International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 5, pp: 618-622 www.ijaem.net ISSN: 2395-5252

Long Short-Term Memory Used for Enhancing Bilingual Machine Translation Approach

Manish Rana, Mohammad Atique
Research Scholar, Department of Graduate Science and Engineering, Sant Gadge Baba Amravati University, Amravati, India
Professor, Department of Computer Science Graduation, Sant Gadge Baba Amravati University, Amravati, India
Corresponding author: Manish Rana

Date of Submission: 25-08-2020 Date of Acceptance: 05-09-2020

ABSTRACT: Long-Term Memory (LSTM) is a specific neural network (RNN) organization designed to model temporary sequences and their long-distance dependencies more accurately than conventional RNNs. This paper, has examined the structures of the LSTM RNN and made some changes in its performance better. LSTM RNNs work better than DNNs. Here, some changes have been made to the gate count and remove some unnecessary elements for the standard LSTM design. This structure effectively uses model parameters than other imaginable, rapidly changing, and exudes an in-depth neural network feed with the order of the largest parameters. LSTM final loss is less than the final LSTM loss.
Keywords- Short-Term Memory, LSTM, Recurrent neural network, RNN.

I. INTRODUCTION:

Deep Neural Network (DNN) is an extremely detailed model that can read a very sophisticated vector-to-vector map. Recurrent Neural Network (RNN) is a DNN set for data sequencing, and as a result

The RNN is also extremely specific. RNNs maintain a vector activation step for each step, which makes the RNN extremely deep. Their depth, too, makes them difficult to train due to explosions and disappearing gradient problems [3] [13] [14].

There have been many attempts to address the difficulty of RNN training. The missing gradients were successfully answered by Hochreiter & Schmidhuber (1997), who developed the formulation of Long Short-Term Memory (LSTM), which is resistant to gradient extinction. LSTM turned to make it easier to use, it has become the standard way to deal with the problem of perishable gradient. Other efforts to overcome the problem of extinct gradient include the use of secondorder power algorithms [18] [19] and familiarity with RNN instruments that ensure that the gradient does not end [23], and stop studying repeating instruments completely [15] [16]] and the careful detection of RNN parameters [25] [26] In contrast to the disappearing gradient problem, the explosive gradient problem was simply easier to deal with by simply forcing more severe delays than the normal gradient [20] [23].

The criticism of LSTM's design is that it is ad-hoc and that it has a large number of objects whose purpose is not immediately apparent. Because of this, it is also not clear whether LSTM is well-built, and it is possible that there are better facilities.

Motivated by this critique, we tried to find out if the LSTM structure was right with a detailed search of the properties. We have found some properties such as the Gated Recurrent Unit (GRU) [6] That surpassed LSTM and GRU's many functions, or a variant of LSTM yielded better results when used to quit. In addition, by adding 1 bias to the LSTM forgettable gateway. We can close the gap between LSTM and better facilities.

Long-term Memory:

In this section we will briefly describe the construction of LSTM. Number 1 traditional formation of LSTM.The Standard RNN suffers from explosion and decay of gradients [3] [13]. Both of these problems are caused by the RNN multiplication environment, its gradient equally equal to the recurring weight matrix raised to high power. These duplicate matrix forces cause gradient to increase or decrease significantly over the number of steps. While learning can suffer if the gradient is reduced by a large factor too often, gradient cutting is most effective whenever the gradient has a small frequency most of the time.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 5, pp: 618-622 www.ijaem.net ISSN: 2395-5252

The full description of LSTM includes St computing circuitry and data extraction circuits from St Unfortunately; different doctors use a different version of LSTM. For this function, we use the LSTM format described directly below. It is similar to the construction of [10] but without mineral connections:

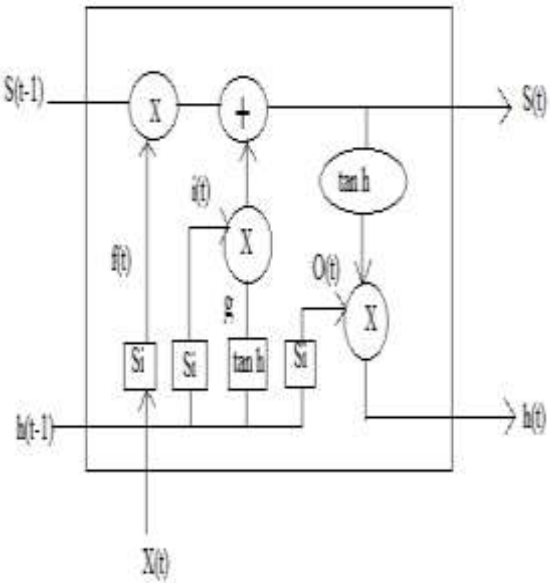

Figure 1 of LSTM buildings

In the above structure of LSTM the symbols are defined as,
S (t-1): The state of the previous cell
h (t-1): The hidden state of the previous cell
f (t): Forget the gate
(t): Gateway to information
Si: Sigmoid function
X (t): Current input
X: Vector repetition, in this paper must be written *.
O (t): Result
S (t): The current state of the cell
h (t): The current hidden state of a cell

In this LSTM structure the state of the cell storage state. Based on the current installation LSTM decides how much of the previous data should be deleted. This action was performed with the help of Forget the gate. Once the previous information has been removed new information is added to the cell using the Information gate. The figures are,

i (t) = Si (WxiXt + Whi ht-1 + bi)
f (t) = Si (WxfXt + Whf ht-1 + bf)
g = tanh (WxgXt + Whght-1 + bg)
(t) = Si (WxoXt + Who ht-1 + bo)
I-S (t) = S (t-1) * f (t) + i (t) * g
h (t) = tanh (S (t)) * O (t)

The 'W' weight vector is started randomly. 'B' bias value is also started randomly. All weight loads are renewed after each repetition.
The architecture of LSTM learns more and more about training and good performance in the long run and short-term memory.

II. METHOD:

In order to work on this art we create inclusion data in the program. The created database will be in the range defined by the LSTM build standard. The improved architecture has made two changes to the standard architecture that helps LSTM work more efficiently.

First, there is a general amount of preconceived notion of deletion and new additional information was determined separately which is why it is not possible to use other full details. For new buildings the amount of data to be removed is calculated according to the number of new details that need to be added. So in the new Architecture initially the gate details will count new details to add and based on the gate release information forget the gate will calculate the amount of details to forget.

Secondly, Due to the use of the 'tanh' function when calculating h (t) some use full details in the loss which is why we decided to delete this 'tanh' function. After removing the 'tanh' from the h (t) equation the construction of buildings is more accurate and the error rate is reduced.

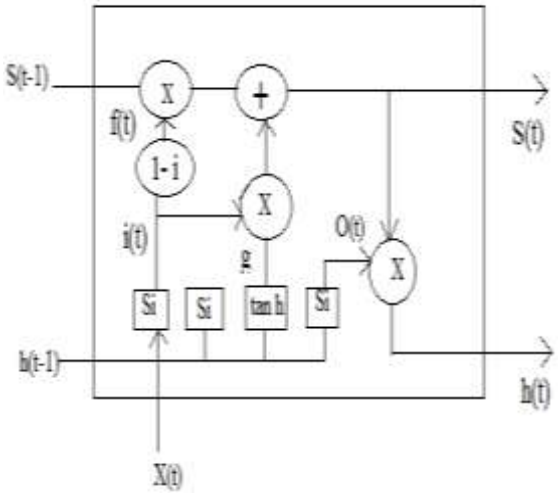

Figure 2 Modified LSTM format

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 5, pp: 618-622 www.ijaem.net ISSN: 2395-5252

After the conversion of new statistics is highlighted as,
i (t) = Si (WxiXt + Whi ht-1 + bi)
f (eg) = (1 - i)
g = tanh (WxgXt + Whght-1 + bg)
(t) = Si (WxoXt + Who ht-1 + bo)
I-S (t) = S (t-1) * f (t) + i (t) * g
h (t) = S (t) * O (t)

Algorithm / construct modification will not only work with accuracy but will reduce the value time of execution.

III. OUTCOME AND DISCUSSION:

We used LSTM in python language and worked with many variations but this modified version of the building gives us high accuracy. We also worked on a lot of modified construction by changing common functions, changing equations and reconnecting different gates with different gates but all construction does not perform well compared to standard construction.

Normal LSTM is made for data collection of 0 to 99 iterations and the result is displayed as a output screen. This short screen only showed the last part of the product with the last loss at the end of the output.

```
cur iter: 98
y_pred[0] : -0.500087106205
y_pred[1] : 0.200498204588
y_pred[2] : 0.0994746051746
y_pred[3] : -0.499579491524
loss: 7.08662403825e-07
cur iter: 99
y_pred[0] : -0.500096072911
y_pred[1] : 0.200463300125
y_pred[2] : 0.0995059890975
y_pred[3] : -0.499595630139
loss: 6.31438766408e-07
```

Figure 3 General Outcome of LSTM

In figure 3 after the last repetition the final loss is 6.31438e-07. This loss is less than all other losses of various properties except for number 2 LSTM loss losses.

Now, the modified LSTM when used to insert the same data the result is displayed as the output screen screen. This short screen only showed the last part of the product.

```
cur iter: 98
y_pred[0] : -0.499712732281
y_pred[1] : 0.199925730777
y_pred[2] : 0.100122240059
y_pred[3] : -0.500299187737
loss: 1.92494594059e-07
cur iter: 99
y_pred[0] : -0.499731288362
y_pred[1] : 0.19992779384
y_pred[2] : 0.100116124116
y_pred[3] : -0.500280730381
loss: 1.69714030648e-07
```

Figure 4 Transformed LSTM Effect

In figure 4 after the last repetition the final loss is 1.6971403e-07. LSTM final loss is less than the final LSTM loss.

	Standard LSTM	Modified LSTM
Loss	6.31438e-07	1.6971403e-07

Table 1 Loss difference

IV. CONCLUSION:

The standard LSTM configuration works best for RNN by handling the gradient extinction problem. LSTM architecture is incomplete. To make it more accurate we tried on it by changing its design so we came up with a new LSTM design that works better than standard construction. LSTM performance.

REFERENCES:

[1]. Amit, Daniel J. Modeling brain function: The world of neural attraction networks. Cambridge University Press, 1992.
[2]. Bayer, Justin, Wierstra, Daan, Togelius, Julian, and Schmidhuber, J¨urgen. It alters cellular structures in memory learning sequentially. For Artificial Neural Networks-ICANN 2009, pages 755-764. Springer, 2009.
[3]. Bengio, Joshua, Simard, Patrice, and Frasconi, Paolo. Learning long-term dependence on gradient birth is difficult. Neural Networks, IEEE Transaction on, 5 (2): 157-166, 1994.
[4]. Bergstra, James, Breuleux, Olivier, Bastien, Fr´ed´eric, Lamblin, Pascal, Pascanu,

Razvan, Desjardins, Guillaume, Turian, Joseph, Warde Farley, David and Bengio, Joshua. Theano: CPU and GPU expression expression compiler. At the Proceedings of the Python for Scientific Computing Conference (SciPy), June 2010.

[5]. Boulanger-Lewandowski, Nicolas, Bengio, Joshua, and Vincent, Pascal. Modeling a temporary dependence on high sequence: Application for polyphonic music production and recording. ArXiv preprint arXiv: 1206.6392, 2012.

[6]. Cho, Kyunghyun, van Merrienboer, Bart, Gulcehre, Caglar, Bougares, Feti, Schwenk, Holger, and Bengio, Joshua. Reading phrase presentations using rnnencoderdecoder for mathematical translation. arXiv preprint arXiv: 1406.1078, 2014.

[7]. Chung, Junyoung, Gulcehre, Caglar, Cho, KyungHyun, and Bengio, Joshua. Powerful testing of repeated neural networks in modeling sequences. arXiv preprint arXiv: 1412.3555, 2014.

[8]. Collobert, Ronan, Kavukcuoglu, Koray and Farabet, Cl .ement. Torch7: A machine-like space for machine learning. At BigLearn, NIPS Workshop, number EPFL-CONF-192376, 2011.

[9]. Gers, Felix A, Schmidhuber, J¨urgen, and Cummins, Fred. Learns to forget: Continuous prediction of lstm. Neural Census, 12 (10): 2451-2471, 2000.

[10]. Cemetery, Alex. It creates a sequence of repeated neural networks. arXiv preprint arXiv: 1308.0850, 2013.

[11]. Greff, Klaus, Srivastava, Rupesh Kumar, Koutn´ık, Jan, Steunebrink, Bas R, and Schmidhuber, J¨urgen. Lstm: odyssey search space. arXiv preprint arXiv: 1503.04069, 2015.

[12]. Hinton, Geoffrey E and Shallice, Tim. Lesioning network attraction: a diagnosis of dyslexia found. Psychological Review, 98 (1): 74, 1991.

[13]. Hochreiter, Sepp. Untersuchungenzudynamischenneuronalenn etzen. Master's Thesis, Institut fur Informatik, TechnischeUniversitat, Munchen, 1991.

[14]. Hochreiter, Sepp and Schmidhuber, J¨urgen. Long-term memory. Neural Census, 9 (8): 1735-1780, 1997.

[15]. Jaeger, Herbert. An echo-style approach to analyzing and training neural networks is repetitive — with the erratum note. Bonn, Germany: German National Research Center for GMD Technical Report, 148: 34, 2001

[16]. Jaeger, Herbert and Haas, Harald. Harnessing Nonlinearity: Predicting disruptive systems and saving power over wireless connections. Science, 304 (5667): 78-80, 2004.

[17]. Marcus, Mitchell P, Marcinkiewicz, Mary Ann, and Santorini, Beatrice. Creating a large corpus described in English: Penntreebank. Computer languages, 19 (2): 313-330, 1993.

[18]. Martens, James. In-depth learning through free hessian optimization. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pages 735-742, 2010.

[19]. Martens, James Sutskever and Ilya. Studying neural networks repetitively and doing good non-hessian. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pages 1033-1040, 2011.

[20]. Mikolov, Tom´a ˇs. Mathematical Language Models are based on Neural Networks. PhD thesis, PhD thesis, Brno University of Technology, 2012. -, 2012.

[21]. Mikolov, Tomas, Karafi´at, Martin, Burget, Lukas, Cernock` y, Jan, and Khudanpur, Sanjeev. A network model based on a standard network. At INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010, pages 1045-1048, 2010.

[22]. Mikolov, Tomas, Joulin, Armand, Chopra, Sumit, Mathieu, Michael, and Ranzato, Marc'Aurelio. Learning long memory in repetitive neural networks. arXiv preprint arXiv: 1412.7753, 2014.

[23]. Pascanu, Razvan, Mikolov, Thomas, and Bengio, Joshua. It is difficult to train neural networks over and over again. arXiv preprint arXiv: 1211.5063, 2012.

[24]. Plaut, David C. Semantic and the engagement of the social networking network. Continuation of the 17th annual conference of the community of cognitive science, volume 17, pages 37-42. Pittsburgh, PA, 1995.

[25]. Sutskever, Ilya, Martens, James, Dahl, George, and Hinton, Geoffrey. With the importance of initiation and momentum in in-depth learning. In Proceedings of the 30th International Conference on Machine

Learning (ICML- 13), pages 1139–1147, 2013.

[26]. Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc VV. Sequential learning sequences for neural networks. In Advances in Neural Information Processing Systems, pages 3104-3112, 2014.

[27]. Zaremba, Wojciech and Sutskever, Ilya. Learning to do. arXiv preprint arXiv: 1410.4615, 2014.

[28]. Zaremba, Wojciech, Sutskever, Ilya, and Vinyals, Oriol. Normal neural network duplication. arXiv preprint arXiv: 1409.2329, 2014.

Author Biography

Author Detail:

Manish Rana, PhD scholar Department of Computer Science, Sant Gadge Baba Amravati University, Amravati. He has more than twelve years Teaching Experience. His area of interest includes Artificial Intelligence, Machine translation and soft computing. He has published 5 International Journal and 3 Papers in national Conference. (manishrana@live.com)

Dr. Mohammad Atique , is presently working as Professor, P.G. Department of Computer Science, Sant Gadge Baba Amravati University, Amravati. He has around 37 publications to his credit in International/National Journal and conferences. His area of interest includes Artificial Intelligence, Machine translation and soft computing.
(mohd.atique@gmail.com)