SEMINAR REPORT
ON
RECURRENT NEURAL NETWORK

A report submitted in partial fulfilment of the requirement for the award of

The degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

Submitted By:-

Ayush Singh Rawat

1501051120

2015-2019

Department Of Information Technology

DIT UNIVERSITY, DEHRADUN

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.

2018-2019

ABSTRACT

Imagine you have a normal feed-forward neural network and give it the word "Neural Network" as an input and it processes the word character by character at the time it reaches the character "l", it has already forgotten about previous learnings which makes it almost impossible for this type of neural network to predict what character would come next, that is when RNN comes in picture it is most efficient is times series/ time series, speech, text, financial data, audio, video, weather and thiss reasearch paper is about just that.

1. INTRODUCTION

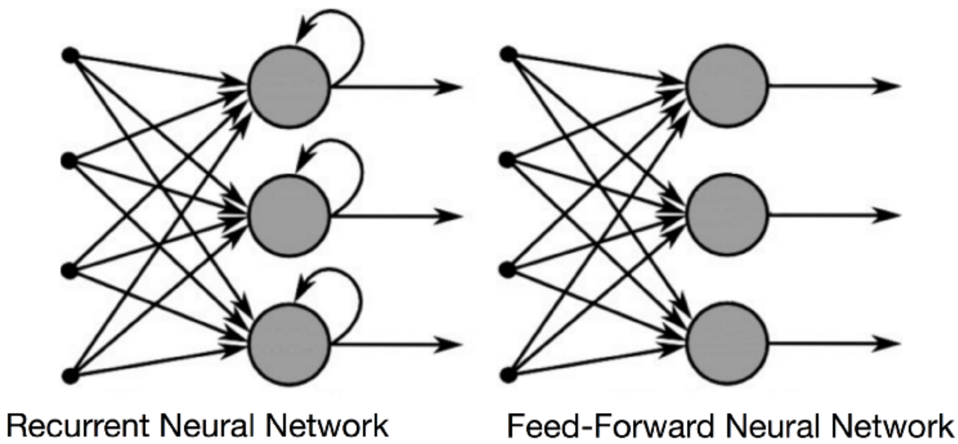

Figure 1(a): Basic Structure of RNN and Feed forward neural network

A Neural Netwroks Consists of layers, 1 input layer 1 output layers and can have multiple hidden layers. But as we can see in the figure a Feed forward neural network has only one way forward and there is no connection backwards, backpropogation can only be done once a whole cycle has completed but in a Recurrent neural network every layer is connected to both ways which makes it easier to backpropogated faster and plus the RNN is such a promising algorithm because it has it's own memory, in fact the only neural networks that have an internal memory. RNN's are not a new research they are a topic of 1980's but have been possible since few years because of the computational, graphical power and handling of massive amount of data.

2. WHEN TO USE A RNN?

RNN is best used for sequential data like time series, speech, text, financial data, audio, video, weather etc because they find better contexts in these kind of data as compared to other algorithms and they work better with sigmoid and tanh activation function than other neural networks.

2.1 WHY TO USE RNN?

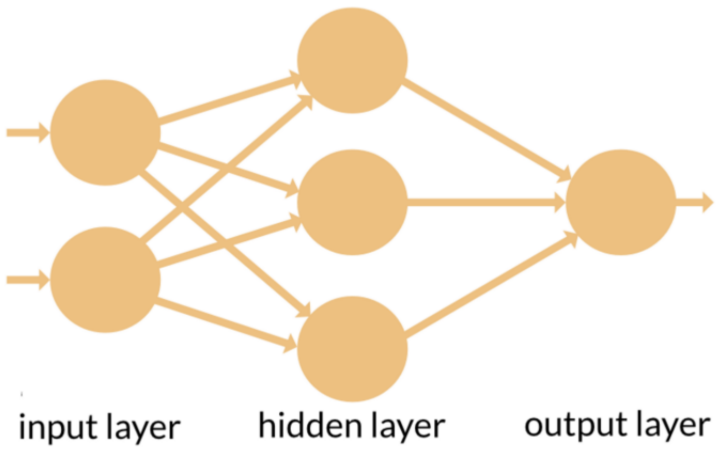

Figure 2(a): A simple neural network

1: In a feed forward neural network the information only moves in one directionas shown in the figure whereas in RNN the information cycles through a loop. When it makes a decision, it takes into consideration the current input and also what it has learned from the inputs it received previously.

2: Feed-Forward Neural Networks have no memory of the input they received previously and are therefore bad in predicting what's coming next they remember nothing rather than the training data but because of internal memory RNN predicts better than any other model in a time series prediction.

3: Weight matrix in a Feed forward neural networks takes inputs and then produces the output whereas RNN's apply weights to the current and also to the previous input.

4: Also note that while Feed-Forward Neural Networks map one input to one output whereas RNN's can map one to many(predicting Stock prices), many to many(Translation of Language) and many to one(Classifying a voice like in Google home and Apple siri).

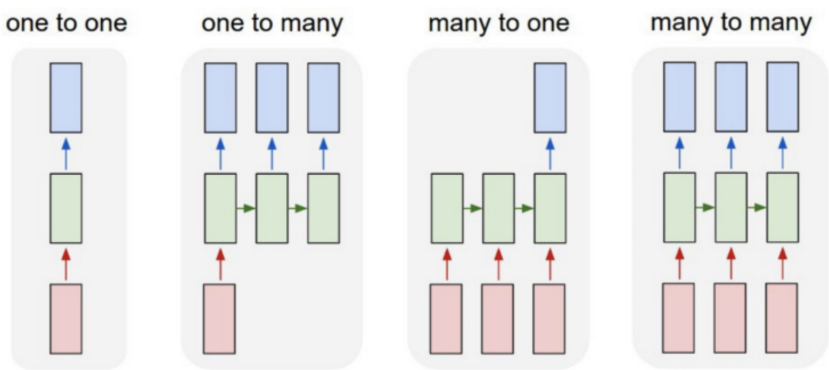

Figure 2(b): RNN's One to many, Many to one and many to many map

2.2 BACKPROPOGATION IN RNN

The report does not covers how propogations are done it is out of context in this report, Forward-Propagation is to get the output of your model and check if this output is correct or incorrect, to get the error. Backward-Propagation, which is nothing but going backwards through your neural network to find the the error with adjusting weights, which enables you to find the optimal weights(Gradient Descent), how a Neural Network learns during the training process.

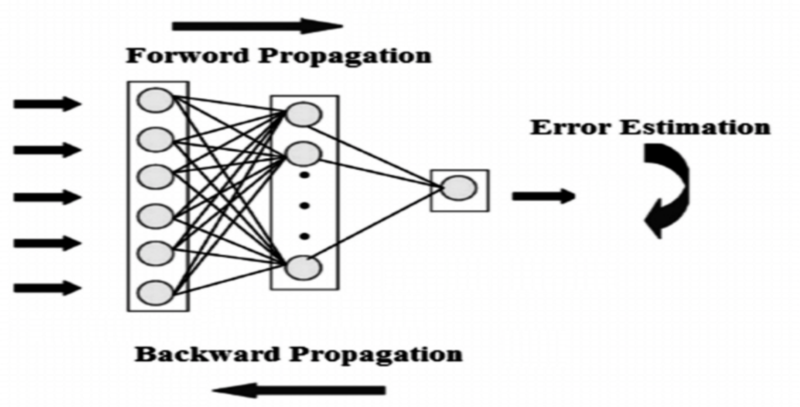

Figure2 (c) : Forward Propagation and Backward Propagation in a neural network

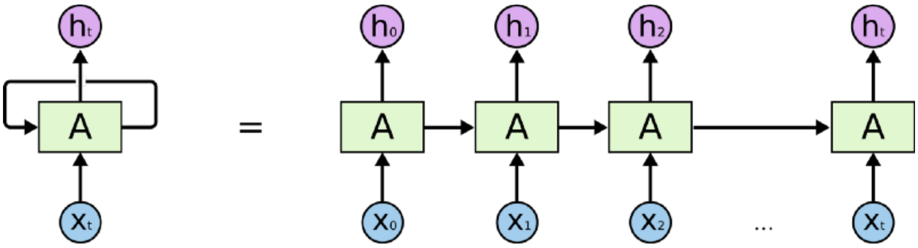

Figure2(d): A recurrent neural network

This Figure shows why a RNN is seen as a sequence of Neural Networks, If Backpropogation is done since the error of a given timestep depends on the previous timestep. Within BPTT the error is back-propagated from the last to the first timestep, while unrolling all the timesteps. This allows calculating the error for each timestep, which allows updating the weights and can be computationally expensive when you have a high number of timesteps.[1]

2.3 ISSUES IN RNN

There are two major obstacles RNN's have. For that we first need to know what gradient is as shown in the figure a gradient descent is nothing but a cost function in which we find the global minimum of the graph which helps us get the optimal weights for optimal accuracy in our output(prediction).

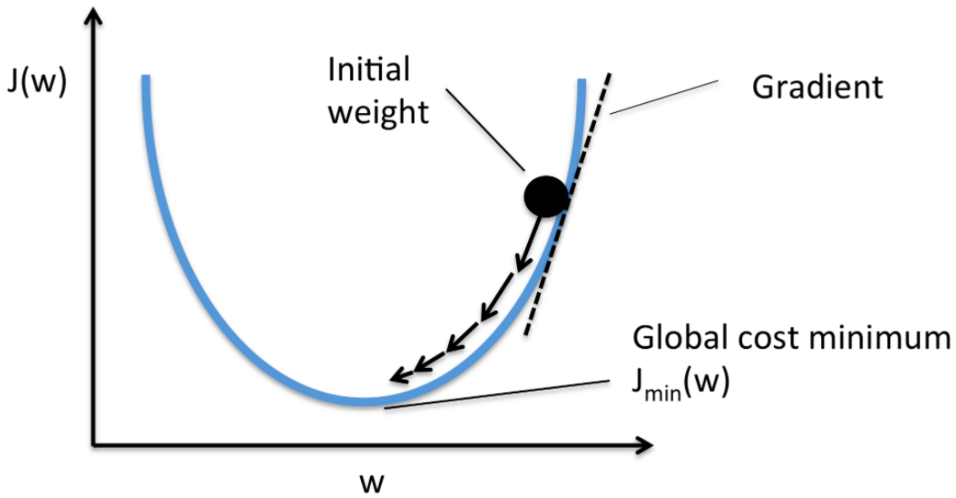

Figure 2(e): Figure of a Gradient Decsent

A: Exploding Gradients

When the algorithm assigns high importance to the weights and can be solved by reducing gradients.

B: Vanishing Gradients

When the values of a gradient are too small and is solved through the concept of Long-short term memory.

3. Long-Short Term Memory

Long Short-Term Memory (LSTM) networks have more memory than that of RNN because of as we all know RNN experiences vanishing gradients because it has a very small memory. LSTM can read, write and delete information from its memory and to do so LSTM has gates which decide whether or not to store information, based on the importance it assigns to the information. The assigning of importance happens through weights, which are also learned by the algorithm. This simply means that it learns over time which information is important and which not.

In an LSTM you have three gates: 1: input, 2: forget and 3: output gate. These gates determine whether or not to let new input in (input gate), delete the information because it isn't important (forget gate) or to let it impact the output at the current time step (output gate).

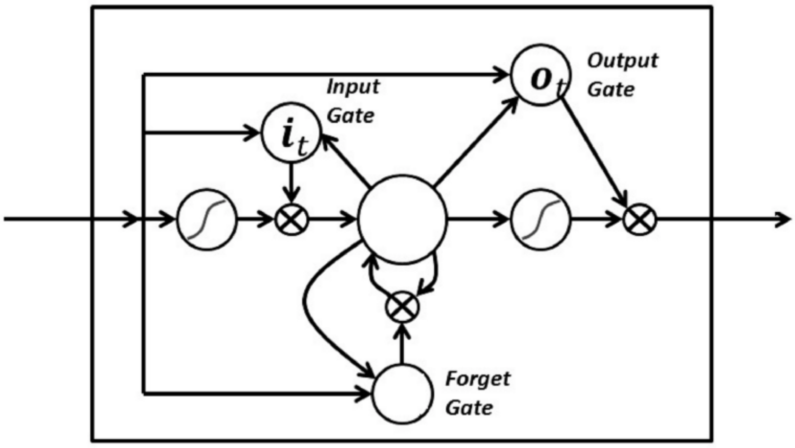

Figure 3(a): Three gates of LSTM

The gates in a LSTM are sigmoids, meaning that they range from 0 to 1. The fact that they are analog, enables them to do backpropagation with it. The problematic issues of vanishing gradients is solved through LSTM because it keeps the gradients steep enough and therefore the training relatively short and the accuracy high.

4. DESCRIPTION OF DATASET

Dataset consists of following files:

- prices.csv: raw, as-is daily prices. Most of data spans from 2010 to the end 2016, for companies new on stock market date range is shorter. There have been approx. 140 stock splits in that time, this set doesn't account for that.
- prices-split-adjusted.csv: same as prices, but there have been added adjustments for splits.
- securities.csv: general description of each company with division on sectors
- fundamentals.csv: metrics extracted from annual SEC 10K fillings (2012-2016), should be enough to derive most of popular fundamental indicators.

Figure 2 demonstrates the dataset and assessment is done on Spyder IDE with python 3.6.4.

Table 1: Description of dataset for first 5 rows

	Date	Open	High	Low	Close
0	1970-01-02	0.30627	0.30627	0.30627	0.30627
1	1970-01-05	0.30627	0.31768	0.30627	0.31385
2	1970-01-06	0.31385	0.31385	0.30996	0.30996
3	1970-01-07	0.31385	0.31385	0.31385	0.31385
4	1970-01-08	0.31385	0.31768	0.31385	0.31385

5. RESULTS

5.1 Output using Feed forward neural networks

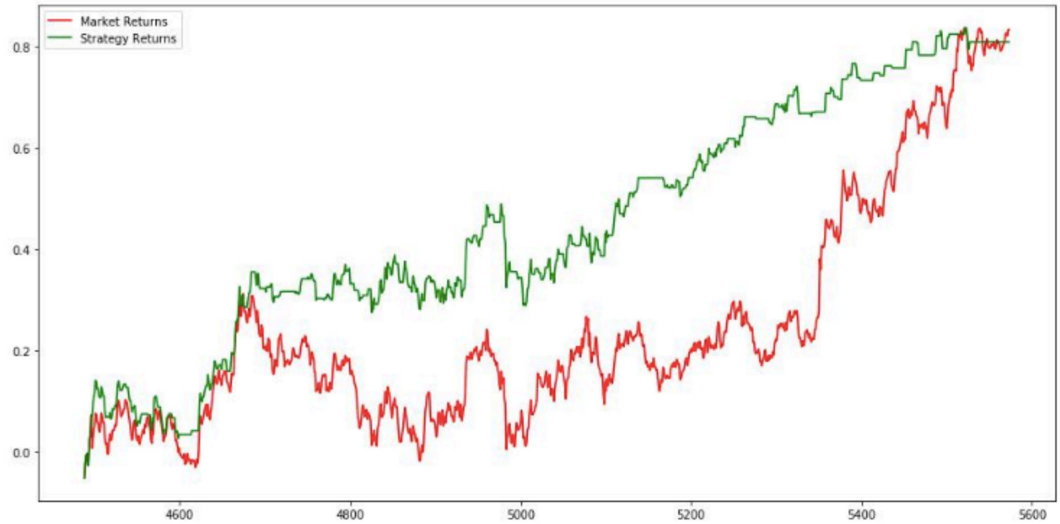

Figure 5(a): Applying ANN

5.2 Output using RNN

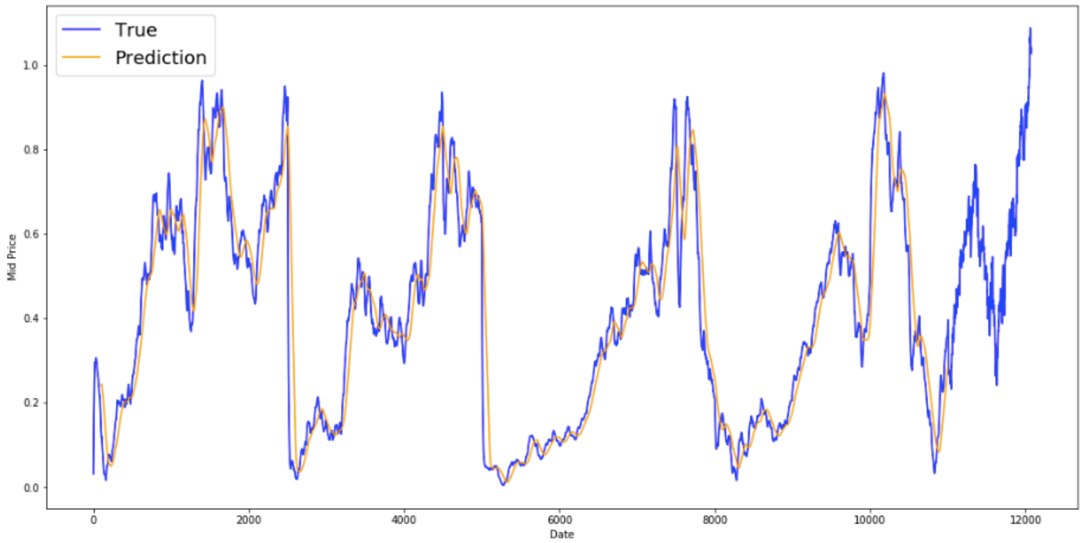

Figure 5(b): Accuracy RNN

6. APPLICATIONS

1. Language modeling character and word level LSTM's are used
2. Machine Translation also known as sequence to sequence learning
3. Image captioning (with and without attention, eg: Automatic tagging whileuploading pictures on facebook)
4. Hand writing generation(Predicting next word based on learning).
5. Image generation using attention models - my favorite
6. Question answering (Virtual Assistance)
7. Assistants like Apple's Siri, Samsung's Bixby, Google and Amazon's Alexa

7. CONCLUSION

Now we have proper understanding of how a Recurrent Neural Network works, which enables us to decide if it is the right algorithm to use for a given Machine Learning problem, As we can see that the accuracy of RNN in time series prediction outperforms than that of ANN by a vast different.

Specifically, we learned whats the difference between a Feed-Forward Neural Network and a RNN, when we should use a Recurrent Neural Network, how Backpropagation and Backpropagation Through Time works, what the main issues of a RNN are, how a LSTM works and what are the applications of RNN.

References

1. Michael Auli, Michael Galley, Chris Quirk, and Geoffrey Zweig. Joint language and translation modeling with recurrent neural networks, pages 1044–1054, 2013.
2. Pierre Baldi and Gianluca Pollastri. The principle design of large-scale recursive neural network architectures–DAG-RNNs and the protein structure prediction problem. The Journal of Machine Learning Research, 4:575–602, 2003.
3. Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgements. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, 2005.
4. Justin Bayer, Daan Wierstra, Julian Togelius, and Ju¨rgen Schmidhuber. Evolving memory cell structures for sequence learning. In Artificial Neural Networks–ICANN 2009, pages 755–764. Springer, 2009.
5. Richard K. Belew, John McInerney, and Nicol N. Schraudolph. Evolving networks: Using the genetic algorithm with connectionist learning. In In. Cite-seer, 1990.
6. Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. Neural Networks, IEEE Transactions on, 5(2):157–166, 1994.

7. Yoshua Bengio, R´ejean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. The Journal of Machine Learning Research, 3:1137–1155, 2003.

8. Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. Advances in optimizing recurrent networks. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pages 8624– 8628. IEEE, 2013.

9. James Bergstra, Olivier Breuleux, Frederic Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In Proceedings of the Python for Scientific Computing Conference (SciPy), volume 4, page 3. Austin, TX, 2010.

10. Avrim L. Blum and Ronald L. Rivest. Training a 3-node neural network is NP-complete. In Machine Learning: From Theory to Applications, pages 9–28. Springer, 19.