Retail Sales Forecasting using Neural Networks

Bhavish Kumar, Sai Praharsha Devalla
School of Information Studies, Syracuse University
Dr. CK Mohan
Department of Electrical Engineering & Computer Science, Syracuse University

Abstract

Effective Store & Inventory management is crucial for successfully operating a retail store. Proactive anticipation of the upcoming demand will help the retailer's stock up on products, accordingly, resulting in revenue & profit maximization. In this paper we propose to build forecast models by using neural networks to help the retail store accurately predict the sales for the upcoming week. We have compared and contrasted the performance of LSTMs[1], GRUs[2], RNNs[3] (BPTT), FFNNs[4] (Plain Backpropagation) along with regular Linear Regression and Random Forest Regression forecast models. The Mean Squared Error (MSE) and Computational Effort metrics have been used for model comparison.

1. Introduction

Making accurate demand forecasts is crucial for retailers in order to enhance their inventory planning, competitive pricing etc. Understocking of inventory will negatively affect the store profits whereas overstocking may result in losses caused by expiration of perishable products. Both understocking and overstocking are a consequence of poor demand estimation. Typically, the retail business is highly volatile, with a large fluctuation in sales caused by holiday season, store performance and several other internal/external factors. As a result, sales/demand forecasting of retail stores is a highly challenging process and requires the use of some advanced & powerful techniques. "The data in this domain also carries various challenges, such as highly non-stationary historical data, irregular sales patterns, sparse sales data, highly intermittent sales, etc." [1] Moreover, the sales of stores located in a region can be correlated to each other in a way that an increase in sales of one store may cause a decrease/increase in sales of another store. The weekly fluctuations/trend in store level sales of this dataset will be discussed in the upcoming data description section.

The forecasting techniques such as Exponential Smoothing [2] and ARIMA [3] are univariate techniques which take only the target variable (sales) into consideration and forecasts each time series separately by ignoring other predictor attributes. Thus, we are interested in performing forecasting on the multivariate "Cross-sectional time series"[4] by using RNNs, LSTMs and GRUs which are a special group of neural networks (NN) that are well suited for multivariate time series forecasting problems. [5,6,7,8]

2. Problem & Data Description

This dataset contains data of 45 Walmart stores located in different regions. Each store contains many departments, and Walmart is interested in projecting the sales for each department in each store at a weekly level. The goal of this project is to predict the weekly sales of Walmart stores and departments at a store & department level by considering the sales of the previous 'N' weeks and other external factors such as 'Temperature', 'Fuel Price', 'Store Type', 'Store Size', 'Consumer Price Index', 'Unemployment Rate', 'Is Holiday Week' etc. which could have a potential impact on the sales.

The Time Series Dataset contains 3 CSV files (*'features.csv', 'stores.csv' & 'train.csv'*) that have all the predictor attributes and the target attribute (*'weekly sales'*) data available across the 3 files. This dataset contains ~421K rows of weekly sales for 99 departments and 45 stores of 3 store types across 143 weeks. The *'train.csv'* is a historical dataset which covers weekly sales of every store & department between 2010-02-05 to 2012-11-01. It contains the attributes, 'store number', 'department

[1] Long Short Term Memory Networks

[2] Gated Recurrent Unit Networks

[3] Recurrent Neural Networks with Backpropagation Through Time

[4] Feed Forward Neural Networks with Plain Backpropagation

number', 'weekly sales', 'IsHoliday', and 'Date'. "The stores.csv file contains anonymized information about the 45 stores, indicating the type and size of store." [9]. The features.csv file contains additional data related to the store, and regional activity for the given dates such as 'Temperature', 'Fuel Price', 'Consumer Price Index' and 'Unemployment Rate'.

From the below shown Fig. 1 We can observe a fluctuating seasonal trend across all the 20 stores. As expected, we see a periodic huge peak in sales between the weeks 40 to 50 & 90 to 100

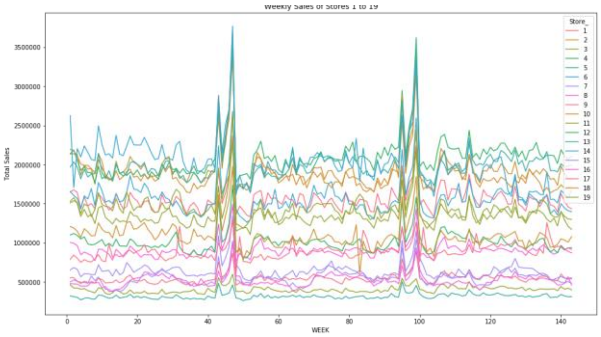


Fig. 1 Weekly Sales of individual stores 1 to 20

Another interesting observation can be made from the below shown Fig. 2, which shows that average sales is highest for Stores A, followed by B Stores and lastly Store C. Thus, we can be assured that Store Type has a significant impact on sales.

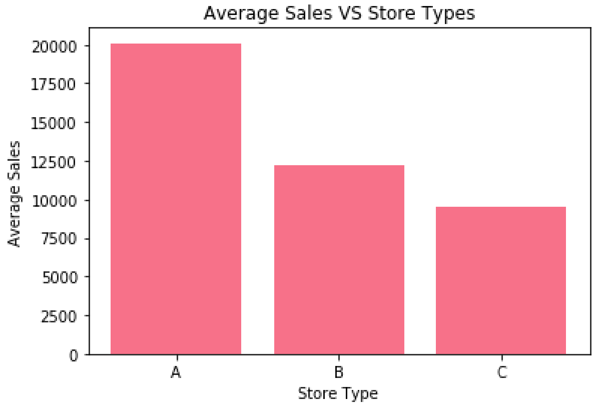
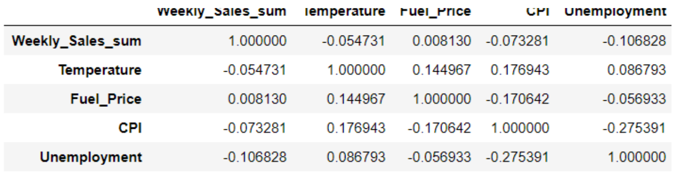


Fig. 2 Average Sales across 3 Store Types



	Weekly_Sales_sum	Temperature	Fuel_Price	CPI	Unemployment
Weekly_Sales_sum	1.000000	-0.054731	0.008130	-0.073281	-0.106828
Temperature	-0.054731	1.000000	0.144967	0.176943	0.086793
Fuel_Price	0.008130	0.144967	1.000000	-0.170642	-0.056933
CPI	-0.073281	0.176943	-0.170642	1.000000	-0.275391
Unemployment	-0.106828	0.086793	-0.056933	-0.275391	1.000000

Fig. 3 Correlation Matrix of dataset

From the above shown Correlation Matrix, we can observe that the 'Unemployment' rate variable shares the relatively strongest correlation with the 'weekly sales' target variable, whereas 'Fuel Price' has the relatively weakest correlation.

3. Approach & Algorithms Used

- The first step was Data Wrangling where all the data quality issues such as missing values, duplicates & outliers were taken care of. In our dataset, the 'stores' & 'weekly sales' csv files did not have any data quality issues. "Since a lot of Machine Learning and Deep Learning algorithms require the data to be numeric"[10], the categorical variable 'Store Type' of the stores dataset was One Hot Encoded to obtain 3 binary variables. Next, the 'features' dataset had more than 50% missing values in the Markdown attributes, which were dropped to avoid inducing bias by doing imputation. The attributes 'CPI' & 'Unemployment Rate' had 585 missing values which were imputed with the median values of the column. Imputing with median may not be the most accurate imputation and may induce bias in the data and also in the estimates produced by the model. But the main advantage of this method of imputation over other methods such as KNN[5] imputation or MICE[6] imputation is that this method is computationally very inexpensive and works very fast in comparison to KNN or MICE imputation, especially when dealing with large datasets. Because of this reason, we decided to go ahead with the median method of imputation. We then merged the 3 cleaned dataframes to pull in all the attributes

[5] K Nearest Neighbors imputation, finds K nearest neighbors for a missing datapoint from the entire dataset and fills the missing datapoint with the most frequently occurring value amongst the K nearest neighbors

[6] Multiple Imputation by Chained Equations

into a single main Dataframe which would be used for the next step.

- On the next step we performed Exploratory Data Analysis as explained in the Data Description section above.
- The original data which is at store-department level had to be cut down on the granularity by aggregating the data to store level for producing Store Level Forecasts. We scaled the data using a MinMax scaler since, "many machine learning algorithms work better when features are on a relatively similar scale and close to normally distributed."[11]
- We also created a new 'Week of the Year' attribute (1 to 52) to capture the seasonality in the data and the impact of time of the year on sales.
- Next, we Trained the following Forecast Models on the aggregated Store level dataset:
    - Initially we used regular Linear Regression & Random Forest Regression to help draw out a comparison & be able to infer that special Neural Network algorithms will be able to forecast better.
    - Next, we experimented with Feed Forward Neural Networks (FFNN) using Plain Backpropagation
    - We then experimented with special Neural Network models such as Recurrent Neural Networks (RNN) using Backpropagation Through Time (BPTT), Long Short Term Memory (LSTMs) & Gated Recurrent Unit (GRUs) Networks. These special neural network models are specially designed and meant for Multivariate Time Series Forecasting problems like this.
- All the Store Level Forecasting models involved the following two types of data preprocessing & experimentation in common:
    - **Approach One - 45 Models:** In the first approach, we treated each store as a different entity and trained 45 distinct models for each of the 45 stores using the corresponding store's data subset. The moving windows were used on each subset to create attributes containing lagged values, which were also used as predictor attributes to train the multivariate model. The Moving Window (MW) strategy transforms a time series dataset into a regular dataset with input attributes and the output attribute.
    - **Approach 2 - Single Unified Model:** In the second approach, we used the unified dataset containing all 45 stores to train one single forecast model common for all the stores. But moving window lagged attributes were created on each store wise subset to avoid providing lag values of one store to another store. All the moving window processed subsets were unified to create one single dataset used to train the single unified forecast model common for all 45 stores.
- The shape of the processed (generated with Moving Windows) Train & Test datasets used were different for special Neural Networks (LSTMs, GRUs & RNNs) and regular FFNN & Regression models. For LSTMs, GRUs and RNNs, the processed train & tests datasets used were 3 Dimensional objects where the 3 dimensions are [batch_size(no of observations per batch); moving window sequence length of each attribute; number of attributes]. The Target object was 2D [Batch size; Number of Target attributes].

  Whereas for regular FFNNs & Regression models the Train & Test datasets used were regular 2D dataframes having all the explanatory attributes along with the lag variables as input and a 1D target attribute.
- After completing Store Level forecasting, we lowered the granularity and Trained the LSTM & GRU Forecast Models on unaggregated store-department level dataset to make Store-Department Level forecasts. Here only the single unified model approach was used since it is not feasible to train 45 x 99 = 4,455 individual models for 4,455 *Store-Department* subsets. For this department-store level forecasting,

we Experimented with LSTM & GRU models which were identified as best performing models for Store level forecasting. We created moving window sliders of sequence length 10 for every store-dept subset separately, one at a time to avoid leaking past data from one dept to another dept. Then we concatenated all the processed 3D Numpy Arrays to create one single Train & Test datasets which was used to create train & test loader objects.

- Lastly, we compared & contrasted the evaluation results to draw conclusions

Discussing each of the used algorithms:

3.1. Multiple-Linear Regression

Linear regression is a way to model a relationship between an explanatory variable and a dependent variable by fitting a linear equation to the data. If in case there are multiple explanatory variables it is termed as multiple linear regression. The model tries to fit the regression line in such a way that sum of squared errors is minimized.

$$\widehat{y_i} = b_0 + b_1 x_{i1} + \ldots + b_p x_{pi}$$

$b_0\ is\ intercept, b_1, \ldots, b_p\ are\ the\ coefficients$

We have experimented with 3 different variants of Linear Regression models for Store Level Forecasting. Two variants under the “45 Models” category which are:

- **Model 1A**: Using all input attributes & lagged moving window sequence of length 5
- **Model 1B:** Using only ‘Week of the year’ & ‘Is Holiday’ attributes with lagged moving window sequence length 5

The 1 variant under the single unified category which used sequences of length 5 is:

- **Model 2:** 1 unified model with moving window sequence of length 5

From the below Fig. 4, we can observe that Linear Regression is not capable of solving this problem as the store level forecasted sales do not align with the actual sales. Hence, we try other more advanced techniques.

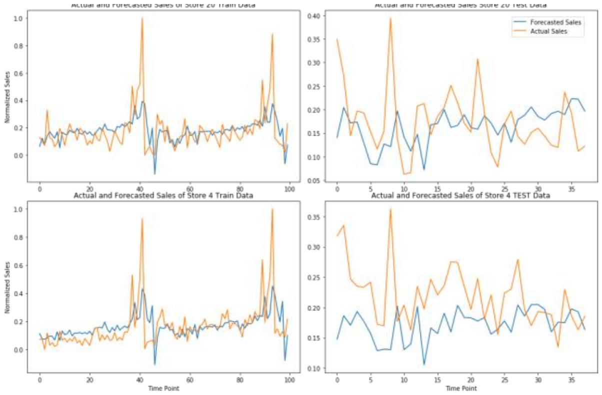


Fig. 4 Actual Vs Forecasted Sales using Linear Regression “Model 2” on two stores

3.2. Random Forest Regression

Random Forest Regressor is an ensemble method consisting of multiple decision trees which combines the predictions made by multiple decision trees and each node split is generated using a selected number of input features [12]. The decision trees with bagging, forms a special case of random forest which gives the randomness to the model compared to bagging by randomly selecting the samples to build the individual tree with replacement. The primary unit of random forest is decision trees, it uses the splitting criteria as residual sum of squares for regression models.

$$RSS = \sum_{left} (y_i - y_L^*)^2 + \sum_{right} (y_i - y_R^*)^2$$

where $y_L^* = mean\ y(value)\ for\ left\ node$

$$y_R^* = mean\ y(value)\ for\ right\ node$$

We have experimented with the same 3 variants as that of Linear Regression models for Store Level Forecasting using Random Forest Regression models also. The same 3 methods as Linear Regression of “Model 1A”, “Model 1B” & “Model 2” have been used for RF Regression.

From the below Fig. 5, we can observe that the unified Random Forest Regression model is overfitting on the train data as it is doing well with the store level forecasts on the Train data alone and performing relatively poorly on the test data. Hence, we proceed to other more suitable techniques.

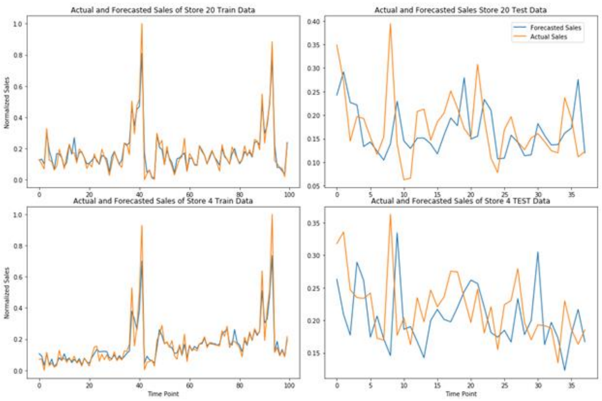


Fig. 5 Actual Vs Forecasted Sales using unified RF model "2" on two stores

3.3. Feed Forward Neural Networks

A Feed Forward Neural Network is a regular Neural Network where the information moves only in the forward direction from the input nodes to the hidden nodes and to the output nodes as shown in the below figure.

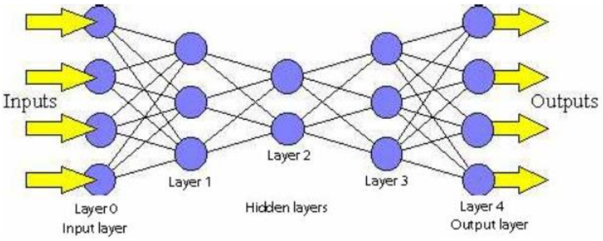


Fig. 6 FFNN with fully connected layers

The FFNN is trained in batches over multiple epochs depending on the number of epochs required for the model to converge. Training happens through backpropagation by applying Gradient Descent on the Error (Output - Target) using Chain Rule. "The goal with backpropagation is to update each of the weights in the network so that they cause the actual output to be closer to the target output, thereby minimizing the error for each output neuron and the network as a whole."[13] Again, the Mean Squared Error is used as a Loss Function since the target variable ('sales') is a continuous numeric variable. Ultimately, the aim of a FFNN is to learn an approximate function 'f' which maps a set of input values to output values.

We have experimented with 4 different variants of FFNNs for Store Level Forecasting. Two variants under the "45 Models" category which are:

- **Model 1A**: Using all input attributes & lagged moving window sequence of length 10
- **Model 1B:** Using only 'Week of the year' & 'Is Holiday' attributes with lagged moving window sequence length 10

The 2 variants under the single unified category which used sequences of length 5 are:

- **Model 2A:** Shallow Network with 1 hidden layer
- **Model 2B:** Deep Network with 3 hidden layers

The Train & Test MSE along with the required computational effort for each of these variants will be discussed in the Results section.

From the below shown Actual (orange line) & Forecasted Sales (blue line) vs time graph in Fig 7. produced for model variant 1B (best performing FFNN variant), it is clearly evident that FFNNs are not suitable for this kind of a time series forecasting problem. Hence, we explore other special neural networks next.

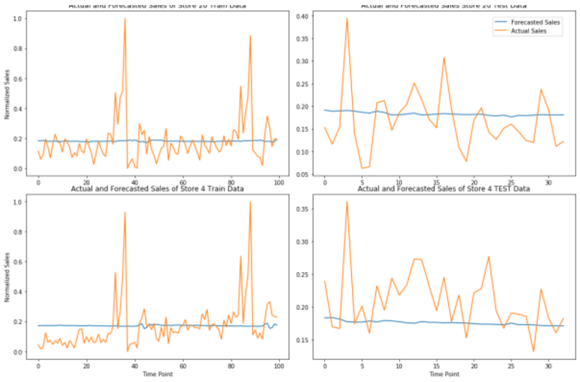


Fig. 7 Actual & Forecasted Sales of 2 stores using FFNN model '1B'

3.4. Recurrent Neural Networks with Backpropagation Through Time

These are a special type of Neural Networks that use Backpropagation Through Time (BPTT) to calculate the gradients and update the weights. BPTT is different from traditional plain Backpropagation which is used in FFNNs. With BPTT, the gradients are calculated at every time instant from a given time instant all the way till time instant t = 0.

We have experimented with 3 different variants of RNNs for Store Level Forecasting. Two variants under the “45 Models” category which are:

- **Model 1A**: Using lagged moving window sequences of length 5
- **Model 1B:** Using lagged moving window sequences of length 10

Model 2: The 1 variant under the “single unified category” which used sequences of length 10 was used to train a Neural Network with 1 RNN layer and 1 fully connected output layer.

From the below shown Actual (orange line) & Forecasted Sales (blue line) vs time Fig 8. produced for the “model 2” variant (best performing RNN variant), we can observe that RNNs have produced the best results in comparison to all the other models so far and it is doing a better job at detecting the fluctuations in sales. Although, there is scope for improvement

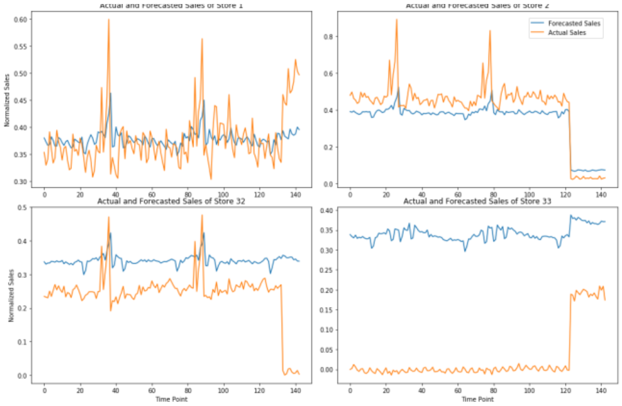


Fig. 8 Actual & Forecasted Sales of 2 stores using RNN model ‘2’

One major drawback of RNNs is the vanishing/exploding gradient problem which can severely limit the model performance. This happens when either a very small gradient value or a large gradient value gets multiplied repeatedly (Chain Rule) when backpropagating all the way till time instant t = 0. This causes the resultant gradient value to be either very large or very small and the model diverges from the minima as a consequence. “This has led to the development of LSTMs and GRUs, two of the currently most popular and powerful models used in Multivariate Time Series Forecasting and NLP”.[14]

3.5. Long Short Term Memory Networks

LSTMs are a special kind of RNNs which are very well suited for Multivariate Time Series Forecasting problems like these. They overcome the problem of vanishing/exploding gradient and are also capable of storing previous information for a long period of time. The LSTMs have forget gates and input gates which cause the Stability-Plasticity trade off. This tradeoff is about the proportion of newly arriving information to be considered and proportion of old information existing in the memory to be considered. Higher the stability, higher proportion of the current state is maintained.[15]

For Store Level Forecasting using LSTMs, we have experimented with the same 3 variants as that of RNNs. Even for LSTMs, the variant “Model 2” performed much better than the other 2 variants. The below Fig 9. shows the Actual & Forecasted Sales vs Time for “model 2” variant (best performing LSTM variant).

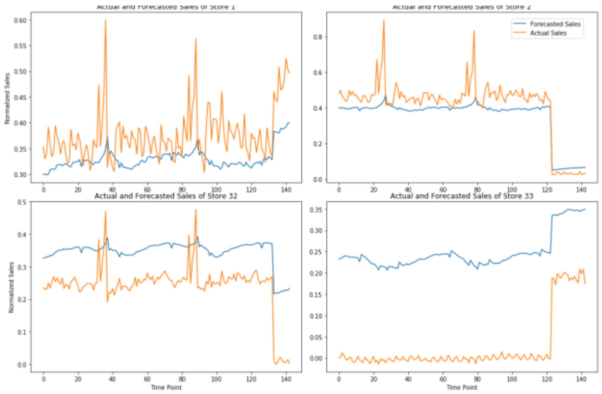


Fig. 9 Actual & Forecasted Sales of 2 stores using LSTM model ‘2’

Since LSTMs were one of the best performing models for Store level forecasts, we have also used the LSTM network model for our Department-Store level forecasts. We trained a single unified 1 layer LSTM network model with 15,400 weight updates to obtain the Actual & Forecasted Sales results as shown in the below Fig 10.

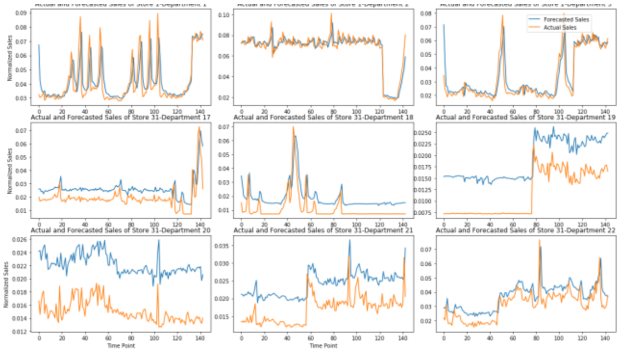


Fig. 10 Actual & Forecasted Sales of 6 store-departments using unified LSTM model ‘2’

We can observe that the 1 layer LSTM network model is doing a pretty good job at forecasting the sales even at the department-store level and is able to do a good job with detecting all the peaks and dips in sales. Hence, we can be assured that LSTMs are capable of solving this problem of forecasting sales at a store and department-store level, which the other models were not able to do.

3.6. Gated Recurrent Unit Networks

GRUs also work in a very similar way to LSTMs and can be considered as a variant of LSTM. Even GRUs also help in solving the problem of vanishing/exploding gradients and they also produce equally excellent results. Even GRUs use, update gate and reset gate similar to LSTMs. Again, the update gate and reset gate vectors decide on the information that should be passed to the output. “The special capability of GRUs is that they can keep information from long ago, without it being lost through time and they can also remove information which is irrelevant to the prediction.”[16]

For Store Level Forecasting using GRU, we have experimented with the same 3 variants as that of LSTM, with one additional 4th variant of “single unified model” category. With this 4th variant “2B”, we experimented by adding an additional GRU layer to check if the already good performance improves even further. Even for GRUs, the variant “Model 2A” (single unified model with 1 GRU layer) performed much better than the other variants. Adding an extra GRU layer did not make any significant difference to the performance.

The below Fig. 11 shows the Actual & Forecasted Sales vs Time for “model 2A” variant (best performing GRU variant).

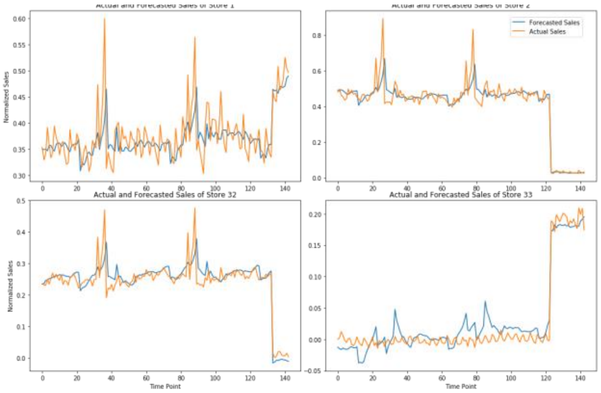


Fig. 11 Actual & Forecasted Sales of 4 stores using GRU model ‘2A’

Once again, we observe a pretty good overlap with the Actual & Forecasted sales at Store level, upon using a unified GRU model with 1 layer.

Once again, since GRUs were also one of the best performing models for Store level forecasts, we have experimented with a similar single unified model on networks having 1 GRU layer and 2 GRU layers for our Department-Store level forecasts. Here again, we observed that adding an additional GRU layer increased the computational effort but did not make any difference to the model performance.

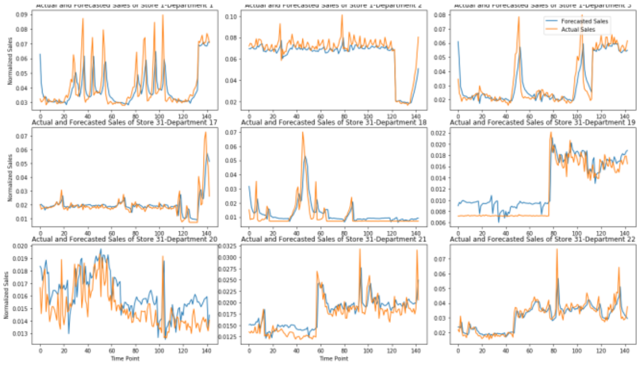


Fig. 12 Actual & Forecasted Sales of 6 store-departments using unified GRU model ‘1’

From Fig. 12, we can observe that the 1 layer GRU network model is also doing a good job at forecasting the sales at the department-store level and similar to LSTM, GRU is also capable of solving this problem of forecasting sales at a store and department-store level.

4. Results

In this section we assess the Train & Test Mean Squared Error (MSE) & the computational effort (Number of Weight Updates) for each variant of every algorithm discussed in the above section. The MSE is the loss function for this problem which we are trying to optimize. We built Linear regression and random forest regression models to draw out a comparison between these and Neural networks forecasting ability.

The definition of each model variant has been defined in the above section under the corresponding algorithm.

First let us look at the results of every variant of an algorithm individually and then compare the results across multiple algorithms with a single table.

4.1. Linear Regression

Out of the 3 Linear Regression variants, the Model 2 (unified model) has the best MSE performance for Store Level Forecasts as shown in Table 1. So the model 2 is run through 30 trials to obtain the mean and standard deviation of train_mse and test_mse for comparing it with other algorithms.

Methods	Train MSE	Test MSE
Model 1A	0.015096	0.011964
Model 1B	0.016425	0.007986
Model 2	0.009678	0.007462

Table 1. Linear Regression Store Level Forecast results

4.2. Random Forest Regression

Once again, the Model 2 has the best performance for Store Level Forecasts, out of the 3 methods even for Random Forest models.

Methods	Train MSE	Test MSE
Model 1A	0.015096	0.011964
Model 1B	0.016425	0.007986
Model 2	0.009678	0.007462

Table 2. Random Forest Regression Store Level Forecast results

4.3. Feed Forward Neural Networks

In the case of a FFNN, Model 1B (45 separate models) has better performance than the other 3 variants. This is different from every other model where the unified model performs better. However, these MSE results are very poor and as mentioned above, we can infer that FFNNs are not suitable for this problem.

Methods	Train MSE	Test MSE	NO of Weight Updates
Model 1A	0.027492	0.019572	2700
Model 1B	0.048121	0.017240	2700
Model 2A	0.026903	0.026977	300
Model 2B	0.026864	0.027655	300

Table 3. FFNN Store Level Forecast results

4.4. Recurrent Neural Networks

The Model 2 variant has the best MSE performance out of the two and it takes the least amount of computational effort too. Here again, model 2 is run through 30 trials to obtain the mean and standard deviation of train_mse and test_mse for comparing it with other algorithms.

Methods	Train MSE	Test MSE	NO of Weight Updates
Model 1A	0.016149	0.018166	4500
Model 1B	0.024598	0.022617	3000
Model 2	0.004076	0.003316	80

Table 4. RNN Store Level Forecast results

4.5. LSTMs

Now coming to the store level forecasts made with LSTM models. Out of the 3 variants, even LSTMs show the best results for Model 2 (single unified model) which produced the best results and is also the most computationally inexpensive variant too. So, we have also run 30 trials of the single unified LSTM model and taken the mean and standard deviations of Train and Test MSE to make a fair comparison with other models.

Methods	Train MSE	Test MSE	NO of Weight Updates
Model 1A	0.023502	0.021669	1800
Model 1B	0.031340	0.021027	900
Model 2	0.007351	0.012752	80

Table 5. LSTM Store Level Forecast results

4.6. GRUs

Out of the 4 different variants here, once again, both the 45 models methods are not doing as well as the single unified model method. Amongst the 2 single unified models also, we can observe that increasing the number of layers is not making any significant difference to the model performance. The single unified model with 1 GRU layer is yielding the best results and hence we have run 30 trials of this model to obtain the mean & standard deviations of MSEs.

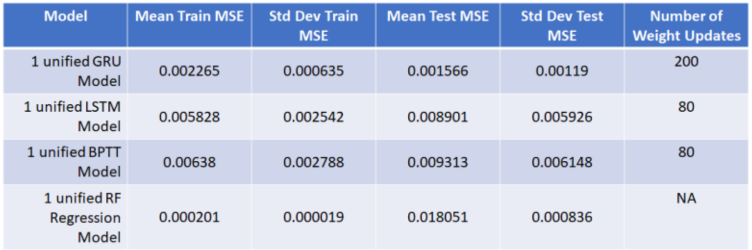
Methods	Train MSE	Test MSE	NO of Weight Updates
Model 1A	0.028789	0.032823	900
Model 1B	0.030605	0.034109	900
Model 2A	0.003316	0.003070	200
Model 2B	0.004607	0.003707	200

Table 6. GRU Store Level Forecast results

Next, we compare all the different algorithms used for Store Level Forecasting.

4.7. Comparison of Best Store Level Forecast Models

The below table shows the Top 4 Store Level Forecast models and this table can be used to compare and contrast the best performing models. 30 trials have been run for each model and their Mean and Standard deviation of MSEs have been recorded in this table shown below. The unified GRU model has the best results with the lowest mean test MSE. Also, the computational effort is only marginally higher for GRU models, while the mean & standard deviation of MSE is much better. The Random Forest Regression model has a very low train MSE, but a high Test MSE which means that the RF regression model is overfitting on train data.



Model	Mean Train MSE	Std Dev Train MSE	Mean Test MSE	Std Dev Test MSE	Number of Weight Updates
1 unified GRU Model	0.002265	0.000635	0.001566	0.00119	200
1 unified LSTM Model	0.005828	0.002542	0.008901	0.005926	80
1 unified BPTT Model	0.00638	0.002788	0.009313	0.006148	80
1 unified RF Regression Model	0.000201	0.000019	0.018051	0.000836	NA

Table 7. Store Level Forecast results of top models

4.8. Comparison of Department-Store Level Forecast Models

Now coming to a lower level of granularity, by assessing the results of our Department-Store level forecasts models.

As discussed in the above Approach section, the 3 experiments with GRU & LSTM models and their corresponding results have been listed in the below Table 8. Once again, we took the mean and standard deviation of 10 MSEs obtained through 10 trials run on each model. Based on the results, we can see that unified GRU & LSTM models with 1 layer have the best performance and take minimum computational effort too. Once again, increasing the number of layers makes no difference to the MSE scores even for the store-department level forecasts.

The low standard deviation of MSE results also means that the models have a good stability and the results do not fluctuate a lot with every run.

Model	Mean(Train MSE)	Mean(Test MSE)	StdDev (Train MSE)	StdDev (Test MSE)	No Weight Updates
GRU 1 layer	0.00019	0.00023	0.00013	0.00026	15400
GRU 2 layers	0.00033	0.00048	0.00007	0.00012	67375

LSTM1 layer	0.00024	0.00017	0.00006	0.00013	15400

Table 8. Department - Store Level Forecast results

Thus, we conclude that single layer LSTM & GRU models trained on this data with sliding window sequence of length 10 produce the best results in terms of both MSE & Computational Effort. Therefore, the specially designed LSTM & GRU neural network models are well suited for this kind of a multivariate time series forecasting problem. Both store level & department-store level sales estimates can be successfully made using the approaches & methods discussed above.

5. Discussion

One of the biggest challenges was creating an aggregated Store level Dataset containing weekly sales at a store level as we had a more granular data which would result in building too many models. Another major challenge was in the data preprocessing section where the moving window lags had to be created for every store and store-department subsets separately. Later the processed subsets had to be unified for training a unified single model.

For the future work CNN-LSTM networks can be built to try and forecast the sales of the stores. Also, we can try to forecast the sales for a department across all the stores by aggregating to department level in addition to the store level aggregation which we have already implemented. In this way vendors can know which department is generating higher revenue and based on that they can make suitable changes to improve the sales. Moreover, we can also try to implement sequence to sequence learning to forecast the weekly sales, for multiple weeks into the future.

Acknowledgements

We would like to thank our professor Dr CK Mohan for being an excellent instructor who helped us gain all the knowledge that is required to complete this project. He also guided us through the entire project by providing complete instructions on the ways to approach this project.

References

[1]. *Sales Demand Forecast in E-commerce using a Long Short-Term Memory Neural Network Methodology*[Online]. https://arxiv.org/pdf/1901.04028.pdf

[2]. Hyndman, R. et al., 2008. Forecasting with Exponential Smoothing: The State Space Approach, Springer Science & Business Media.

[3]. Box, G.E.P. et al., 2015. Time Series Analysis: Forecasting and Control, John Wiley & Sons.

[4].[Online].https://analystprep.com/cfa-level-1-exam/quantitative-methods/time-series-data-vs-cross-sectional-data/.

[5].[Online].http://www.wildml.com/2015/10/recurrent-neural-networks-tutorial-part-3-backpropagation-through-time-and-vanishing-gradients/

[6].[Online].http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[7].[Online].http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[8].[Online].https://www.semanticscholar.org/paper/Learning-to-Forget%3A-Continual-Prediction-with-LSTM-Gers-Schmidhuber/11540131eae85b2e11d53df7f1360eeb6476e7f4?p2df

[9].[Online].https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data

[10].[Online].https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/

[11].[Online].https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02#:~:text=MinMaxScaler%20preserves%20the%20shape%20of,MinMaxScaler%20is%200%20to%201.

[12].Tan PN, Steinbach M, Karpatne A, Kumar V (2013) Introduction to data mining, 2nd edition. Pearson, London.

[13][Online].https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

[14][Online].http://www.wildml.com/2015/10/recurrent-neural-networks-tutorial-part-3-backpropagation-through-time-and-vanishing-gradients/

[15][Online].http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[16][Online].https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be

APPENDIX

After submitting the Project Implementation on 1st December, we incorporated the following additional changes:

- Took the Computational Effort in the form of number of weight updates into consideration, while comparing the models and evaluating the results.
- Conducted multiple trials of every model and recorded the Mean & Standard Deviations of MSEs to make a fair comparison between multiple algorithms.
- Implemented LSTM network models for Department-Store level forecasts, in addition to the already implemented GRU models

Link to GIT Repository: https://github.com/bhavish2207/RETAIL-SALES-FORECASTING-using-Recurrent-Neural-Networks

Link to Presentation Video: https://www.youtube.com/watch?v=7uKPiMz0ey8&t=608s