**CNNpred: CNN-based stock market prediction using a diverse set of variables**

Financial markets are the beating heart of the global economy, with billions of dollars changing hands every day. Clearly, a strong forecast of market behaviour in the future would be quite useful in a variety of situations. Because stock markets play such a significant influence in economic growth (Beck & Levine, 2004), understanding their behaviour and forecasting their future can be extremely beneficial in achieving economic objectives.

Previously, numerous algorithms has been used for feature extraction, the most common being artificial neural networks and SVMs. Other ML methods include: random forests, linear discriminant analysis, quadratic discriminant analysis, logistic regression and evolutionary computing algorithms. Deep Learning models have now shown higher success than these algorithms. An important part of designing an algorithm for stock market prediction is feature extraction. The algorithm should be able to understand the important features which affect the stock market prediction automatically and then give an accurate prediction. Technical indicators like volume, momentum, rate of change etc. are certain variables that are often used for prediction. However, these may not be the best set of features. Hence, the author suggests a CNN based approach to automate the process of feature extraction and subsequent prediction.

The author suggests two frameworks for the purpose of prediction:

**2D-CNNpred:** The first one postulates that a single model is true for the prediction of future fluctuations in all markets. In simple words, the historical data of any market, when fed to this model will be able to successfully predict the future trends.

**3D-CNNpred**: This approach follows that the mechanisms that dictate the future behaviour of a certain market will be, even slightly, different for different markets. Here, data from all the markets is fed as input to a model designed for a particular market to estimate its future fluctuations.

**Initial variable set**

The initial variables include 82 features which can be categorised broadly into eight categories: primitive variables (close price and day of the week), technical indicators (technical features extracted from historical data like moving average), world stock markets (effect of other countries’ stock markets), exchange rate of US dollar in major countries, commodities (price of gold, wheat etc.), Big U.S. companies (stock prices of major companies like Apple), future contracts (effect of future prices of stocks) and other useful variables (eg: treasury bill rates). Data from 5 markets: S&P 500 index, NASDAQ Composite, Dow Jones Industrial Average, NYSE Composite, and RUSSELL 2000 have been used for the purpose.

**Model formulation**

Previous 60 day’s data is used for the prediction of the 61st day’s movement. The design of the 2 models have been presented below.

**2D-CNNpred:** This model takes a 60x82 size input vector (no. of days x no. of initial variables). It is passed through a convolution layer with 8 filters of size 1x82. This layer tries to extract the important feature for each day. The output from this layer is then passed into another convolution layer of size 3x1which is inspired from the famous candlestick patterns like Three Line Strike and Three Black Crows which try to find meaningful patterns in three consecutive days. Next, there is a max pooling layer with size 2x1 which is a common setting for the pooling layers. It is followed by another convolution layer of size 3x1 to construct even more complex features and a max pooling layer of size 2x1. Finally, the layers are flattened and fed into a dense network where a sigmoid function is used to map the input to a probability of whether the stock price is going to go up on the next day. The final output is discretised as 0 (probability < 0.5) or 1 (probability >= 0.5) based on this.

**3D-CNNpred:** The input in this case is 3D tensor with size 60 (no. of previous days) x 5 (no. of markets) x 82 (no. if initial features). A convolution layer (1x1) is used first to extract the important features from the set of initial features. This is followed by another convolution layer of size (3x5) to extract durational features across markets. The next max pooling layer (2x1), convolution layer (3x1) and another max pooling layer (2x1) is same as in 2D-CNNpred. The final prediction is made using a dense layer fed into a sigmoid activation function.

The activation function for all the layers (except the last dense layer) was ReLU and adam optimizer with a batch size of 128 was used to train the network.

**Results and Discussion**

The models were tested against three other baseline models and the buy and hold strategy. Both models showed better performance in terms of F-measure, Sharpe Ratio and Certainty Equivalent. On increasing the transaction costs, the Sharpe Ratio and CEQ decreased but was still better than the buy and hold strategy. The value of investing $1 in each of the strategy also showed that the 2D-CNNpred and 3D-CNNpred gave better returns than the rest.

CNNs are powerful tools and it is necessary to carefully select the filter sizes in case of financial markets. Common CNN filters of 3x3 or 5x5 used in image classification applications may not be a good choice in case of finance. It can be proposed that these models can be used as subsystems of future trading simulation models.

With the development of more powerful computers and decreasing computational time, it may be possible to build larger CNNs which may be able to give better predictions by extracting more complex features which, otherwise is impossible for even the experts in finance to capture and understand. Deep Neural Networks are an important part in the trading ecosystem and the future looks to be a very promising one.