**SoC Summer 2022 Final Documentation**

**CNN Based Stock Market Prediction**

**Shivprasad Kathane**

**Keywords: CNN, Stock Market Prediction, Convolutional Neural Network, AlexNet, NIFTY50, Deep Learning, Machine Learning, Feature Extraction**

|  |  |  |
| --- | --- | --- |
| **Team Member Name** | **Roll Number** | **Email-Id** |
| Sautrik Chaudhuri | 213040032 | 213040032@iitb.ac.in |

**Brief Description**

|  |
| --- |
| Financial Markets are the backbone of any economy. Therefore, prediction of the fluctuations in the stock market is valuable in a multitude of fields. However, the stock market is almost impossible to predict completely accurately given the vast number of influencing variables. Therefore, it becomes important to understand the features that influence the stock market the most. In this project, this part of extraction of important features has been done using Convolutional Neural Networks. The main idea of this project is based on a paper by Hoseinzade and Haratizadeh (2019) who proposed a CNN architecture which uses market information from different markets, along with certain technical indicators of a certain market, to predict the future fluctuation in stock prices in that market. The method was further improved by exploring famous CNN architectures like LeNet-5 and AlexNet. Results were found to be satisfactory and such CNN based models may be part of trading systems in the future. |

**Progress**

|  |
| --- |
| The project was divided into phases by our mentor. Every week, we were supposed to work on some important part of the project and then discuss it with him at the end of that week. The project was spread over 10 weeks and a brief overview of each week’s work is presented below:  Week 1-3: We worked on understanding the various concepts related to Time Series Modelling. This included concepts like seasonality, trends and long-term oscillations. Next, we learnt about standard modelling techniques like AutoRegressive Model, ARIMA, SARIMA, Exponential Smoothing and so on. Finally, we learnt about the basics of ML and DL including regression, classification, ensemble modelling, ANNs etc. These concepts were applied on a small project for ‘Predicting Credit Card Default using ML’.  Week 5: This week focused on diving deeper into Deep Learning models like CNN. We focused on learning the different terms associated with the network like filter/kernel, max pooling, average pooling, convolution and so on. As the final project was based on a 3D CNN architecture, we also got our hands on another small project on ‘CNN based image classification for the presence of viral pneumonia in CT scans’.  Week 6: During this week, we focused on reading the paper by [Hoseinzade and Haratizadeh (2019)](https://www.sciencedirect.com/science/article/pii/S0957417419301915). We learnt how they used 2 CNN model architectures 2D-CNNpred and 3D-CNNpred in order to use information from different markets to come up with a prediction in fluctuation for a certain market. The summary of the paper can be found [here](https://github.com/sautrikc/Seasons-of-Code/blob/main/Week%206_Paper%20Summary.docx).  Week 7-9: These weeks were spent in implementing the model described in the paper and further improving it. I looked up commonly used CNN architectures like LeNet-5 and AlexNet in order to improve the model accuracy. These architectures were modified according to the application at hand as these were originally designed for image classification.  Week 10: In the final week, we took up the task of incorporating information from other markets like NIFTY50 in order to test the robustness of our model. Further, a simple web application was made which predicted the whether the prices would go up or down based on a user input of historical market data.  The progress has been summarized in the files below:  [Seasons-of-Code/Week 7\_Summary.docx at main · sautrikc/Seasons-of-Code (github.com)](https://github.com/sautrikc/Seasons-of-Code/blob/main/Week%207_Summary.docx)  [Seasons-of-Code/Week 8\_Summary.docx at main · sautrikc/Seasons-of-Code (github.com)](https://github.com/sautrikc/Seasons-of-Code/blob/main/Week%208_Summary.docx)  [Seasons-of-Code/Week 9\_Summary.docx at main · sautrikc/Seasons-of-Code (github.com)](https://github.com/sautrikc/Seasons-of-Code/blob/main/Week%209_Summary.docx)  [Seasons-of-Code/Week\_10\_Incorporating\_additional\_markets\_.ipynb at main · sautrikc/Seasons-of-Code (github.com)](https://github.com/sautrikc/Seasons-of-Code/blob/main/Week_10_Incorporating_additional_markets_.ipynb)  The major challenges that I faced during this project was that, being a beginner in this field, it took me some time to grasp the concepts and implement it properly in code. Further, there were some compatibility issues while deploying the model using Flask as well as training the model using TensorFlow. I used Google Colab’s TPU for training purposes and had to use a friend’s system for running my Flask application.  The model can be further improved by hyperparameter tuning and ensemble techniques. However, the major drawback for this is the training time and hence I hope to continue improving my model in the future. The html interface can also be improved for a better user experience. |

**Results**

|  |
| --- |
| The feature variables include technical indicators, exchange rate of US dollar, price of commodities like gold, stock prices of Big US companies and others. In total, there were 82 feature variables. In the paper, two different architectures 2D-CNNpred and 3D-CNNpred were used. It involved a CNN architecture as follows: Conv layer (1x82), Conv layer (3x1), MaxPooling layer (2x1), Conv layer (3x1), MaxPooling layer (2x1) and Dense Layer. 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.  Implementation of the paper can be found at: [Seasons-of-Code/Week\_7\_Implementation\_of\_the\_paper.ipynb at main · sautrikc/Seasons-of-Code (github.com)](https://github.com/sautrikc/Seasons-of-Code/blob/main/Week_7_Implementation_of_the_paper.ipynb)  I looked up into famous CNN architectures like LeNet-5 and AlexNet in order to improve the model architecture. These models were originally designed for Image Classification tasks and hence had to be modified to suit financial data. The CNN architecture used was: Conv (1x82), Conv (3x1), MaxPooling (2x1), Conv (3x1), MaxPooling (2x1), Conv (3x1), Conv (3x1), Conv (3x1), MaxPooling (2x1), Dense (4096), Dense (4096) and Dense (1). Each Conv layer had 8 filters and a droprate of 0.1 was used to prevent overfitting. Adam Optimizer with a batch size of 128 was run for 20 epochs. The Mean Absolute Error Loss Function was used and the F1 score was used as the metric for evaluating our model.  Implementation of my model can be found at: [Seasons-of-Code/Week\_9\_Implementing\_AlexNet.ipynb at main · sautrikc/Seasons-of-Code (github.com)](https://github.com/sautrikc/Seasons-of-Code/blob/main/Week_9_Implementing_AlexNet.ipynb)  The F1 score improved from 0.64 to 0.699 which is a significant improvement when it comes to stock market prediction. The model was tested on new data from NIFTY50 which also yielded an F1 score of 0.697.  A web interface was made and the model was deployed using Flask. The python application can be found [here](https://drive.google.com/file/d/1e4GSTK1-GBAeui_H-FMuqmD7SQ53yqYJ/view?usp=sharing). In order to run the file, the [templates](https://github.com/sautrikc/Seasons-of-Code/tree/main/templates) folder needs to be downloaded along with a set of [model](https://drive.google.com/file/d/1Geb878SrcgZ4tYICqFVP22DuGOyK-jvA/view?usp=sharing) weights and kept in the same directory before running the application in VS code.  **Link for Github Repository:** [sautrikc/Seasons-of-Code: This repository stores all the code which is a part of the project 'CNN based stock market prediction' (github.com)](https://github.com/sautrikc/Seasons-of-Code)  **Link for PPT:** [Seasons-of-Code/Final\_PPT.pptx at main · sautrikc/Seasons-of-Code (github.com)](https://github.com/sautrikc/Seasons-of-Code/blob/main/Final_PPT.pptx)  **Link for Video:** [Seasons-of-Code/Project\_Video.mp4 at main · sautrikc/Seasons-of-Code (github.com)](https://github.com/sautrikc/Seasons-of-Code/blob/main/Project_Video.mp4)  The model can be further improved by better choice of hyperparameters which would require a higher amount of training time. I will take up this work in the future for further improvement. |

**Learning Value**

|  |
| --- |
| I learnt a lot from this project and have summarized a few of the topics below:   1. Time Series Modelling 2. Regression, Classification, Ensemble Modelling 3. Evaluation metrics like F1score, Precision and Recall 4. Working with imbalanced datasets 5. Convolutional Neural Network architecture and related terminologies 6. Famous CNNs like LeNet-5, AlexNet, VGG-16, Inception-v1 7. Model Deployment using Flask 8. HTML 9. Knowledge of financial markets and metrics like Sharpe Ratio and Certainty equivalent 10. Libraries like Tensorflow, Numpy, Pandas and TA-Lib |

**Software used**

|  |
| --- |
| The following software were used during the course of the project:   1. Jupyter Notebooks 2. Google Colab 3. VS Code 4. HTML |

**Suggestions for others**

|  |
| --- |
| Develop a basic understanding of the different optimization methods and research on various hyperparameter tuning techniques. There will be issues faced, but believe me, you will learn a lot at the end of it. Also remember that training is going to take up a lot of time, so plan accordingly. |

**References and Citations**

|  |
| --- |
| **Citations:** Hoseinzade, E. and Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, 129, pp.273–285. doi:10.1016/j.eswa.2019.03.029.  **Links:**   1. [Time series-Introduction](https://towardsdatascience.com/time-series-introduction-7484bc25739a#:~:text=A%20time%20series%20is%20a%20series%20of%20data%20points%20ordered,the%20future%20is%20being%20predicted) 2. [Time Series Definition](https://www.investopedia.com/terms/t/timeseries.asp) 3. [What is Time Series Data? | Definition, Examples, Types & Uses](https://www.influxdata.com/what-is-time-series-data/) 4. [The Complete Guide to Time Series Data](https://www.clarify.io/learn/time-series-data) 5. [Time Series Analysis and Forecasting: Examples, Approaches, and Tools](https://www.altexsoft.com/blog/business/time-series-analysis-and-forecasting-novel-business-perspectives/) 6. [Taxonomy of Time Series Forecasting Problems](https://machinelearningmastery.com/taxonomy-of-time-series-forecasting-problems/) 7. [10 Challenging Machine Learning Time Series Forecasting Problems](https://machinelearningmastery.com/challenging-machine-learning-time-series-forecasting-problems/) 8. [Time Series Forecasting Methods, Techniques & Models | InfluxData](https://www.influxdata.com/time-series-forecasting-methods/) 9. [11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)](https://machinelearningmastery.com/time-series-forecasting-methods-in-python-cheat-sheet/) 10. [The Complete Guide to Time Series Analysis and Forecasting | by Marco Peixeiro | Towards Data Science](https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775) 11. [Welcome To Colaboratory](https://colab.research.google.com/?utm_source=scs-index) 12. [Machine Learning - Grundy](https://wiki.wncc-iitb.org/index.php/Machine_Learning) 13. [Deep Learning - Grundy](https://wiki.wncc-iitb.org/index.php/Deep_Learning) 14. [UCI Machine Learning Repository: default of credit card clients Data Set](http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients) 15. [Machine Learning for Finance Guide - A Real-Life Example - AlgoTrading101 Blog](https://algotrading101.com/learn/machine-learning-for-finance-guide/)   [Credit Card Default: a very pedagogical notebook | Kaggle](https://www.kaggle.com/code/lucabasa/credit-card-default-a-very-pedagogical-notebook)   1. [Deep Learning Basics](https://colab.research.google.com/github/lexfridman/mit-deep-learning/blob/master/tutorial_deep_learning_basics/deep_learning_basics.ipynb) 2. [An Intro to Git and GitHub for Beginners (Tutorial) (hubspot.com)](https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners) 3. [Keras vs Tensorflow vs Pytorch [Updated] | Deep Learning Frameworks | Simplilearn](https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article) 4. [TensorFlow - Grundy (wncc-iitb.org)](https://wiki.wncc-iitb.org/index.php/TensorFlow) 5. [TensorFlow basics  |  TensorFlow Core](https://www.tensorflow.org/guide/basics) 6. [Deep Dive into Math Behind Deep Networks | Piotr Skalski | Towards Data Science](https://towardsdatascience.com/https-medium-com-piotr-skalski92-deep-dive-into-deep-networks-math-17660bc376ba) 7. [Deep Learning Tensorflow Tutorial](https://github.com/Kulbear/deep-learning-coursera/blob/master/Improving%20Deep%20Neural%20Networks%20Hyperparameter%20tuning%2C%20Regularization%20and%20Optimization/Tensorflow%20Tutorial.ipynb) (requires [tf\_utils.py](https://github.com/bighuang624/Andrew-Ng-Deep-Learning-notes/blob/master/assignments/assignment2-3/tf_utils.py) file) 8. [Basic classification: Classify images of clothing  |  TensorFlow Core](https://www.tensorflow.org/tutorials/keras/classification) 9. [A Beginner's Guide To Understanding Convolutional Neural Networks – Adit Deshpande – Engineering at Forward | UCLA CS '19 (adeshpande3.github.io)](https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/) 10. [Gentle Dive into Math Behind Convolutional Neural Networks | by Piotr Skalski | Towards Data Science](https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9) 11. [Convolutional Neural Network (CNN)  |  TensorFlow Core](https://www.tensorflow.org/tutorials/images/cnn) 12. [Time series forecasting  |  TensorFlow Core](https://www.tensorflow.org/tutorials/structured_data/time_series) 13. [Stock Price Time Series Forecasting using Deep CNN (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2021/08/hands-on-stock-price-time-series-forecasting-using-deep-convolutional-networks/) 14. [CNNpred: CNN-based stock market prediction using a diverse set of variables - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S0957417419301915) 15. [Practical Statistics for Data Scientists](https://www.researchgate.net/profile/Janine-Zitianellis/post/Can_anyone_please_suggest_a_books_on_machine_learning_using_R_Programming/attachment/613a5b83647f3906fc975a71/AS%3A1066204907204608%401631214467436/download/Practical+Statistics+for+Data+Scientists+50%2B+Essential+Concepts+Using+R+and+Python+by+Peter+Bruce%2C+Andrew+Bruce%2C+Peter+Gedeck.pdf) 16. [Python Data Science Handbook](https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/Index.ipynb#scrollTo=6K97if3hwgZ2) 17. [Stock Prediction with ML: Feature Engineering — The Alpha Scientist](https://alphascientist.com/feature_engineering.html) 18. [Stock Prediction with ML: Feature Selection — The Alpha Scientist](https://alphascientist.com/feature_selection.html) 19. [Using CNN for financial time series prediction (machinelearningmastery.com)](https://machinelearningmastery.com/using-cnn-for-financial-time-series-prediction/) 20. [http://www.deeplearningbook.org](http://www.deeplearningbook.org/) 21. [A guide to an efficient way to build neural network architectures: Hyper-parameter selection and tuning for Dense Networks using Hyperas on Fashion-MNIST | by Shashank Ramesh | Towards Data Science | Part 1](https://towardsdatascience.com/a-guide-to-an-efficient-way-to-build-neural-network-architectures-part-i-hyper-parameter-8129009f131b) 22. [Practical Aspects of Deep Learning | Deep Learning | WnCC Wiki](https://wiki.wncc-iitb.org/index.php/Deep_Learning#Practical_Aspects_of_Deep_Learning) 23. [Loss Functions and Optimization Algorithms | by Apoorva Agrawal | Medium](https://medium.com/data-science-group-iitr/loss-functions-and-optimization-algorithms-demystified-bb92daff331c) 24. [Regularization Techniques | Regularization In Deep Learning](https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/) 25. [Metrics to Evaluate your ML Algorithm | by Aditya Mishra | Towards Data Science](https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234) 26. [Improving Performance of Convolutional Neural Network! | by Dipti Pawar | Medium](https://medium.com/@dipti.rohan.pawar/improving-performance-of-convolutional-neural-network-2ecfe0207de7) 27. [Ensemble Learning Methods for Deep Learning Neural Networks](https://machinelearningmastery.com/ensemble-methods-for-deep-learning-neural-networks/) 28. [Popular CNN Architectures](https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d) 29. [Transfer Learning](https://machinelearningmastery.com/transfer-learning-for-deep-learning/) 30. [Cross Validation in Time Series](https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4) 31. [How to Save and Load Your Keras Deep Learning Model (machinelearningmastery.com)](https://machinelearningmastery.com/save-load-keras-deep-learning-models/) 32. [Simple way to deploy machine learning models to cloud | by Tanuj Jain | Towards Data Science](https://towardsdatascience.com/simple-way-to-deploy-machine-learning-models-to-cloud-fd58b771fdcf)   **Repositories:**  <https://github.com/hoseinzadeehsan/CNNpred-Keras> |

**Disclaimer**

|  |
| --- |
| All the codes, text, data, images used in this project belong to their respective owners and I do not claim any right over them. Any such resource used in this project is for educational and research purposes only. All information provided are in good faith, however we make no representation or warranty of any kind, express or implied, regarding the accuracy, validity, reliability, availability or completeness of any information. UNDER NO CIRCUMSTANCES SHALL WE HAVE ANY LIABILITY TO YOU FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF THIS PROJECT OR RELIANCE OF ANY INFORMATION PROVIDED BY THIS PROJECT. YOUR USE AND RELIANCE ON ANY INFORMATION PROVIDED BY THIS PROJECT IS SOLELY AT YOUR OWN RISK.  Copyright Disclaimer under section 107 of the Copyright Act 1976, allowance is made for “fair use” for purposes such as criticism, comment, news reporting, teaching, scholarship, education and research. Fair use is a use permitted by copyright statute that might otherwise be infringing. |

**Licenses**

|  |
| --- |
| The information on licenses for software used are provided below:  **Tensorflow:** <https://github.com/tensorflow/tensorflow/blob/master/LICENSE>  **Google Colab:** <https://colab.research.google.com/pro/terms/v1>  **Anaconda**: <https://www.anaconda.com/eula-anaconda-distribution>  **Python:** Python software and documentation are licensed under the [PSF License Agreement](https://docs.python.org/3/license.html#psf-license).  **Github:** <https://docs.github.com/en/site-policy/github-terms/github-terms-of-service>  **VS Code**: [License - Visual Studio Code](https://code.visualstudio.com/license) |