Modeling Stock Market Returns using Deep Learning

Devansh Gupta* (2019160)
Divyansh Rastogi* (2019464)
Rupanshu Yadav* (2019475)

All authors denoted by * contributed equally, the ordering of naming is alphabetical.

Problem Statement

More than five trillion dollars worth of stocks are traded every single day. Investors can maximize their profits by investing rationally based on technical analysis. The time series prediction problem is complex due to the widely accepted semi-strong efficient market hypothesis and their volatile & non-stationary structure. Although these financial markets are shown to be predictable to a certain extent.

Stock market analysis methods fall into two categories:

1. Technical analysis involves the prediction of stock prices using historical data, primarily price and volume.
2. Fundamental analysis makes predictions based on the intrinsic value of the investment by studying macroeconomic features, industry features, and company's features, including financial reports, competitors, & viability.

Statistical models such as ARIMA, STAR are widely used for time series prediction purely based on the historical price of a stock. Machine learning has been intensively researched for its use in financial market forecasting where Decision trees, Xgboost, & Naïve Bayes all have shown to be beneficial in stock market price prediction. Deep learning methods & frameworks have also proven to be helpful in time series analysis with its ability to model complex non-linear relationships.

Literature Review

- Volatility & non-stationary nature of stock markets
- Partial predictability of financial markets
- Predictive relationships of numerous financial & economic variables
- Statistical & Econometric ML methods:
  - ARIMA
  - SVM
  - Boosted Regression Trees
  - Logistic Regression
- Feature Selection & Extraction / Data Transformation Methods:
  - Genetic Algorithms + ANN / SVM
  - Autoencoders + DNN / SVM
  - PCA + ANN / DNN
- Deep CNN's
- Deep CNN's + LSTM
- GANs

Dataset

Dataset Description

The dataset contains several daily features of S&P 500, NASDAQ Composite, Dow Jones Industrial Average, RUSSELL 2000, and NYSE Composite from 2010 to 2017.

The dataset covers features from various categories of technical indicators, futures contracts, price of commodities, important indices of markets around the world, price of major companies in the U.S. market, and treasury bill rates.

Relevant papers:

- *CNN-Pred: CNN-based stock market prediction using a diverse set of variables.*
- *U-CNNpred: A Universal CNN-based Predictor for Stock Markets*

Data Set Characteristics:	Sequential, Time-Series	**Number of Instances:**	1985
Attribute Characteristics:	Real	**Number of Attributes:**	84
Associated Tasks:	Classification, Regression	**Missing Values?**	Yes

https://archive.ics.uci.edu/ml/datasets/CNNpred%3A+CNN-based+stock+market+prediction+using+a+diverse+set+of+variables

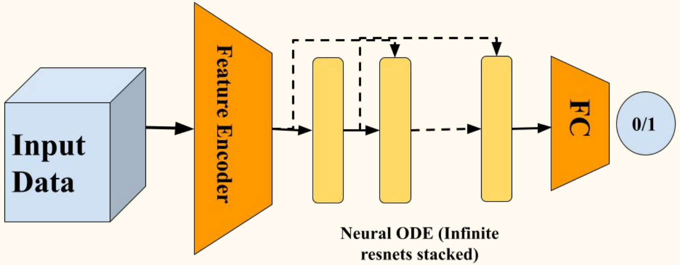
Baselines

- ARIMA
- PCA + ANN
- CNN-Pred 2D & 3D
- CNN-LSTM

Results

Neural ODE Classifier

1. A class of models which are trained by solving an ordinary differential equation rather than backpropagating gradients
2. The discrete resnet expressions are made continuous via ODE solving thus facilitating continuous time steps.
3. We mainly use it as an infinite depth residual network to make an end to end classifier for classifying the stock trends



Structure of Neural ODE Classifier

Market Name	Average	Maximum
NASDAQ	**0.48**	**0.579**
NYSE	**0.531**	**0.567**
S&P500	**0.542**	**0.622**
RUSSELL	**0.517**	0.584
DJI	**0.519**	0.589

Regression metrics on the test set using CNN-LSTM model trained for 100 Epochs on the CNN-Pred Stock market Dataset

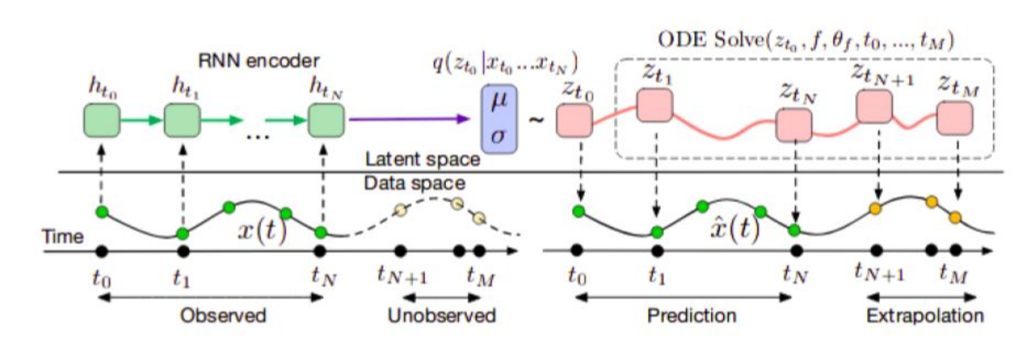
Market Name	Average	Maximum
NASDAQ	0.438	**0.55**
NYSE	**0.567**	**0.624**
S&P500	**0.541**	**0.556**
RUSSELL	**0.491**	0.495
DJI	**0.521**	**0.527**

F-Scores on the test set using infinite depth classifier using Neural ODE trained for 15 Epochs on the CNN Pred Stock market Dataset on single market data as input.

Bold entries denote an improvement over the results obtained on the CNNPred2D baseline

Neural ODE VAE

1. Learns to generate realistic time series based on samples of time series given, hence learning the dynamics to generate appropriate stock prices
2. Use variational inference to sequentially encode the data using an RNN and decode the data using a Neural ODE and making inferences through the decoder.
3. Trained on sequences consisting of closing prices of 60 time steps and doing inference with the first 59 time-steps as inputs and getting the closing price of the 60th day.



https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf

Market Name	MAE	RMSE	Max Closing Price
NASDAQ	2177.418	2251.247	341682.12
NYSE	2461.052	2538.778	388508.06
S&P500	799.763	815.964	24420.32
RUSSELL	1350.78	1355.90	12414.37
DJI	7288.813	7455.721	2980330.8

Initial results on the Regression metrics on the test set using ODE VAE model trained for 150 Epochs on the sequential closing price data of the CNNPred Stock market Dataset using a GRU encoder and a Neural ODE Decoder

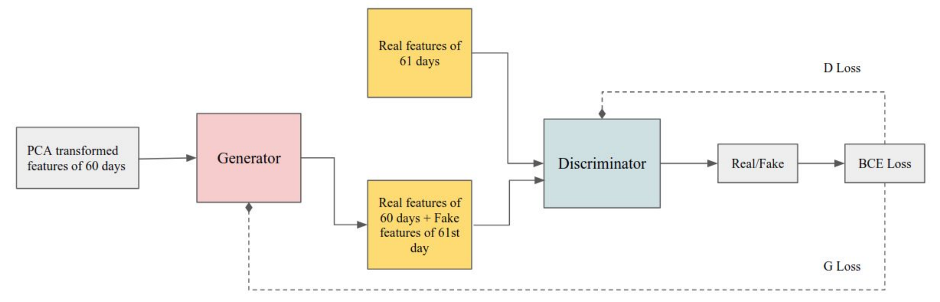
GANs

- The GAN model builds upon the idea to predict PCA transformed features as utilized in PCA+ANN Baseline. The model i.e. the generator is trained in an adversarial setting.

- The model is tested on the S&P 500 index. Initially data is preprocessed with outlier adjustments and then the data is split in 80/20 ratio for train and test set respectively. Furthermore, the data is standardized with the mean and variance of the training dataset.

- The generator utilizes a shallow LSTM single layer network followed by a fully connected layers comprising of TanH and ReLU activations along with batch normalization. The generator's aim is to take features of past 60 days and generate the new features of the 61st day. The discriminator is a deep CNN network that takes in features of 61 days and predicts whether the this trend is real or fake. The loss function used to train the discriminator is BCE Loss.

- The accuracy on the train set and the test set were observed to be 0.545 & 0.504 respectively.



Generative Adversarial Network

Conclusion

Conclusion

- We investigated the feasibility of GANs and Neural ODEs in capturing the distributions of real stock market data.
- On comparison with traditional PCA+DNN, GANs performed slightly worse. This could be majorly attributed to the careful tuning required for training a GAN. It should be noted that GANs are significantly harder to tune compared to CNN and LSTM.
- Neural ODEs as infinite depth classifiers gave superior results as compared to our baselines but the ODE-VAE results were slightly inferior, but we hypothesized that since such generative models require a lot of examples to learn, hence a better dataset would be ideal in our scenario.