Updating the Forget Gate of the LSTM to Improve Pattern Learning

Preston Robertson

Department of Industrial and Systems Engineering, Mississippi State University, Mississippi State, MS, 39762 USA

Abstract

The Long Short Term Memory(LSTM) model is a neural network that specializes in time-series based data... The goal of this project is to improve the forget gate of the LSTM model to improve the pattern learning of the model.

Keywords: Example, example, example, example.

1. Introduction

The problem of analyzing time-series based data has been a goal of researchers, since normal artificial neural network learning can not be directly applied. The architecture of a standard neural network or a convolutional neural network(CNN) can not change or update as time goes on after training the model [1]. For example, the artificial neural network or CNN will handle image data but can not have a video as the input data. This is due to the inability to vectorize video data compared to image data. That is where the recurrent neural network(RNN) is implemented [2]. The architecture of the recurrent neural network allows for a feed-forward network that learns from results from previous nodes. The RNN models can also not have a video format as input data; however, the model processes each image in the video data in such a way that the model still learns as if the images were still in video formats. This change to architecture allows for new types of data to be analyzed, such as: video, speech, vibrations, robot control, sign language translation, etc [3].

The RNN model has a common issue of vanishing gradient when [4]. This vanishing gradient refers to the learning slowdown of the model. This vanishing gradient occurs due to the models infinite node nature. For example, cell 1 will have a larger impact on the results of cell 2 than cell 43's results impact on cell 44. This is due to the amount of information being transferred and eventually the old information will overshadow the new inputs. The Long Short Term Memory(LSTM) model is an attempt to fix this issue through adding a variable named "cell state" [2]. The LSTM was initially made to revolutionize speech detection and is even used in popular services such as Apple's Siri. The LSTM helps fix the

Email address: pgr41@msstate.edu (Preston Robertson)

vanishing gradient problem by giving each cell the ability to forget old information. Before the discussion of how the LSTM model forgets, it is important on how the information is translated through out each cell. Each cell has its own output (which is the objective of the artificial neural network) and a cell state that is a recorded value transferred through each cell [5]. This how each cell forgets data. Each cell takes the previous layer's output and the current cell state and compares them. The more these values do not match, the more the cell forgets its cell state. This process is called the forget gate by learning through the updates of weights and biases [6].

The forget gate has had significant impact in the analysis of speech data [7], which is partially the reason why the accuracy of speech recognition in products such as Amazon's Alexa has increased. However, the LSTM model is not yet perfect. The forget gate has issues itself. The cell state is a single value propagating through each cell [8]. When the cell state drops to a significantly low value in the forget gate, then essentially all information before is lost. This is an issue due to the pattern nature of data sets. For example, let's say the LSTM model is analyzing stock prices. If the stock price has a significant decline then model will forget the old information and adjust to the new information. However, a researcher may notice the pattern be a seasonal issue (such as the stocks of a swimming wear company). The LSTM model will never find this pattern and will have significant error when the pattern repeats. The second issue with the cell state being a single value, keeps the vanishing gradient problem. Since the third cell will have a larger impact on the cell state value than the impact of the 45th cell in the LSTM model. The objective of the paper is to reduce the vanishing gradient problem through adding more values similar to the cell state.

2. Literature Review

The first LSTM model was proposed in 1997 as a solution to the RNN models difficulty to handle long term dependencies in data [9]. The model has found great success in speech recognition data, DBLSTM-HMM by Alex Graves at the University of Toronto [10]. The original LSTM model initially only added the cell state to the RNN model. This model architecture was very popular in the early 2000s before the invention of the forget gate. then later the forget gate was added by Gers, Schmidhuber, and Cummins in the year 2000 [11]. The motivation for this addition to the RNN model was due to the over saturation that forms in the model. Due to the inability to forget data, the cell state would not properly update to represent the new data. Let's say that there is an original LSTM model that predicts where Jim eats lunch. Jim went to his favorite restaurant for the last 5 years, but then it closed down for the winter season. Without the ability to forget, the LSTM model will continue predict Jim will eat at the restaurant despite that not being the correct output value. This is due to the over saturation of the previous data. This over saturation happens in several data sets [12] such as predicting stocks, since there are outside factors that the model can not account for, the model will be wrong about trends. However, with

the ability of the model to forget the old data, then the model can adapt to the new trends despite not getting all the information.

3. Methodology

To test the proposed method, both solutions will be tested along side a GRU, normal RNN, and LSTM with a forget gate. Each of these models will be tested using five different data sets. The first data set will be a simple function, most likely the Sine function since that seems to be an industry standard. The next data set will be a random companies stocks. The next two data sets will be image based data sets attempting to detect some identity in the image. The final data set will be a speech-based data set. Each of these models were selected to cover the common uses of the time-series based models.

3.1. Mathematical Model

To further development of the LSTM model by attempting to add the capability of pattern learning. This pattern learning would allow for new forms of analysis. Such as conversation tracking and seasonal stats analysis. This update will come from two different solutions. These solutions will be tested and analyzed to determine which theoretical properties perform better in each data set selected. Since this is the project proposal, the mathematical models have not yet been made for either solution.

3.2. Solution 1

Implementing a multi-track system where inputs are taken at specific intervals (on top of the LSTM model). Adding in a second cell state that keeps the output from the previous cell states and implements that value to the forget gate. These values will store an amount of values up to k which will be pre-determined by the researcher and the data set. The way this solution would work is that the researcher would set a k value that will call back k nodes and implement that output into the current forget gate. This calling function would allow for better pattern learning. For example, the ability to recall the same date from the previous year's value. This is an easy example to show the usefulness of this solution. The consequences of this change would be the increase to the expense of computational power due to the vector of values saved through each iteration.

3.3. Solution 2

This solution involves keeping cell states and calling back cell states with similar values to the previous output. In mathematical terms, find where ht = mt (mt is where the cell states are stored). The next output (or the m(t+k) value) from the previous cell state is then inserted into the current forget gate. This theoretically would allow for trends to be found through-out the data set by effectively calling previous cell states. This would greatly increase the pattern learning of the model since it effectively taking the derivative (or rate of change) at similar node states.

4. What's Next

4.1. For this Project

The next steps in the process is first and foremost to make final versions of the mathematical models that can work in this forget gate. The next step is deciding the final data sets, these will be determined by popularity. Some have already been decided, but were not included since not all have been found. The next step would be to implement the new forget gate into the LSTM models and running through the methods discussed above. This is not necessary, but improvements to my current document for the final project is the last step. Since the beginning of this paper, I have learned a lot more on LaTex and will be able to make the document more organized. After the major hurdle of making the mathematical models, the project should finish decently quickly afterwards since the rest of the process is standardized, and will use previously made models. This project should finish on time.

4.2. For Future Work

Possible improvements to the given solutions that are outside the scope of this project. The improvement to the first solution would be taking the derivative of the cell states around the callback node with respect to node number. This would be useful information since the current cell would learn the jump size between each output keeping the rate of change the same. Since this value is called back from a preset k value, this would theoretically be more value able information. This is due to the learning rate of the previous cells and the direction of the previous season would be more valuable than the specific values. Solution 2 would benefit from the previous cell trends rather than the specific value. However, since it is an average of the previous values with the same cell state value it is not as bad as the first solution using specific values. This does not mean that the specific numbers are better than the trends. Since the objective is to improve pattern learning, it is important to track general trends. These have not been implemented due to the complex problem of data analysis within a neural network model and translating the trend values to be in the same domain as the cell states. These problems are above this paper, but do not seem impossible for future work.

References

[1] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[2] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[3] H. Xiao, M. A. Sotelo, Y. Ma, B. Cao, Y. Zhou, Y. Xu, R. Wang, and Z. Li, "An improved lstm model for behavior recognition of intelligent vehicles," *IEEE Access*, vol. 8, pp. 101 514–101 527, 2020.

[4] S. Hochreiter and J. Schmidhuber, "Lstm can solve hard long time lag problems," *Advances in neural information processing systems*, vol. 9, 1996.

[5] F. Qian and X. Chen, "Stock prediction based on lstm under different stability," in *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. IEEE, 2019, pp. 483–486.

[6] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[7] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A c-lstm neural network for text classification," *arXiv preprint arXiv:1511.08630*, 2015.

[8] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.

[9] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[10] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 2013, pp. 273–278.

[11] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[12] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.