

# Neural Network Applications: LSTM Recurrent Neural Network for Local Weather Prediction (December 2017)

Derek B. Munday

**Abstract—** The purpose of this effort is to apply machine learning to enhance weather predictive capabilities. A recurrent neural network with long short-term memory was trained to predict a set of weather data given the 48 preceding hours of weather data sets. This paper will discuss the methods used to create the trained model and some preliminary results from the model.

**Index Terms—** machine learning; recurrent neural network; LSTM; adam optimizer; xavier initialization;

## I. INTRODUCTION

MANKIND has been trying to determine the patterns of the weather for all of recorded history and surely earlier as well. The sheer complexity of the system makes it difficult to model. Over time, mankind has been working on demystifying our planets weather. Physics, the mathematical modeling of the physical universe and how it moves, and more specifically fluid dynamics and thermodynamics have given us great insight into our planets weather system. Instruments have also been developed to measure the attributes of the heterogeneous fluid we call the atmosphere so that weather can be predicted to higher degrees of accuracy. These instruments have accumulated decades if not centuries of historical weather data that can be used to improve our weather models. Why then is the weather man still attributed the adage of being the one profession that is paid to be wrong? The simple answer is that while we have made great strides in understanding the weather, we are not done yet.

As was mentioned earlier, Earths weather system is very complex. It has taken [entire lifetimes of highly dedicated scientists](#) [1] to produce even portions of the field of Physics. Some of the limitations of the scientific forefathers were the tools available. Highly complex systems frequently require highly complex models. In Isaac Newton's day for example, it would be highly impractical to try and calculate the temperature for tomorrow by hand even if a suitable model did exist. In fact this was actually [attempted](#) in the 1920 but the advent of computers has reduced the calculation time to less than that of the forecast period itself.

Computers then have reduced the time it takes to calculate our models so that their accuracy can be tested, but what about actually developing the models. We mentioned before that entire lifetimes have been poured into mathematically modeling the physical world. Wouldn't it be beneficial if computers could do for model creation what computers have done for model calculation? As it turns out an amalgam of computer science, data science, and statistics called machine learning is well suited to this very thing.

To be specific a subset of machine learning called Artificial Neural Networks (ANN) is well suited to helping to rapidly create model solutions like those that we need to predict weather patterns. ANN's are "computing systems inspired by the biological neural networks that constitute animal brains." [2] ANN's can be thought of as modeling specific mental abilities. Whereas the neural network of a person is required to be very broad in its purpose, an ANN can be used to solve very specific albeit complex problems such as weather prediction.

This paper describes a particular example of using an ANN to analyze a large amount of weather data from a single location. The results of the ANN is a model of the detected patterns and structures within the data.

## II. METHODS

### A. Data

There is much raw weather instrument data available today. The [National Oceanic and Atmospheric Administration's \(NOAA\) National Centers for Environmental Information \(NCEI\)](#) is responsible for preserving, monitoring, assessing, and providing public access to the Nation's treasure of climate and historical weather data and information. [3] Intricacies and patterns in the data are subtle and difficult to discern using standard analysis and visualization methods. New ways of analyzing the data need to be utilized to give further insight into how the Earth's weather system behaves.

The location of the weather data collection site chosen for this effort is the Middle Georgia Regional Airport (KMCN). Hourly weather instrument readings for the last three days can be found for free [here](#), while up to ten years of historical weather instrumentation for this site can be requested from NOAA. The ten years of historical data will be used to train

an ANN. Since NOAA requires a request be made to access its data, the three most recent days of observations (which are always available online) should be used to predict weather at this location.

## B. Environment

### 1) Tensorflow Dataflow Framework

Tensorflow was chosen as the dataflow framework. Tensorflow is an open-source symbolic math library commonly used for machine learning applications such as ANNs. Google Brain developed Tensorflow as the successor to DistBelief which is Google's proprietary machine learning system. [4]

Tensorflow was chosen over other frameworks (such as scikit-learn) because it allows the ability to take advantage of graphics processing units (GPU) or even tensor processing units (TPU) for more efficient training. [5] GPU/TPUs were not used for this example because the example looks at a relatively small part of the overall system, however there is nothing inherent in the code which would restrict this example to CPU processing. There is a fine [tutorial](#) for setting up Tensorflow to use either CPUs or GPU/TPU. It is important that GPU/TPU processing ability be available to build upon the work outlined here.

### 2) Python Scripting Language

Python was chosen as the programming language for many reasons. One of the primary reason for choosing Python was because "it is sufficiently simple to be taught to school children with great success." [6] Another reason for choosing Python is that "Python has also emerged as one of the leading open platforms for data science" [6] Since Python is widely used in the data sciences it has a huge [online community](#) devoted to furthering the Python open-source knowledgebase. Finally, and perhaps most importantly, python was chosen because it is the only supported language of Tensorflow which is covered by the [API stability promises](#).

### 3) Jupyter Notebook Application

The Jupyter Notebook Application produces documents that are a combination of rich text and computer code. The rich text contains human readable analysis descriptions and results. The code, which is collocated and interspersed throughout the document is modifiable and executable within the document. A notebook was created for the development of this example.

### 4) Tensorboard

Tensorboard is a suite of applications for inspecting and understanding your Tensorflow runs and graphs. [7] There are many tools to graph the outputs of a Tensorflow model. In addition to creating scalars, histograms, images, audio, and graphs Tensorboard forces you to apply good design concepts by scoping your Tensorflow network so that it can be visualized.

## C. Data Normalization

One of the most basic equations for an ANN node can be expressed as follows:

$$y = Wx + B \quad (1)$$

In (1),  $y$  is the output of the node while  $x$  is the input. Now the ANN has no control over the inputs in fact we want it to

learn about the inputs and not modify them. For actual learning to take place though the ANN needs to modify the equation of this node so that it can produce the proper outputs. That is where the variables 'W' and 'B' come into play. 'W' and 'B' stand for weight and bias respectively. The node weights the input and adds a bias to produce the desired output. The weight and bias are the keep variable modified by the ANN to "learn" the data.

All that said the input data itself can vary wildly when you take into account that it will be used for multiplication. The day of the year input can go as high as 366 while the lowest nonzero value for precipitation is 0.01. Now a human knows that the amount of precipitation last hour has much more meaning as to whether it will rain next hour than does which day of the year it is. Each of these values plugged into the equation above are going to differ significantly though. This is not necessarily a problem with getting the right answer so much as it is a problem with time. If an ANN can get the right result it will take a long time to work out the weights and biases to harmonize the variability of these two features. To solve this variability we normalize the data.

The inputs for the first layer of the model are normalized between the values negative 0.5 and positive 0.5. A custom normalization for the raw data was created using the maximum and minimum values found for each feature.

Another issue in deep neural networks is internal covariate shift between layers. Normalizing the distribution of layer input saves valuable processing time that would otherwise be wasted adapting each layer to the new distribution. [8]

## D. Model Selection

The model that is chosen for machine learning depends largely on the attributes of the data that you want an ANN to learn and the intended results that you want the ANN to produce. Since we want our ANN to predict future values we have chosen a Recurrent Neural Network (RNN) to use with our data. RNNs are being applied to various problems such as speech recognition, language modeling, translation, and image captioning to name a few. [9]

The RNN is an effective tool for finding structure in sequential data. [10] Our data from KMAC is indeed sequential, but RNNs suffer from a unique problem. An exploding gradient is a problem with generic RNNs that refer to a large increase in the norm of a gradient during training. [11] To avoid the exploding gradient problem we used a specific type of RNN model called a Long Short-Term Memory (LSTM). LSTM is a gradient-based method which enforces constant error flow through gated learning units. [12]

## E. Loss Function

We have chosen the  $L_2$  Loss Function for the purposes of identifying the difference between the outputs of our model and the labels used while training said model.  $L_2$  was chosen over  $L_1$  due to the fact that  $L_2$  tends to reduce the effects of outliers in the data. [13]

## F. Optimization

We have chosen an Adam Optimizer. Adam optimization was chosen due to it being "an algorithm for first-order gradient-based optimization of stochastic objective functions,

based on adaptive estimates of lower-order moments.” [14]  
Which sounds sufficiently awesome.

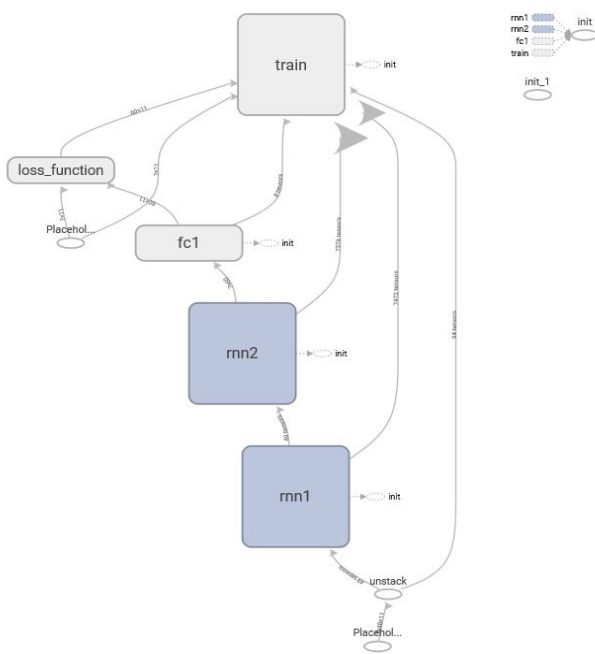
### G. Model Execution

This Tensorflow model consists of three layers. The first and second layers are static RNN layers each composed of sixty Layer Normalized Basic LSTM Cells. The final layer is a standard fully connected layer with a linear activation function. Xavier initialization was used to initialize the weights and biases for the final layer.

The model was trained using randomly populated batches of one hundred inputs. Each input consisted of 48 time steps which each had 11 features. Over 67,000 unique inputs were sampled from to create the batches. The training session executed 10,000 iterations which processed one batch each.

## III. RESULTS

### A. Tensorflow Model Graph

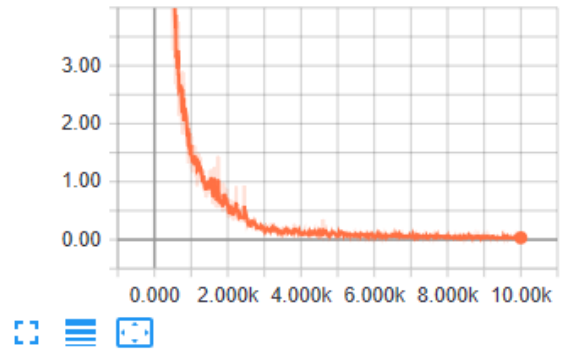


Tensorboard produces graphs of the model itself. The above graph is interactive within the Tensorboard interface. Each cell can be double clicked to investigate nested details of the model. An issue we ran into while navigating the graphs was the Tensorboard server dropping out. Double clicking a cell would cause the tool to freeze up and the browser would report a lack of response from the local host. This problem made working with the Tensorboard suite of tools very difficult to work with.

### B. Loss Optimization

As can be seen in the graph below the Adam Optimizer does indeed lower the loss of the predictions of the model.

loss\_function/loss\_function



## IV. CONCLUSIONS AND FURTHER WORK

The example laid out previously is only the beginning of an effort to analyze a complex system such as the Earth's weather. There is much left unfolded from the topics above. LSTMs are but one class of many neural networks that are useful in analyzing data for the purpose of predicting. Initialization and optimization routines also need further research to improve on the work presented here. Furthermore, the model described in this paper has room for expansion. Algorithms to evaluate the testing data loss need to be developed to verify the accuracy of the model. The model needs to continue to be tested against current real weather readings to determine how the model needs to be improved. Since the model produces outputs that are of the same shape and structure as the inputs, the outputs can be fed back into the model to create predictions even further into the future. All of these topics require further research and development.

## V. REFERENCES

- [1] Wikipedia, "History of physics - Wikipedia," Wikimedia Foundation, Inc., 23 November 2017. [Online]. Available: [https://en.wikipedia.org/wiki/History\\_of\\_physics](https://en.wikipedia.org/wiki/History_of_physics). [Accessed 10 December 2017].
- [2] Wikipedia, "Artificial neural networks - Wikipedia," Wikimedia Foundation, Inc., 10 December 2017. [Online]. Available: [https://en.wikipedia.org/wiki/Artificial\\_neural\\_network](https://en.wikipedia.org/wiki/Artificial_neural_network). [Accessed 10 December 2017].
- [3] NOAA, "National Centers for Environmental Information (NCEI) formerly known as National Climatic Data Center (NCDC) | NCEI offers access to the most significant archives of oceanic, atmospheric, geophysical and coastal data," Department of Commerce, [Online]. Available: <https://www.ncdc.noaa.gov/>. [Accessed 10 December 2017].
- [4] Google Brain, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operation Systems Design and Implementation (OSDI '16)*, Savannah, GA, 2016.

- [5] S. Raschka, "Machine Learning FAQ," 2017. [Online].  
] Available:  
<https://sebastianraschka.com/faq/docs/tensorflow-vs-scikitlearn.html>. [Accessed 10 December 2017].
- [6] C. Rossant, Learning Ipython for Interactive Computing  
] and Data Visualization - Second Edition, Packt  
Publishing, 2015.
- [7] LearningTensorFlow.com, "Learning TensorFlow ::  
] Visualisation with TensorBoard," [Online]. Available:  
<https://learningtensorflow.com/Visualisation/>. [Accessed  
10 December 2017].
- [8] S. Ioffe and C. Szegedy, "Batch Normalization:  
] Accelerating Deep Network Training by Reducing,"  
[Online]. Available:  
<http://proceedings.mlr.press/v37/ioffe15.pdf>. [Accessed  
10 December 2017].
- [9] C. Olah, "Understanding LSTM Networks - colah's blog,"  
] 27 August 2015. [Online]. Available:  
<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Accessed 10 December 2017].
- [1] R. Pryzant, "Evaluating Tensorflow," [Online]. Available:  
0] <https://pdfs.semanticscholar.org/08ea/577896374a6a3f315f0c15519778ab56b92e.pdf>. [Accessed 10 December  
2017].
- [1] R. Pascanu, T. Mikolov and Y. Bengio, 21 November  
1] 2012. [Online]. Available:  
<https://pdfs.semanticscholar.org/728d/814b92a9d2c6118159bb7d9a4b3dc5eeaaeb.pdf>. [Accessed 10 December  
2017].
- [1] S. Hochreiter and J. Schmidhuber, "nc.dvi," 1997. [Online].  
2] Available:  
<http://www.bioinf.jku.at/publications/older/2604.pdf>.  
[Accessed 10 December 2017].
- [1] R. Shukla, "L1 vs. L2 Loss function - Rishabh Shukla,"  
3] 28 July 2015. [Online]. Available:  
<http://rishy.github.io/ml/2015/07/28/l1-vs-l2-loss/>.  
[Accessed 10 December 2017].
- [1] D. P. Kingma and J. L. Ba, 30 January 2017. [Online].  
4] Available: <https://arxiv.org/pdf/1412.6980.pdf>. [Accessed  
10 December 2017].