t-1} + \ldots + \beta_q Z_{t-q}$$

where β_i are constants. The $z's$ are usually scaled so that $\beta_0 = 1$.

4.1.5 Moving average convergence divergence (MACD)

Moving average convergence divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of prices. The MACD is calculated by subtracting the 26-day exponential moving average (EMA) from the 12-day EMA. A nine-day EMA of the MACD, called the "signal line", is then plotted on top of the MACD, functioning as a trigger for buy and sell signals.

4.1.6 AutoRegressive Models

AutoRegressive is a stochastic process used in statistical calculations in which future values are estimated based on a weighted sum of past values. An AutoRegressive process operates under the premise that past values have an effect on current values.

Mathematical Formulation:

Suppose that Z_t is a purely random process with mean zero and variance σ_Z^2. Then a process X_t is said to be an AutoRegressive process of order p (abbreviated to an $AR(p)$) if,

$$X_t = \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} + Z_t$$

This is rather like a multiple regression model, but X_t is regressed on past values of X_t rather than on separate predictor variables. A process considered $AR(1)$ is the first order process also known as *Markov Process*, meaning that the current value is based on the immediately preceding value. An $AR(2)$ process has the current value based on the previous two values.

4.1.7 Mixed AutoRegressive - Moving Averages (ARMA) Models

There are many random processes for which it is impossible to build a universally valid model through either the moving average or the purely AutoRegressive model. By integrating the AutoRegressive and moving average models, the ARMA (Mixed AutoRegressive Moving Average Models) model was obtained. This is deemed to be the most suitable one for economic forecasts, when the evolution of the exogenous variables is unknown.

Mathematical Formulation:
A mixed AutoRegressive-moving average process containing p AR terms and q MA terms iss said to be an ARMA of order (p, q), given by:

$$X_t = \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} + Z_t + \beta_0 Z_t + \beta_1 Z_{t-1} + \ldots + \beta_q Z_{t-q}$$

The importance of ARMA is process lies in the fact that a stationary time series may often be adequately modelled by an ARMA model involving fewer parameters than a pure MA and AR process by itself.

4.1.8 AutoRegressive Integrated Moving Averages (ARIMA) Models

In most cases time series are not stationary. In order to fit a stationary models discussed above, it is necessary to remove non-stationary sources of variation. If the observed time series is not stationary in the mean then we can difference the series. Differencing is widely used, if X_t is replaced by $\nabla^d X_t$ in above equations, then the model we obtain are capable of describing certain type of non-stationary series. Such models are called integrated models.

Mathematical Formulation:
Writing general AutoRegressive Integrated Moving Averages model with parameters p, q, d as $ARIMA(p, q, d)$.

$$W_t = \nabla^d X_t$$

$$W_t = \alpha_1 W_{t-1} + \ldots + \alpha_p W_{t-p} + Z_t + \beta_0 Z_t + \beta_1 Z_{t-1} + \ldots + \beta_q Z_{t-q}$$

4.1.9 Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) Process

The *generalized AutoRegressive conditional heteroskedasticity (GARCH)* process describes an approach to estimate volatility in financial markets. There are several forms of GARCH modeling, The general process for a GARCH model involves three steps. The first is to estimate a best-fitting AutoRegressive model. The second is to compute autocorrelations of the error term. The third is to test for significance.
Mathematical formulation of GARCH is a very nice one but a bit long to describe it here. It is available on Wikipedia and many other sources, but the main idea is that ARCH or GARCH tries to model the Variance of a random process using ANOVA

4.1.10 Granger Causality

The Granger's Causality test is a statistical hypothesis test which helps in determining whether a time series is useful in predicting future value of another time series. It's mathematical formulation

is based on linear regression model for stochastic processes. Ordinary regression reflects just correlation but causality can be tested by measuring the ability to predict the future values of a time series using prior values of another time series. We say that a variable X that evolves over time *Granger-causes* another evolving variable Y if predictions of the value of Y based on its own past values and on the past values of X are better than predictions of Y based only on its own past values. There are two underlying principles for causal relationships:

1. The cause happens prior to its effect.
2. The cause has unique information about the future values of its effect.

Mathematical Formulation:
The test of identification of causal effect of X on Y is given by,

$$P[Y_{t+1} \epsilon A | I(t)] \neq P[Y_{t+1} \epsilon A | I_{-X}(t)]$$

where P is the probability, A is an arbitrary non-empty set, and $I(t)$ and $I_{-X}(t)$ respectively denote the information available as of time t in the entire universe, and that in the modified universe in which X is excluded. If the above hypothesis is accepted, we say that X Granger-causes Y.

4.1.11 General Discussion

The basic idea in time series analysis is that we have a time series and we need to predict it's future values using some mathematical analysis. So we know that time series has 4 components namely, trend, Seasonality, Cyclic component and the random changes/irregular components. So in order to predict accurate future values we need to get a hold on all of these components. we can decompose any time series in these components based on two models,

1. ***Additive Model:*** $y_t = T_t + C_t + S_t + I_t$, An additive model would be used when the variations around the trend does not vary with the level of the time series
2. ***Multiplicative Model:*** $y_t = T_t \times C_t \times S_t \times I_t$, multiplicative model would be appropriate if the trend is proportional to the level of the time series.

Where T is trend, C is cyclic component, S is seasonal component, I is the irregular component. So we can always assume such a decomposition exists or at least a reasonable approximation exists. Now to model the trend we need to get handles on the mean, as it represents general trend. Models like Moving Averages, MACD, ARIMA, while ARIMA being most popular and widely used for such purposes as they provide nice insight and are also very widely studied. Seasonality and Cyclic components remains the same over time and are unchanged in short term. Now the component i.e the random changes are modelled by us using Brownian motion. So as we know the random component is not predictable so we want to obtain certain bounds on it so as to predict a range in which future values are highly likely to lie in. The best model for this Purpose is the $GARCH$ there are several modifications to it but the main formulation remains the same. So now we have modelled the Mean and the variance of the time series, so we can effectively predict a range for future values.

Most of the above models are linear models, these models are not always sufficient to satisfactorily model real world phenomena. There is an increasing interest in developing non linear models for time series analysis as linear models are not able to identify higher order correlations. In order to capture higher order correlation we need to introduce some sort of non-linearity to our models to make better prediction, so people have developed non-linear versions of ARIMA and GARCH model to make them work even when we drop the linearity hypothesis. Other model include non-linear autoregressive processes, threshold models, bi-linear models. We can deploy even neural networks to model the non-linearity. Hidden Markov Models are also used to model time series data but we are not discussion HMM as they were done in class.
Some of the interesting features that the nonlinear models may offer are:

1. The prediction interval does not increase in time.
2. The distribution of forecast errors is not a normal one.

The above analysis was purely mathematical, but we know that there are other factors which affect stock prices of company, like Psychological factors, Public opinions, Expert Opinions, Supply demand, Big Financial news, Company specific new, all these thing are not included in out analysis. According to the EMH also stock market prices are largely driven by new information, i.e. news, rather than present and past prices. It would be great if we can get a hold of such information and model it effectively because that would basically reduce the random component ass it included such factors only. If such information is modelled we can obtain tighter bounds on variance of the time series as some of the random changes are no more random. In order to make use of such information like twitter tweets, News articles, expert opinions and Social Sentiment of stock, we make use of deep learning architecture that we will describe in the next section.

4.2 Stock Prediction: NLP Techniques

4.2.1 News Corpus Analysis

Overview
It has been shown that the financial market is informationally efficient [Fama, 1965] i.e. stock prices reflect all known information, and the price movement is in response to news or events. There may be positive news, negative news or news to which market may not react at all. For example, News event like: "Google has decided to cease operations in China" should have negative impact on the stock prices. On the other hand, Event like:"USA and CHINA has signed a 26 billion dollar business deal" should have positive impact on the market.Therefore, an investor often keep track of these news to predict their effect on stock market.There are certain news which has local impact while some news have global impact. Certain news can have so negative impact that it can overshadow the impact of positive news like, "Tsunami , Earthquake and Nuclear leak hits Japan.".

With the recent rise in Natural Language Processing techniques, we have got the power to find the correlation between these events and market movement.
Event Representation
In order to find the correlation between the news events and stock market prediction, We are, first, required to represent these events in some structured forms. some of the existing work , uses Bang-of-words model, Structured tuple representation and the most recent event Embeddings.

BOW model: This model assumes the whole collection of news article as a collection of tokens. This is the easiest of all the representation and gives good enough prediction. The results of such a system using BOW is provided in the result section. Major drawback of BOW model is that it neglects the semantic relationship between the actors present in the event. This results in reduced predictive power. For example, the event "Microsoft posted solid end-quarters result while Google fails" will be converted into collection of tokes ['Microsoft','Posted','Solid','End-quarter','results','Google','fails']. This doesn't describes who posted good results and who didn't.

Structured Tuple Representation: This representation represents a news event as a vector of actors ,action and object. For example, the event "Microsoft Sues Google" would be represented as [actor="Microsoft",action="Sues",object="Google"]. This representation recognizes the semantic meaning in the paper. However, this representation will treat two tuple [actor="Microsoft",action="posted",object="result"] and [actor="Google", action="plummets",object="shares"] differently. We require a presentation that would have similar representation for two very near events.

Event Embeddings: Event embeddings are dense vector that are trained in such a way that two vector(as shown above) will have similar representation , even if the words are different.This is achieved by passing the tuples as an input to a Neural Tensor Network(described in Model section).

4.2.2 Social Media Analysis

Overview
Behavioral economics tells us that emotions can profoundly affect individual behavior and decision-making. As described in the section 2, the state of market is highly dependent on how much investors showing interest in the market. Therefore, it would be appropriate to say that there does exist some correlation between emotion of investor and market movement. One way to tap the emotions of crowd is to analyse the behaviour of people on social platform such as Twitter, Facebook etc.It is

often said that social platforms are mirror showing the sentiment of crowd. In this age, people often express their emotions in some form or another on these social platforms. We can use these platform to analyze the emotion of crowd.

Emotion Representation
There are many techniques proposed in the recent years, One such technique uses LSTM (described in section 2) to find the sentiments of crowd. There is a paper by Johan Bollen,Huina Mao,Xiao-Jun Zeng that uses OpinionFinder and GPOMS to convert twitter Data into a a finite dimensional vector , each dimension representing one of the emotion [sad, happy, calm,alert,vital,sure]. These vectors then can be used as a time series to predict another Stock price time series or the can be used as an input to a NN to help predict stock price.

4.2.3 General Discussions

Sentiment analysis and News corpus analysis plays a very important role in predicting the stock price. This can be seen in the work done by Ding[2014]. However, Stock market sometime acts in a very unpredicted manner. The above mentioned techniques only form the smaller part of the data required to accurately predict the market. However, with future improvement in technology and increase in availability of data, We are expecting a better accuracy.

5 Deep Architectures

5.1 Long Short-term Memory Networks

Overview
Long short-term memory (LSTM) is a *recurrent neural network (RNN)* architecture (an artificial neural network) proposed in 1997 by Sepp Hochreiter and Jrgen Schmidhuber. A LSTM network is universal in the sense that given enough network units it can compute anything a conventional computer can compute, provided it has the proper weight matrix, which may be viewed as its program.

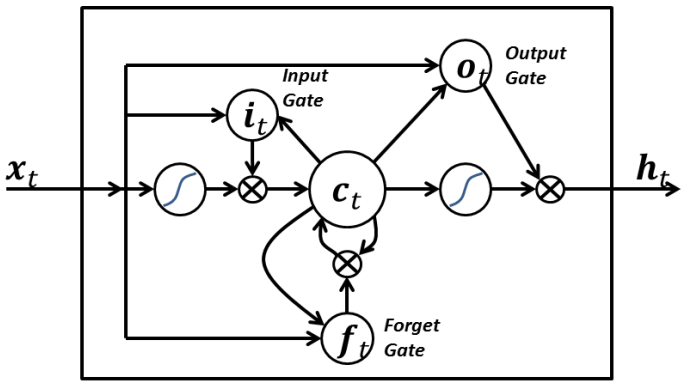


Figure 2: A simple LSTM block with only input, output, and forget gates. LSTM blocks may have more gates.

Architecture
A LSTM network is an artificial neural network that contains LSTM units instead of, or in addition to, other network units. A LSTM unit is a recurrent network unit that excels at remembering values for either long or short durations of time. The key to this ability is that it uses no activation function within its recurrent components. Thus, the stored value is not iteratively squashed over time, and the gradient or blame term does not tend to vanish when Backpropagation through time is applied to train it.

LSTM blocks contain three or four ”gates” that they use to control the flow of information into or out of their memory. These gates are implemented using the logistic function to compute a value

between 0 and 1. Multiplication is applied with this value to partially allow or deny information to flow into or out of the memory. For example, an "input gate" controls the extent to which a new value flows into the memory. A "forget gate" controls the extent to which a value remains in memory. And, an "output gate" controls the extent to which the value in memory is used to compute the output activation of the block.

Mathematical Formulation

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$
$$c_t = f_t o c_{t-1} + i_t o \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$
$$h_t = o_t o \sigma_h(c_t)$$

Variables:
x_t :input vector
h_t:output vector
c_t:cell state vector
$W, U \& b$: parameter matrices and vector
f_t : forgot gate vector
i_t : input gate vector
O_t : output gate vector

Activation function:
σ_g:sigmoid function
σ_c:hyperbolic tangent

5.2 Convolutional Deep Neural Network

Overview
A Convolutional Neural Network (CNN) is comprised of one or more convolutional layers (often with a subsampling step) and then followed by one or more fully connected layers as in a standard multilayer neural network. The architecture of a CNN is designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal). This is achieved with local connections and tied weights followed by some form of pooling which results in translation invariant features.

Architecture
A CNN consists of a number of convolutional and sub-sampling layers optionally followed by fully connected layers. The input to a convolutional layer is a m x m x r image where m is the height and width of the image and r is the number of channels, e.g. an RGB image has r=3. The convolutional layer will have k filters (or kernels) of size n x n x q where n is smaller than the dimension of the image and q can either be the same as the number of channels r or smaller and may vary for each kernel. The size of the filters gives rise to the locally connected structure which are each convolved with the image to produce k feature maps of size mn+1. Each map is then sub-sampled typically with mean or max pooling over p x p contiguous regions where p ranges between 2 for small images (e.g. MNIST) and is usually not more than 5 for larger inputs. Either before or after the sub-sampling layer an additive bias and sigmoidal nonlinearity is applied to each feature map. The figure below illustrates a full layer in a CNN consisting of convolutional and sub-sampling sub-layers. Units of the same color have tied weights.

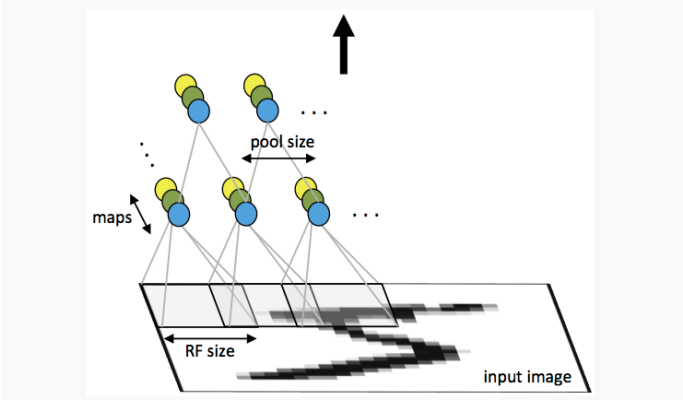


Figure 3: First layer of a convolutional neural network with pooling.

Backpropogation

Let $\delta^{(l+1)}$ be the error term for the $(l+1)^{th}$ layer in the network with a cost function $J(W,b;x,y)$ where (W,b) are the parameters and (x,y) are the training data and label pairs. If the l^{th} layer is densely connected to the $(l+1)^{th}$ layer, then the error for the l^{th} layer is computed as

$$\delta^{(l)} = ((W^{(l)})^{T(l+1)})f(z^{(l)}$$

and the gradients are

$$\Delta_{W^{(l)}} J(W,b;x,y) = \delta^{(l+1)}(a^{(l)})_T$$
$$\delta_{b^{(l)}} J(W,b;x,y) = \delta^{(l+1)}$$

If the l^{th} layer is a convolutional and sub-sampling layer then the error is propagated through as

$$\delta_k^{(l)} = upsample((W_k^{(l)})^T{}_k^{(l+1)})f(z_k^{(l)})$$

Where k indexes the filter number and $f'(z_k^{(l)})$ is the derivative of the activation function. The upsample operation has to propagate the error through the pooling layer by calculating the error w.r.t to each unit incoming to the pooling layer. For example, if we have mean pooling then upsample simply uniformly distributes the error for a single pooling unit among the units which feed into it in the previous layer. In max pooling the unit which was chosen as the max receives all the error since very small changes in input would perturb the result only through that unit.

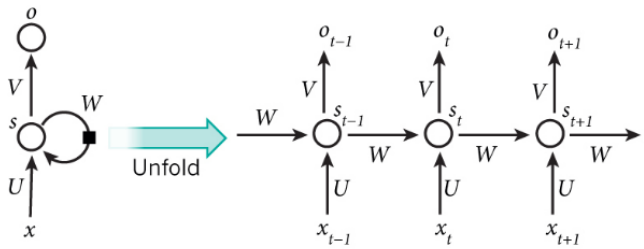
Finally, to calculate the gradient w.r.t to the filter maps, we rely on the border handling convolution operation again and flip the error matrix $\delta_k^{(l)}$ the same way we flip the filters in the convolutional layer.

$$\Delta_{W_k^{(l)}} J(W,b;x,y) = \sum_{i=1}^{m}(a_i^{(l)})rot90(\delta_k^{(l+1)},2)$$
$$\Delta_{b_k^{(l)}} J(W,b;x,y) = \sum_{a,b}\delta_k^{(l+1)}$$

Where $a^{(l)}$ is the input to the l-th layer, and $a^{(1)}$ is the input image. The operation $(a_i^{(l)})\delta_k^{(l+1)}$ is the valid convolution between i^{th} input in the l^{th} layer and the error w.r.t. the k^{th} filter.

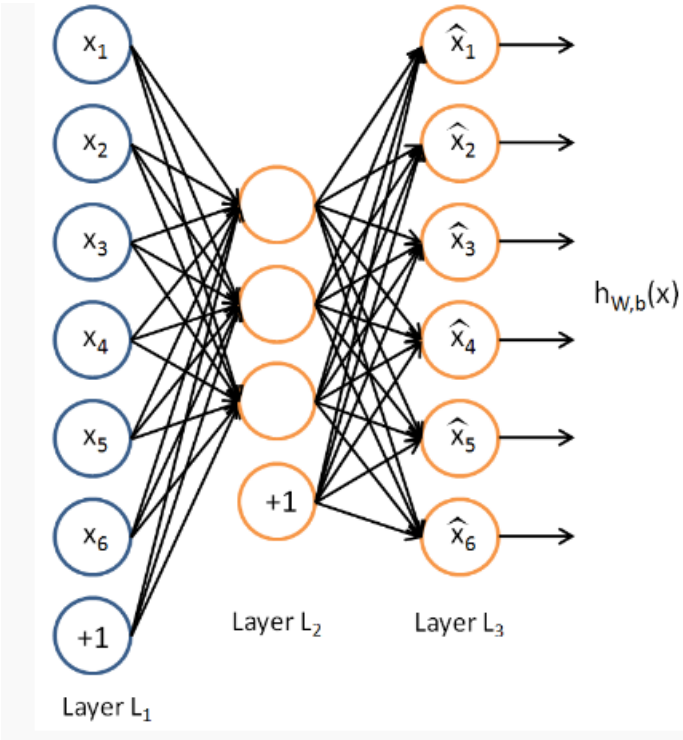
5.3 Other Networks

Recurrent Neural Network (RNNs)
A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. Unlike feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs. This makes them applicable to tasks such as unsegmented connected handwriting recognition[1] or speech recognition.[2]



Autoencoder

An autoencoder is an artificial neural network used for unsupervised learning of efficient codings. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for the purpose of dimensionality reduction. Recently, the autoencoder concept has become more widely used for learning generative models of data.



The autoencoder tries to learn a function $h_{W,b}(x)x$. In other words, it is trying to learn an approximation to the identity function, so as to output x^1 that is similar to x.

Deep Belief Networks (DBNs)

DBN are graphical models which learn to extract a deep hierarchical representation of the training data. They model the joint distribution between observed vector x and the l hidden layers h^k as follows:

$$P(x, h^1, ..., h^l) = (\prod_{k=0}^{l-2} P(h^k|h^{k+1}))P(h^{l-1}, h^l)$$

where $x = h^0, P(h^{k-1}|h^k)$ is a conditional distribution for the visible units conditioned on the hidden units of the RBM at level k, and $P(h^{l-1}, h^l)$ is the visible-hidden joint distribution in the top-level RBM.

6 Paper Summary

6.1 Stock Price Prediction: Corpus Analysis(Event Embeddings)

Title : Deep Learning for Event-Driven Stock Prediction
Author : Xiao Ding , Yue Zhang , Ting Liu , Junwen Duan

6.1.1 Abstract

In this paper, Authors have proposed a deep learning method for event-driven stock market prediction. First, events are extracted from news text, and represented as dense vectors, trained using a novel neural tensor network. Second, a deep convolutional neural network is used to model both short-term and long-term influences of events on stock price movements.

6.1.2 Introduction

It has been shown that the financial market is informationally efficient [Fama, 1965] stock prices reflect all known information, and the price movement is in response to news or events. Instead of using simplier models such as bags-of-words, noun phrases, and named entities [Kogan et al., 2009; Schumaker and Chen, 2009], this paper uses new representation called Event Embeddings to capture structured relations in the events. First , A more structured representation is extracted from the text using OPEN IE. For example, The structured representation of the news title " Microsoft sues Motorolla" would look like (Actor = Microsoft, Action = sues, Object = motorolla).

These structured representation have some relational flaws as mentioned in above section. Therefore, Author has come up with a novel Neural Tensor Network that produces Event embeddings. For predictive Model, authors are using a convolutional neural network (CNN) to perform semantic composition over the input event sequence, and a pooling layer to extract the most representative global features. Then a feed-forward neural network is used to associate the global features with stock trends through a shared hidden layer and a output layer.

6.1.3 Neural Tensor Network for Learning Event Embeddings

Event representation and Extraction

An event is represented as a tuple E = $(O1, P, O2, T)$,where P is the action, $O1 is the actor and O2$ is the object on which the action is performed. T is the timestamp of the event, which is mainly used for aligning stock data with news data. For example, the event Jan 13, 2014 - Google Acquires Smart Thermostat Maker Nest For for 3.2 billion dollars.is modeled as: (Actor = Google, Action = acquires, Object = Nest, Time = Jan 13, 2014).

Neural Tensor Network

The input of neural tensor network is word embeddings and the output is event embeddings. The initial word representation of d-dimensions (d = 100) from is learned using large-scale financial news corpus, using the skip-gram algorithm [Mikolov et al., 2013]. From Figure 1, $R_1 \in \mathrm{R}^d$ is computed by:

Algorithm 1: Event Embedding Training Process

Input: $\mathcal{E} = (E_1, E_2, \cdots, E_n)$ a set of event tuples; the model $EELM$

Output: updated model $EELM'$

random replace the event argument and got the corrupted event tuple

$\mathcal{E}^r \leftarrow (E_1^r, E_2^r, \cdots, E_n^r)$

while $\mathcal{E} \neq [\,]$ **do**

  $loss \leftarrow max(0, 1 - f(E_i) + f(E_i^r) + \lambda \|\mathbf{\Phi}\|_2^2$

  if $loss > 0$ **then**

    $Update(\mathbf{\Phi})$

  else

    $\mathcal{E} \leftarrow \mathcal{E}/\{E_i\}$

return $EELM$

where $T_1^{[1:k]} \in \mathrm{R}^{dXdXk}$ is a tensor, and the bilinear tensor product $O_1 T_1 P$ is a vector r $\in \mathrm{R}^k$, where each entry is computed by one slice of the tensor $(r_i = O_1^T T_1^{[i]} P, i = (1, , k)$. The other parameters are a standard feed-forward neural network, where $W \in R^{kX2d}$ is the weight matrix, $b \in R_k$ is the bias vector, and $f = tanh$ is the activation function.R_2 and U are computed in the same way asR_1 .

Training

For training, more than 10 million events were extracted from Reuters financial news and Bloomberg financial news as the training data for event embeddings. The training algorithm, described above repeats for N iterations over the training examples.The loss of two event tuples are calculated as: $loss(E, E^r) = max(0, 1f(E) + f(E^r)) + \lambda ||\Phi||_2^2$ where Φ = (T_1 , T_2 , T_3 , W, b) is the set of parameters. The standard L_2 regularization weight λ is set as 0.0001. If the loss $loss(E, E_r) = max(0, 1f(E) + f(E_r))$ is equal to zero, the algorithm continues to process the next event tuple. Otherwise, the parameters are updated to minimize the loss using the standard back-propagation (BP) algorithm.

6.1.4 Deep prediction Model

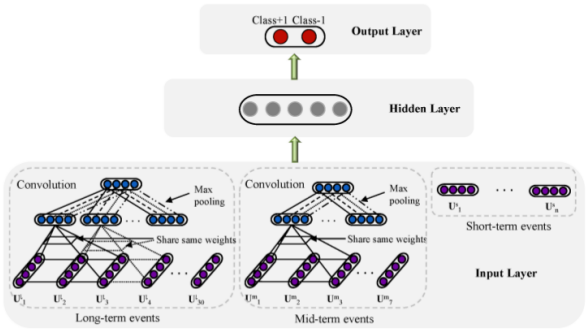


Figure 4: Architecture of the prediction model based on a deep convolutional neural network

The input to the model is a sequence of event embeddings, where events are arranged in chronological order. Embeddings of the events on each day are averaged as a single input unit (U). The output of the model is a binary class, where Class +1 represents that the stock price will increase, and Class -1 represents that the stock price will decrease. As shown in Figure 3, for long-term (left) and mid-term (mid- dle) news, the narrow convolution operation is used to com- bine l (l = 3) neighbour events.Authors have used max pooling layer on top of the convolutional layer, which

forces the network to retain only the most useful local features produced by the convolutional layer. Note that the convolution

Formally, given a series of input event embeddings U = (U_1 , U_2 , , U_n), where U_i $\in$R^d , a one-dimensional convolution function takes the dot product of the weight vector W_1 $\in$R^l with each l events (sliding window) in U to obtain Q

$$Q_j = W_1^T \, U_{jl+1:j}$$

The max pooling is performed over Q, where Q(j,) is the j-th row of matrix Q, as :

$$V_j = maxQ(j, .) \quad ,$$

Authors have used a feedforward neural network to correlate the feature vector and the stock prices.Formally, let the values of the output layer be $y_{cls}(cls+1, 1)$, its input be net_{cls} , and Y be the neuron vector of the hidden layer; then:$y_{cls} = f(net_{cls}) = \sigma(\mathrm{W}_3^T Y)$ and Y = $\sigma(\mathrm{W}_2^T V^C$), $where \sigma is the sigmoid function$,$\mathrm{W}_2$ is the weight vector between the hidden layer and the feature layer, and W_3 is the weight vector between the neuron cls of the output layer and the hidden layer.

6.1.5 Results

For evaluation purpose, Companies from Standard and Poor's (SP500) were chosen.This paper was compared with the paper by Luss[2012] and by Ding[2014]. Luss[2012] used BOW model for event representation and Support Vector Machine for prediction. Ding[2014] uses Strutured Event Tuple for event representation and Convolutional neural Network. This paper uses Event Embeddings for event representation and CNNs for prediction.

Paper	Index prediction
Luss[2012]	56.38%
Ding[2014]	58.83%
EB-CNN	64.21%

It is clear from the table that the performance of this method was better than the previous work done in this field.

Below is the graph for accuracy of some of the companies.

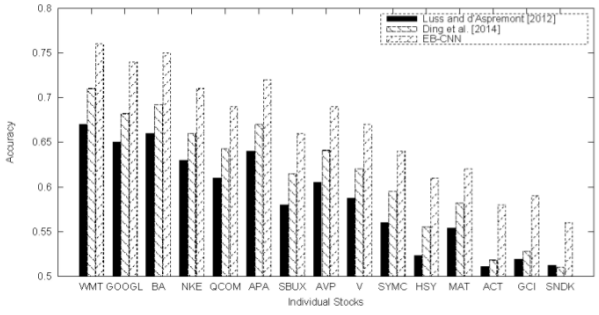


Figure 5: results of individual stock prediction

6.2 Stock Price Prediction: Time Series

Title : Twitter mood predicts the stock market
Author : Johan Bollen, Huina Mao1, Xiao-Jun Zeng

The paper is motivated by the observations from behavioral Economics that emotions can affect individual behaviour and decision making. The aim is to verify that whether this applies to society at

large, does the public mood and opinion is correlated to financial Markets and Stock price movement. The figure below demonstrates what we are going to do in this paper.

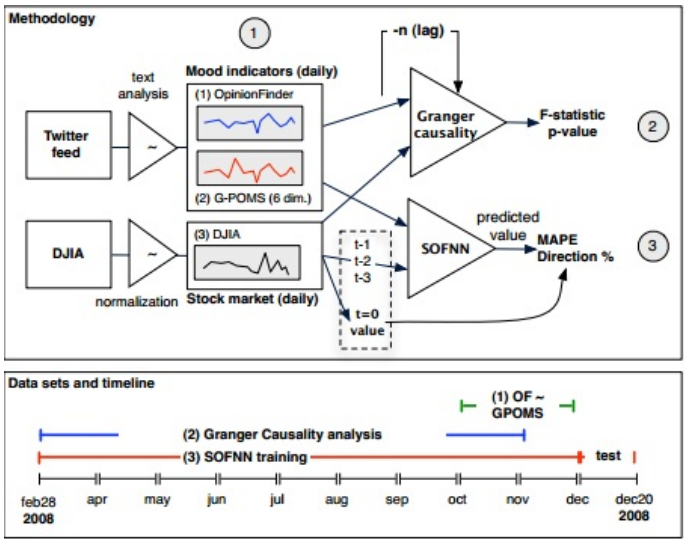


Figure 6: Diagram outlining 3 phases of methodology and corresponding data sets: (1) creation and validation of OpinionFinder and GPOMS public mood time series from October 2008 to December 2008 (Presidential Election and Thanksgiving), (2) use of Granger causality analysis to determine correlation between DJIA, OpinionFinder and GPOMS public mood from August 2008 to December 2008, and (3) training of a Self-Organizing Fuzzy Neural Network to predict DJIA values on the basis of various combinations of past DJIA values and OF and GPOMS public mood data from March 2008 to December 2008.

The author uses twitter data of about 10 months from year 2008 and analysed it using two different tools first is the *Opinion Finder* and the second one is *Google profile of Mood states(GPOMS)*. Opinion finder is a tool which measure positive and negative mood while the GPOMS has six different attributes namely, Calm, Alert, Sure, Vital, Kind and Happy. Both of these tools generate time series data of varying public mood in time. In total seven time series are obtained six from GPOMS and one from Opinion Finder. The author verified the OF and GPOMS time series against large social events and found changes in public moods before and after elections, at thanksgiving. Qualitative relationship between GPOMS and OF was also found out by the author using a simple regression model to find out that Sure, Vital and Happy components of GPOMs are significantly correlated with OF moods, while the rest are not. The main idea used in the paper is that we have several time series of different moods and one time series of Stock data, specifically Dow Jones Industrial average(DJIA) and we want to determine whether these mood time series are useful in predicting stock price of DJIA. We earlier mentioned about *Granger Causality Test* which is a very useful test for such situations. Now Bi-variate Granger Causality analysis is done for different moods and DJIA price and the null hypothesis was rejected which stated that mood time series doesn't predict DJIA values. The observation included that Calm time series had Highest Grangers Causality with DJIA values.
Granger causality analysis suggests a predictive relation between certain mood dimensions and DJIA. However, Granger causality analysis is based on linear regression whereas the relation between public mood and stock market values is almost certainly non-linear. To better address this non linear effect and asses the contribution that public mood assessments can make in predictive models of DJIA value, the performance is compared with the performance of a *Self-organising Fussy Neural Network(SOFNN)* which predict DJIA bassed on:

1. the past 3 days of DJIA values
2. the same combined with various permutations of our mood time series.

SOFNN in particular is a five layer hybrid neural network with the ability to self-organize its own neurons in the learning process. To predict the DJIA value on day t, the input attributes of our

SOFNN include combinations of DJIA values and mood values of the past n days. Several conclusion are drawn from the result of SOFNN, First, adding positive/negative sentiment obtained from OF has no effect on prediction accuracy compared to using only historical DJIA values. This confirms the results of our Granger causality analysis. Second, when calm series is used as an input we obtain highest prediction accuracy. In this paper highest accuracy of 87.6% was observed using clam as an input to SOFNN.
The conclusion of the paper are that public mood as measured from large-scale collection of tweets posted on twitter.com is correlated or even predictive of DJIA values. The calmness of the public (measured by GPOMS) is thus predictive of the DJIA rather than general levels of positive sentiment as measured by OpinionFinder. A SelfOrganizing Fuzzy Neural Network trained on the basis of past DJIA values and our public mood time series furthermore demonstrated the ability of the latter to significantly improve the accuracy of even the most basic models to predict DJIA closing values.

7 Conclusion

As a part of Course project- Stock Price Prediction using Deep Learning, We thoroughly performed the literature survey and explored the kind of work done in the two fields of stock market prediction. We studied various Deep Learning and Time Series Models that are being used for prediction task. We have done thorough study of several research paper to gain better understanding of this field and have summarised the following two papers in this report.

1. Deep Learning for Event-Driven Stock Prediction.
2. Twitter mood predicts the stock market

With all of our survey , we came to know that such models do work in the real life and are often employed by companies such ass Goldman sachs and Gartner Inc. However, these tools are still unreliable to be fully trusted with the investor's money as these are not descriptive mathematical which are studied thoroughly. These models are still kind of black boxes and we don't have a complete understanding of how they works and what is happening inside the network, but surely we do use them as an one of the indicator of how the market will perform.
Just to bring to reader notice, nowadays most of the trading is done algorithmically and there are very very fast algorithms that trade thousands of time in a fraction of seconds. Financial firms deploy their own *High Frequency trading* Algorithms to trade effectively and to even fool other trading algorithms, such an act was known as the great flash crash of 210 where the DJIA dropped by 1000 i.e about 9% and then kept decreasing for 30 minutes and then attained it's normal prices. This was caused due to two high frequency algorithms while fooling each other and after some time they terminated and market got back to normal. These algorithms works great but at times they collapse and with them the market collapses.
Further in the project, We are planning to use this knowledge to implement a model to predict stock prices. We are planning to use Deep Architectures and time series techniques for this purpose. We will compare our work against the existing work done in that field.

Acknowledgement

This work was done under the guidance of Annervaz and Chiranjib Bhattacharyya. We would also like to thank Sharmistha and Adhirupa for their assistance in the course.

References

[1] Twitter mood predicts the stock market. - $JohanBollen^{1,*}$,$HuinaMao^{1,*}$,$Xiao - JunZeng^{2}$.
[2]Exploiting Social Relations and Sentiment for Stock Prediction - Jianfeng Si ,Arjun Mukherjee ,Bing Liu ,Sinno Jialin Pan ,Qing Li,Huayi Li
[3]Deep Learning for Event-Driven Stock Prediction - Xiao Ding , Yue Zhang , Ting Liu , Junwen Duan
[4]Deep Learning for Multivariate Financial Time-Series
[5]Investigation Into The Effectiveness Of Long Short Term Memory Networks For Stock Price Prediction. * Sites like wikipedia and investopedia and Quant.stackexchange were used to understand the concepts.