Foreign Exchange Rate Prediction (Indian Rupees): Using Deep Learning Technologies

Nikhil Naik – nnaik2@stevens.edu

Abstract

This report describes in detail the methodologies and technologies used in final project for course CS 583: Deep Learning conducted under the guidance of Professor Jia Xu and Dr. Abdul Khan. The primary objective of this project was to develop a robust deep learning models capable of accurately predicting Foreign Exchange Rates (Indian Rupees) - a challenging task within the realm of time series analysis. A key focus of the experimentation was the evaluation of various transfer learning techniques and their impact on the overall performance of the models. The project involved the creation and comparison of several RNN and LSTM based regressor models which are Fine-tuned on India's Forex data. Performance assessments were conducted using prominent metrics such as Mean Squared Error, Mean Absolute Error and R-squared. Furthermore, the project investigates hyperparameter tuning to optimize the best-performing model to enhance its predictive capabilities. The project repository can be found here.

1. Introduction

The Foreign Exchange Rate, or forex rate, represents the relative value of one currency compared to another and is crucial for international trade and financial transactions. Predicting forex rates poses a significant challenge due to the complex and dynamic nature of the factors influencing currency fluctuations. Multiple variables, such as geopolitical events, economic indicators, and market sentiment, contribute to the volatility of exchange rates. Additionally, the interconnectivity of global markets and the presence of intricate patterns in time series data make it challenging to capture and model accurately.

Moreover, time series data presents its unique set of challenges stemming from temporal dependencies, non-stationarity, seasonality, trends, noise, complex patterns, limited historical information, and the dynamic influence of external factors. Addressing these intricacies calls for specialized models such as RNN and LSTM, designed to account for the sequential nature of input data.

In addition to the primary goal of constructing an accurate prediction model, this project undertakes a comparative analysis of various transfer learning techniques to identify the most effective one. The methodologies encompass the creation of a pretrained model using 3 different types of dataset formats (Explained in detail in method section) which incorporating data from multiple countries. Subsequently, we fine-tune the models with India's economic indicators and Forex data. The refined models are then applied to predict future Forex rates, incorporating a specified 30-day lag time to forecast the rate for the 31st day.

Ultimately, the best performing model (LSTM), underwent a crucial refinement through hyperparameter tuning during the fine-tuning phase. In this process, the foundational layers of the pre-tuned model remained fixed, while the subsequent layers which were being fine-tuned with Indian Forex data underwent optimization. Parameters such as activation functions, node counts in each layer, and the final optimizer function were meticulously selected using the Keras-tuner library. This meticulous approach resulted in the development of a final model that significantly outperformed its counterparts created through manual trial and error. Further details on these steps will be provided in subsequent sections of this report, offering an insight into the enhancement achieved through transfer learning and hyperparameter tuning.

3. Methods

This section provides a comprehensive overview of the processes, technologies, and implementation details employed throughout the course of this project. The methodology encompasses the entire lifecycle of the project, detailing the steps taken to achieve the objectives set forth.5. Experiments.

3.1 Data Cleaning

We meticulously curated the dataset to ensure its accuracy, consistency, and reliability. This involved identifying and rectifying errors, resolving inconsistencies, handling missing values, and addressing outliers.

- Standardizing Date Format

To align disparate date formats across multiple datasets, we standardize the date values by making them universally consistent and easily distinguishable, bridging variations like '%d-%m-%Y' and '%Y-%m-%d' for seamless integration.

- Interpolation of Null Values

Addressing missing data points, we employed linear interpolation techniques. This method intelligently filled null values by referencing neighboring data or predefined criteria within the dataset.

- Dataset Merging

Combining datasets from various countries was essential. We orchestrated the merging process using Pandas' 'merge_asof' function, opting for a left join merge approach with a tolerance window of two days

- Data Normalization and Scaling

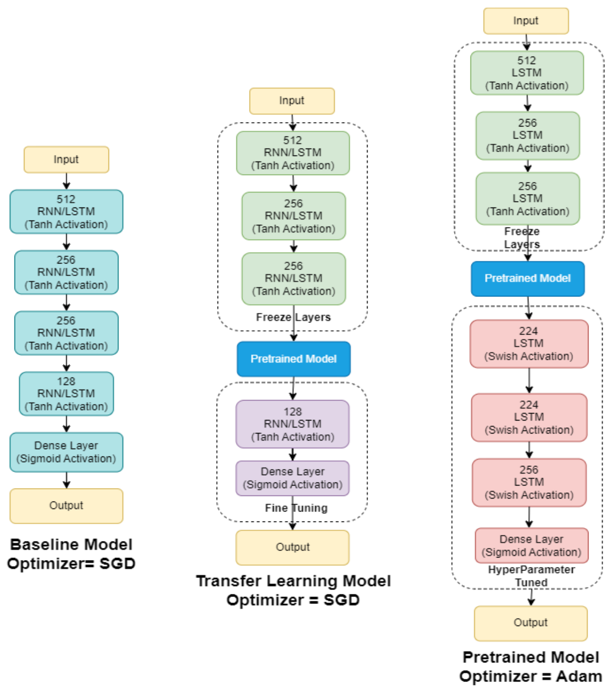
Leveraging MinMaxScaler, we standardized numerical features within a predetermined range, typically between 0 and 1. This ensured uniformity while preserving the relative distributions, enabling accurate comparisons across different variables. to form a comprehensive merged dataset.

3.2 Exploratory Data Analysis

Our comprehensive EDA delved deep into dataset characteristics, distributions, and inter-variable relationships. This exploration unearthed intricate patterns, anomalies, and trends, empowering informed decisions for subsequent analyses and modeling. We specifically conducted country-wise data visualization, generating 2x2 subplots and illustrative boxplots to visualize column distributions effectively.

3.3 Model Development

Employing Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) architectures, we embarked on model development. Initially, we formulated a baseline model with RNN/LSTM layers, yielding modest outcomes. Subsequently, we engineered separate pre-trained models based on datasets with varying target data availability: no target data, partial target data, and substantial data. Utilizing transfer learning, we fine-tuned these models by freezing initial layers and augmenting them with additional RNN/LSTM layers plus a Dense layer. Hyperparameter tuning, performed using Random Search Keras Tuner, significantly enhanced the model's accuracy, refining our predictive capabilities.



4. Datasets

The influential determinants impacting currency price include GDP, Consumer Price Index (CPI), Interest Rates, and Relative Forex. Our analysis encompassed datasets from eight distinct countries: the USA, India, China, Russia, South Africa, Poland, Brazil, and Australia.

- Data Sourcing:

We obtained information from varied sources to compile our datasets:

- Normalized GDP: Sourced from the Federal Reserve Economic Data (FRED) at https://fred.stlouisfed.org/.
- Consumer Price Index: Acquired from the same FRED source (https://fred.stlouisfed.org/).
- Interest Rates: Gathered from the FRED repository (https://fred.stlouisfed.org/).
- Forex Data: Extracted from the Federal Reserve data available at https://www.federalreserve.gov/.

- Temporal Span and Dataset Size:

Our dataset spans from January 1, 1996, to December 30, 2022, encompassing around 7000 instances. This substantial corpus provided a robust foundation for our analysis, enabling comprehensive insights into currency price dynamics across the selected countries.

5. Tools and Technologies:

- Platform for Python Code: Utilized Google Colab as the coding environment to execute Python scripts.
- GPU for Model Training: Employed the GTX 1650 GPU to facilitate the training process for our models, optimizing computational performance.
- Pandas Library Usage: Leveraged functions like 'merge_asof' and interpolation within the Pandas library to manage data merging and interpolation tasks effectively.
- TensorFlow/Keras Implementation: Utilized the Sequential method along with layers like LSTM, RNN, and Dense within TensorFlow/Keras for model architecture and development.
- Matplotlib for Data Visualization: Utilized Matplotlib extensively for data visualization, employing its graphical capabilities to present insights and results effectively.
- DateTime Standardization: Incorporated the datetime library to standardize datetime formats, ensuring consistency and uniformity across the dataset.
- Seaborn for Visual Summaries: Utilized Seaborn to create informative boxplots, visually summarizing the variability of dataset values, aiding in data exploration.
- Scikit-learn for Data Scaling and Metrics: Employed Scikit-learn to scale the data using MinMaxScaler and access various metrics available in the library, facilitating data preprocessing and evaluation of model performance.
- Hyper-Parameter Tuning: Used Keras_tuner library for hyper-parameter tuning.

6. Experiments

This section describes in detail, the different dataset formats used in pretraining step, Fine tuning of various models, The training steps involved in creating these models and the hyperparameter tuning for these models.

6.1 Dataset Formats

The experiments conducted in this project aimed to compare the baseline model's performance with three distinct types of transfer learning methods. To facilitate these models, four datasets were created. Let's delve into the details of each dataset.

- Baseline Models Dataset

The baseline model serves as a simplistic approach to model building. Each row in the training dataset for this model encompasses economic indicators and forex information from all countries. Essentially, each row represents a specific date, and the corresponding values for all countries on that given date are arranged in columns. The training feature set is constructed by concatenating the information from the last 30 days' rows, forming a comprehensive set of features. The target for training this feature set is the forex value of the Indian Rupee on the 31st day. There were no transfer learning steps involved in the type of model building.

- Relational – Knowledge Transfer Learning Datasets

In our approach to transfer learning, the primary goal was to develop a generalized pretrained model capable of predicting the forex rate for any country. This pretrained model was later fine-tuned to specifically accommodate and predict India's forex rate. The creation of these models involved crafting a dataset where each row represented the economic indicators and forex data for a given date in a particular country. The final training features were created introducing a 30-day lag for each country to incorporate historical context, and defining the target variable as the forex value for that country on the 31st day. This meticulous dataset preparation served as the foundation for training a generalized model through transfer learning, which was subsequently fine-tuned to tailor its predictions to the unique characteristics of India's forex rate.

Within the framework of Relational Knowledge Transfer Learning, our experimentation involved testing three distinct types of datasets, each incorporating varying degrees of India's data during the pretrained model building phase. The specifics of these datasets are outlined below.

o Zero-Inclusion Pretraining:

In this type of transfer learning, we utilize a dataset during the pretrained model building phase that deliberately excludes any data related to our target country, i.e., India. This distinctive approach allows the model to establish a foundation without direct exposure to the patterns of India's data, allowing model to generalize before the fine tuning phase.

o Partial-Inclusion Pretraining:

In this type of transfer learning, we utilize a dataset during the pretrained model building phase that includes some of the data related to our target country, i.e., India. This approach allows the model to include some of the patterns of the target county’s data during its generalization stage. Note, the partial data of target country is again seen by the model during fine tuning phase.

o Full-Inclusion Pretraining:

In this type of transfer learning, we utilize a dataset during the pretrained model building phase that contains the entire dataset including data related to our target country, i.e., India. The model first learns to generalize on every country’s data including India and is then finetuned again on India’s dataset to enhance the models understanding of India’s data.

Note: In all three methodologies described above, the finetuning step involves complete data from the India’s dataset.

6.2 Training Methodology

- For baseline model

RNN and LSTM baseline models were created using the baseline dataset mentioned in previous section. Since the dataset only consisted of few thousand rows and few hundred features, we train the model for 30 epochs. We observe that the LSTM baseline model converges within 20 epochs and RNN model acts sporadically. There are no finetuning steps involved with baseline model.

- For Pretrained model

For all three-dataset type mentioned in previous section, a RNN and a LSTM model is built separately. These models took the longest to train as they had more than forty thousand training entries with 4 features. This time the epochs were again set to 30 to keep consistent with the baseline model for fair comparison. With better hardware resources the epochs could have been increased to see if the final model performance increases.

- For Finetuning model

To refine the pretrained model obtained from the previous steps, a targeted approach was implemented. Specifically, the first three layers of the model were frozen (trainable = False), preserving their learned features. Meanwhile, the last dense layer and the penultimate RNN/LSTM layer were substituted with similar but untrained counterparts. This strategic replacement aimed to tailor the model to the nuances of India's economic indicators and forex values.

Subsequently, the model underwent fine-tuning exclusively on India's economic indicators and forex data. Given the relatively small dataset of seven thousand rows and four features, this fine-tuning phase was remarkably swift. The total number of epochs executed during this step was set to 70, ensuring an effective refinement of the model's predictive capabilities. Following this fine-tuning process, the resulting models were deemed final and ready for evaluation.

6.3 Hyper-Parameter Setting

In the final stage of our model building experimentation, we employed the Keras-Tuner library to optimize the hyperparameters of the best-performing model obtained from the fine-tuning stage. The parameters of the pretrained model were retained, while the subsequent layers underwent a tuning process. The list of parameters considered for tuning and their respective choices is detailed below:

- LSTM Layers Activation: relu, sigmoid, swish
- Total LSTM Layers: 0, 1, 2, 3
- Dense Layer Activation: relu, sigmoid
- Optimizer: sgd, rmsprop, adam, Adadelta

To ensure efficiency and prevent unnecessary computation time, early stopping was implemented. This mechanism halted the tuning process for parameters that did not exhibit satisfactory performance. A total of 25 trials were executed, each comprising 2 executions to identify the optimal set of hyperparameters that maximized the model's predictive capabilities.

7. Results

The performance evaluation of the various models implemented in our study is summarized in the table below. Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2) are presented for each model, providing insights into their predictive capabilities. Notably, the HP Tuned LSTM model outperforms others, demonstrating superior predictive accuracy with a lower MSE, MAE, and a positive R^2 value, indicating a better fit to the data.

Sr.No	**Model**	**MSE**	**MAE**	**R^2**
1	Baseline RNN	0.048	0.208	-4.73
2	Baseline LSTM	0.026	0.149	-2.082
3	RNN (Substantial Target Data)	0.011	0.092	-0.325
4	LSTM (Substantial Target Data)	0.014	0.107	-0.748
5	RNN (Partial Target Data)	0.013	0.105	-0.585
6	LSTM (Partial Target Data)	0.011	0.095	-0.420
7	RNN (No Target Data)	0.010	0.083	-0.196
8	LSTM (No Target Data)	0.010	0.089	-0.260
9	HP Tuned LSTM	0.004	0.059	0.446

We also plotted the prediction value to showcase clearly the performance of the best model

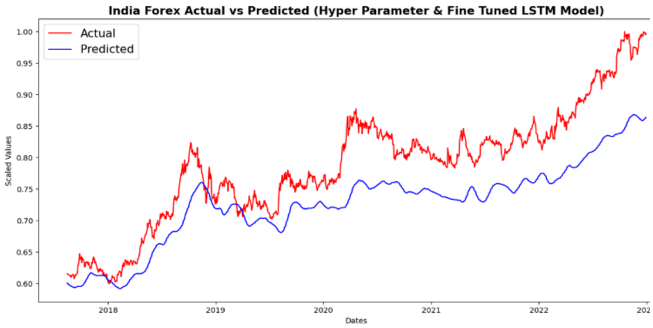


Fig: HyperParameter Tuned LSTM plot

The plot for loss functions and other Actual vs Predicted plots can be seen in the Notebook in the project repository.

8. Problems and Issues

- Notable Variables Exerting Influence on Forex Exchange Rates:

Our study acknowledges the impact of numerous influential parameters extending beyond our analytical scope. Parameters like Inflation, Government Debts, Political Stability, and Economic Recession significantly influence Forex Exchange Rates. Unfortunately, owing to the unavailability of pertinent data, our analysis was confined to the incorporation of select variables, including Normalized GDP, CPI, and Interest Rates, leaving these essential factors unexplored and unaccounted for in our predictive models.

- Computational Restraints and Model Complexity

The intricacy of our models was notably constrained by the computational resources at our disposal. This limitation hindered our ability to explore and construct more sophisticated models. Access to enhanced computational power would have afforded us the opportunity to delve into more intricate model architectures, enabling a deeper analysis of the dataset's complexities and fostering the development of more robust predictive models.

9. Conclusion

In conclusion, our exploration into deep learning models revealed a promising capacity to accurately capture trends in Foreign Exchange rates. Interestingly, the incorporation of information from multiple countries as predictive features for Forex Rate prediction yielded suboptimal results, highlighting the importance of tailoring models for specific countries. Notably, the application of transfer learning significantly enhanced model performance, emphasizing its efficacy in fine-tuning models for precise country predictions. Throughout our experimentation, LSTMs consistently outperformed RNN models, underscoring their superiority in this context. The outcomes of hyperparameter tuning suggested that the initial model complexities were insufficient, emphasizing the need for more intricate model architectures to achieve superior performance in Forex Rate prediction.