
Error Modelling of Demand Patterns to Improve Forecasting Accuracy

School of Computer Science and Engineering

Author: Sa Ziheng

Supervisor: Jagath Chandana Rajapakse



Research Gap + Objectives



Irregularly Structured Demand Patterns



Error Modelling

- Use Residuals as Features
- Study its Impact
- On LSTM and Transformer

Environment Setup

- Python Version 3.9.7
- Pytorch Version 2.0.1
- GPU Tesla P100-PCIE-16GB
- Random Seed = 1345



Experimental Flow

1. Select a random time series data from a dataset
2. Perform hyperparameter tuning for baseline LSTM and transformer networks with the chosen time series data
3. With the hyperparameters obtained from step 2:
 - 3.1. conduct Experiment 1.
 - 3.2. conduct Experiment 2.
 - 3.3. conduct Experiment 3.
4. Select a random subset of time series data from the dataset
5. Repeat step 3 for each time series data.
6. Compare results between Experiments 1, 2 and 3.



Data Description

M5-Competition dataset from Kaggle

- *sales_train_validation.csv* (30490 x 1919 (1913))
- *sales_train_evaluation.csv* (30490 x 1947 (1941))
- *calendar.csv*
- *sell_prices.csv*

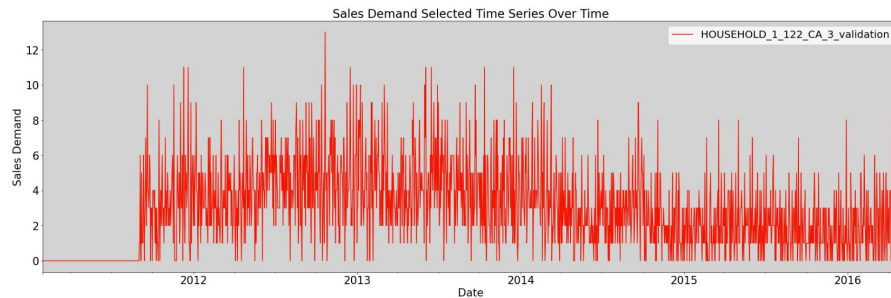
Use 1913 time steps as training/ validation data and reserve 28 time steps as testing data.

Data Selection

Select a random time series data (row **6780**), which displays reasonable variability.

Select data from 1 September 2011 onwards (1698 timestamps), due to constant 0 sales from the beginning

Randomly select another **450** rows of time series data.



Data Preprocessing

- Min-max normalization to $[-1, 1]$
- Convert to single-step format
- Batch into data loaders

Hyperparameter Tuning (LSTM)

3-Fold Cross Validation, with initial hyperparameters: $N = 512$, $L = 1$, $W = 28$.

Batch Size (BS)	16, 32, 64, 128, 256, 512
Hidden Neurons Number (N)	32, 64, 128, 256, 512, 1024
Hidden Layers Number (L)	1, 2, 3, 4, 5, 6
Window Size (W)	7, 28, 91, 182, 364

Optimal hyperparameters: $BS = 16$, $N = 512$, $L = 2$, $W = 91$

Hyperparameter Tuning (Transformer)

3-Fold Cross Validation, with initial hyperparameters: $N = 512$, $L = 1$, $W = 28$.

dmodel	64, 128, 512, 1024
h	2, 4, 8
N	2, 4, 6, 8
Batch Size (BS)	16, 32, 64, 128, 256
Window Size (W)	7, 28, 91, 182, 364

Optimal hyperparameters: $dmodel = 64$, $h = 4$, $N = 2$, $BS = 32$, $W = 28$

Experiment 1: Baseline Models

Given 1698 time stamps available, we use 1670 days as training set and the last 28 days as validation set.

$$\bar{x}_{i+1} = f_{baselineLSTM}(x_i, x_{i-1}, x_{i-2}, \dots, x_{i-90}; \theta)$$

LSTM

- LSTM network is trained on 1579 (1670 - 91) sets of windows and validated against the last 28 days of data.
- Predict 1607 days of demand to obtain train score.
- Forecast 28 days into the future with newly predicted demand continuously updated into the window to obtain test score.

Experiment 1: Baseline Models

Given 1698 time stamps available, we use 1670 days as training set and the last 28 days as validation set.

$$\bar{x}_{i+1} = f_{baselineTransformer}(x_i, x_{i-1}, x_{i-2}, \dots, x_{i-27}; \theta)$$

Transformer

- Transformer network is trained on 1642(1670 - 28) sets of windows and validate against the last 28 days of data.
- Predict 1670 days of demand to obtain train score.
- Forecast 28 days into the future with newly predicted demand continuously updated into the window to obtain test score.

Experiment 2:

Error Models + predicted residuals

General Idea

Utilize residuals obtained from respective network models in experiment 1 and we use these residuals as feature in addition to the sales demand.

$$\hat{x}_{i+1}, \hat{e}_{i+2} = f_{errormodel}(x_i, x_{i-1}, x_{i-2}, \dots, x_{i-L+1}; e_{i+1}, e_i, \dots, e_{i-L+2}; \theta),$$
$$e_i = x_i - \bar{x}_i$$

Residuals are shifted left by one time stamp.

Residuals are Min-max normalized to $[-1, 1]$.

Experiment 2

Training Phase

L window is used -> first L - 1 residuals are unavailable.

Use sample mean to fill up empty values, given by the equation:

$$\bar{e} = \frac{1}{n} \sum_{i=L+1}^T e_i$$

Experiment 2

$T - L - 1$ training/ validation data remaining instead of the original $T - L$ training/ validation data, due to residuals being shifted left by one timestamp.

Input:

x_1	...	x_{L-1}	x_L	x_{L+1}	...	x_{T-L-1}	x_{T-L}	...	x_{T-2}
\bar{e}	...	\bar{e}	e_{L+1}	e_{L+2}	...	e_{T-L}	e_{T-L+1}	...	e_{T-1}

Output:

			\hat{x}_{L+1}	\hat{x}_{L+2}	...	\hat{x}_{T-L}	\hat{x}_{T-L+1}	...	\hat{x}_{T-1}
			\hat{e}_{L+2}	\hat{e}_{L+3}	...	\hat{e}_{T-L+1}	\hat{e}_{T-L+2}	...	\hat{e}_T

Target:

			x_{L+1}	x_{L+2}	...	x_{T-L}	x_{T-L+1}	...	x_{T-1}
			e_{L+2}	e_{L+3}	...	e_{T-L+1}	e_{T-L+2}	...	e_T

T = the total timestamps for training data, L = the lookback period, \hat{x}_i = the predicted sales demand and \hat{e}_i = the predicted residuals.

Experiment 2

Testing Phase

Newly predicted output is continuously updated into the window whenever actual sales or residuals data is unavailable.

Input:

x_{T-L+1}	...	x_{T-1}	x_T	\hat{x}_{T+1}	...	$\hat{x}_{T+\tau-L}$	$\hat{x}_{T+\tau-L+1}$...	$\hat{x}_{T+\tau-1}$
e_{T-L+2}	...	e_T	\hat{e}_{T+1}	\hat{e}_{T+2}	...	$\hat{e}_{T+\tau-L+1}$	e_{T-L+1}	...	$\hat{e}_{T+\tau}$

Output:

				\hat{x}_{T+1}	\hat{x}_{T+2}	...	$\hat{x}_{T+\tau-L+1}$	$\hat{x}_{T+\tau-L+2}$...	$\hat{x}_{T+\tau}$
				\hat{e}_{T+2}	\hat{e}_{T+3}	...	$\hat{e}_{T+\tau-L+2}$	$\hat{e}_{T+\tau-L+3}$...	$\hat{e}_{T+\tau+1}$

T = the total timestamps for training data, L = the lookback period, \hat{x}_i = the predicted sales demand, \hat{e}_i = the predicted residuals and τ = the future timestamp period = 28.

Experiment 3: $\hat{x}_{i+1}, \hat{e}_{i+2} = f_{\text{error model}}(x_i, x_{i-1}, x_{i-2}, \dots, x_{i-L+1}; e_{i+1}, e_i, \dots, e_{i-L+2}; \theta)$

Error Models + sample mean residuals

General Idea

Training phase is the same as Experiment 2.

During testing phase, sample mean residuals instead of predicted residuals are used whenever the residual data is unavailable.

Input:

x_{T-L+1}	...	x_{T-1}	x_T	\hat{x}_{T+1}	...	$\hat{x}_{T+\tau-L}$	$\hat{x}_{T+\tau-L+1}$...	$\hat{x}_{T+\tau-1}$
e_{T-L+2}	...	e_T	\bar{e}	\bar{e}	...	\bar{e}	\bar{e}	...	\bar{e}

Output:

				\hat{x}_{T+1}	\hat{x}_{T+2}	...	$\hat{x}_{T+\tau-L+1}$	$\hat{x}_{T+\tau-L+2}$...	$\hat{x}_{T+\tau}$
				\hat{e}_{T+2}	\hat{e}_{T+3}	...	$\hat{e}_{T+\tau-L+2}$	$\hat{e}_{T+\tau-L+3}$...	$\hat{e}_{T+\tau+1}$

T = the total timestamps for training data, L = the lookback period, \hat{x}_t = the predicted sales

Analysis Method

Single-item

Visualizations and comparison between **RMSE** and **R2** score.

450-item

Compare test statistics and confidence intervals

- Paired 2-group comparison with Hotelling's T-squared statistic with 95% confidence
 - Test for the null hypothesis where both evaluation metrics are the same.
- Bonferroni simultaneous confidence intervals (SCIs)
 - Obtain the 95% simultaneous confidence interval of the difference in evaluation metrics.

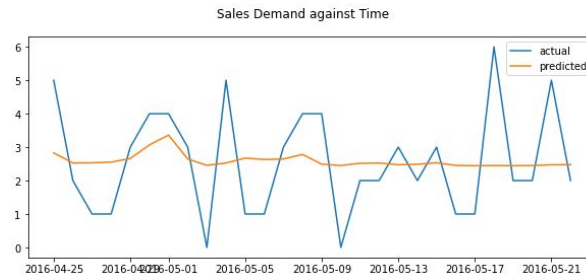
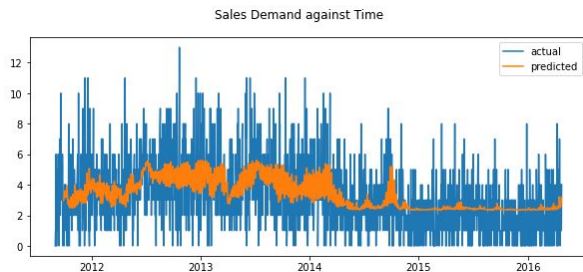


Summary of Test Score: 1 Item

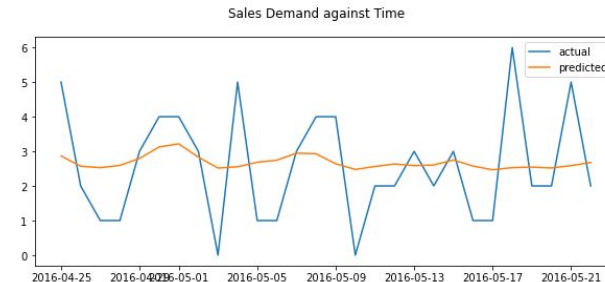
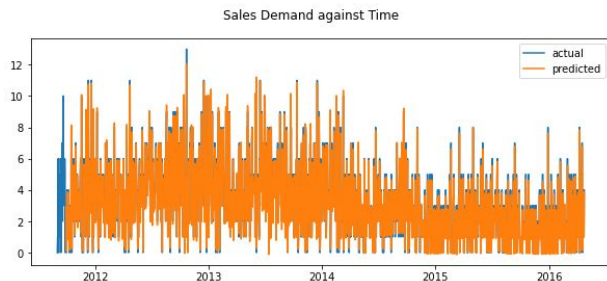
Model	Test RMSE	Test R2
Experiment 1: LSTM	1.60280	-0.04464
Experiment 2: LSTM	2.88533	-2.38533
Experiment 3: LSTM	2.53887	-1.6211
Experiment 1: Transformer	1.51024	0.07252
Experiment 2: Transformer	1.50423	0.07989
Experiment 3: Transformer	1.5105	0.07226

Experiment 1/2: Transformer

Experiment 1 Train Set and Test Set:



Experiment 2 Train Set and Test Set:



Summary of Comparison: 450-item

Basic Comparisons

Comparison	Difference in Test RMSE		Difference in Test R2		
	>	<	>	<	=
1. Experiment 1 – 2: LSTM	27	423	412	27	11
2. Experiment 1 – 3: LSTM	136	314	307	132	11
3. Experiment 1 – 2: Transformer	131	319	313	126	11
4. Experiment 1 – 3: Transformer	136	314	307	132	11

Summary of Comparison: 450-item

Paired 2-Group Comparison

Comparison	Null Hypothesis
1. Experiment 1 – 2: LSTM	Rejected
2. Experiment 1 – 3: LSTM	Not Rejected
3. Experiment 1 – 2: Transformer	Rejected
4. Experiment 1 – 3: Transformer	Not Rejected

Summary of Comparison: 450-item

95% Bonferroni SCIs




Comparison	Difference in Test RMSE SCIs	Difference in Test R2 SCIs
1. Experiment 1 – 2: LSTM	{-1.03591, -0.65436}	{2.95151, 7.07173}
2. Experiment 1 – 3: LSTM	{-0.21150, -0.01189}	{-0.15113, 1.69016}
3. Experiment 1 – 2: Transformer	{-0.12818, -0.01858}	{0.02199, 0.20830}
4. Experiment 1 – 3: Transformer	{-0.04201, 0.02205}	{-0.03294, 0.12772}

Conclusion

- Error modelling does not consistently lead to accuracy improvements.
- Sample mean residuals generally serve as a better estimate than predicted residuals.
- Attention-based model generally produces better results with error modelling as compared to LSTM.



Limitations

- Same hyperparameters for different time series data
 - Choice of time series models
- 
- 
- 

Future Improvements

- Obtain optimal hyperparameters for each time series data
 - Optuna
 - Hyperopt
 - Bayesian optimization
- Choice of time series models
 - Informer
 - Autoformer
- Include noise in residuals feature to prevent overfitting

Thank You

