Deep Investment in Financial Markets using Deep Learning Models

Saurabh Aggarwal
Computer Science Graduate,
Software Developer,
New Delhi 110026

Somya Aggarwal
Student at San Jose State University,
San Jose, CA 95192,
United States of America

ABSTRACT
The aim of this paper is to layout deep investment techniques in financial markets using deep learning models. Financial prediction problems usually involve huge variety of data-sets with complex data interactions which makes it difficult to design an economic model. Applying deep learning models to such problems can exploit potentially non-linear patterns in data. In this paper author introduces deep learning hierarchical decision models for prediction analysis and better decision making for financial domain problem set such as pricing securities, risk factor analysis and portfolio selection. The Section 3 includes architecture as well as detail on training a financial domain deep learning neural network. It further lays out different models such as- LSTM, auto-encoding, smart indexing, credit risk analysis model for solving the complex data interactions. The experiments along with their results show how these models can be useful in deep investments for financial domain problems.

General Terms
Prediction Analysis, Portfolio selection, Financial Markets

Keywords
Machine learning, deep learning, artificial intelligence, neural network, credit risk, stock market

1. INTRODUCTION
Financial market problems usually involve data from multiple sources which makes them difficult to analyze. Deep Learning, being an important module in artificial intelligence can be used for deep investments. To gain insights and to perform deep investments, it is required to get a model that can learn complex relationships and features from data inputs and hence can lead to predictions for the output variables, for instance, can be the assets under management return value. The deep learning prediction models have great advantages over the traditional models such as: a) Over-fitting is avoided. b) Increase in sample fit for complex interactions c) input data set can be easily expanded to include all the features of possible relevance to the problem set.

2. RELATED WORK
[1] Fama, 1965: It has been shown that the financial markets are informationally-efficient. The prices and the behavior of the stocks reflect all the known information and the price movement is the result of any news or event. Tracking the behavior of stock price movements have now been done by deep learning using neural networks.

[2] Deep Learning for event driven stock prediction, X Ding, 2015: The paper discusses the deep learning method for event driven stock market prediction. Here the events are extracted from news text as vectors and then trained as novel neural tensor network. It further uses a deep convolutional neural network to model both long and short term influences of events on stock price movements. The experimental results here depict 6% improvements on S&P 500 index predictions.

[3] An Artificial Neural Network Approach for Credit Risk Analysis, 2010: The paper analyzes the ability of the artificial neural network model developed to forecast the credit risk of a panel of Italian manufacturing companies. It compares the architecture of the artificial neural network model developed in the paper with the one developed in 2004 and hence showing the difference between the two using neural networks.

[4] Dropout: A Simple Way to Prevent Neural Networks from Over-fitting: The paper shows that the dropout improves the performance of neural networks on supervised learning tasks in vision, speech recognition, document classification and computational biology, obtaining state-of-the-art results on many benchmark data sets. The key idea explained here is to randomly drop units from the neural-network during training which prevents units from co-adapting.

[**5] Nonparametric Approach to Pricing and Hedging Derivative Securities:** Here, a non-parametric method for estimating the pricing formula of a derivative asset using learning networks is discussed. To illustrate the practical relevance of the network pricing approach, it is applied to the pricing and delta-hedging of S&P 500 future options.

3. FINANCIAL MODELS USING DEEP LEARNING

3.1 Architecture
A fundamental machine learning problem, with an input variable X, is used to find predictor of an output Y. The mapping Y= F(X) is defined as input-output mapping where the input provided is high-dimensional. In classification problems the mapping is, F: X → Y where Y ∈ {1…, K} indexes categories for K cases. Deep learning is a form of machine learning but is distinguished by passing through different layers of abstraction called neurons, the learned features of the data. Deep learning is hierarchical, such that at every layer factors are extracted from features, and a deeper level's factors become the next level's features. Let $Z^{(l)}$ denote the l-th layer, hence $X = Z^{(l)}$ and gives final output (Y). [6] So, deep learning architecture can be defined as:

$$Z^{(0)} = f^{(1)}(W^{(0)} + b^{(0)})$$

$$Z^{(1)} = f^{(2)}(W^{(1)} + b^{(1)}) \ldots..$$

$$Y(X) = W^{(L)} Z^{(L)} + b^{(L)}$$

$W^{(l)}$ are weight matrices, and $b^{(l)}$ are threshold or activation levels. The activation functions $f^{(l)}$ is crucial in deciding a good predictor model. Commonly used activation functions

for input x are RELU (Rectified linear units) max{x, 0} [7] and sigmoidal (e.g., 1/ (1 + exp (−x)), cosh(x), or tanh(x)). The Layers are each of the (L) transformations with the output as the $(L+1)^{th}$ layer when an input (X) is given.

3.2 Training a Deep Neural Architecture

Training a deep neural net involves a) training b) validation and c) testing. In training set, the weights of the network are adjusted. The over fitting is minimized in validation set along with model selection. The testing is then performed to evaluate the actual prediction. Then it is required to solve for (W, b) once the size, activation functions and the depth of learning routine have been chosen where: [8]

$$W = (W_0, W_1 \ldots\ldots W_L),\ b = (b_0, b_1 \ldots\ldots\ldots b_L)$$

(W, b) are the parameters for learning that are computed during training. To achieve this training data set D is:

$$D = \{Y^{(i)}, X^{(i)}\}^{T} \text{ for } \{i = 1, 2, 3 \ldots T\}$$

And a loss function $L(Y, Y'')$. Hence solve,

$$\text{Args min}_{w,b} \frac{1}{T} \sum_{i=1}^{T} L(Y_i, Y^{W,b}(X^i))$$

$\emptyset(W, b)$, a regularization penalty is added to avoid over fitting. Hence, the equation to be solved is (Eq-1):

$$\text{Args min}_{w,b} \frac{1}{T} \sum_{i=1}^{T} L(Y_i, Y^{W,b}(X^i)) + \lambda\emptyset(W, b)$$

Here, λ is a key parameter that signifies the amount of regularization. Often, too little regularization leads to over fitting and poor out of sample performance.

3.3 Auto-encoder

An auto-encoder is a deep learning model, which approximates X by itself (for, X = Y) thereby, trains the architecture. An auto-encoder also puts most cost-effective representation of X. Here for {X1, X2 . . .} training data set, target values are set as Yi = Xi. [9] A static auto-encoder with two linear layers can be defined as deep learner as following:

$$Z^{(2)} = W^{(1)} X + b^{(1)}$$

$$a^{(2)} = f_2(Z^{(2)}),$$

$$Z^{(3)} = W^{(2)} a^{(2)} + b^{(2)},$$

$$Y = W^{(w,b)}(X) = a^{(3)} = f_{(3)}(Z^{(3)})$$

Here, a (2) and a (3) are called the activation levels. The aim is to learn weight matrix $W^{(1)}$ and $W^{(2)}$. Commonly, $a^{(1)} = X$ is set. In a two layer deep learning model $W^{(1)}$ and $W^{(2)}$ are simultaneously based on input data set X. For a financial time series (Y_T), a dynamic one layer auto-encoder can be defined as following:

$$Y_T = W_x X_t + W_y Y_{T-1} \quad \text{and}$$

$$WY_T = \begin{pmatrix} X(t) \\ Y(t-1) \end{pmatrix}$$

As shown above, the equation encodes and the weight matrix W decodes the vector Y_t into Y (t-1) which is its history state and X (t) which is its current state. Since, the model is already in predictive form, hence deep learning's auto encoding shows nicely that there's no need to model the variance and covariance matrix. The major benefit that auto-encoding does is it avoids over fitting for the complex interactions in the financial data. For Example:

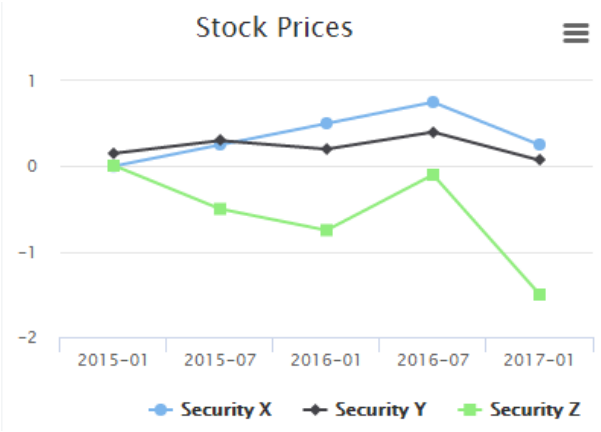

Figure1 (a): Stock Prices for Security X, Security Y and Security Z plotted half yearly between 2015 and 2017 before auto-encoding. The x-axis displays the date range while y-axis displays the stock prices ($) multiplied by a common factor for range.

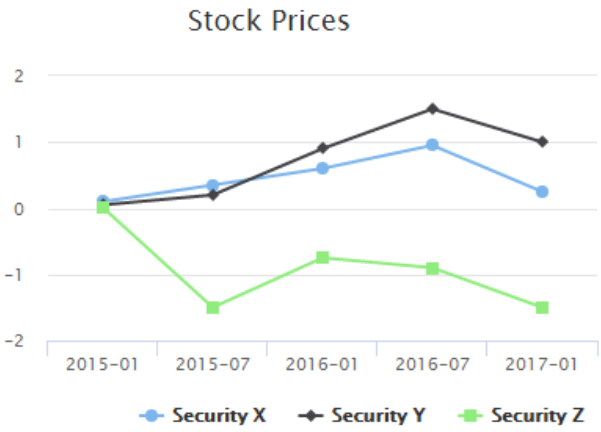

Figure1 (b): Stock Prices for Security X, Security Y and Security Z plotted half yearly between 2015 and 2017 after auto-encoding. The x-axis displays the date range while y-axis displays the stock prices ($) multiplied by a common factor for range.

As displayed in Fig-1(a). Security-X takes an upward trend in July-2016 before being auto-encoded. After applying auto-encoding technique, Fig-1(b) shows that Security-Y instead takes and upward trend for July-2016, which clearly shows that over fitting, is avoided in this scenario.

3.4 Long Short Term Memory Models (LSTMs)

The RNN's can be defined as rectified neural networks that can easily learn complex dynamics via deep learning equations [10]:

$$Z_t = W_{xz} X_t + W_{zz} + b_x$$

$$Y_T = f(W_{hz} Z_t + b_z)$$

Here, N is the hidden units with Z_t as the hidden layer. X_t is the input with Y_T as the output over the time period t. As per (Dean et al. 2012 and Lake et al. 2016), Rectified Neural networks have proven successfully on the tasks of text generation and speech recognition. As a form of recurrent neural networks, the Long-short-term-memories (LSTMs) give a solution by incorporating memory units. This helps the network to learn, when it has to update hidden states with the

new information and when to forget the previous hidden states. For a Financial application, architecture for an LSTM model can be defined as:

$$P_t = \sigma\,(W^T\,[S_{T-1,}\,X_T] + b_{f,}$$

The information can be removed or added from the memory state with the help of layers with the help of sigmoidal function $\sigma(x) = (1 + e^{-x})^{-1}$ for the portfolio function P(t). The forget gate can be defined as the gate used to forget the information for the previous cell state for the hidden state M_T

$$F_{T\,=}\,P_T \otimes M_{T-1}$$

Where, $\otimes$ signifies a point wise multiplication. For a financial application of portfolio selection, the function P_T when given weight matrix W, gives a new cell state as the sum of previous cell state passed through a forget gate for the selected components of the $[S_{T-1,}\,X_T]$ vector.

3.5 Credit Risk Analysis Model

The credit risk analysis is defined as a mechanism being adopted in financial companies to know the credit worthiness of a portfolio, companies or any other entity before investments are being made. The goal of deep learning models is usually, the feature representation of a high dimensional input space. For example, in image processing the feature representation can be faces, edges and finally pixels represented as different layers of a neural network. From a deep learning neural network, the feature extraction is the main output. The non-linear contrast of input variables gives hence the insights for the tendency of firms to default.

A test was conducted for a variable number of inputs to understand the risk factor. [11] The input-variables used for neural network were: A) Return on Investments (ROI) B) ROA (Return on assets) which signifies how profitable a portfolio is with respect to the assets under its holdings. C) % of capital D) % of equity E) ROE, is defined as return on equity F) dividend and the net income. G) Overall depreciation rate H) YTM (yield to maturity) signifies the net worth of yield being processed.

The sample space was divided into three classes- Safe, Vulnerable to risk and third one being risky. Portfolios (P1, P2 and P3) from a financial data set were chosen with H, defined as security (S) holdings over a period of time (T). Following results were produced for comparison of the three portfolios based on inputs infused in the neural network.

Table-1: Deep Learning Neural Network for Credit Risk Analysis Results for the variable inputs of portfolios (P) with securities (S) held over a time range (T)

Portfolios	Safe	Vulnerable	Risk-Factor
Portfolio P1	80.6%	19.4%	0%
Portfolio P2	77.9%	20%	2.1%
Portfolio P3	90%	2.7%	7.3%

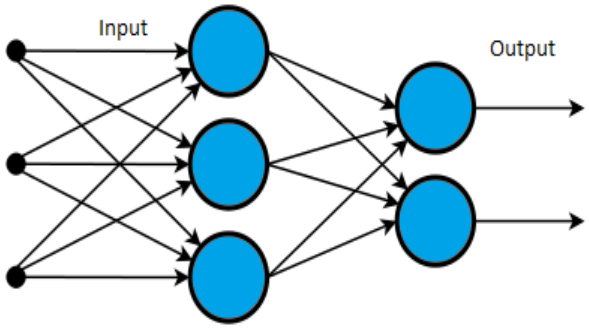

Figure: Neural Network showing Crediting Rating Analysis with Input I(t) and output O(t).

Hence, the results shows the Portfolio (P1) being least on risk factor (0%), while Portfolio P3 being most risky at a risk factor of about 7.3%, that will surely help the investment management firms to decide the portfolios before any decisions have been made based on the input factors and their weights.

3.6 Smart Indexing

When the aim is to approximate or replicate a stock index from collection subset of a number of stocks, the following two methods or approaches can be chosen from :a) Choose a small group of stocks which have in the past given a performance very similar to that of the observed index. b) Choose a small group of stocks, which have historically represented a large part of the total aggregate information of all. For experimental analysis, top five companies listed on S&P500(x) index were considered for a financial year namely App(x), Mob(x), General(x), Chive(x) and IB(x). The actual names were masked for analysis. The aim was to setup a Deep feature policy (DFP) which through some hidden layers can approximate the S&P500(x) based on top stocks at the index For this, a neural network with five input layers and two hidden layers was designed. The company's market capitalization (in dollars), assets under management value, annualized return and percentage of change in value for the year were considered. The total returns of index for the year were calculated as CAGR (Compound Annual Growth Rate, Annualized Return) which is ((Index end value)/(Index start value))^(1/ (number of years)) – 1 which came out to be 10.01%, while the experimental value calculated was 11.96% with an approximation error of 1.95%. To improve the predictive performance, the regularization added to the loss functions should be estimated by cross validation and hence teaching the model to calibrate itself to the training data [12].

Often, by hit and trial, one can find a small sub-set of stocks which gives a reasonable linear approximation [13] of the considered index. The deep learning allows converting the input set of data through a sequence of adaptive linear layers, which means that, in training, even non-linear relationships can also get identified. Since all the hidden layers gives a new meaning to the input features, the resulting approximation can be said to be a deep feature policy (DFP) [14]. With a given diverse set of input data, a DFP can be trained to approximate the data being targeted, to close accuracy. It can be noted that many classic models have been focusing on in-sample approximation quality, due to their shortcomings. In contrast to that deep learning addresses out-of-sample performance as their optimization target.

4. CONCLUSION

The author demonstrated experimental conclusions for estimating credit risk analysis for selection of portfolios with securities for investments. The author also explained the effects of over fitting and on applying auto-encoding, that the results can sometimes be very different than estimated. The smart indexing models and LSTMs as explained in sub-sections helps to find the non-linear relationships in a complex financial data set from multiple sources. Hence, the deep learning models can be very useful for decision making in investments in financial markets.

5. FUTURE WORK

The next steps in future will be to analyze further more complex data interactions in financial markets such as comparison of hedge funds. The deep learning models discussed in this paper along with other models will be applied to hedge funds data set to bring out comparison analysis based on their risk factor.

6. REFERENCES

[1] The behavior of stock market prices, by Eugene F. Fama, Journal of Business, Volume 38, Issue 1, Jan 1965, 34-105.

[2] Deep Learning for Event-Driven Stock Prediction, by Xiao Ding, Yue Zhang , Ting Liu , Junwen Duan, Harbin Institute of Technology, China, Singapore University of Technology and Design

[3] An Artificial Neural Network Approach for Credit Risk Management By- Vincenzo Pacelli1, Michele Azzollini, Journal of Intelligent Learning Systems and Applications, 2011, 3, 103-112

[4] Srivastava et al.: Dropout: a simple way to prevent neural networks from overfitting, Journal of Machine Learning Research, 15, 1929-1958, 2014.

[5] J. M. Hutchinson, A. W. Lo, and T. Poggio: A Nonparametric approach to pricing and hedging derivative securities via learning networks, Journal of Finance, Vol. 48(3), pp. 851-889, 1994.

[6] Learning Deep Architectures for AI, Yoshua Bengio, Foundations and Trends in Machine Learning

[7] Rectified Linear Units Improve Restricted Boltzmann Machines, By Vinod Nair and Geoffrey E. Hinton

[8] Learning Deep Architectures for AI, By Yoshua Bengio, Foundations and TrendsR in Machine Learning Vol. 2, No. 1 (2009) 1–127

[9] Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks, By Quoc V. Le, Stanford.

[10] Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling, Has¸im Sak, Andrew Senior, Franc¸oise Beaufays, Google

[11] An Artificial Neural Network Approach for Credit Risk Management Vincenzo Pacelli1, Michele Azzollini Journal of Intelligent Learning Systems and Applications, 2011, 3, 103-112

[12] Deep Learning in Finance, By- J. B. Heaton, N. G. Polson, J. H. Witte, 23-Feb 2016

[13] Deep vs. Shallow Networks: an Approximation Theory Perspective By Hrushikesh N. Mhaskar1 and Tomaso Poggio, Aug 2016

[14] Learning Deep Feature Representations with Domain Guided Dropout for Person Re-identification By Tong Xiao ,Hongsheng Li ,Wanli Ouyang ,Xiaogang Wang