Predicting Future Sales

Enver Evci
School of Computer Engineering
Middle East Technical University
Cankaya, Ankara
Email: enverevci@gmail.com

***Abstract*—In this paper, I have compared performances of some of the most popular recurrent neural networks while forecasting total sales for every product and store in the next month. LSTM, GRU and RNN have been used to measure accuracy. Results that are taken from these methods was compared with root mean square error.**

From the results, we can say that GRU gives us the most accurate results. Also, LSTM and GRU showed that they are more robust to vanishing and exploding gradient threat that RNN. Also RNN and GRU has some overfitting issues compared to LSTM.

1. Introduction

Many events in our day-to-day lives, such as the movement of stock prices, sales, temperature, traffic rate are measured over a period of time can be shown as an example for timeseries. We can analyse these timeseries data and make predictions about future. Time series is anything which can be seen sequentially at regular interval like any time interval we want over the time. Time series data is important when you are predicting something which is changing over the time using past data. In time series analysis the goal is to estimate the future value using the behaviours in the past data.[1]

We use timeseries prediction models to forecast the value that is missing or unknown. Timeseries has time values for each entry and a target value that can be anything like a label or value. The output of the model is the predicted value for our requested time step.

In this paper, I am trying to test the some of the deep learning methods for timeseries forecasting. Dataset from the contest named "Predicting Future Sales" at Kaggle website has been used for this purpose. Dataset was provided by one of the largest Russian software firms - 1C Company.

For the evaluation part I am going to use root mean square error. Also, there is a maximum epoch limit for both methods which is 200.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(Predicted_t - Actual_t)^2}$$

What made us think if a model is fit is the RMSE that is close for both our testing data and training data.

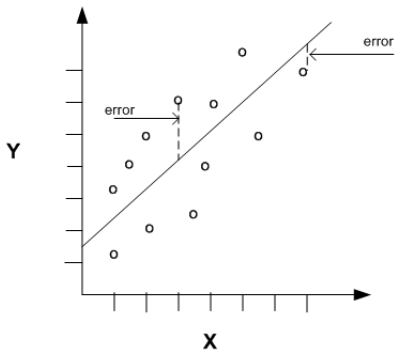


Figure 1. RMSE (Tayal, May 2014,"Performance Analysis of Regression Data Mining Techniques Implemented on Breast Cancer Dataset")

date	date_block_num	shop_id	item_id	item_price	item_cnt_day
02.01.2013	0	59	22154	999	1
03.01.2013	0	25	2552	899	1
05.01.2013	0	25	2552	899	-1
06.01.2013	0	25	2554	1709.05	1
15.01.2013	0	25	2555	1099	1
10.01.2013	0	25	2564	349	1
02.01.2013	0	25	2565	549	1
04.01.2013	0	25	2572	239	1
11.01.2013	0	25	2572	299	1
03.01.2013	0	25	2573	299	3
03.01.2013	0	25	2574	399	2
05.01.2013	0	25	2574	399	1
07.01.2013	0	25	2574	399	1
08.01.2013	0	25	2574	399	2
10.01.2013	0	25	2574	399	1

Figure 2. Sale Dataset.

2. Literature Survey

Zachary C. Lipton, David C. Kale, Charles Elkan, Randall Wetzel (2015) evaluated the ability of LSTMs to recognize patterns in multivariate time series of clinical measurements. They collect the measurement of some clinical data from the patients and tried to predict the diagnoses. Their method outperform some of the best known techniques. Zheng Zhao, Weihai Chen, Xingming Wu, Peter C. Y. Chen, Jingmeng Liu (2017) have tried to predict the traffic for intelligent transportation systems in order to make users efficiently decide which route to travel or which travel mode to select or departure time using LSTM network. Rui Fu, Zuo Zhang, Li Li (2016) also tried to predict the traffic

item_name	item_id	item_category_id
! ВО ВЛАСТИ НАВАЖДЕНИЯ (ПЛАСТ.) D	0	40
!ABBYY FineReader 12 Professional Edition Full [PC, Цифровая версия]	1	76
***В ЛУЧАХ СЛАВЫ (UNV) D	2	40
***ГОЛУБАЯ ВОЛНА (Univ) D	3	40
***КОРОБКА (СТЕКЛО) D	4	40
***НОВЫЕ АМЕРИКАНСКИЕ ГРАФФИТИ (UNI) D	5	40
***УДАР ПО ВОРОТАМ (UNI) D	6	40
***УДАР ПО ВОРОТАМ-2 (UNI) D	7	40
***ЧАЙ С МУССОЛИНИ D	8	40
***ШУГАРЛЭНДСКИЙ ЭКСПРЕСС (UNI) D	9	40
*ЗА ГРАНЬЮ СМЕРТИ D	10	40
*ЛИНИЯ СМЕРТИ D	11	40
*МИХЕЙ И ДЖУМАНДЖИ Сука любовь	12	55
*СПАСАЯ ЭМИЛИ D	13	40

Figure 3. Item Dataset.

flow with GRU and LSTM and they showed these methods outperforms ARIMA. Fazle Karim, Somshubra Majumdar, Houshang Darabi, Shun Chen (2017) have showed the increase of success of fully convolutional networks with LSTM sub-modules for time series classification. Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag and Yan Liu (2018) created GRU-D ,a new version of GRU which make predictions by using multivariate time series with missing values and outperformed state-of-art methods.

3. Problem Definition and Algorithm

3.1. Task Definition

The problem I am trying to solve is predicting future sales for every item in every shops with the given timeseries. I tried to predict the sales of the next month by using the data for 30 months.

3.1.1. Dataset. As it is mentioned, we have two dataset as ”csv” files. One of the datasets contains rows which include date, shop id, item id, date block number, price and sale amount of a single day. However, some of the items were not sold on some of the date blocks. Therefore, if we group the data by date block number, we can see that number of rows are not equal for every shop-item. There is no record for a missing shop-item in every date block. Another information about dataset is some records have ”-1” or negative numbers as number of sale on a day. That means customers return those items so that item count as non-saled item on that shop.

Also, 1C Company prepared this csv file by using 34 months of data and they discritezed it by these data blocks that one block contains one month of information. Therefore, we have 34 date block. Rows in a single block contains daily records and these records are not regularly seperated. That means, date blocks have different number of rows.

Size of this data is 2935849, 6.

Other csv file contains item information like item id, item category id and item name.

Size of this data is 22170, 3

3.1.2. Preprocessing Stage. I have tried some different data preprocessing techniques but the last one actually worked. First of all, I have concatanete item categories to the sales dataset by using the information from item dataset. After getting categories, I have removed date so that I can only use date block numbers. Then, I scale the data with Minimum Maximum Scaler. However, this data gave me low scores. I, later, thought that scaling shop id, item id and category id was a wrong thing to do because there is no relation like if you have higher item id result should change with that information. I thought them as nominal features.

After examining the data, I saw that there is also a seasonality effect so that knowing which month I am processing data is important. Therefore, before deleting date feature, I took the month information and used it as a new feature. By this way, now, I can make the networks learn the effects of different monts in a year for the sale amount of specific item.

As I mentioned before, I thought item, shop, item category ids as nominal features so I decided to encode them with base-n-encoder. I did not used one-hot encoder because maximum item id and shop id are so high that if I create columns for all of them I would have;

#max-item-id + #max-shop-id + #max-item-category-id

extra columns. By using base-n-encoder, I have reduced the number of possible features. And after encoding the features, I have group all the rows so that every row contains monthly sale amount for a certain item id, shop id, date block number and month.

After feeding this preprocessed data, I got low results again but higher than other tries with different hyperparameters. Then, I thought that I should lower the number of features for faster and accurate convergence. In addition, since there may be no sales information for a specific item-shop in some date blocks, I have added all item-shop records to all date blocks even though there was no sale for that product. These empty records have 0 as number of sales for that month. Therefore, now, all date blocks have the same number of records.

Finally, I have sorted whole data by grouping date block number, month, item id, shop id, item category and removed item id, shop id and item category features. Reason behind this action is, since all of the shop-item pairs now have the same index in each date blocks, we can still have the index information about these shop-item and item category. Therefore, neural networks can follow the pattern between these indexes. And the last step, I decided not to scale number of monthly sales. Because, month and date block number information was not scaled and since they have bigger change, they become more important features for networks and results were produced as very low scaled version of test data.

3.2. Algorithm Definition

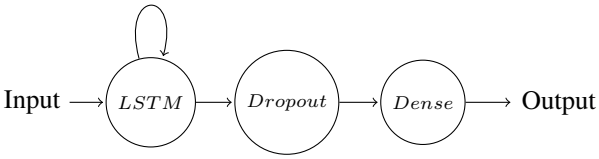
There are three deep learning methods I compare to see the accuracy and performance. I have used them to evaluate and predict next month sales predictions for the given timeseries. I have used LSTM, GRU and RNN while

predicting future sales. I have fed the networks with training set that contains information belongs to date block 0 to date block 30. And to calculate loss I gave networks one-month-shifted train evaluation set that contains only number of monthly sales of every shop-item pairs. For validation and test sets I shifted one month again. That means validation set is between month 1-31 and test set is between month 2-32.

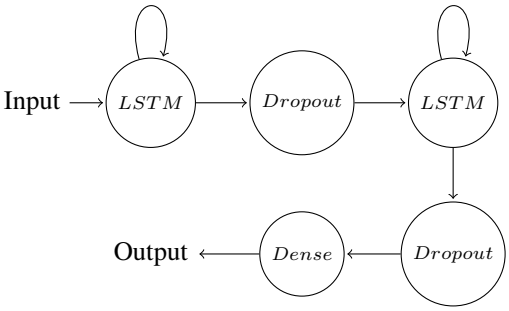
3.2.1. LSTM. Long Short-Term Memory networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems[2].

I haved trained a sequence-to-sequence regression LSTM network to predict the next month sales. I have used two different architectures for evaluation even though I have tried more deep networks. Deeper networks resulted low scores so I eliminated them from evaluation stage.

Model 1:



Model 2:



Model 1 has one LSTM layer layer that takes a sequence and pass the output to the dense layer which have 30 nodes as the output. Model 2 has two LSTM networks that first LSTM layer return sequences to second LSTM layer after Dropout layer. Second LSTM layer which has half of the hidden units that first LSTM has pass its output to Dense layer which has 30 outputs.

3.2.2. GRU. Introduced by Cho, et al. in 2014, GRU (Gated Recurrent Unit) aims to solve the vanishing gradient problem which comes with a standard recurrent neural network. GRU can also be considered as a variation on the LSTM because both are designed similarly and, in some cases, produce equally excellent results.[3]

GRU network has been again trained as a sequence-to-sequence regression to predict the next month sales. I have used two different architectures for evaluation. When it gets deeper results were not so good. It gets unnecessaryly complexed and make it difficult to predict next month.

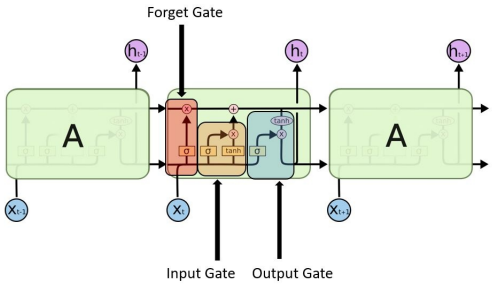
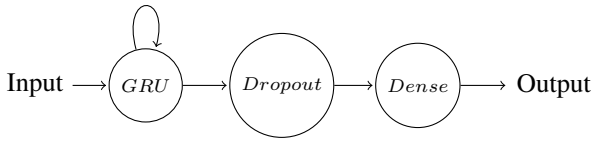


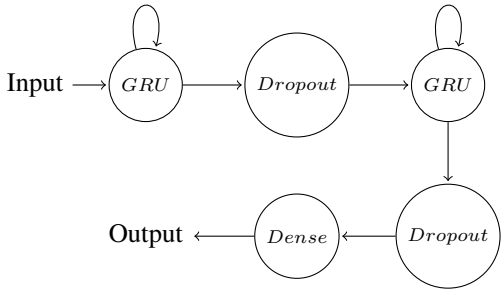
Figure 4. LSTM cell schema (2016 Feb, Retrieved from https://feature.engineering/difference-between-lstm-and-gru-for-rnns/)

Again we can see the models below;

Model 1:



Model 2:



I have used same architecture in GRU experiment with LSTM since GRU is a variation of LSTM. I wanted to see how different results can be obtained.

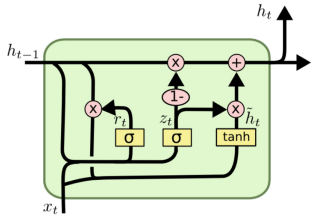


Figure 5. GRU cell schema (2016 Feb, Retrieved from https://feature.engineering/difference-between-lstm-and-gru-for-rnns/)

3.2.3. RNN. A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit temporal dynamic behavior for a time

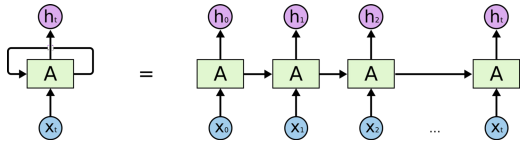
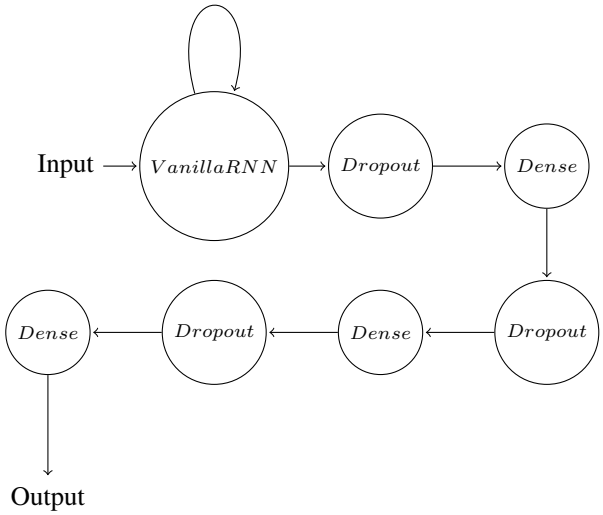


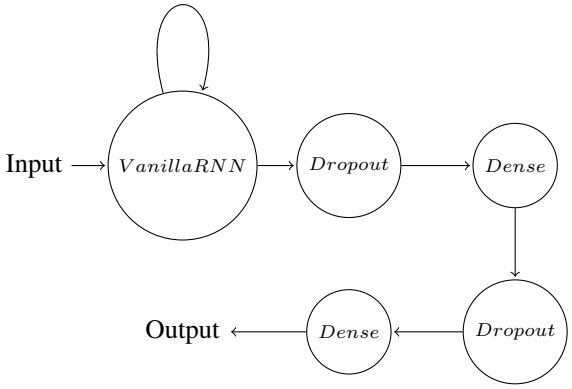
Figure 6. GRU cell schema (Pranoy Radhakrishnan, 2017 Aug, Retrieved from https://towardsdatascience.com/introduction-to-recurrent-neural-network-27202c3945f3)

sequence. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs.[4] Vanilla RNN network has been trained as a sequence-to-sequence regression to predict the next month sales. I have used three different architectures for evaluation. When it gets deeper results were not so good. It gets unnecessaryly complexed and make it difficult to predict next month.

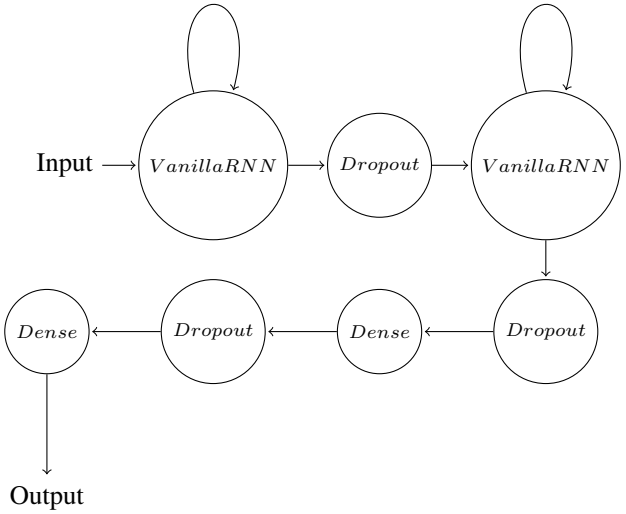
Again we can see the models below;
Model 1:



Model 2:



Model 3:



In RNN we have the danger of vanishing or exploding problems. Also, during training, I have encountered exploding in model 3 and when I tried to use more hidden units in RNN layers. I have used half the number of hidden units of RNN in Dense layers and second RNN layer also have been tried both same hidden size and half of the hidden size of the first RNN layer. All layers except the last Dense layer which has 30 outputs have ReLU activation function.

All of three different deep learning methods have Dense in their last layer with activation function linear which just gives the output score without any activation function.

4. Experimental Evaluation

4.1. Methodology

For all of the methods, I have tested the accuracy between the test sample and the prediction of our three different kinds of network with Root Mean Square Error. As we can understand from the name, lower values shows that better fitting between prediction and real values.

Normally in the dataset we have 2935849 entries. I wanted to reduce the number of entries to 10000 in order to make it more simple and focus on only learning other than the computational power. Also, since I wanted to compare the results in a fair way based on both accuracy and performance, I have decided a maximum epoch number which is 200.

After getting the results, I have examined all of the outputs which are loss functions and the graph that shows the predicted values and the real values for the next month. And you can see the hyperparameters that were tried and best results for each of the methods.

4.2. Results

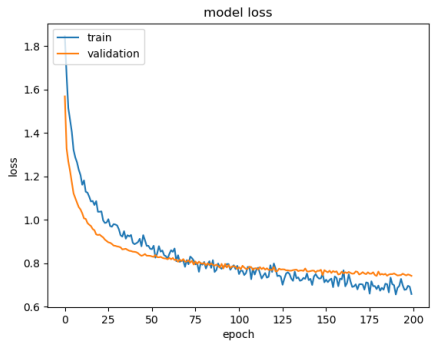


Figure 7. LSTM train and validation loss graph for only 200 epoch

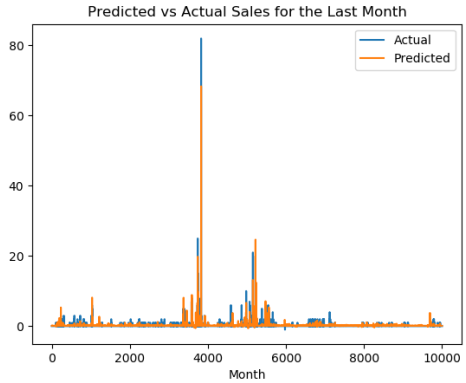


Figure 8. LSTM prediction for month 32

For LSTM, I have tried different hyperparameters with different models.

Model: 1, 2
Batch Size: 100
Learning rates: 0.01, 0.001, 0.0001
Hidden Unit: 256, 512
Learning Rate Decay: 0.001, 0.0001
Dropout Rate: 0.2, 0.5
Optimizer: Adam
Loss Function: Mean Squared Error

Best parameters;
Model:1, Learning rate: 0.001, Hidden Unit: 256, Learning Rate Decay: 0.0001, Dropout Rate: 0.5
RMSE: 0.399081

For GRU, I have tried different hyperparameters with different models.

Model: 1, 2
Batch Size: 100
Learning rates: 0.01, 0.001, 0.0005
Hidden Unit: 64, 256, 512

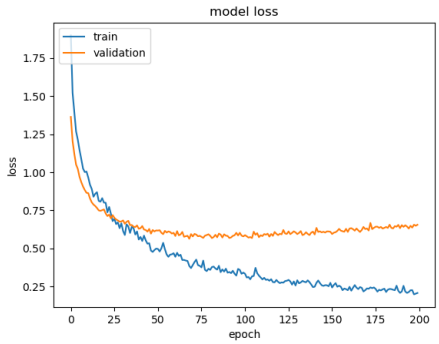


Figure 9. GRU train and validation loss graph for only 200 epoch

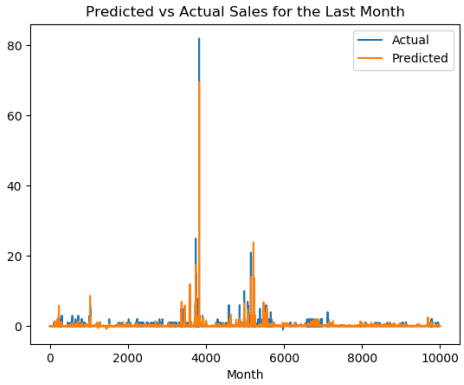


Figure 10. GRU prediction for month 32

Learning Rate Decay: 0.001, 0.0001
Dropout Rate: 0.2, 0.5
Optimizer: Adam
Loss Function: Mean Squared Error

Best parameters;
Model:2, Learning rate: 0.001, Hidden Unit: 256, Learning Rate Decay: 0.0001, Dropout Rate: 0.5
RMSE: 0.389967

For RNN, I have tried different hyperparameters with different models.

Model: 1, 2, 3
Batch Size: 100
Learning rates: 0.001, 0.0001
Hidden Unit: 64, 128, 256, 512
Learning Rate Decay: 0.001, 0.0001
Dropout Rate: 0.4, 0.5
Optimizer: Adam
Loss Function: Mean Squared Error

Best parameters;
Model:2, Learning rate: 0.001, Hidden Unit: 256, Learning Rate Decay: 0.001, Dropout Rate: 0.5
RMSE: 0.468627

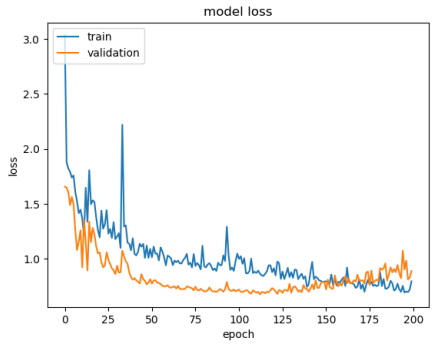


Figure 11. RNN train and validation loss graph for only 200 epoch

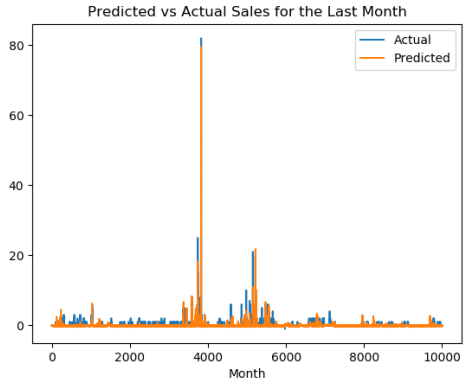


Figure 12. RNN prediction for month 32

4.3. Discussion

It can be understood from the results that LSTM and GRU gives us the most accurate results while RNN gives the worst. Although LSTM and GRU has almost similar points, GRU seems to be overfit. I have tried to increase the hidden unit size and lower the learning rate but still I had the overfit results. To test, I also trained some of the models from each of LSTM, RNN and GRU with 300 epoch to see if I can get a different results based on comparison between them. However, results were same again even though they both improved a little. When I saw the loss graph for both of them, I saw their loss decay were small.

Maybe using different architectures than I used were change the results and considering using CNN would be also an option. However, I did not want to make LSTM and GRU architecture so differently in order to compare since they both designed similarly.

5. Conclusion

As a conclusion, we can say that GRU gives us the most accurate results. Also, LSTM and GRU showed that they are more robust to vanishing and exploding gradient threat. While training the RNN, I saw that when I increased the hidden units or depth of network, gradient was exploded and I got results as infinity more quickly. In addition, LSTM and GRU decreases the loss more stable generally and they converge faster than RNN. Also I saw while testing that increasing the complexity of network harms LSTM less than the other two networks.

References

[1] Bista (2016, May), *5 Statistical Methods For Forecasting Quantitative Time Series*, Retrieved from https://www.bistasolutions.com/resources/blogs/5-statistical-methods-for-forecasting-quantitative-time-series/

[2] Jason Brownlee (2017, May), *A Gentle Introduction to Long Short-Term Memory Networks by the Experts*, Retrieved from https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/

[3] Simeon Kostadinov (2017, Dec), *Understanding GRU networks*, Retrieved from https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be

[4] Amit Shekhar (2018, April), *Understanding The Recurrent Neural Network*, Retrieved from https://letslearnai.com/2018/04/14/understanding-the-recurrent-neural-network.html