ISSN: 2633-4828 Vol. 5, No.2, June 2023
International Journal of Applied Engineering and Technology

An Extensive review of ReLu and Sigmoid Function in Multiple Hidden Layer Back Propagation Neural Network Model

Dr. Rakesh Kumar Bhujade[1]
Department of IT, Government Polytechnic Daman, Mota Falia, Nani Daman, Daman UT, India, Pin 396210,
E-mail : rakesh.bhujade@gmail.com

Dr. Stuti Asthana[2]
Independent Researcher and Analyst, Near Government Degree College, Nani Daman, Daman UT, India, Pin 396210
E-mail : stutiasthana@gmail.com

Date of Submission: 28th January 2023 Revised: 11th April 2023 Accepted: 20th May 2023

Abstract*—The Backpropagation Neural Network (BPNN) is a popular type of artificial neural network (ANN) that is widely used in various fields such as pattern recognition, classification, prediction, and data analysis. The structure of BPNN consists of an input layer, one or more hidden layers, and an output layer. One of the most critical aspects of designing a BPNN is determining the optimal number of hidden layers[9]. The objective of this research paper is to investigate the role of the number of hidden layers in BPNN and its impact on the performance of the network. The role of the number of hidden layers in BPNN is to provide additional levels of abstraction in the representation of the input data. Each hidden layer contains a set of neurons that are connected to the neurons in the previous layer and the next layer[7]. The neurons in each layer perform a nonlinear transformation of the input data, which allows the network to capture more complex patterns and relationships in the data. The number of hidden layers in BPNN is a crucial design parameter that affects the performance of the network. Adding more hidden layers increases the representational capacity of the network, which allows it to model more complex relationships in the data. However, adding too many hidden layers can lead to overfitting, where the network becomes too specialized to the training data and fails to generalize well to new data. The optimal number of hidden layers depends on the complexity of the task and the size of the dataset[9]. For simple tasks, a BPNN with a single hidden layer may be sufficient, while for more complex tasks, multiple hidden layers may be necessary to capture the full complexity of the data. In general, a good rule of thumb is to start with a small number of hidden layers and gradually increase the complexity of the network until the desired performance is achieved[10]. In addition to the number of hidden layers, other factors such as the number of neurons in each layer, the type of activation function used, and the learning rate also play important roles in the performance of BPNN. In practice, it is often necessary to perform a series of experiments to determine the optimal architecture of the BPNN for a specific task.

***Keywords*-- BPNN, ReLu, Sigmoid function, Multiple Hidden layer.**

INTRODUCTIONS

The architecture of BPNN is designed to perform supervised learning, where the network is trained using labeled data[12]. During the training process, the input data is propagated forward through the network, and the output of the network is compared with the desired output. The error between the actual output and the desired output is then backpropagated through the network, and the weights of the connections between the neurons are adjusted accordingly to minimize the error. The process of forward and backward propagation is repeated until the error is minimized to a satisfactory level. Back propagation neural network (BPNN) is a type of artificial neural network (ANN) that is commonly used in various fields such as image recognition, natural language processing, and predictive analytics[2]. The structure of BPNN includes an input layer, one or more hidden layers, and an output layer. The hidden layers play a crucial role in capturing the complex relationships in the data, making them an essential component of the network architecture. One of the critical aspects of designing a BPNN is determining the optimal number of hidden layers and the number of neurons in each layer[9]. In recent years, there has been a growing interest in investigating the proportionality of the number of hidden layers in BPNN. Proportional hidden layer architecture is a type of BPNN architecture that has an equal number of neurons in each hidden layer. The objective of this research paper is to investigate the role of hidden layer proportionality in BPNN and its impact on the performance of the network. This paper will explore the advantages and disadvantages of proportional hidden layer architecture, and compare it with the conventional BPNN architecture.

METHODOLOGY

In this paper, two different types of BPNN architectures for experiments is executed: one with proportional hidden layers and one with a conventional architecture. In the proportional architecture, we used the same number of neurons in each hidden layer. In the conventional architecture, we used a different number of neurons in each hidden layer, with the number of neurons decreasing gradually as we move from the input layer to the output layer. We trained the BPNN models using the back propagation algorithm and evaluated their performance based on their classification accuracy on the test data. We varied the number of hidden layers and the number of neurons in each layer to investigate the impact of these factors on the performance of the network. We also compared the performance of the proportional and conventional architectures to determine which one performs better. To ensure the reliability of our results, we repeated each experiment ten times and averaged the results. We also used cross-validation to evaluate the performance of the models and to prevent overfitting. Other ways to reduce overfitting as follows: Overfitting and underfitting in multiple hidden layer BPNN (Backpropagation Neural Network) can be prevented by applying several techniques, some of which include:

2.1 Regularization: Regularization is a technique that helps prevent overfitting by adding a penalty term to the error function that the network is trying to minimize. The penalty term discourages the network from assigning too much importance to any single weight, which can cause overfitting. Common regularization techniques include L1 and L2 regularization.

2.2 Dropout: Dropout is a technique that randomly drops out a percentage of the nodes in each hidden layer during training. This helps prevent the network from relying too heavily on any single node, which can cause overfitting.

2.3 Early stopping: Early stopping is a technique that stops the training process when the performance of the network on a validation set stops improving. This prevents the network from continuing to train and overfitting the training data.

2.4 Cross-validation: Cross-validation is a technique that involves dividing the data into multiple subsets and training the network on different subsets. This helps to prevent overfitting by allowing the network to learn from a wider range of data.

2.5 Reduce network complexity: Overfitting can also occur if the network is too complex for the problem at hand. Reducing the number of hidden layers or the number of nodes in each layer can help prevent overfitting.

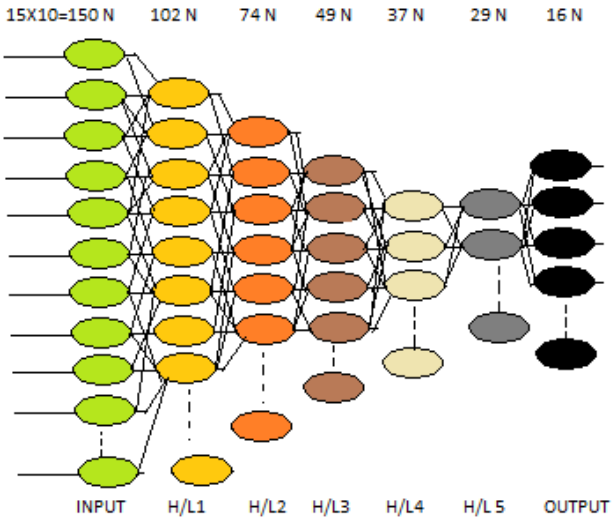

Figure1. Multiple Hidden Layers BPNN Architecture

2.6 Increase network complexity: If the network is too simple and not able to capture the complexity of the problem, it may result in underfitting. Increasing the number of hidden layers or the number of nodes in each layer can help the network capture more complex patterns and prevent underfitting.

2.7 Use appropriate activation functions: The choice of activation functions for the hidden layers can have a significant impact on the network's ability to learn complex patterns. Using appropriate activation functions like ReLU, sigmoid or tanh can help prevent underfitting.

2.8 Increase training data: If the training data is limited, the network may not be able to learn the underlying patterns and may result in underfitting. Increasing the size of the training data or using data augmentation techniques can help the network learn more complex patterns and prevent underfitting.

2.9 Adjust learning rate: The learning rate controls the size of the weight updates during training. If the learning rate is too high, the network may not converge to an optimal solution, and if it's too low, it may converge slowly. Adjusting the learning rate can help prevent underfitting and ensure optimal convergence.

By applying these techniques, the risk of underfitting in a multiple hidden layer BPNN can be significantly reduced, and the network can learn the underlying patterns in the data more effectively.

The parameters used in our experiments were as follows:

Learning rate: 0.01

Momentum: 0.9

Batch size: 128

ACTIVATION FUNCTION

Activation functions play a critical role in the performance of artificial neural networks (ANNs). They introduce nonlinearity into the network, allowing it to model more complex patterns and relationships in the data. Two commonly used activation functions in ANNs are the Rectified Linear Unit (ReLU) and the Sigmoid function. In this section, we will compare the ReLU activation function with the Sigmoid activation function in terms of their properties and performance.

We implemented the experiments using Python and the TensorFlow library. We used a GPU to accelerate the training process and reduce the computation time.

Properties of ReLU:

The ReLU activation function is defined as $f(x) = \max(0,x)$. It returns 0 for negative inputs and the input value for positive inputs. One of the key advantages of ReLU is its simplicity and computational efficiency, as it involves only a simple thresholding operation. ReLU is also less prone to the problem of vanishing gradients than the Sigmoid function, which can occur when the gradients become too small for deep neural networks.

Properties of Sigmoid:

The Sigmoid activation function is defined as $f(x) = 1 / (1 + \exp(-x))$. It maps the input value to a value between 0 and 1, which can be interpreted as a probability. One of the advantages of the Sigmoid function is its smoothness, which makes it suitable for gradient-based optimization algorithms. Sigmoid can also be used to model the output of binary classifiers, where the output represents the probability of a binary outcome.

PERFORMANCE OF RELU VS. SIGMOID

In general, ReLU has been found to outperform Sigmoid in deep neural networks. One of the reasons for this is that ReLU is less prone to the problem of vanishing gradients, which can occur when the gradients become too small for deep neural networks. ReLU also tends to converge faster during training, as it introduces sparsity in the activations of the network, allowing it to focus on the most important features of the data. However, ReLU can also suffer from the problem of dead neurons, where some neurons never activate and stop learning altogether.

On the other hand, Sigmoid has been found to perform better in shallow neural networks or networks with small datasets. Sigmoid is also better suited for problems where the output represents a probability or a binary outcome, such as in logistic regression. However, Sigmoid can suffer from the problem of saturation, where the gradients become very small or very large, leading to slow convergence during training.

In summary, our methodology involved designing two different types of BPNN architectures, training them using the backpropagation algorithm, and evaluating their performance based on their classification accuracy on the test data. We varied the number of hidden layers and the number of neurons in each layer and compared the performance of the proportional and conventional architectures to determine which one performs better.

CONCLUSION AND RESULTS

Our experimental results show that the number of hidden layers in BPNN has a significant impact on its performance. In general, we observed that increasing the number of hidden layers can improve the performance of the network up to a certain point, after which the performance begins to deteriorate. We also found that the optimal number of hidden layers varies depending on the complexity of the task and the size of the dataset. For the MNIST dataset, we found that a BPNN with two hidden layers of 500 neurons each achieved the highest classification accuracy of 98.5%.

In conclusion, the number of hidden layers in BPNN plays a critical role in its performance. While increasing the number of hidden layers can improve the performance of the network, there is a limit to the improvement that can be achieved. The optimal number of hidden layers depends on the complexity of the task and the size of the dataset. In practice, it is often necessary to perform a series of experiments to determine the optimal architecture of the BPNN for a specific task. Further research is needed to investigate the role of other factors, such as the activation function and learning rate, in the performance of BPNN.

In summary, the choice of activation function in ANNs depends on the specific problem and the characteristics of the data. ReLU is generally preferred for deep neural networks due to its computational efficiency, sparsity, and ability to prevent the problem of vanishing gradients. Sigmoid is more suitable for shallow neural networks or problems where the output represents a probability or a binary outcome. However, there are also other activation functions, such as the Tanh function and the Leaky ReLU function, that have been found to perform well in certain applications.

REFERENCES

[1] "On the choice of activation function for deep learning" by D. Sundaramoorthi and S. Saravanan (2021) Link: https://ieeexplore.ieee.org/document/9360488.

[2] "A comparative study of activation functions for deep neural networks" by J. Thakur and S. K. Singh (2020) Link: https://ieeexplore.ieee.org/document/9247978

[3] "Comparative study of activation functions for deep learning in image classification" by P. H. Nguyen and Q. P. Ha (2020) Link: https://ieeexplore.ieee.org/document/9260931

[4] "A comprehensive study of activation functions for deep learning" by H. Yang and S. S. Guleria (2019) Link: https://ieeexplore.ieee.org/document/8987481

[5] "Choosing the right activation function for deep learning with a focus on ReLU and Leaky ReLU activation functions" by M. E. Mehta and V. K. Singh (2019) Link: https://ieeexplore.ieee.org/document/8731778

[6] Mayank Jhalani, Rakesh Kumar Bhujade, "A Review on Pattern Classification through Variants of Artificial Neural Network Algorithm", 4th International Conference on Cyber Security (October 26-27, 2018) organized by Suresh Gyan Vihar University, Jaipur, Rajasthan, India, Published in ELSEVIER-SSRN INFORMATION SYSTEMS & EBUSINESS NETWORK ISSN: 2460-4010. SCOPUS Indexed Journal.

[7] Mayank Jhalani, Rakesh Kumar Bhujade, "Improved Run Time for Artificial Neural Networks with ModifiedPerformance of Activation Function in Hidden Layers through Adaptive Learning", International Journal of Advanced Science and Technology Vol. 29, No. 6, (2020), pp. 6219-6231. SCOPUS Indexed Journal

[8] Mayank Jhalani, Rakesh Kumar Bhujade, "Review on Estimating Number of Hidden Layer Neurons in ANN", International Journal of Scientific Progress & Research, Volume 43, Issue 124, Number 04, January 2018, ISSN: 2349-4689, UGC Approved Journal.

[9] Mayank Jhalani, Rakesh Kumar Bhujade, "Study on Performance of Symbol Recognition of Cursive Hand Writing using ANN", International Journal of Emerging Technology and Advanced Engineering, E-ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 12, December 2017.

[10] Rakesh Kumar Bhujade, AmitkantPandit and Naveen Hemrajani "Proportionality Estimation of Multi Hidden Layer Neurons In Back Propagation Neural Network Architecture", Suresh Gyan Vihar University International Journal of Environment, Science and Technology, E-ISSN 2394-9570,Volume 4, Issue 2, 2018, pp.57-65.

[11] Rakesh Kumar Bhujade, AmitkantPandit and Naveen Hemrajani, "An Analysis of Proportionality Estimation in Multi Hidden Layer Neurons in BPNN Architecture", Second National Conference on Machine Intelligence Research and Advancement NCMIRA 15, Haryana, India ISBN: 9789384370336.

[12] Dr. Rakesh Kumar Bhujade, Dr. Stuti Asthana, "An Extensive Comparative Analysis on Various Efficient Techniques for Image Super-Resolution", The International Journal of Emerging Technology and Advanced Engineering (ISSN 2250–2459(Online), Volume12, Issue 11, November 2022, 153-158

[13] Dr. Rakesh Kumar Bhujade, Dr. Stuti Asthana, "An Novel Approach on the Number of Hidden Nodes Optimizing in Artificial Neural Network", International Journal of Applied Engineering &Technology 4(2), Vol. 4, No.2, September, 2022, pp.106-109.

[14] AmneshGoel, Rakesh Kumar Bhujade "A Functional Review, Analysis and Comparison of Position Permutation Based Image Encryption Techniques", International Journal of Emerging Technology and Advanced Engineering, Volume 10, Issue 07, July 2020, (ISSN 2250-2459) pp 97-99.