B. Ghosh, I. Kalyan Dutta, A. Carlson and M. Totaro. (Oct. 20-22, 2020). Mathematical Modeling and Prediction of Neural Network Training based on RC Circuits, *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, Montreal, QC, Canada, 2020, pp. 1-4, doi: https://doi.org/10.1109/ISNCC49221.2020.9297244]

ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/344787842

Mathematical Modeling and Prediction of Neural Network Training based on RC Circuits

Conference Paper · October 2020

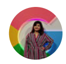
DOI: 10.1109/ISNCC49221.2020.9297244

CITATIONS
0

READS
135

4 authors:

Bhaskar Ghosh
University of Louisiana at Lafayette
12 PUBLICATIONS **47** CITATIONS



Indira Kalyan Dutta
Arkansas Tech University
15 PUBLICATIONS **54** CITATIONS

Albert H. Carlson
Austin Community College
27 PUBLICATIONS **43** CITATIONS

Michael Totaro
University of Louisiana at Lafayette
46 PUBLICATIONS **320** CITATIONS

SEE PROFILE

SEE PROFILE

SEE PROFILE

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project An Intro to Local Entropy and Local Unicity View project

Project Lightweight Cryptography and Polymorphic Encryption as a Security Solution for the Data Associated with Internet of Things Devices View project

All content following this page was uploaded by Bhaskar Ghosh on 21 October 2020.

The user has requested enhancement of the downloaded file.

Mathematical Modeling and Prediction of Neural Network Training based on RC Circuits

Bhaskar Ghosh*, Indira Kalyan Dutta†
University of Louisiana at Lafayette
Louisiana, USA
*bhaskar.ghosh1@louisiana.edu,
†indira.dutta1@louisiana.edu

Albert Carlson
Kyle, Texas
ltzap1@gmail.com

Michael Totaro
University of Louisiana at Lafayette
Louisiana, USA
michael.totaro@louisiana.edu

***Abstract*—A recent study [1] and our experimental results show that a Neural Network(NN) often learns the most important features early during training. This results in accuracy during training to follow similar curves in each case. The magnitude of weight updates are the highest during the early iterations and weight updates stabilize with continued training of the NN. In this paper, we examine the training curve of a neural network with respect to an Iteration Weight value of τ. Using τ, we can predict the approximate maximum accuracy and the required number of training examples to reach that specific accuracy level. The model resembles the well understood Resistance-Capacitor(RC) Charging Circuit and appears to act accordingly. Our motivation stems from the fact that predicting the number of training iterations for a model to reach a desired level of accuracy has not been attained. As such, our work should be useful for researchers in their efforts to improve their training designs.**

***Index Terms*—Neural Networks, Training, Electromagnetism, Prediction**

I. Introduction

Considerable research has been done in attempting to understand the performance of Neural Network (NN) training [2]–[4]. Initially, the NN undergoes the maximum change in accuracy with respect to a small number of training examples. The greatest impact and the highest gradient change to accuracy occurs in the early training iterations. As one drills deeper into the training process, the accuracy tends to oscillate at later stages as the weights are fine-tuned; however, the curve widely retains the same general trend of the curve.

With an increasing number of training examples, there appears to be a change in the magnitude of accuracy, which gradually decreases and reaches an asymptotic accuracy line. Figure 1 illustrates the evaluation accuracy of a ResNet-20 architecture [1] with respect to the number of training iterations. The shape of the curve is characteristic of NN training. We have observed the same curve in our training research. Moreover, we observe further that the same curve occurs for the RC Charging Circuit curve, as shown in Fig. 4.

To the best of our knowledge, mathematically modelling the accuracy with respect to the training progression of a NN has not been done. We draw from the concepts of electromagnetism and RC Charging as motivation to address this issue. We seek to predict an approximate maximum accuracy that can be achieved by the NN, as well as the number of training iterations required for a NN training, based on initial training values, in order to reach a desired level of accuracy. Further, we are guided by the principles and concepts of abstract algebra [5] and topology [6], which tell us that we can apply the same solution to problems with the same mathematical description [7]. We will use the RC Charging Circuit equation to model the training graph, as the NN training graph is mathematically observed as similar to the RC Charging Circuit graph shown in Fig. 4, as follows:

- Mathematically model the training of a NN.
- Predict the maximum achievable accuracy of NN training based on the RC Charging Circuit.
- Predict the number of iterations required to reach the maximum predicted accuracy.

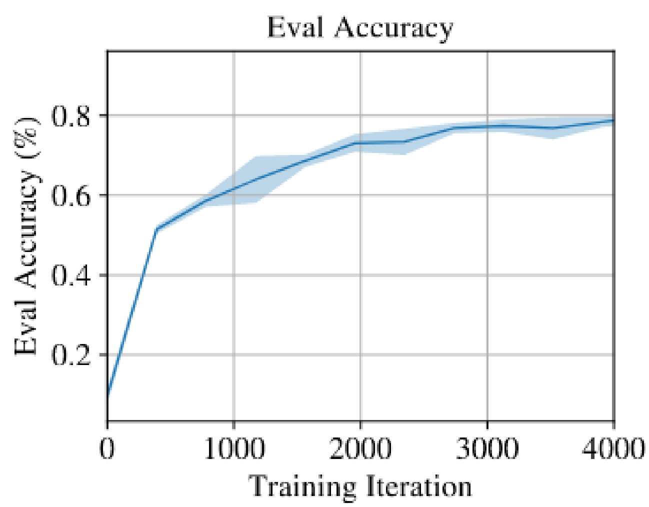


Fig. 1. Training vs Accuracy

II. Background

A. Neural Networks

Neural Networks (NNs) [8] can be defined as a structure that contains an input layer and an output layer, with computations taking place in the output layer, which is also referred to as a *perceptron* or a *single layer neural network*. Similar to the operations of a biological brain, the input layer transmits data that it receives as input to the output layer for the computations to take place and an output is presented. Often, intermediate layers are placed in the network as additional computation resources and are generally abstracted from the user. These layers are known as *hidden layers*. When a network contains more than a single computational layer, it is known as a *multilayer neural network*. NNs consist of a network of neurons in multiple layers (e.g. fully-connected, convolutional, pooling, etc.), whereby each layer has an associated set of weights that are applied on features of the input to transform them into a desired results for the subsequent layers [9], [10].

B. Neural Networks Training

For multilayer NN training, optimal weights are calculated for the features by a method known as *backpropagation*. NNs have a predefined loss function, or an objective function, that it tries to minimize to increase the training accuracy, which will increase the overall accuracy of the NN. The gradient update to the loss function is calculated by backpropagation, which has two main phases: the *forward pass* and the *backward pass*, as shown in Fig. 2.

- A *forward pass* takes place when an input is fed to the NN input layer followed by a cascade of computations across the next layers with current weights, and an output is generated by the output layer. This output is then compared with training values and the weight updates are computed based on the derivative of the loss function with respect to the output.
- A *backward pass* is conducted after the calculation of the gradient of the loss function. Once the gradients are calculated, the values are sent to each layer where the weights associated with each neuron are updated. Since updates are conducted in the opposite direction, it is called a backward pass [10].

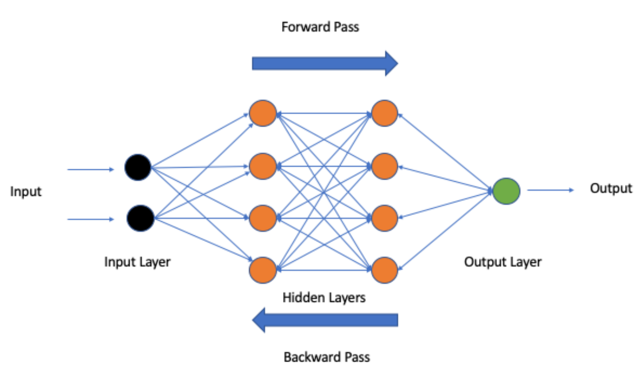


Fig. 2. Training in a Neural Network

Next, we define several terms that are related to training. When a NN completes training over the whole data set, it is known as one *epoch*. Networks are not generally trained over a whole dataset, instead datasets are divided into smaller instances or *batches* before the training is performed. *Batch size* is the number of batches over which the data are divided. The number of batches that must be iterated, in order to complete one epoch is known as an *iteration*. For example, if we have 500 training instances and we create a batch size of 10 batches, we would have to complete 50 iterations to complete one epoch [11].

As previously mentioned, the magnitude of weight changes due to backpropagation is highest during the first few iterations of training. The magnitude of change slowly decreases over a period of time and the network finally stabilizes. This phenomenon is illustrated in Fig. 1, where we see a sharp rise initially in training accuracy from the high magnitude of weight updates and the gradual stabilizing in later iteration phases [1]. We shall later discuss the concept of Resistance and Capacitance Charging, which is used to model our training prediction.

C. Resistance and Capacitance Charging Circuit

A Resistance and Capacitance Circuit (RC circuit) contains a voltage source ϵ, a resistor R and a capacitor C, which stores an electric charge in a circuit. The circuit shown in Fig. 3 allows the capacitor to be charged.

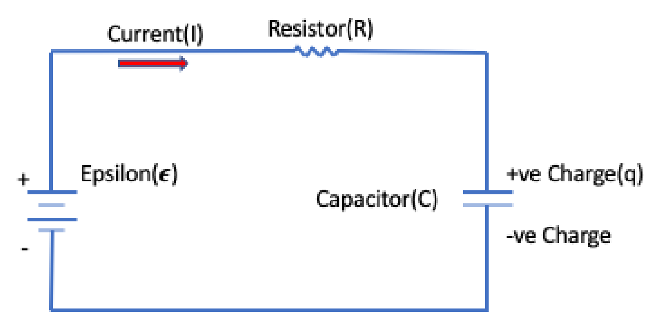


Fig. 3. Charging of a Capacitor in a RC Circuit

We use Eq. 1 in order to understand the voltage difference between the capacitors. The equation is as follows:

$$V_C(t) = \epsilon(1 - e^{-\frac{t}{\tau}}) \quad (1)$$

where ϵ is the highest voltage, t is the time and $\tau = RC$, which is known as the **time constant**. The graph of charging over time can be seen in Fig. 4 [12].

III. Proposed Model

From Fig. 1 and Fig. 4, we can observe the similarity of the graphs. The goal is to find the relationship between the time constant τ and the number of training iterations required.

A. Value of τ

The value of τ is a characteristic of the shape of the curve. 1τ is estimated at the point where the accuracy reaches the value of 63.2% [12]. At the value of 1τ, it becomes possible to accurately estimate the remainder of the curve. We will designate τ as the *Iteration Weight* of the curve. The accuracy increase for each τ changes on the curve. τ represents a

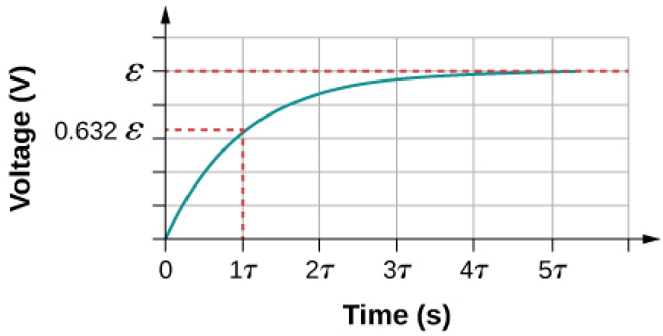


Fig. 4. Voltage difference across Capacitor

reduction of 63.2% at the maximum accuracy achievable given the characteristics of the problem. Once we have reached 63.2%, we can choose the correct maximum curve. There exists an infinite number of accuracy curves between 0% and 100% accuracy. For simplicity, we will consider the family of curves in 5% maximum increments, although any number of curves could be considered. At the point represented by 63.2% accuracy, the curves are sufficiently separated to make it possible to identify the maximum accuracy for the problem.

B. Predicting Maximum Accuracy

We will use Eq. 1 to establish a mathematical model to predict maximum accuracy of a NN training based on the trend at τ. Using the equation, we will select from the range of curves to establish the maximum accuracy. The maximum achievable accuracy should be 100%; however, given the curve type, it is theoretically impossible to reach 100% accuracy. Thus, we are searching for the asymptotic upper limit for accuracy, or the ceiling function denoted by $\lceil A \rceil$, where A is the accuracy. The minimum accuracy we seek is 65%. Although, potentially any value could be reached.

Using the following form of Eq. 1:

$$t = -\ln\left(1 - \frac{A_t}{A_{max}}\right) \tag{2}$$

where t is a fraction of the training iteration τ, A_t is the accuracy at training iteration t, and A_{max} is the maximum accuracy of the graph. We compute which of the family of graphs is the correct maximum with a maximum accuracy range of 65% to 100% at intervals of 5%, as seen in Fig. 5. The solution for t in the Eq. 2 represents the training iteration for the accuracy point with respect to the differing maximum accuracy. We will use these curves as a reference in order to approximate the maximum possible accuracy for the training curve that we want to predict. We will also use these curves to understand the training iteration for any desired accuracy level. If we look at the derivative with respect to accuracy as per Table I, we can move to the graph which it fits approximately, based on the derivative of that point on the curve.

In order to find the maximum possible accuracy for a graph under consideration, we consider the accuracy point of 63.2% for the graph. We will calculate the average rate of change for

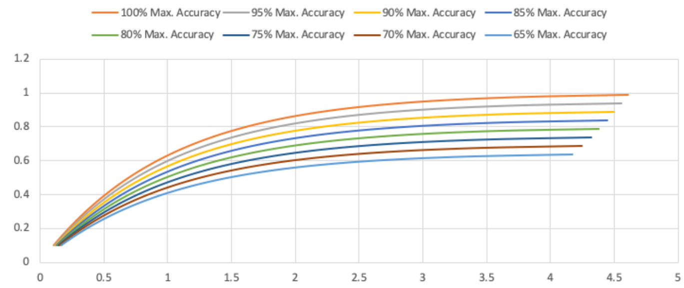


Fig. 5. Graphs for different accuracy range

TABLE I
MAXIMUM ACCURACY WITH CORRESPONDING SLOPE AT τ

Maximum Accuracy	Derivative at Tau
65%	0.03233
70%	0.0839
75%	0.134
80%	0.1845
85%	0.2346
90%	0.2846
95%	0.3347
100%	0.3847

the graph at that point using Eq. 3. To find the derivative of f at the point $x = a$ we use:

$$f'(a) = \lim_{x \to a} \frac{f(x) - f(a)}{x - a} \tag{3}$$

We calculate the derivative of the function of the curve at $f'(0.632)$ which gives the rate of instantaneous change from $\frac{\delta y}{\delta x}$. Using the graph in Fig. 1 as an example, the approximate best secant value of which is 0.2 at τ. The derivatives of the reference graphs were calculated at the same percentages, as shown in Table I.

Next, comparing the derivatives of the reference graph and the example research graph at τ, we estimate the maximum possible accuracy for the graph. The result lies between the 80% and 85% maximum accuracy curves. From Table I, the derivative value at the 80% accuracy curve is 0.1845 and the derivative value at the 85% accuracy curve is 0.2346. The calculated derivative of the example was 0.2, which is closer to the derivative value for the 80% accuracy curve. Therefore, the selected maximum curve is between 80% and 85%. For purposes of calculation we use the fitted floor function of 80% maximum value.

C. Predicting the Curve

After determining the approximate maximum possible accuracy, we use Eq. 2 with the τ value to estimate the number of required training examples to reach that maximum. Selecting the desired final accuracy, matching that value of τ with the desired accuracy, and plugging those values into the equation, results in the estimate for the number of training examples required. Noting that this is just an approximation, the user can multiply the number of examples by an appropriate factor.

Overlaying the study example in Fig. 1 with our predicted curve with maximum accuracy of 80%, the maximum variation from the actual curve is bounded at 10%, as seen in Fig. 6. Generally speaking, this is in excellent agreement for the curve, as the asymptotic accuracy is very close to 80%, as predicted. This was applied to several examples and similar levels of accuracy were achieved, an example of which is shown in Fig. 7.

Fig. 6. Predicted Accuracy vs Ground Truth

Fig. 7. Predicted Accuracy vs Ground Truth

IV. Analysis of Results

Training is a continuous process that is never "completed." In Fig. 2, there are oscillations and noise in the accuracy, which means there could be unsatisfactory training examples. Smoothing the curve shows the actual trend of learning, given the training set in the architecture, the architecture of the NN, and the nature of the problem itself. It is unclear at this time what the contribution of each of these factors is with respect to accuracy. The goal is to verify our model. Our aim is to check our model by stopping the training at a point where the accuracy has reached 63.2%. We find that the accuracy prediction at this point in training can be an early indicator of the maximum possible accuracy.

To increase accuracy, a modified or deeper architecture might be required. More complex architectures, for example, variant models of Generative Adversarial Networks [13] might demonstrate significant accuracy increase. Changing the dataset may also affect the accuracy; however, the measure of change remains an open issue for research. This leads us to the conclusion that better examples should be used early in the training process. Is it possible to decrease the number of training examples based on quality? Can the quality of an example be measured by accuracy if that example is applied as the initial training examples? The exact mechanisms that cause oscillation and noise in accuracy should also be studied. Finally, it is important to know how early in the training process can prediction be made accurately.

V. Conclusion

Based on our model, we can anticipate the highest possible accuracy and the number of training examples needed for a NN to attain a desired level of accuracy. We do this based on the RC Charging Equation, or more precisely, on the value of τ. Researchers will need to train their NNs initially for several training iterations, in order to reach the accuracy value of τ before we can make our estimations. In our experimental work, we use NNs to check our model; our experimental results were encouraging and appear accurate.

References

[1] J. Frankle, D. J. Schwab, and A. S. Morcos, "The Early Phase Of Neural Network Training," *ICLR 2020*, pp. 1–20, 2020.

[2] C. Zhang, B. Recht, S. Bengio, M. Hardt, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2019.

[3] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?," in *Advances in Neural Information Processing Systems*, 2018.

[4] A. Achille, S. Soatto, and M. Rovere, "Critical learning periods in deep networks," in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[5] N. Lord and J. B. Fraleigh, "A First Course in Abstract Algebra," *The Mathematical Gazette*, 1989.

[6] P. J. Hilton and J. L. Kelley, "General Topology," *The Mathematical Gazette*, 1957.

[7] A. H. Carlson, *Set Theoretic Estimation Applied to the Information Content of Ciphers and Decryption*. PhD thesis, University of Idaho, USA, 2012.

[8] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.

[9] H. Zhu, M. Akrout, B. Zheng, A. Pelegris, A. Jayarajan, A. Phanishayee, B. Schroeder, and G. Pekhimenko, "Benchmarking and Analyzing Deep Neural Network Training," *2018 IEEE International Symposium on Workload Characterization, IISWC 2018*, pp. 88–100, 2018.

[10] C. C. Aggarwal, *Neural Networks and Deep Learning*. Cham: Springer International Publishing, 2018.

[11] "Epoch vs iteration when training neural networks - Stack Overflow." https://stackoverflow.com/questions/4752626/epoch-vs-iteration-when-training-neural-networks, June 2020. [Online].

[12] S. Ling, J. Sanny, and W. Moebs, *OpenStax University Physics - Volume 2*. XanEdu Publishing Inc; 1st edition, 2016.

[13] B. Ghosh, I. K. Dutta, M. Totaro, and M. Bayoumi, "A Survey on the Progression and Performance of Generative Adversarial Networks." Submitted for Publication, In Press, 2020.

View publication stats