# BOUNDNESS OF A NEURAL NETWORK WEIGHTS USING THE NOTION OF A LIMIT OF A SEQUENCE

Dr. Hazem Migdady

Department of Mathematics and Computer Science, Tafila Technical University, P.O. Box 179, Tafila 66110

***ABSTRACT***

*feed forward neural network with backpropagation learning algorithm is considered as a black box learning classifier since there is no certain interpretation or anticipation of the behavior of a neural network weights. The weights of a neural network are considered as the learning tool of the classifier, and the learning task is performed by the repetition modification of those weights. This modification is performed using the delta rule which is mainly used in the gradient descent technique. In this article a proof is provided that helps to understand and explain the behavior of the weights in a feed forward neural network with backpropagation learning algorithm. Also, it illustrates why a feed forward neural network is not always guaranteed to converge in a global minimum. Moreover, the proof shows that the weights in the neural network are upper bounded (i.e. they do not approach infinity).*

***KEYWORDS***

*Data Mining, Delta Rule, Machine Learning, Neural Networks, Gradient Descent.*

## 1. INTRODUCTION

Mitchell (1997) argue that "Neural network learning methods provide a robust approach to approximating target functions, and they are among the most effective learning methods currently known". Moreover, the author believes that "backpropagation learning algorithm has proven surprisingly successful in many practical problems". LeCun, et al. (1989), Cottrell (1990) and Lang, et al. (1990) provided experimental results support the efficient characteristic of backpropagation learning algorithm with neural networks.

A feed forward neural network is considered as a black box since there is no certain interpretation or anticipation of its weights behavior. The weights of a neural network are considered as the learning tool. During the training process of a neural network, the weights are repeatedly modified, since the main characteristic of a neural network is: "those connections between neurons leading to the right answer are strengthened by maximizing their corresponding weights, while those leading to the wrong answer are weaken" (Negnevitsky, 2005).

DOI : 10.5121/ijdkp.2014.4301 

"Each input node is connected with a real-valued constant (i.e. weight) that determines the contribution of that input to the output. Learning a neural network involves choosing values for the weights. Hence, the learning problem faced by Backpropagation is to search a large hypotheses space defined by all possible weight values for all the units in the network" (Mitchell, 1997).

The gradient descent technique is among the common techniques that are used to perform the search process that was mentioned by Mitchell (1997) by optimizing the weights of a neural classifier, which is achieved by applying the delta rule that is used to find out the amount by which the current value of a weight will be updated.

Mathew (2013) believes that "The most popular neural network algorithm is backpr opagation, a kind of gradient descent method. Backpropagation iteratively process the data set, comparing the network's prediction for each tuple with actual known target value to find out an acceptable local minimum in the NN weight space in turns achieves the least number of errors". Moreover, in a related context, Kumar, et al. (2012) mentioned that backpropagation algorithm "learns by recursively processing a set of training tuples, comparing the network's observed output for each tuple with the actual known class attribute value. For each training tuple, the weights are edited so as to reduce the mean squared error between the network's output and the actual class label or value. These changes are propagated in backward direction through each hidden layer down to the first hidden layer. After running the process repetitively, the weights will finally converge, and the training process stops".

Even though a neural classifier is considered as a robust learning machine, it is not guaranteed to converge to a global minimum; instead it is possible to stuck in a local minimum. In this notion a proof is provided that helps to understand the characteristics and the nature of a neural network weights, which in turn can be used to interpret the whole behavior of a neural classifier.

The remainder of this paper is organized as the following: the next section mentions the concept of the gradient descent technique. This section contains two subsections show that the delta rules for the input and hidden weights approaches zero. The paper ends with the conclusion.

## 2. GRADIENT DESCENT TECHNIQUE AND THE DELTA RULE

In this section the gradient descent technique and the delta rule will be introduced. We will show that the delta rule approaches zero. Hence, taking into consideration that the delta rule is used to update the values of the weights in a neural classifier, this implies that the weights are upper bounded and they do not approach infinity.

The learning task in a feed forward neural network with backpropagation training algorithm is achieved by the optimizing of the neural network weights in order to fit the data points in a dataset. This can be achieved by applying the gradient descent technique which mainly based on choosing the weights that minimize the error (i.e. the difference between the actual output and the target). Russell and Norvig (2010) provided some details about the error function and called it as: *Loss* function.

According to Negnevitsky (2005) Equation (1) below shows the major equation to update a neural network weights.

$$w_{i+1} = w_i + \Delta w_i \qquad (1)$$

where $w_i$ is the weight at the current training iteration $i$, and $w_{i+1}$ is the weight at next training iteration $i+1$. $\Delta w_i$ is the amount of change in $w_i$. $\Delta w_i$ is known as the delta rule which is used to update the weights in the neural network. The delta rule is not identical for all weights in the neural network, since there are two kinds of weights: (1) Input Weights (connect the input layer to the output layer) and (2) Hidden Weights (connect the hidden layer to the output layer).

**(1) DELTA RULE FOR HIDDEN WEIGHTS**

Equation (2) below illustrates the delta rule $\Delta w_j$ that is used to update the hidden weights in a feed forward neural network.

$$\Delta w_j = \alpha (y_D - y_A)\, y_A (1 - y_A)\, y_{jw_j} \qquad (2)$$

where $y_D$ and $y_A$ are the target and the actual outputs on the output layer respectively, where $y_D \in \{0,1\}$. $y_j$ is the output of the hidden neuron $j$. $w_j$ is the hidden weight that connects the hidden neuron $j$ to the output layer in the neural network. $\alpha$ is a small positive value that is known as the learning rate. $y_A$ is calculated using the logistic regression function as equation (3) illustrates.

$$y_A = \frac{1}{1 + e^{-z}} \qquad (3)$$

where $z$ is a multiple regression function, $y_1 \ldots y_n$ are the outputs of the hidden neurons, and $w_1 \ldots w_n$ are the weights that connect the hidden layer to the output layer. The major aim here is to prove that the delta rule in (2) approaches zero, since this implies that the weights will be stable in some level, which means that the weights are bounded and they do not approach infinity. Such feature causes the neural classifier to stuck in a local minimum or a global minima.

As mentioned before, the desired output takes two values only: $y_D \in \{0,1\}$.

(1) when $y_D = 1$, there are three cases: $y_j = 0$, $y_j = 1$, and $y_j \in (0,1)$.
Now let $wj_i$ be the weight of the hidden output $y_j$ at iteration $i$. Thus:

$$\begin{aligned} \Delta wj_i &= \alpha\, y_j\, (1 - y_A)\, y_A (1 - y_A) \qquad (4) \\ &= \alpha\, y_j\, (1 - y_A)^2\, y_A \\ &= \alpha\, y_j\, y_A\, (y_A - 1)^2 \\ &= C_1\, y_A\, (y_A - 1)^2 \end{aligned}$$

where $C_1 = \alpha\, y_j$

$$= C_1 \left(\frac{1}{1 + e^{-z}}\right)\left(\frac{1}{1 + e^{-z}} - 1\right)^2$$

Case (I): when $y_j \in (0,1)$ => $C_1 = \alpha\, y_j$ is positive (note that $\alpha$ is a small positive value such that $\alpha = 0.1, 0.01, 0.001$)

- if $w_j \rightarrow +\infty$ then $z = y_j w_j + z^{\sim} \rightarrow +\infty$
  where $z^{\sim} = w_0 + w_1 y_1 + \cdots + w_{j-1} y_{j-1} + w_{j+1} y_{j+1} + \cdots + w_n y_n$
  $\Rightarrow e^{-z} \rightarrow 0$
  $\Rightarrow \frac{1}{1+e^{-z}} \rightarrow 1$
  $\Rightarrow (\frac{1}{1+e^{-z}} - 1) \rightarrow 0$
  $\Rightarrow \Delta wj_i = C_1 \left(\frac{1}{1+e^{-z}}\right)\left(\frac{1}{1+e^{-z}} - 1\right)^2 \rightarrow 0$
  $\Rightarrow \Delta wj_i \rightarrow 0$

Thus, as $wj_i$ increases ($wj_i \rightarrow +\infty$), then the sequence $\Delta wj_i$ decreases and approaches to 0 ($\Delta wj_i \rightarrow 0$). This result can be used with the notion of limit of a sequence, which is considered as the most basic concept among different concepts of limit in real analysis see Bartle and Sherbert (2000).

According to Bartle and Sherbert (2000), the notion of a limit of a sequence implies that: when a sequence converges to 0, then for each $\varepsilon > 0$ $\exists\, n^* \in \mathbb{N}$ such that if $i \geq n^*$ then $|\Delta wj_i - 0| < \varepsilon$. That means for any $\varepsilon > 0$ there is some number $n^*$ such that all the terms of the sequence $\Delta wj_i$ that are produced after the term $\Delta wj_{n^*}$ (*i.e.* $\Delta wj_{n^*+1}, \Delta wj_{n^*+2}, \ldots$) will be less than $\varepsilon$. Hence, if $\varepsilon = 1 \times 10^{-10}$ for instance, then there is $n^* \in \mathbb{N}$ such that all the terms $\Delta wj_{n^*+1}, \Delta wj_{n^*+2}, \ldots$ are less than $\varepsilon$. In other words $\Delta wj$ will not exceed $1 \times 10^{-10}$ for any term that is produced after the term $n^*$.

- if $wj \rightarrow -\infty$ then $z = y_j w_j + z^{\sim} \rightarrow -\infty$
  where $z^{\sim} = w_0 + w_1 y_1 + \cdots + w_{j-1} y_{j-1} + w_{j+1} y_{j+1} + \cdots + w_n y_n$
  $\Rightarrow e^{-z} \rightarrow +\infty$
  $\Rightarrow \frac{1}{1+e^{-z}} \rightarrow 0$
  $\Rightarrow \left(\frac{1}{1+e^{-z}} - 1\right) \rightarrow -1$
  $\Rightarrow \Delta wj_i = C_1 \left(\frac{1}{1+e^{-z}}\right)\left(\frac{1}{1+e^{-z}} - 1\right)^2 \rightarrow 0$
  $\Rightarrow \Delta wj_i \rightarrow 0$

So as $wj_i$ decreases ($wj_i \rightarrow -\infty$), then the sequence $\Delta wj_i$ decreases and approaches to 0 ($\Delta wj_i \rightarrow 0$). This result also satisfies the notion of limit of a sequence which was mentioned earlier when $w_j \rightarrow +\infty$, and it proves that $\Delta wj$ is a sequence approaches 0.

Case (II): when $y_j = 1 \Rightarrow C_1 = \alpha\, y_j$: $\alpha\, y_j \in (0,1)$ (note that $\alpha$ is a small positive value such that $\alpha = 0.1, 0.01, 0.001$)

- if $w_j \rightarrow +\infty$ then $z = y_j w_j + z^{\sim} \rightarrow +\infty$
  $\Rightarrow e^{-z} \rightarrow 0$
  where $z^{\sim} = w_0 + w_1 y_1 + \cdots + w_{j-1} y_{j-1} + w_{j+1} y_{j+1} + \cdots + w_n y_n$
  $\Rightarrow \frac{1}{1+e^{-z}} \rightarrow 1$
  $\Rightarrow \left(\frac{1}{1+e^{-z}} - 1\right) \rightarrow 0$
  $\Rightarrow \Delta wj_i = C_1 \left(\frac{1}{1+e^{-z}}\right)\left(\frac{1}{1+e^{-z}} - 1\right)^2 \rightarrow 0$

$$\Rightarrow \quad \Delta wj_i \to 0$$

So as $wj_i$ increases ($wj_i \to +\infty$), then the sequence $\Delta wj_i$ decreases and approaches to 0 ($\Delta wj_i \to 0$). This result also satisfies the notion of limit of a sequence which was mentioned earlier.

- if $w_j \to -\infty$ then $z = y_j w_j + z^{\sim} \to -\infty$

$$\Rightarrow \quad e^{-z} \to +\infty$$
$$\Rightarrow \quad \frac{1}{1+e^{-z}} \to 0$$
$$\Rightarrow \quad \left(\frac{1}{1+e^{-z}} - 1\right) \to -1$$
$$\Rightarrow \quad \Delta\, wj_i = C_1 \left(\frac{1}{1+ e^{-z}}\right)\left(\frac{1}{1+ e^{-z}} - 1\right)^2 \to 0$$
$$\Rightarrow \quad \Delta wj_i \to 0$$

So as $wj_i$ decreases ($wj_i \to -\infty$), then the sequence $\Delta wj_i$ decreases and approaches to 0 ($\Delta wj_i \to 0$). This result also satisfies the notion of limit of a sequence which was mentioned earlier.

Case (III): when $y_j = 0$ this implies that the delta rule is also 0. Recall the delta rule from equation (2):

$$\Delta wj_i = \alpha\, y_j\, (1 - y_A)\, y_A (1 - y_A)$$
$$\Rightarrow \quad \Delta wj_i = 0 \text{ since } y_j = 0$$

And this also satisfies the notion of a limit of a sequence.

(2) when $y_D = 0$, there are also three cases: $y_j = 0$, $y_j = 1$, and $y_j \in (0,1)$.

Now let $wj_i$ be the weight of the hidden output $y_j$ at iteration $i$. Thus:

$$\Delta wj_i = \alpha\, y_j\, (0 - y_A)\, y_A (1 - y_A)$$
$$= \alpha\, y_j\, (y_A^2)(y_A - 1)$$
$$= C_1\, y_A^2\, (y_A - 1)$$

where $C_1 = \alpha\, y_j$

$$= C_1 \left(\frac{1}{1 + e^{-z}}\right)^2 \left(\frac{1}{1 + e^{-z}} - 1\right)$$

Hence, by using the same argument above when $y_D = 1$ we will end up that: when $y_j \in (0,1)$ or $y_j \in \{0,1\}$ then $\Delta wj_i \to 0$ when $w_j \to +\infty$ or $w_j \to -\infty$.

Since $\Delta wj_i \to 0$ for large $i$, then:

$$\Rightarrow \quad (wj_{i+1} = wj_i + \Delta wj_i) \to 0$$
$$\Rightarrow \quad (wj_{i+1} - wj_i = \Delta wj_i) \to 0$$
$$\Rightarrow \quad (wj_{i+1} - wj_i) \to 0$$
$$\Rightarrow \quad (wj_{i+1} \to wj_i) \text{ for large } i.$$

**(2) DELTA RULE FOR INPUT WEIGHTS**

Equation (5) below illustrates the delta rule $\Delta w_k$ that is used to update the input weights.

$$\Delta w_k = \alpha \cdot x_k \left(y_j(1-y_j)\right)\left(\sum_{j=1}^{l}(\Delta w_j)(w_j)\right) \qquad (5)$$

where $x_k$ an input variable, $y_j$ the output of the hidden neuron $j$, and $l$ is the number of the hidden neurons in the neural classifier.

Since the delta rule $\Delta w_k$ in equation (5) depends on the amount $\Delta w_j$ which approaches zero for large $i$ (as proved earlier), then the delta rule $\Delta w_k$ also approaches zero for large $i$. Moreover, the hidden output $y_j$ affects the value of the delta rule $\Delta w_k$, but $y_j$ is a hidden output of the logistic regression function at hidden neuron $j$, thus the values of $y_j$ are: $y_j \in \{0,1\}$, or $y_j \in (0,1)$. Thus, by applying the former argument with the delta rule for hidden weights, it is easy to show that when $y_j \in \{0,1\}$ then $\Delta w_k \to 0$.

By now we need to show that $\Delta w_k$ approaches 0 when $y_j \in (0,1)$:

let $wk_i$ be the weight of the input $x_k$ at iteration $i$. Thus:

$$\Rightarrow \Delta wk_i = C_1\, y_j(1-y_j)$$

where $C_1 = \alpha\, x_k\left(\sum_{j=1}^{l}(\Delta w_j)(w_j)\right)$

$$\Delta wk_i = C_1 \left(\frac{1}{1+e^{-z}}\right)\left(1-\frac{1}{1+e^{-z}}\right)$$

- If $w_k \to +\infty$ then $z = x_k w_k + z^{\sim} \to +\infty$
  where $z^{\sim} = w_0 + w_1x_1 + \cdots + w_{k-1}x_{k-1} + w_{k+1}x_{k+1} + \cdots + w_nx_n$
  $\Rightarrow e^{-z} \to 0$
  $\Rightarrow \frac{1}{1+e^{-z}} \to 1$
  $\Rightarrow (1-\frac{1}{1+e^{-z}}) \to 0$
  $\Rightarrow \Delta\, wk_i = C_1\left(\frac{1}{1+e^{-z}}\right)(1-\frac{1}{1+e^{-z}}) \to 0$
  $\Rightarrow \Delta wk_i \to 0$

Thus as $wk_i$ increases ($wk_i \to +\infty$), then the sequence $\Delta wk_i$ decreases and approaches to 0 ($\Delta wk_i \to 0$). This result satisfies the notion of limit of a sequence that was mentioned earlier.

- If $w_k \to -\infty$ then $z = x_k w_k + z^{\sim} \to -\infty$
  where $z^{\sim} = w_0 + w_1x_1 + \cdots + w_{k-1}x_{k-1} + w_{k+1}x_{k+1} + \cdots + w_nx_n$
  $\Rightarrow e^{-z} \to +\infty$
  $\Rightarrow \frac{1}{1+e^{-z}} \to 0$
  $\Rightarrow \left(1-\frac{1}{1+e^{-z}}\right) \to 1$
  $\Rightarrow \Delta\, wk_i = C_1\left(\frac{1}{1+e^{-z}}\right)\left(1-\frac{1}{1+e^{-z}}\right) \to 0$
  $\Delta wk_i \to 0$

So as $wk_i$ decreases ($wk_i \to -\infty$), then the sequence $\Delta wk_i$ decreases and approaches to 0 ($\Delta wk_i \to 0$). This result also satisfies the notion of limit of a sequence which was mentioned in the previous case when $w_k \to +\infty$.

Thus, we proved that $\Delta wk$ is a sequence approaches to 0 for large $i$, since it satisfies the notion of limit of a sequence.

Since $\Delta wk_i \to 0$ for large $i$, then:

$$\Rightarrow (wk_{i+1} = wk_i + \Delta wk_i) \to 0$$
$$\Rightarrow (wk_{i+1} - wk_i = \Delta wk_i) \to 0$$
$$\Rightarrow (wk_{i+1} - wk_i) \to 0$$
$$\Rightarrow (wk_{i+1} \to wk_i) \text{ for large } i.$$

Therefore, both sequences in (2) and (5) approach zero, which implies that the weights in the neural network are bounded.

## 3. CONCLUSION

In this notion a proof was introduced that identifies the upper bounds of a neural network weights. This was achieved by applying the notion of a limit of a sequence on the delta rule which is part of the gradient descent technique. The result showed that the weights in a neural classifier are upper bounded (i.e. they do not approach infinity) since the amount of the delta rule is decreased though training epochs and it approaches zero. This, in turn, minimizes the change of the weights amounts which satisfies the notion of a limit of a sequence. Such result helps to understand the behavior of a neural network classifier and explains why a neural network is not always guaranteed to find the global minimum in the solution surface, and stuck in a local minimum.

## REFERENCES

[1] Bartle, R. G., & Sherbert, D. R. (2000). Introduction to Real Analysis. 3rd ed., Wiley. ISBN: 0471321486.

[2] Cottrell, G. W. (1990). Extracting features from faces using compression networks: Face, identity, trol, signals, and systems, 2, 303-314.

[3] Kumar, P., Sehgal, K., V. & N., Chauhan, S., D. (2012). A Benchmark to Select Data Mining Based Classification Algorithms for Business Intelligence and Decision Support Systems. International Journal of Data Mining & Knowledge Management Process (IJDKP), Vol.2, No.5.

[4] Lang, K. J., Waibel, A. H., & Hinton, G. E. (1990). A time-delay neural network architecture for isolated word recognition. Neural Networks, 3, 33-43.

[5] LeCun, Y., Boser, B., Denker, J. S., & Solla, S. A. (1990). Optimal Brain Damage. In D. Touretzky (Ed.), Advances in Neural Information Processing Systems (Vol. 2, pp. 598 - 605).

[6] Mathew, L., S. (2013). Integrated Associative Classification and Neural Network Model Enhanced By Using A statistical Approach. International Journal of Data Mining & Knowledge Management Process (IJDKP), Vol.3, No.4.

[7] Mitchell, T. (1997). Machine learning. Singapore: McGRAW-HILL.

[8] Negnevitsky, M. (2005). Artificial intelligence: a guide to intelligent systems (2nd ed.). Britain: Addison-Wesley.

[9] Russell, S., & Norvig, P. (2010). Artificial intelligence: a modern approach (3rd ed.). USA: Prentice Hall.

## AUTHORS

Dr. HAZEM MIGDADY

• A PhD in data mining and machine learning, with an emphasis on inductive learning from large datasets and patterns.

• Lecturer in the Department of Mathematics and Computer Science.