Limitations of RNN, LSTM and GRU

Krity Haque Charu, *Member, IEEE,* Course:CSE425, *ID: 19101173*

***Abstract*—Neural networks are essentially a set of algorithms that attempt to understand the underlying patterns in a set of data, allowing machines to mimic the critical thinking ability of humans. RNNs (Recurrent Neural Networks), LSTMs (Long-Short Term Memory), and GRUs (Gated Recurrent Units) are examples of neural networks that are used for speech recognition, voice recognition, time series prediction, and natural language processing. As a result, unlike feed forward neural networks, these neural networks are primarily concerned with forecasting. Scientists, on the other hand, face some common issues when working on long sequences of data on RNN backpropagation, which is related to the gradient of neural networks. To address the issue, two additional special RNNs, LSTM and GRU, have been proposed. They can deal with Naive RNN's long-term dependencies. Nonetheless, both LSTM and GRU have some unique shortcomings in the Neural Network family. This paper will focus on the limitations of these three types of Recurrent Neural Networks.**

***Index Terms*—RNN, LSTM, GRU, Gradient, Back-propagation**

I. Introduction

One of the most difficult types of data to handle and the forecast is sequential data. The concept of Recurrent Neural Networks was developed to deal with this type of data. While other networks travel in a linear way during the feed-forward or back-propagation processes, the RNN uses Back-Propagation through time to learn instead of a feed-forward pass. RNN works on the principle of saving the output of a particular layer and feeding this back to the input in order to predict the output of the layer. A feed-forward neural network is unable to memorize previous inputs and considers only the current input.

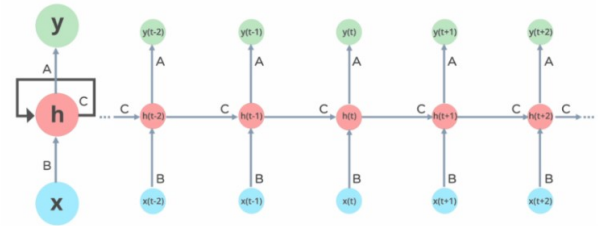

Figure 1 : RNN after unfolding

"x" is the input layer, "A" is the hidden layer, and "h" is the output layer. A, B, and C are the network parameters used to improve the output of the model. At any given time t, the current input is a combination of input at x(t) and x(t-1). The output at any given time is fetched back to the network to improve on the output.The hidden state is updated using the following recurrence relation:

$$h(t) = f_c(h(t-1), x(t)) \tag{1}$$

Here,
h(t)=New state
f_c=Function with Parameter c
h(t-1)=Old state
x(t)=Output Vector at time step t
However, RNN fails to deal with some conditions.For starters, it is incapable of storing information for an extended period of time. To predict the current output, it is sometimes necessary to refer to information that was stored a long time ago. RNNs, on the other hand, are completely incapable of dealing with such long-term dependencies.Second, there is no finer control over which aspects of the context should be carried forward and how much of the past should be 'forgotten.' Other issues with RNNs include exploding and vanishing gradients. which occur during a network's training process via backtracking. On the other hand, Long Short-Term Memory is an upgraded version of RNN that was explicitly designed to avoid long-term dependency problems.The LSTM can remove or add information to the cell state because it is carefully controlled by structures called gates.

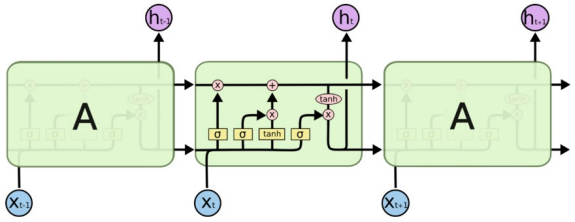

Figure 2 : LSTM

Yet, LSTM has drawbacks removing the Vanishing gradient problem fully. We still have a path from older past cells to the current one. In fact, the path is now even more complicated because it is connected to additive and forget branches. Without a doubt, LSTM and derivatives can learn a lot of long-term information. However, they can only remember sequences of 100s, not 1000s, 10,000s, or more.Lastly, GRU is similar to LSTM but with two gates: less complex than LSTM but has some flaws of its own. Since the inception of RNN,numerous theoretical and experimental works have been published on the subject of this type of RNN since its inception, with many of them reporting on the astounding results achieved across a wide range of application domains where data is sequential. The LSTM network has had a significant impact in language modeling, speech-to-text transcription, machine translation, and other applications where RNN could not reach. GRU Network can handle these applications also. However, each of these networks has some constraints that have yet to be overcome. Thus, knowing the limitations in details is important so that we can be knowledgeable of them before operating on these neural networks. Therefore, the goal of this study is to expose the limitations of RNN, LSTM, and GRU Network.

II. Body

A. Limitations of RNN

RNN was considered one of the best ideas for utilizing memory elements in neural networks. Many problems.Many real-world problems are episodic and heavily reliant on time. The problem occurs when we want to work a longer sequence of data. RNN tends to forget things that came in the very beginning of time steps. The reason for this is that the gradient of this network either gets smaller or exponentially bigger. Hence, RNN founds its greatest disadvantages from this.

1) Short-term Memory Problem: RNN happens to forget previous old information when dealing with a time series.RNNs helps to detect dependencies in sequential data. This means they intend to find correlations between different points within a sequence.When describing Short-term dependencies, it mainly describes a dependence in the recent past. Long-term dependencies on the other hand are correlations between points in time that are far away from each other.Finding such dependencies, on the other hand, makes it impossible for RNNs to recognize patterns in sequential data and use this information to forecast a trend. Moreover, it can't determine the next word when there is a distant relationship of unknown length. Suppose, there is two sentences:
Today, due to my current job situation and family condition, I have to take a loan
Last Year, due to my current job situation and family condition, I had to take a loan
To predict today and last year we must focus on the first two words: Today and Last year. But it has a tendency to forget words that occurred a long time ago due to its short term memory while working with long unknown length of sentences. So we wish to have a long term memory in it.

2) Vanishing Gradient Problem: RNNs suffer from the problem of vanishing gradients and it is the main cause of RNN's short-term memory problem. The gradients mainly carry information used in the RNN.It is the change in loss in respect to the weight. To update any weights and reduce error in RNN, it uses the process called backpropagation through time. Through time refers that, in every recurrence it is like going back in time towards the past. In every backpropagation the value of gradient(dE/dW) can be calculated as the summation of gradients at each time step.

$$\frac{dE}{dW} = \sum_t \frac{dE_t}{dW} \tag{2}$$

If there are hundreds of layers and t = 100 and we are taking the derivative and derivative of the sigmoid is always below 0.25, it would end up calculating all the partial derivatives associated with the network, which is a huge multiplication and end up with a vanishing value such that we can't use them for error calculation.With that, value of gradient become smaller. We know, to update any node's weight the formula is:

$$NewWeight = OldWeight - LearningRate * Gradient \tag{3}$$

So, when the gradient becomes too small, the parameter updates become insignificant. Weight of old nodes and new nodes will hardly be different .Long training time, poor performance, and bad accuracy are the major issues in gradient problems.

3) Exploding Gradient Problem: Exploding gradients is a problem in which the gradient value becomes very big and this often occurs when we initialize larger weights and we could end up with NaN. If our model suffered from this issue we cannot update the weights at all. In an exploding gradient problem, the slope tends to grow exponentially instead of decaying.This problem arises when large error gradients accumulate, resulting in very large updates to the neural network model weights during the training process.

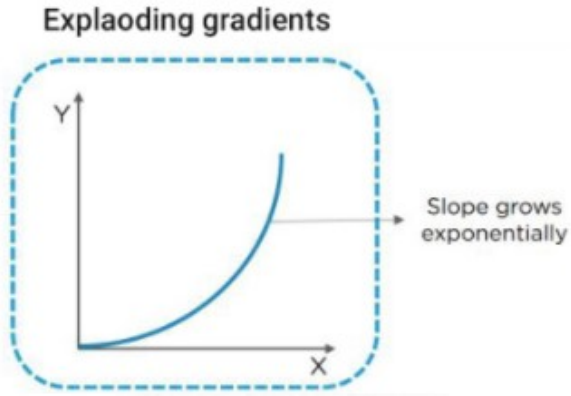

Figure 3 : Exploding gradient

However, gradient clipping is a process that can handle this issue. At a predefined threshold value, we clip the gradient. This will prevent the gradient value from going beyond the threshold and we will never end up in big numbers or NaN.

B. Limitations of LSTM

All RNN are in the form of a chain of repeating modules of a neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.LSTMs also have a chain-like structure, but the repeating module is a bit different structure. Instead of having a single neural network layer, four interacting layers are communicating extraordinarily.LSTM has three gates to filter the necessary and unnecessary keywords that don't required in long run. Forget Gate chooses whether the information coming from the previous timestamp is to be remembered or is irrelevant and can be forgotten. In the input gate, the cell tries to learn new information from the input to this cell. At last, in the output part, the cell passes the updated information from the current timestamp to the next timestamp.Moreover, Long term memory is transferred from Cell Update and Short-term memory is transferred from state update like RNN. Some of the most demanding applications are Language modelling or text generation, image processing, hand and speech recognition, music note predictions, language

translation and many more. Nevertheless, everything in this world comes with its own advantages and disadvantages, LSTMs too, have a few drawbacks.They are described below:

1) Cannot fully diminish the Gradient Problem: Because they solved the problem of vanishing gradients, LSTMs became popular. However, they do not completely remove it. The issue is that the data must still be moved from cell to cell for evaluation. Furthermore, with the addition of new features (such as forget gates), the cell has grown quite complex.

2) Require Many resources: They require a significant amount of time and resources to train and become ready for real-world applications. In technical terms, they require high memory-bandwidth due to the linear layers present in each cell, which the system typically does not provide. As a result, LSTMs become quite inefficient in terms of hardware.

3) Require Powerfull Network than LSTM: As data mining becomes more popular, developers are looking for a model that can remember past information for a longer period of time than LSTMs. The human habit of dividing a given piece of information into small parts for easy recollection inspired this type of model.

4) Affected by different random weight initialization: LSTMs are affected by different random weight initialization and thus behave similarly to feed-forward neural nets. Instead, they prefer small weight initialization.

5) Overfitting Issue: LSTMs are prone to overfitting, and using the dropout algorithm to mitigate this problem is difficult. Dropout is a regularization method that excludes input and recurrent connections to LSTM units from activation and weight updates while training a network.

Moreover,LSTM are difficult to train because they require memory-bandwidth-bound computation, which is a hardware designer's worst nightmare and, as a result, limits the applicability of neural network solutions. In short, LSTM requires four linear layers (MLP layers) per cell to run at each sequence time-step. Linear layers require a large amount of memory bandwidth to compute; in fact, they cannot use a large number of compute units frequently because the system lacks sufficient memory bandwidth to feed the computational units. Furthermore, while adding more computational units is simple, increasing memory bandwidth is difficult (note enough lines on a chip, long wires from processors to memory, etc). As a result,LSTM and variants do not lend themselves well to hardware acceleration.

C. Limitations of GRU

The Gated Recurrent Unit, or GRU, has the same workflow as the RNN, but the difference is in the operation and gates associated with each GRU unit. GRU incorporates two gate operating mechanisms called Update gate and Reset gate to solve the problem encountered by standard RNN.

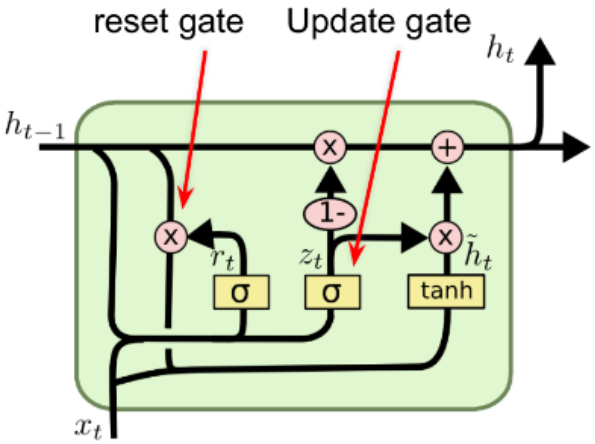

figure 4 : GRU

The update gate is responsible for determining the amount of previous information that needs to pass along the next state. This is really powerful because the model can decide to copy all the information from the past and eliminate the risk of vanishing gradient. The reset gate is used from the model to decide how much of the past information is needed to neglect; in short, it decides whether the previous cell state is important or not. However, there are limitions in GRU to compare to our known Network LSTM. First of all,when dealing with large sequences and accuracy is concerned GRU is not good than LSTM. Also like LSTM, GRU cannot handle the vanishing gradient problem fully. Whereas RNN is faster and less complex. Secondly, GRU has the disadvantages of slow convergence and low learning efficiency. The reason behind this is it requires heavy hardware resources.

III. Conclusion

In this article, we learned about the basic workings and the limitations RNN, LSTM, GRU. Because a normal feed forward neural network cannot deal with previous data and must work only with the incoming one, RNN and later LSTM and GRU were introduced. Despite their widespread use in the neural network field, they have some drawbacks. RNN has a problem with its gradients. Weight cannot be properly updated if it falls below zero. Again, if it becomes exceptionally high, RNN produces incorrect output. LSTM entered the picture to save RNN. However, LSTM cannot solve the gradient vanishing problem clearly. Because of its complex configuration, LSTM necessitates the use of powerful hardware. Also, because LSTM requires more parameters, it is slower. Finally, GRU's limitations were its low convergence rate and slow learning efficiency.

References

[1] Wang, X., Xu, J., Shi, W., Liu, J. (2019, October). OGRU: An optimized gated recurrent unit neural network. In Journal of Physics: Conference Series (Vol. 1325, No. 1, p. 012089). IOP Publishing.
[2] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, 132306.
[3] Team, S. (2018). Recurrent Neural Networks (RNN)-The Vanishing Gradient Problem. URL: https://www. superdatascience. com/blogs/recurrent-neuralnetworks-rnn-the-vanishing-gradient-problem.

[4] Jaiswal,Abhishek.(2022,January 5).Tutorial on RNN — LSTM —GRU with Implementation. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/01/tutorial-on-rnn-lstm-gru-with-implementation/: :text=the
[5] Lendave,Vijaysinh.(2021,August 28).LSTM Vs GRU in Recurrent Neural Network: A Comparative Study.Analytics India Magazine.https://analyticsindiamag.com/lstm-vs-gru-in-recurrent-neural-network-a-comparative-study/
[6] Olah,Christopher.(2015,August 27).Understanding LSTM Networks.Colah's blog.http://colah.github.io/posts/2015-08-Understanding-LSTMs/
[7] Ye,Andre.(2020, September 2020).Long Short-Term Memory Networks Are Dying: What's Replacing It?.Medium.https://medium.com/mlearning-ai/long-short-term-memory-networks-are-dying-whats-replacing-it-5ff3a99399fe
[8] Culurciello,Eugenio.(2018, April 13).The fall of RNN/LSTM.Towards Data Science.https://towardsdatascience.com/the-fall-of-rnn-lstm-2d1594c74ce0: :text=In
[9] SuperDataScience.(2018,August 23).Recurrent Neural Networks (RNN) - The Vanishing Gradient Problem.SuperDataScience.https://www.superdatascience.com/blogs/recurrent-neural-networks-rnn-the-vanishing-gradient-problem
[10] Manu.(2021, January 30).A simple overview of RNN, LSTM and Attention Mechanism.The Startup.https://medium.com/swlh/a-simple-overview-of-rnn-lstm-and-attention-mechanism-9e844763d07b