IT in Industry, Vol. 9, No.1, 2021 Published Online 19-03-2021

UNDERSTANDING AND STUDY OF WEIGHT INITIALIZATION IN ARTIFICAL NEURAL NETWORKS WITH BACK PROPAGATION ALGORITHM

Farhana Kausar[1]. Dr. Aishwarya P.[2] and Dr. Gopal Krishna Shyam[3]
[1]*Research Scholar, Dept. of CSE, Atria IT, VTU, Belgavi, India*
[2]*Research Supervisor, Dept. of CSE, Atria IT, VTU, Belgavi, India*
[3]*Dept of Computer Science, REVA University, Bangalore, India*

Abstract: There are various important choices that need to be assumed when building and training a neural network. One has to determine which loss function to be used, how many layers to be include, what stride and kernel size to use for each layer, which optimization algorithm is best suited for the network and so on. Assuming all the above condition, it decided to initialize the neural network training by different weight initialization techniques. This process is carried out in affiliation or with respect to with random learning rate so that we can get better result. We have calculated the mean test error for newly proposed paradigm and traditional approach. The newly proposed paradigm Xavier Weight Initialization less error in comparison to the traditional approach of Uniform and Gaussian Weight initialization (Random Initialization).

Keywords:
Neural Network, Back Propagation, Weight Initialization, Sigmoid, ReLU.

1. Introduction

Artificial Neural Network is as comparable to the human brain for problem solving skills. It uses Neural Network to decide the best prediction of the system. Neural Network are known for their accuracy and their performance. ANN is inspired by the information and parallel processing model an structure of the brain. The human brain is built up of numbers of neurons that linked with each other in complex pattern. Each neuron receives information from other neurons at the synaptic junction and process the information and transfer it to respective neurons. As in the brain structure, the ANN is often composed of a large number of densely interconnected that can be used for processing, for many kinds of applications like pattern recognition and prediction etc. ANN are characterized by their interconnected network, their node properties and their learning behavior. With the help of ANN, the input is fed into the input nodes and then it is preprocessed in the hidden layers using various function and algorithms and then it is feed to the output layer.

The proposed system aims to initialize the neural networks different weight initialization techniques so that the computational power will be increased and efficiency of program too. We used Backpropagation Algorithm for optimizing the error respective to the weight. Mathematically the neural network is trained using chain rule, where the partial derivates will give an idea to understand the smallest or delta change in the weight. Backpropagation is derived or assessed from Backward propagation of errors. The "backwards" portions of the name backpropagation propose figuring of the gradient continues in reverse through the network organization, with the gradient of the last layer of weights being determined first and the gradient angle of the first layer of weights being determined last.

Section 2 describes Literature survey on weight initialization techniques. Section 3 focuses on the problem statement of how the different weight initialization techniques is mentioned. Section 4 deals with the generalized architecture of artificial neural network with activation functions of Sigmoid and ReLU (Rectified Linear Unit). Section 5 discusses on the experiment performed, results obtained and finally concluded the work.

2. Literature Review

This section reviews related studies on different weight initialization techniques. The training algorithm for backpropagation is considered a method for minimizing the error for weights. During the training phase in the neural network, we apply the backpropagation algorithm to train the neural network until it reaches local minima, where the analysis indicates that the error rate has reached to the minimum error surface.
Sarfaraz Masood [1] Together with the t-Test data suggesting their implied view of providing zero initialization weights in compliance with arbitrary learning rate distresses, it provides a result for the ways to improve the efficiency that can serve as an approach to the conventional irregular weight initialization technique. Bhatia [12] has suggested that weight initialization, which affects the training speed of the network, is one of the most critical factors during training. For each initialization method that is not normal in the weight initialization methods bibliography, the speed, generalization and probability of convergence. In this paper, they have to various parameter using experimental procedure which is time consuming [2].

In [3] proposed the system in which the training strategy depends upon the pre-training of the initial weight for neural network using delta rule, they may use random value for processing. After pre-training, normal BP training is carried out for the completion of network training. For [11] The initial error is significantly smaller and the number of iterations necessary to sustain the error criterion

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

is reduced significantly due to the defined optimum initial weight.. In [4] proposed a system in which optimum range for R was shown to exist in order to reduce the time needed to reach the least of the cost function. Normalization factors were finely defined, that leads to a distribution of the activations independent of the neurons, and to a single non-dimensional amount, the value of which can be quickly found by computer simulation. Mentioned two formula is used to calculate the values of weight

A technique known as statistically regulated activation weight initialization (SCAWI) to and optimum initial weights [13]. The aim of the SCAWI technique is to prevent neuron saturation at an early stage of training. With the desired initial weights assessed, the initial error is significantly lower and the number of iterations necessary to attain the error criterion is significantly reduced. [14]. In [15], A proposal for a simple initialization mechanism that leads to the state of the art in connection with standard stochastic gradient descent (SGD), which highlights the importance of initialization. In [16], The following strengths are present in the weight initialization method 1. Using near optimal initialization of the link weights, we can increase the classification efficiency of the Neural Network; 2. The convergence on well initialized networks of gradient-based methods is typically much faster than on randomly initialized ones. The work [17] It shows that these fundamental properties are fulfilled by the architecture of neural networks. We formally show that these random Gaussian weight networks conduct a distance-conserving data embedding, with a special treatment for in-class and out-of-class data. At the input of the network, different points will possibly have a similar output.. In [18], the author specifies that the initial weight option for feedforward neural networks is an integral part of the training process. It also deals with a specific feedforward neural network topology where the symmetric linear saturated activation function used in the hidden layer is used. The initial error is significantly smaller and due to the optimum initial weight measured; the number of iterations needed to meet the error criterion is significantly reduced. Training such topology is a difficult task, since it is not completely distinguishable between the activation functions. Therefore, a proper initialization method for that situation may be even more important than grappling with the role of sigmoid-activated neural networks. Therefore, multiple initialization possibilities can be checked. As a consequence, specific initialization methods are recommended for implementation, depending on the type of task to be solved. The [19], specifies that the most basic thing to point before a neural network can be trained, apart from the learning algorithm, is where to explore the possibility of iterative learning. This has led to a study of the best possible strategies for weight initialization and their effect on the pace of learning convergence [Nguyen, Widrow 89]. In this paper, we show that there is a correlation between the width of the probability distribution for the weights and the optimum size of the neural network learning stage. A criterion is given that requires the most suitable range of weights to be measured in order to start the procedures of learning. It is also shown that the different weighing layers require different interval sizes. In this work [20], due to the network's approximation properties, the demonstration of the intervals of random weights and biases in Feedforward Neural Network with random Hidden Nodes (FNNRHN) is extremely significant. The basic functions of the surface fitting data generated by a linear combination are the activation functions of the hidden neurons. In order to model the target function with the required precision in its nonlinear regions, the collection of Activation Functions (AFs) for the nonlinear target function should provide nonlinear fragments. The main contribution of this research is to suggest a practical method for generating weights and biases randomly in FNNRHN for the establishment of nonlinear fragments of Activation Functions (AFs) in the input space field containing the data point. The analyses conducted lead to the conclusion that hidden node parameters depend on the input data range and the form of activation function. Ranges for weights and biases, as these parameters have distinct implications, should be treated separately. In addition, the range for the i-th hidden node bias is strictly dependent on the weight of this node. The method proposed helps one to control the flatness and steepness of the AF set and hence the degree of network generalization. Our analysis [22] is Tracking activations (watching for saturation of secret units) and gradients, through layers and through training iterations guided by investigative experiments. The effects of activation function choices (with the assumption that it could affect saturation) and initialization procedure on these are also evaluated (since unsupervised pretraining is a particular form of initialization and it has a drastic impact).

3. Problem Statement

The aim of weight initialization is to prevent layer activation outputs from exploding or disappearing during a forward pass through a neural network. If either happens, the loss gradients will either be too high or too small to flow backwards favorably, and if it is even able to do so at all, the network will take longer to converge. All the weight initialization methods use various technique to initialize the weight of the input network by using various kind of random weight initialization or partitioning technique. In this proposed system, we try to train the model by setting the initial weight to different techniques of the whole network as fixed value. Hence the new mechanism is developed for the initialization of weight using random learning rate and so on. The backpropagation algorithm was used to optimized the error respective to weight. A study of Sigmoid and ReLU activation function which is responsible for the transformation of summed weighted input to the activation of node or the output node and understand the impact of weights an activation on the problem been learned by the Neural Network.

3.1 General Architecture

The Artificial Neural Networks consists of interconnected nodes, which are inspired from human brain. These models are used to study the animals, which help in solving real time problems. The Artificial Neural Network Models are also called as Parallel distributed models or Connectionist Models. An ANN mimics a biological neural network in the human brain. The biological brain is the mechanism by which the nervous system of a living organism operates, allowing the instinctive performance of complex tasks. This nervous system's central processing unit is known as a 'neuron.' The human brain has approximately 10 to 100

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

billi
on neurons, each linked by "synapses" to several others. There are about 100 trillion synapses in the human brain. There are about 100 trillion synapses in the human brain. The human body and its thought processes are regulated by these connections. In short, ANNs aim to replicate the learning processes of the human brain.

It consists of three layers, the input layer, the hidden layer and the output layer as in fig 3.1. The input layer is the passive layer, which is responsible to take input and give to the next layer. The hidden layer is the middle layer between the input and the output layer. It is responsible for weight initialization of the neural network and performs the nonlinear transformation with the activation functions. The hidden layer is responsible for all the feature selection from the input layer. The output layer produces the final results after performing calculations from the previous layers

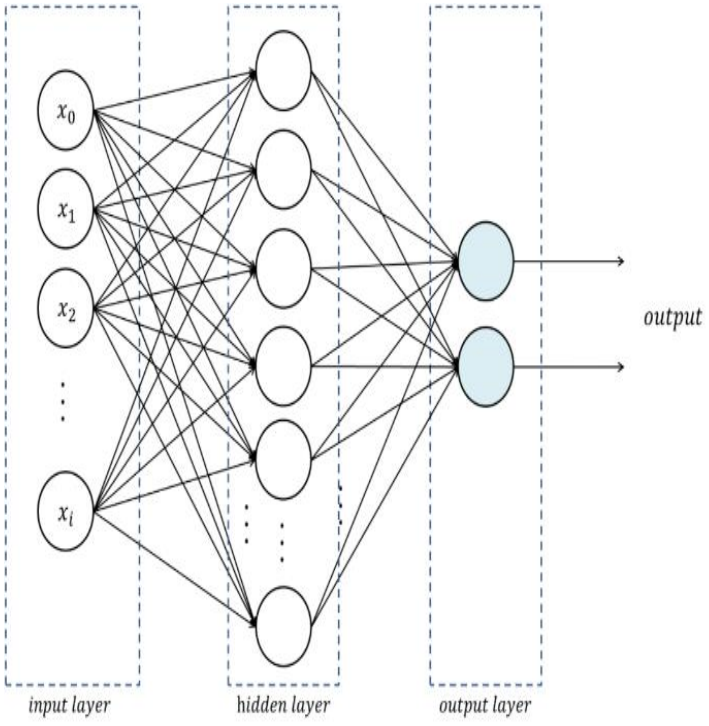

Fig: 3.1 Structure of the Artificial Neural Networks

3.2 Weight Initialization Techniques

Weight initialization is a significant step that has a strong effect on the initialization of weights. A network's training time and the consistency of the resulting model. In fact, improper initialization of weight may prevent the convergence of gradient descent or lead to a poorly performing model. It is crucial to consider this when working with a neural network architecture, as bad outcomes can be wrongly attributed to other aspects of the task modeling process (such as the architecture itself or the quality of the data set). Training time is also impacted by inadequate initialization of weight. The amount of time and computing power required is considered to be one of the drawbacks of learning in artificial neural network to build a model. It is a good idea to start with weight initialization with hyperparameter tuning to create better and faster performing models.

Based on understanding and enhancing initialization of weight across common architectures of networks. It is anticipated that this work will allows neural networks to train more rapidly, which would lead to more experimentation and field developments. By choosing weight values where the initial error is minimal, the initialization schemes provided in this work are expected to help devices with limited resources run typically costly training procedures and allow selection of weights where the error rate is low. Since the process of weight initialization typically has a high impact on a Neural Networks classification results, such a process is of great interest to applied to research. The process of weight initialization typically has a greater effect on the efficiency of the classification than the learning algorithm used with regard to the network architecture and the sample of training.

3.2.1 Zero and One Weight Initialization

If all weights are initialized with 0 or 1, the derivative is the same with respect to the loss feature for every weight w, so all weights in subsequent iterations have the same value. This makes hidden units symmetric for all the n iterations and continues, i.e. setting weights to 0 or 1 does not make it better than a linear model. A significant point to understand and state in mind is that prejudices have no effect whatsoever when initialized with 0.

3.2.2 Random Normal Weight Initialization

The Random Normal weight initialization technique is also known as Gaussian distribution. It is a bell shape curve. The measurement in normal distribution has equal number of measurements below an above the mean value. The general formula for the probability density function of the normal distribution is,

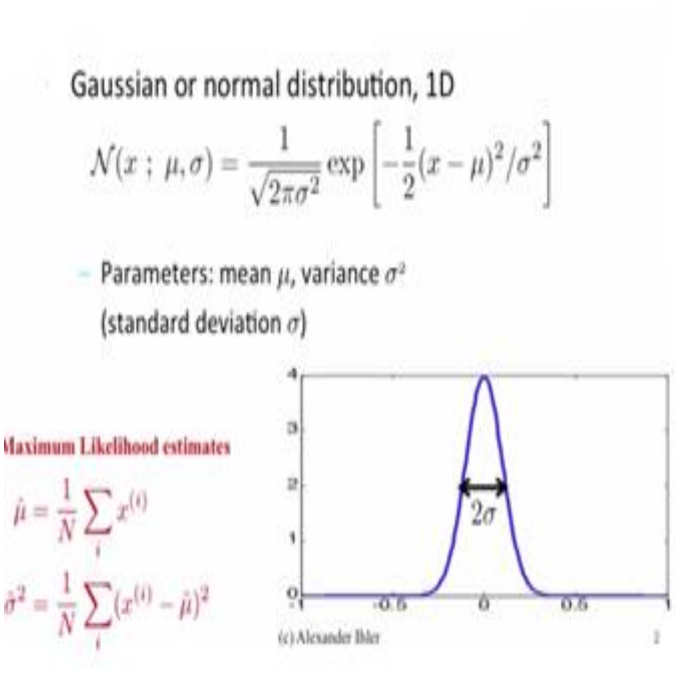

Fig 3.2 Random Normal distribution

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

The Gaussian distribution provides more flexibility over other weight initialization methods. This distribution results into local minima and predicts the quality of the results found.

3.2.3 Random Uniform Weight Initialization

A Uniform distribution, has a constant probability. It is also known as Rectangular Distribution. An alternative way to initialize the weights uniformly from the uniform distribution is the Uniform distribution. Each number has an equal probability of being selected in the uniform distribution. Choosing high values of weights is not the best for the model as it brings problems of bursting and vanishing gradients. Small random numbers, which are similar to 0, are the general way to initialize weights. Starting your weights in the range is always best of [−y,y] where y=1/√$\boxed{n,}$ where n is the number of inputs to a given input layer in neural network.

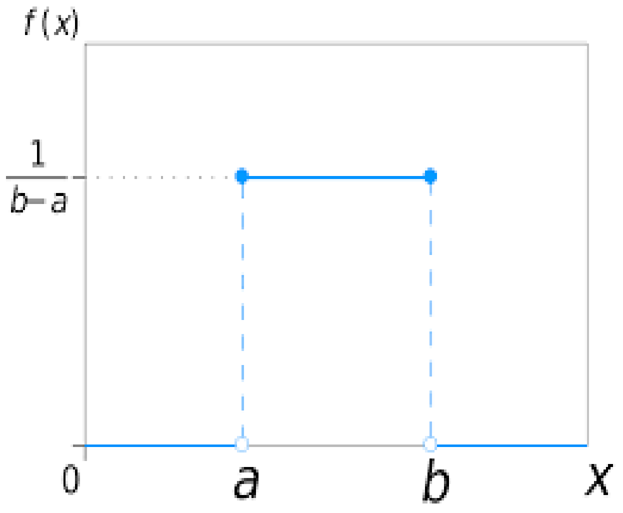

Fig 3.3 Random Uniform distribution

The random initialization, that generates weights are with same probability an is equivalent of random weight generation of Uniform distribution.

3.2.4 Glorot Normal and Uniform Weight Initialization

Xavier Initialization is a heuristic Gaussian initialization that keeps the variance of the input to a layer the same as that of the layer's output. This means that the variation across the network stays the same. It is used to Initialize the network weights so that the functions of neuron activation do not begin in saturated or dead regions. In other words, with random values that are not too small and not too high, we want to initialize the weights. The value depends on the type of the nonlinearity use in the layer.

$$\pm\frac{\sqrt{6}}{\sqrt{n_i + n_{i+1}}}$$

n_i is number of input layers and n_{i+1} is the number of hidden layers.

Xavier Initialization's purpose is to initialize the weights such that the variance of the activations across each layer is the same. This continuous variance prevents the gradient from exploding or disappearing. Let’s take the initialization values, with simplified assumptions,

- Weights and inputs at zero are focused
- The weights and inputs are unique and distributed identically.
- Initialization of biases as zeros

We use the tanh () activation function, which is approximately linear with small inputs: Var(a[l])≈Var(z[l])

With the following initialization rule, which can be applied to all the weights [21]

$$W_{i,j}^{[l]} = N\left(0, \frac{1}{n^{[l-1]}}\right)$$

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

$Var(a_i^{[\ell-1]})$:

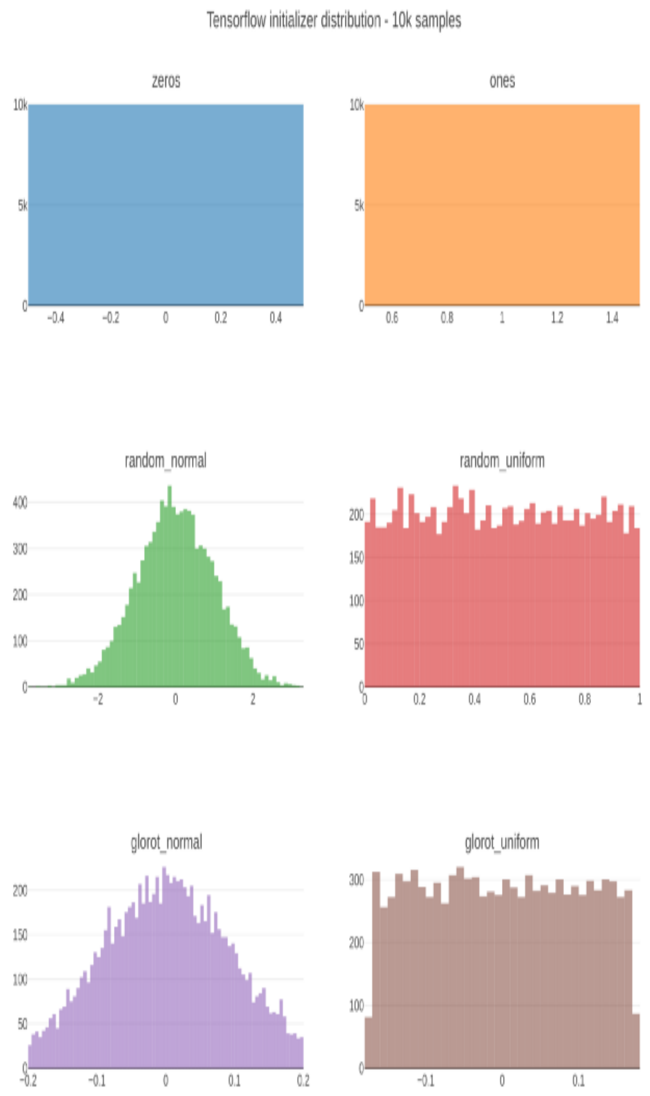

Fig 4.1 Graph for various weight initialization techniques

In brief, if the weights are too small in a network, the signal shrinks as it moves through each layer until it is too small to be useful and the weights in a network start too high, then as it moves through each layer, the signal grows until it is too large to be helpful.

4. Results

The results shown are an understanding of the different types of weight initialization techniques for neural networks in fig 4.1. The loss function is use to quantify whether the neural network model has been trained or not. The fig 4.3 shows the loss function for various weight initialization techniques. The loss function takes the predicted score from our model along with true targets or labels, compares it and gives us some quantitative value of how good or poor those predictions are for the training data.

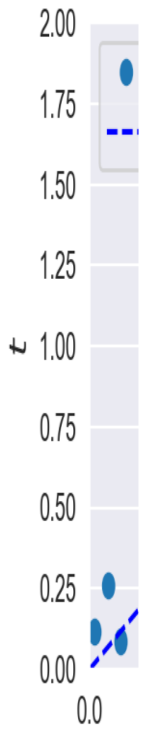

Fig 4.2 Sample Graph for input v/s target function

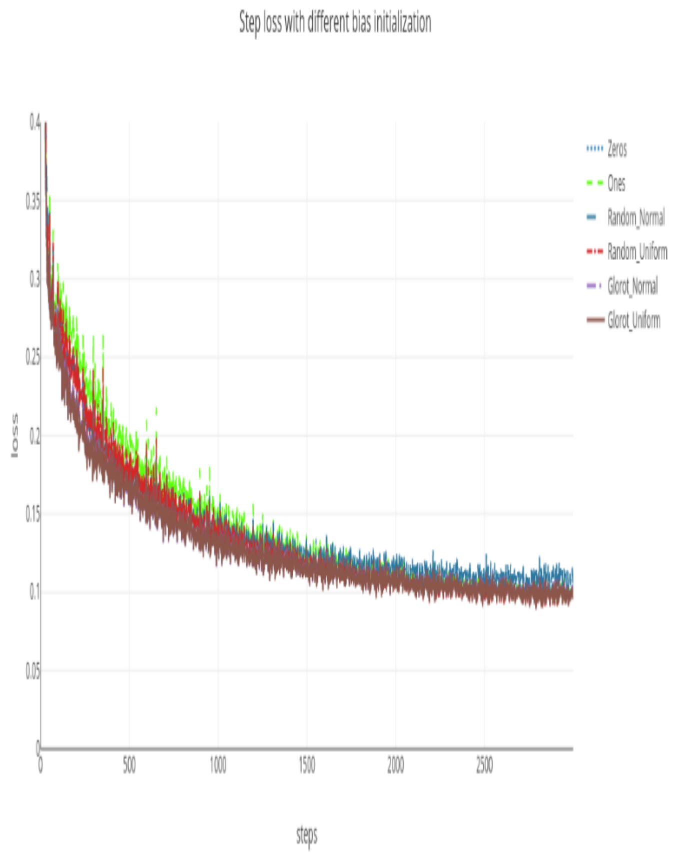

Fig 4.4 Step Loss with different bias initialization

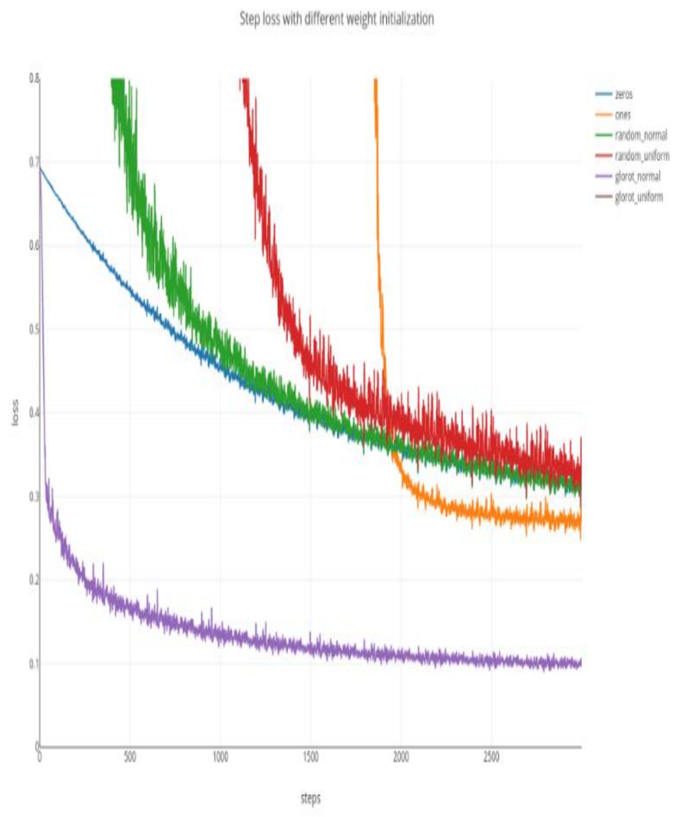

Fig 4.3 Step Loss with different weight initialization

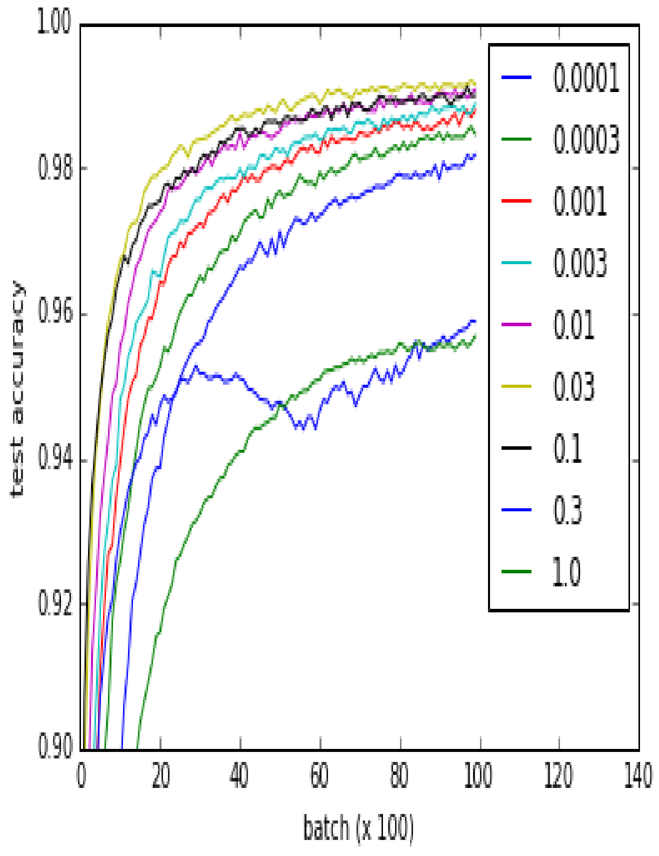

Fig 4.5 Test Accuracy for different Learning rate

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

5. Conclusion

It had been observed that building of a neural network is an important task followed by the weight initialization. The selection of the initial weight is considered an important parameter in the convergence of the error in the Neural Network. Its output depends on the initial weight initialization we define. In this paper was to study and understand the impact of weights initialization techniques along with the Back-Propagation algorithm. The results show the results on the different initialization techniques. For Zero and one's weight initialization the results loss function appears to be flat as there is no change in error gradient. For Random Uniform & Normal was considered under study much better results compared to the zero/ one's initialization. Our study shows that the Glorot Uniform / Normal is the one of the best weight initialization techniques as it assigns the weight for very small and very large weights. The problem of vanishing gradient can be effective be shown in the results.

Xavier initialization, originally proposed by Xavier Glorot and Yoshua Bengio in[22], the initialization technique of weights which attempts to make the variance of a layer's outputs equal to the variance of its inputs. In fact, this concept turned out to be very useful. This initialization naturally depends on the role of layer activation. And Glorot and Bengio considered the logistic sigmoid activation function in their paper, which at that moment was the default option.

The sigmoid activation was later surpassed by ReLU, as it allowed the problem of disappearing/exploding gradients to be solved. As a result, a new initialization technique emerged that applied the same concept (balancing the activation variance) to this new activation function.

References

[1] Masood, Sarfaraz & Chandra, Pravin. (2012). Training neural network with zero weight initialization. ACM International Conference Proceeding Series. 235-239. 10.1145/2381716.2381761.

[2]J. Torres-Sospedra, C. Hernandez-Espinosa and M. Fernandez-Redondo, "Designing a Multilayer Feedforward Ensemble with the Weighted Conservative Boosting Algorithm," 2007 International Joint Conference on Neural Networks, Orlando, FL, 2007, pp. 684-689, doi: 10.1109/IJCNN.2007.4371039.

[3] Li, G., Alnuweiri, H., Wu, Y. "Acceleration ofBackpropagations through Initial Weight Pre-Training withDelta Rule." Proc. of the IEEE Int. Conference on NeuralNetworks, ICNN'93, vol. 1, pp. 580-585, 1993.

[4] Drago, G.P., Ridella, S. "Statistically Controlled ActivationWeight Initialization (SCAWI)." IEEE Transactions. On Neural Networks, vol. 3, no. 4, pp. 627-631,1992.

[5] Kim, Y.K., Ra, J.B.. "Weight Value Initialization forImproving Training Speed in the BackpropagationNetwork." Proc. of Int. Joint Conf. on Neural Networks, vol.3, pp. 2396-2401, 1991.

[6] Chunchland P.S. & Sejnowski T.J.: "TheComputationalBrain", Cambridge, MA: MIT Press, 1992.

[7] K. Hornik, M. Stinchcombe and H. White: "MultilayerFeedforward Networks are UniversalApproximators",Neural Networks, 2, 359-366 (1989).

[8] Rumelhart, D. E., McClelland, J. L., & the PDP researchgroup. (1986). "Parallel distributed processing:Explorations in the microstructure of cognition. Volume I" Cambridge,MA: MIT Press.

[9] K. Hornik, M. Stinchcombe and H. White: "UniversalApproximation of an Unknown Mapping and Its DerivativesUsing Multilayer Feedforward Networks,"NeuralNetworks, 3, 551-560 (1990).
[10] "Comparison of Adaptive Methods for Function Estimationfrom Samples", IEEE Transactions on Neural Networks,vol. 7, no. 4, pp. 969-984, 1996.

[11] Jim Y.F.Yam, Tommy W.S Chow "A Weight Initialization Method to improving the training speed in Feeforward Neural Network" Elsevier NeuroComputing(2000)

[12]MPS Bhatia, Veenu, Pravin Chandra " Impact of Weight Initialization on Training of Sigmoidal FFANN", IJSC.2018, 1692-1695.

[13]. G.P. Drago, S. Ridella, Statistically controlled activation weight initialization (SCAWI), IEEE Trans. Neural Networks 3 (1992) 627}631

[15]. Dmytro Mishkin, Jiri Matas "All You need is a Good Init", published at ICLR 2016

[16]. Celso A. R De. Sousa, "An overview of weight initialization methods for feedforward Neural Networks", RG -2016

[17]. Raja Giryes, Gullimero Sapiro, Alex M Bronstein, "Deep Neural Network awith Random Gaussian Weights: A Universal Classification Strategy", arXiv – March 2016.

[18]. Petr Dolezel, Pavel, Skrabanek, Lunir Gago, "Weight Initialization Possibilities for Feedforward Neural Network with Linear Saturated Activation Functions", Elsevier 2016.

[19]. R Rojas, "Optimal weight initialization for neural networks", International Computer Science Institute, Berkeley, Springer, 1994.

[20]. Grzegroz Dudek, "A method of generating random weights

and biases in feedforward neural networks with random hidden nodes", Poland

[21]. Andrew Ng, Kian Katanforoosh, "Xavier Initialization and Regularization", Stanford.edu

[22]. Xavier Glorot, Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks", AISTATS, 2010

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731