

Reflexis Solution

TEAM: Code2Create

TEAM LEADER: Indraneel Ghosh(Contact: 9929862040)

OTHER MEMBER: Siddhant Kundu(Contact: 9829891415)

College: BITS Pilani, Pilani Campus

Problem Statement Interpretation

Machine Learning Interpretation

- Maximise sales: Find ideal system scheduled hours to maximise sales.
- Analyse effect of changes of manager scheduled hours.
- Make predictions using historical data.



Model Used

Recurrent Neural Networks(RNN)

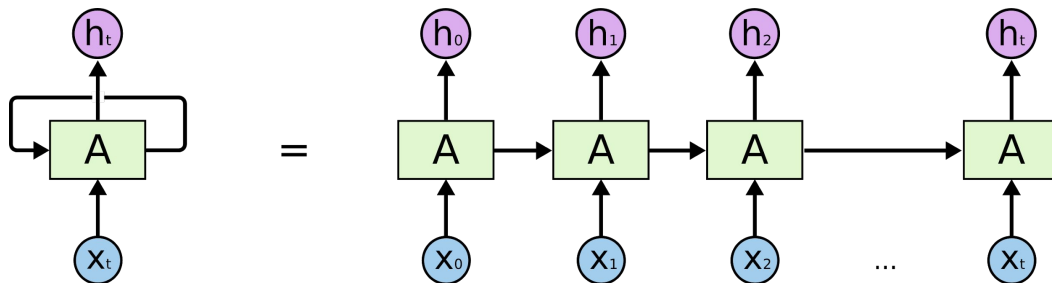
Recurrent Neural Networks(RNN)

- RNN (With LSTM(Long Short Term Memory))(An integrated Machine Learning Approach)
- EDA (Study and clean the given dataset, feature selection)
- We have reasons to believe that the sales depend on seasonality and trends are expected to be present in the data. RNN(with LSTM) captures these trends the best.



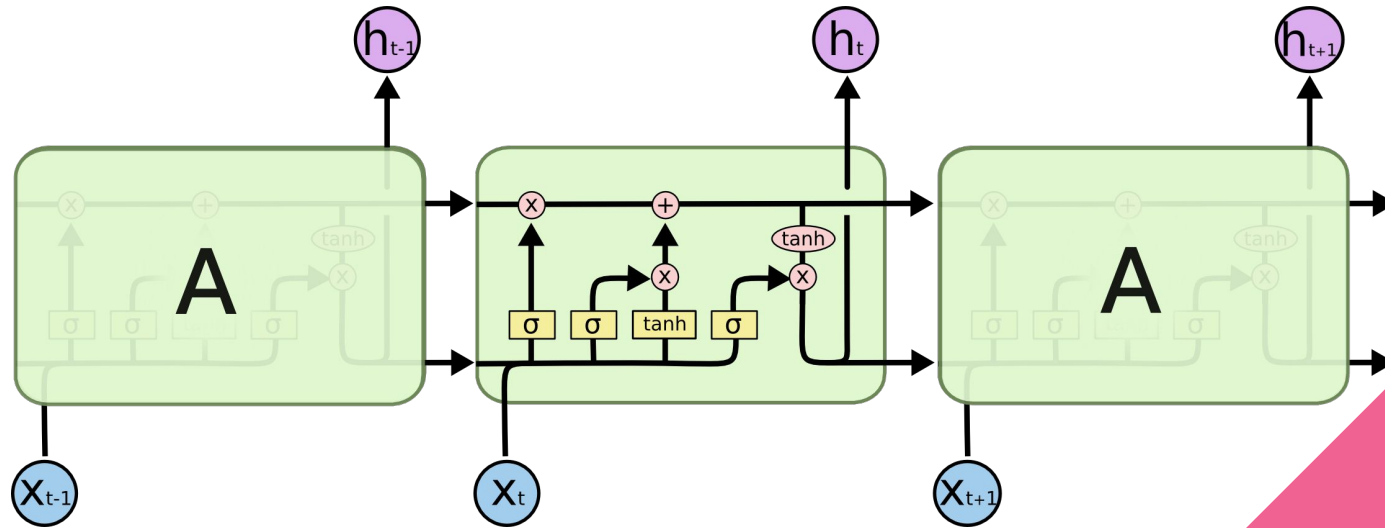
Recurrent Neural Networks

The neural network has previous inputs influencing the earlier outputs., Traditional RNNs are unable to handle “long-term dependencies.”



LSTM(Long Short Term Memory)

LSTMs have a chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.



Bidirectional LSTM

- Bidirectional networks use both past and future as context for present input.
- Analogous to seasonality and trend detection in SARIMA models.
- Intuition: Given an input for a Thursday, we must be able to derive its context as coming after Monday, Tuesday and Wednesday but before Friday, Saturday and Sunday. An input for February can be contextualized as coming after January but before March.



Assumptions

1. Maximum 12 hour work day.
2. Workers work for 7 days a week.
3. Increments made to work hours are in 0.25 intervals



Implementation Details

- Input: Store (Store number)
- RNN Model was designed using keras implementations.
- #Epochs = 50
- Groupby 'Store'(Data grouped by store)
- Batch Size = 8
- cuDNN used to improve performance and reduce training time for the model.



RNN Structure

- RNN Has 3 Layers of LSTM with 64 cells in each layer.
- Every layer has a dropout of 0.25 to reduce overfitting.
- Data is fed as a sequential time series. This Time Series contains an instance of the list [<STORE>, <MANAGER_SCHED_HOURS>, <SYSTEM_SCHED_HOURS>] for every timestamp, which for us is every day for which we have a record.
- The output for this is a list giving a scaled version of the predicted value of ['SALES_ACTUAL'], between -1 and 1.
- Bidirectional LSTMs have been used.



Final Output

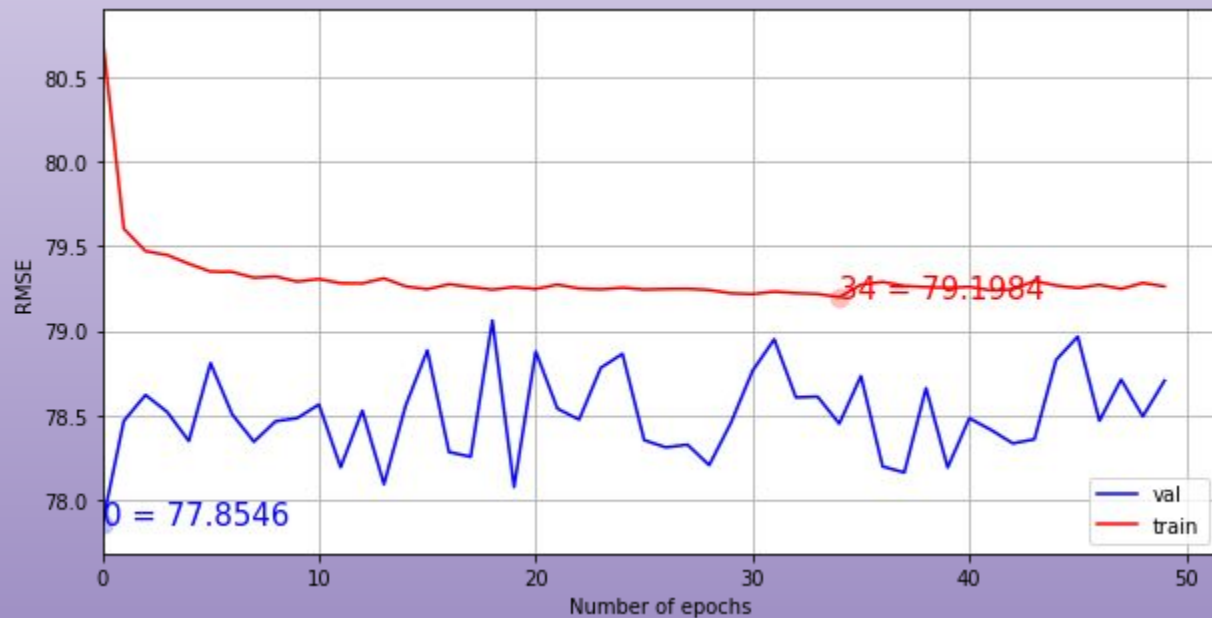
1. A single Dense layer is used at the end for the output generation.
2. Output generated by RNN predicts what the 'SALES_ACTUAL' value would be, given the 'MANAGER_SCHED_HOURS', 'STORE', 'SYSTEM_SCHED_HOURS'.
3. We then iterate over all possible values of 'SYSTEM_SCHED_HOURS' to find the optimal number of scheduled hours so as to maximise profits.
4. We further use this output to determine the effect of changes in manager scheduled hours.



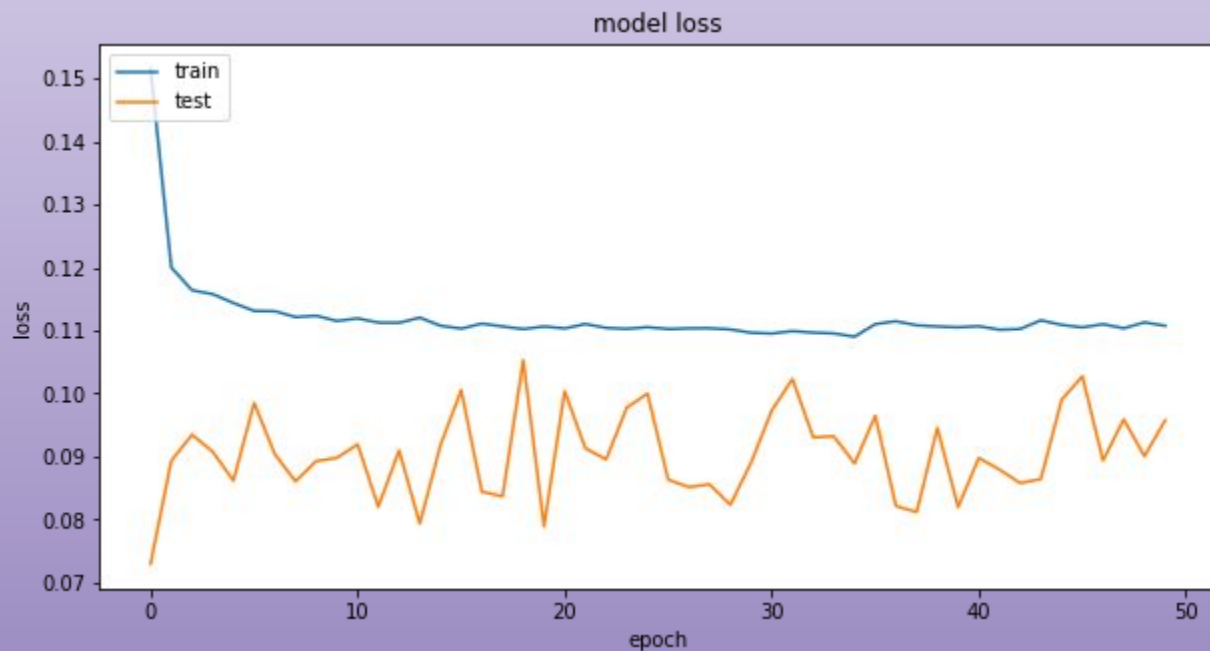


Results(RNN)

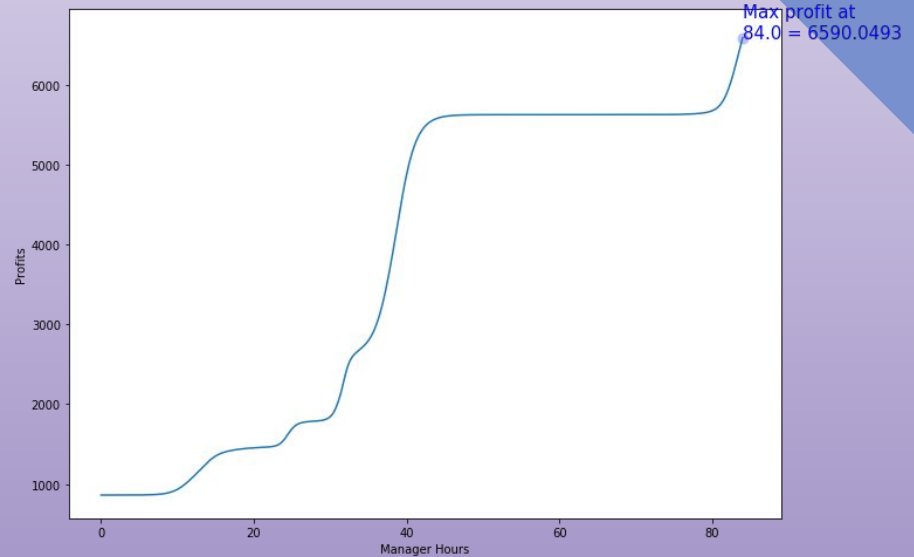
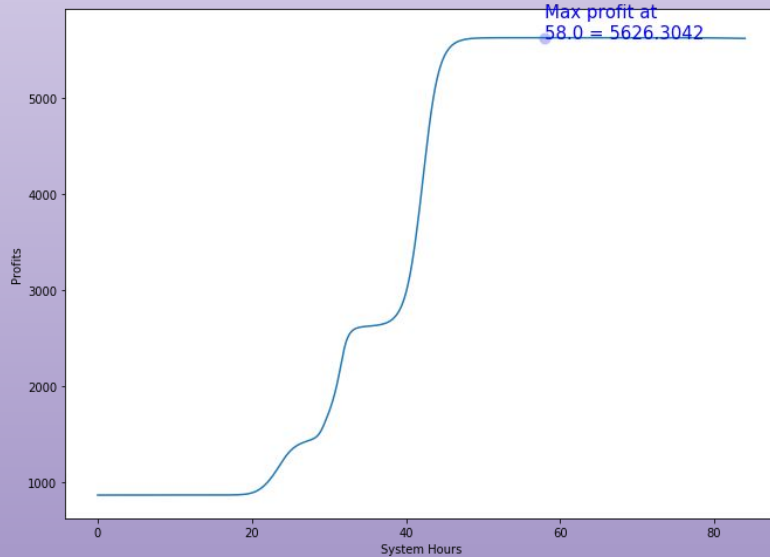
Results (Loss Values as RMSE)



Results (Scaled loss and Model loss)



Results (Ideal Work Hour Predictions)



1st graph - loop over all possible values of 'SYSTEM_SCHED_HOURS' to find the value which gives maximum profits.

2nd graph - for this value of 'SYSTEM_SCHED_HOURS', find the value of 'MANAGER_SCHED_HOURS' that gives maximum profits.

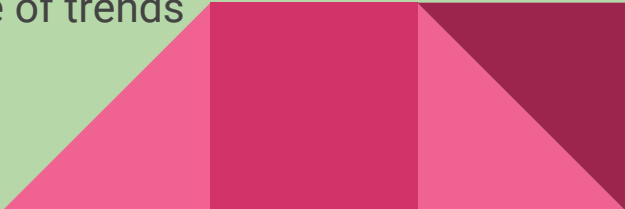
Interpretation of Results

Profits maximization condition :

1. System Scheduled Hours = 58 Hours
2. Manager Scheduled Hours = 84 Hours



Our Opinions

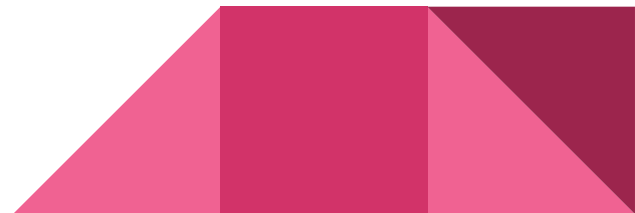
- The dataset provided is too small, which is leading to overfitting.
 - A larger dataset(10000+ entries for every store) can provide better results. This is because the latent features determining the results are highly dependent on the store.
 - The only independent factor whose relationship to system and manager scheduled hours can be mapped is the **date (and by extension, the sale season)**. The other variables in the dataset are dependent variables, and more importantly, they are dependent on the work hours.
 - We have ignored climate and discount data, since due to the low amount of available data points, there is no way of establishing whether changes come because of that info or because of the location data, or because of the presence of trends or seasonality.
- 

Additional Reference

Code Base

- Github: <https://github.com/ighosh98/aic-reflexis.git>
- Kaggle: <https://www.kaggle.com/coder98/reflexis>

[Note: For using Kaggle dataset must be uploaded by user in a zip file in the format originally given by Reflexis Systems in the problem statement. Change input directory format accordingly in the `pd.read_csv` statement]





Thank You