Artificial Neural Network Based Forecasting Daily Stock Market

Ajay Kumar Madkami, Hari Prasadh P ,Manish Kumar kumawat, Murugaraj T,
Partha Sakha Paul,Venkatesh Tentu
Indian Institute of Technology Madras,Chennai-600036

Introduction

Our objective is to forecast daily movements in the stock market, particularly the Nifty 50 index. We aim to utilize a variety of features believed to influence the Nifty 50's performance, employ dimensionality reduction and apply Various Artificial Neural Networks(ANN) to predict whether the stock market will go up or down the next day.

Motivation

Forecasting stock market movements is a challenging task due to the complex interplay of various factors such as economic indicators, investor sentiment, geopolitical events, and market dynamics. Traditional econometric models often struggle to capture the non-linear relationships and dynamic nature of financial markets.

Methodology

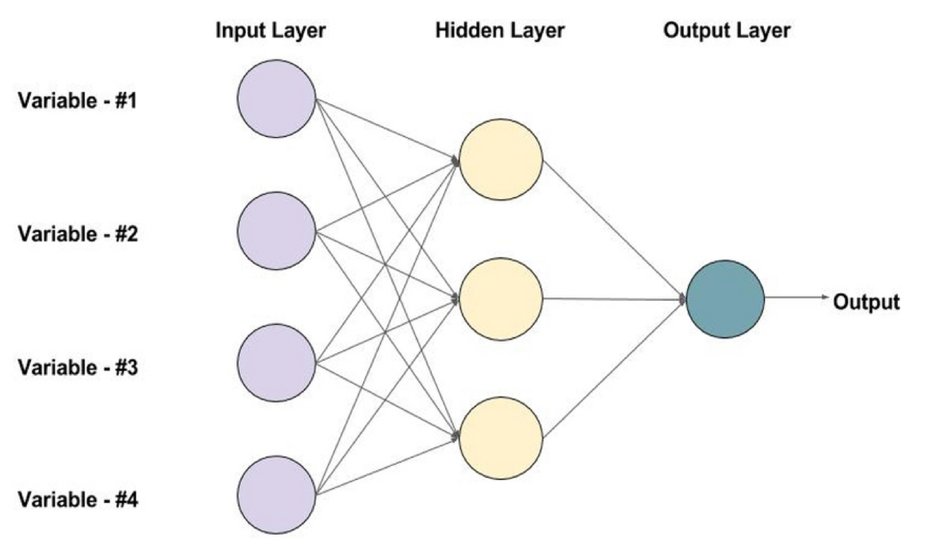
Data Collection and PCA We collect a wide range of features believed to influence the Nifty 50 stock index. The dataset covers historical data for multiple years, allowing us to capture various market conditions and trends.To handle the high dimensionality of the feature space and reduce computational complexity, we apply Principal Component Analysis (PCA) for dimensionality reduction. PCA helps us to identify the most important features while preserving the variance in the data.

FeedForward Neural Network:
An FNN consists of an input layer, one or more hidden layers, and an output layer. Mathematically, the output of an FNN can be represented as follows:

$$\hat{y} = f_n(f_{n-1}(\dots f_1(XW_1+b_1)W_2+b_2\dots)W_n+b_n)$$

where f_i represent the activation function, W_i are weight matrices, b_i are bias and $\hat{y}$ is the predicted output.

FeedForward Neural Network Diagram:

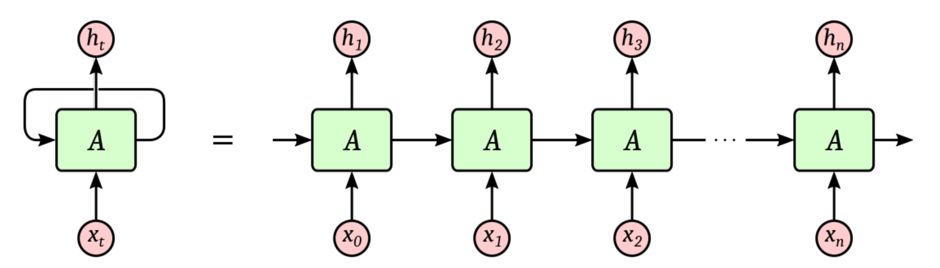


Recurrent Neural Network: An RNN is designed to handle sequential data by maintaining a hidden state that captures information from previous time steps. The output of an RNN at time t is influenced by both the current input x_t and the hidden state h_{t-1}.

$$h_t = f(W_{hh}h_{t-1} + W_{hx}x_t + b_h)$$

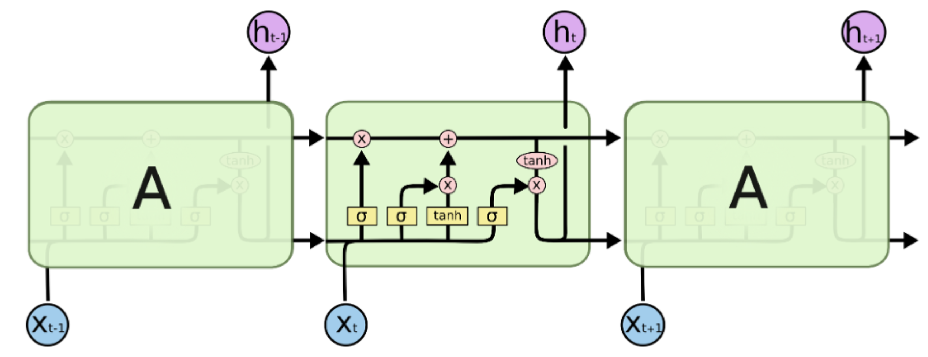
$$\hat{y}_t = g(W_{yh}h_t + b_y)$$

Recurrent Neural Network Diagram:



Methodology(cont.)

Long Short-Term Memory: LSTM networks are a type of RNN designed to address the vanishing gradient problem and capture long-term dependencies in sequential data. They maintain a memory cell that allows information to persist over multiple time steps.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) (Forget\ gate)$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) (Input\ gate)$$

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) (cell\ state)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t (update\ cell\ state)$$

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) (output\ gate)$$

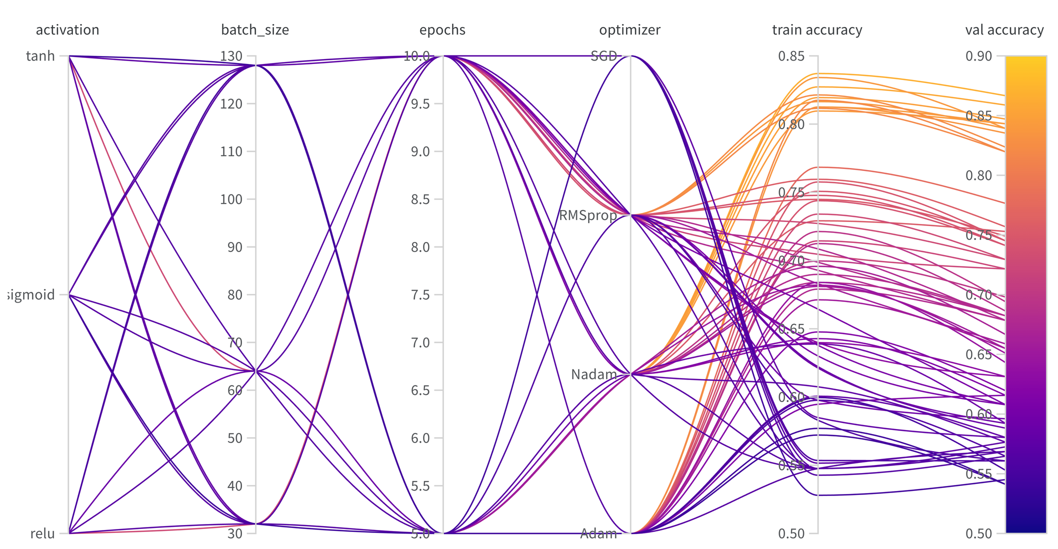
$$h_t = o_t * \tanh\left(C_t\right) (hidden\ state)$$

Model Training and Evaluation: The neural network models are trained using historical data, with a portion of the data reserved for validation. We used WandB to get the accurate hyper parameters.The trained models are evaluated using evaluation metrics.

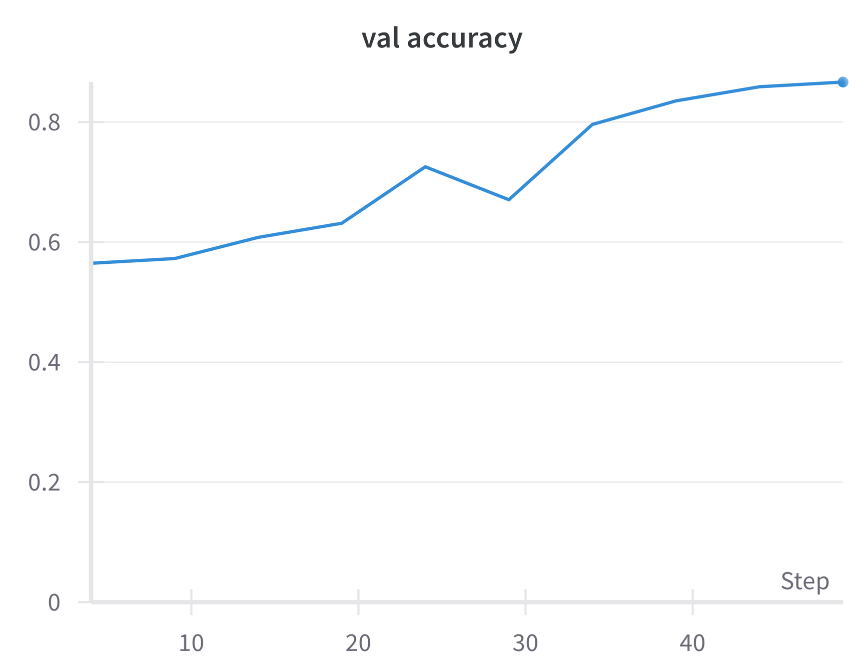
Result

FNN PLOTS:

WandB parallel plot for FNN

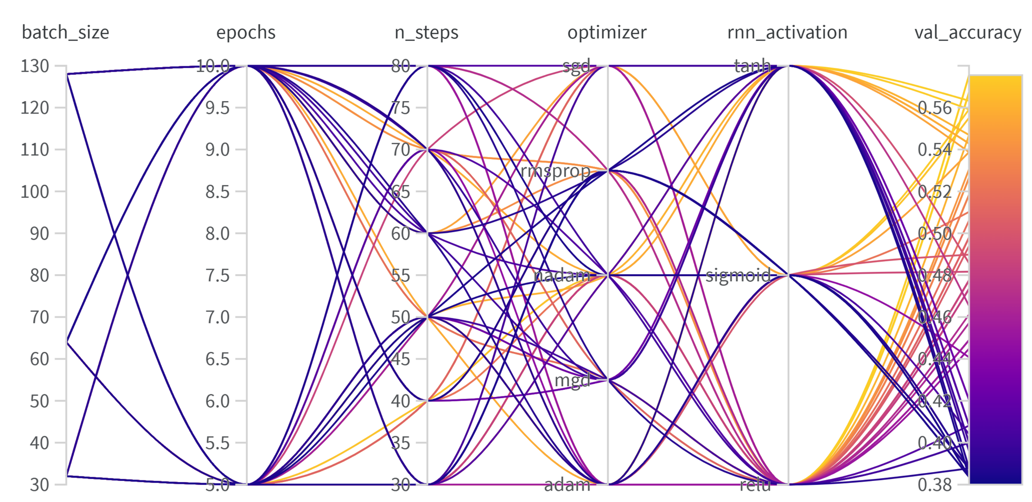


Its corresponding best configuration validation accuracy plot



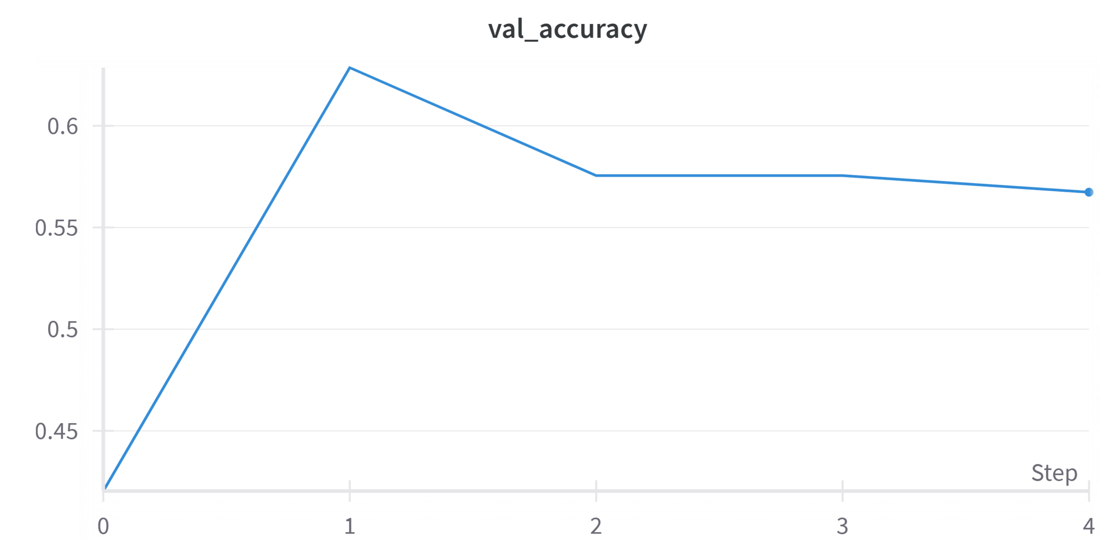
RNN PLOTS:

WandB parallel plot for RNN



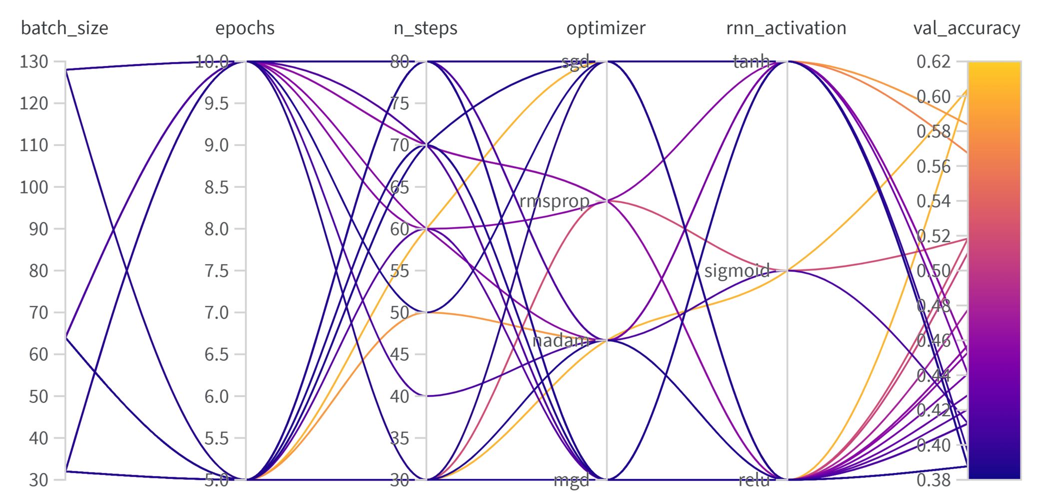
Result(cont.)

Best configuration validation accuracy plot

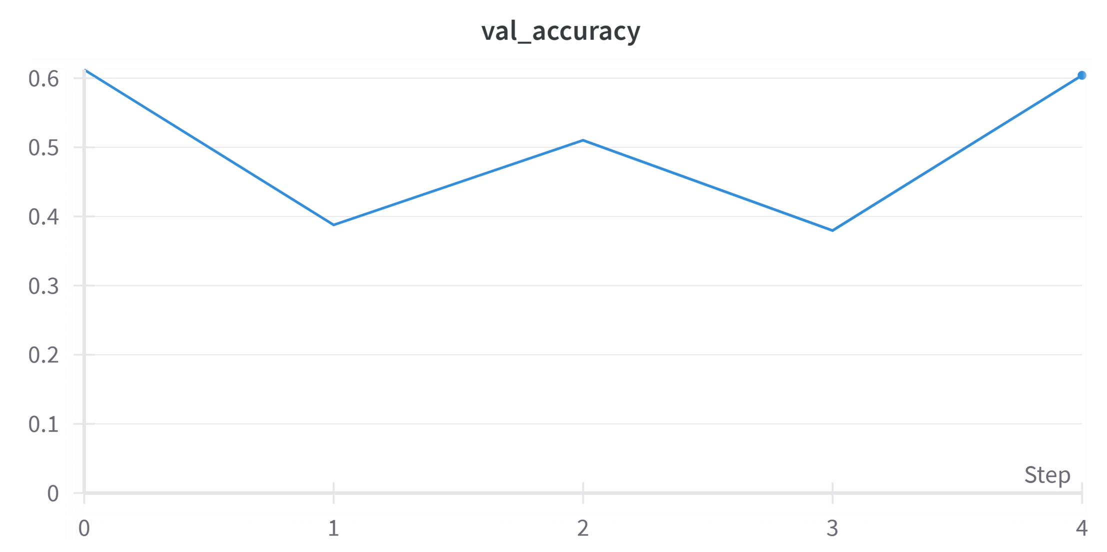


LSTM PLOTS:

WandB parallel plot for LSTM



Best configuration validation accuracy plot



Our experimental results show the performance metrics of the models:

Model	Acc	Precision	Recall	F1
FNN	0.88	0.88	0.88	0.88
RNN	0.57	0.54	0.54	0.53
LSTM	0.60	0.58	0.58	0.58

These metrics provide insights into the performance of each model in terms of its Accuracy(Acc), Precision, Recall, and F1-score.

Conclusion and Future plans

From the results we conclude that LSTM model perform better than the RNN model. Since we have sequential pattern data FNN is used only for comparison. For future research,we increase the feature set, to improve the accuracy and robustness of our predictions, thereby making our models more effective in forecasting stock market trends.
Moreover, we plan to extend our models to address regression problems, enabling them to predict not only the direction but also the approximate values of future market returns.

References

[1] Xiao Zhong , David Enke(2016), Forecasting daily stock market return using dimensionality reduction, Expert Systems With Applications, 67 (2017) 126–139.

[2] Niaki and Hoseinzade: Forecasting S and P 500 index using artificial neural networks and design of experiments. Journal of Industrial Engineering International 2013 9:1.