ISSN:2633-4828 Vol. 4, No.2, September, 2022

International Journal of Applied Engineering & Technology

An Novel Approach on the Number of Hidden Nodes Optimizing in Artificial Neural Network

Dr. Rakesh Kumar Bhujade[1], Dr. Stuti Asthana[2]

[1]HoD (IT) Government Polytechnic Daman(UT), [2]Independent Researcher and Analyst, Daman (UT)

Date of Submission: 21st July 2022 Revised: 14th September 2022 Accepted: 20th October 2022

How to Cite: ***Bhujade R et.al.,(2022). An Novel Approach on the Number of Hidden Nodes Optimizing in Artificial Neural Network. International Journal of Applied Engineering &Technology 4(2), pp.106-109.***

Abstract- **Forecasting, classification, and data analysis may all gain from improved pattern recognition results. Neural Networks are very effective and adaptable for pattern recognition and a variety of other real-world problems, such as signal processing and classification concerns. Providing improved pattern recognition results for predicting, categorizing, and data analysis. To provide correct results, neural networks need sufficient data pre-processing, architecture selection, and network training; nevertheless, the performance of a neural network is reliant on the size of its network. The correct preparation of data, selection of architecture, and training of the network are all required for ANN to provide satisfactory results; yet, the efficacy of a neural network still depends on its size. Fundamental to the field of artificial neural networks is the detection of hidden neurons in neural networks. The random selection of hidden neurons may result in either under fitting or over fitting. Over fitting happens when the network's ability to expand beyond the test data is hindered as a result of an abnormally close fit between the data and the network. After training the ANN with real-world data, the proposed method calculates the approximately optimal number of hidden nodes. The number of hidden nodes is determined depending on the similarity of the input data, as per the specified method.**

INTRODUCTION

NEURONS are single processing units in a massively parallel distributed processor called a "Neural Network." While Nathan Rashevsky initially proposed neural networks in 1938, McCulloch and Pitts made significant improvements to their theory two years later, in 1943. Neurons in the human brain work in a similar way, firing or not firing according on the strength of the signal they receive when stimulated. Neurons communicate with one another through synaptic connections. a response is elicited by the firing of a neuron (whether it is stimulating another neuron or causing an end result, like moving your arm or leg). Natural Language Processing [1] and neural networks have been used to recognise a wide range of items, from Chinese characters to credit checks to gas turbine engine vibrations [2].

An ANN is a computer programme that attempts to mimic the brain's neural network in order to better understand the human mind. A perceptron is a neural network comparable to a neuron.

Synaptic connections allow perceptrons to receive input (stimuli) and produce output (actions). There is a weight given to each connection in the neural network (Wi). Finally, this weight multiplies the input value (Xi).

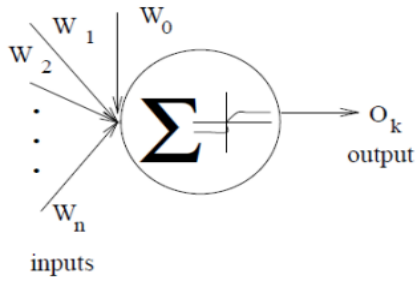

Figure 1.Artificial Neural Network Perceptron

$$I_i = \sum_{i=1}^{m} WiXi (1)$$

$$O_k = Ai = \frac{1}{1+e^{-I_i}} (2)$$

Weighted input values (Ii) to a perceptron are summed up for each connection/synapse that is attached to the perceptron from I = 1 to m The activation value (Ai) is obtained by applying an activation function to the total weighted inputs of the perceptron (eq. 1). Equation 2 shows a commonly used activation function, the sigmoid activation activation function. The sigmoid activation function is nonlinear, much as other activation functions. An important nonlinearity is added to the system by the sigmoid function or "squashing" function, which allows output values to fluctuate from -1 to 1 or zero to 1. With a sufficient number of hidden units, ANNs can estimate any continuous function across a limited input domain with arbitrary accuracy, making them something of a "universal approximator" refer to Figure 1 [4] [5] [6].

BACKWARD PROPAGATION

Training data are required for supervised machine learning algorithms like neural networks. This data may be collected in a variety of ways. Data used for training a neural network helps the network to uncover patterns hidden within the data.

When training ANNs, back propagation, also known as error back-propagation, is often utilized. Back propagation works as follows, assuming that the error is linear and has the form E = (a – i, where E is the activation of the ANN minus the actual input. A feed forward approach is used for the transmission of the training data across the network. This result in the generation of output values, which are then compared to the actual values included inside the dataset. An error is then calculated (using eq. 3) for the output node (s).

Because of this, we have a phrase for it called "Back-Propagation." The intelligence or knowledge of an ANN is stored in the weights of the connections, and by training the network, an ANN determines the appropriate weights for the connections in order to create the appropriate input/output behavior as determined by the data. The intelligence or knowledge of an ANN is stored in the weights of the connections. In eqs 3 and 4, the value k denotes the degree of imprecision associated with the kth output node. In order to determine it, both the actual output value of the network for that node and the anticipated output value are used in the computation. When calculating the errors of the hidden nodes, first the sum of all weights, W_{kh}, are multiplied by the errors of the associated output nodes (k), and then the output of the h_{th} hidden node is used as the final factor (O_h). A particular version of the gradient descent search strategy is known as the backpropagation algorithm. The error back-propagation approach searches for the mean squared error value that is the smallest possible value in space of weights.

$$\delta_k = O_k(1 - O_k)(t_k - O_k) \qquad (3)$$

$$\delta_h = O_h(1 - O_h) \sum_{k \in outputs} W_{kh} \cdot \delta_k \qquad (4)$$

While descending the grade, local minimum issues may arise. By taking little steps in a direction that seems to be most promising at any given moment, gradient descent seeks a global (or overall) solution to a given problem. As soon as you've taken every possible step away from the goal, which is the smallest possible mistake, the back-propagation algorithm stops looking. When the algorithm explores your area, it may be drawn to a local minimum rather than a global minimum due to the "local minimum" problem. A local minimum is the lowest value in its immediate vicinity, but it is not the lowest value in your whole space. The problem of a local minimum may be addressed in a number of ways. This problem may be solved by using step decay. The back-propagation algorithm's step sizes fluctuate with each iteration, and this varies with decay. In order to get to the global minimum, the algorithm must be able to step past local minimums as displayed in Figure 2. As the number of iterations increases, the size of each step reduces, making it easier to zero in on the right answer.

A second approach is to simply re-establish the network with new synaptic weights that are generated at random. A momentum value, on the other hand, takes into account both the most recent weight changes and the most recent error in this training cycle. It is yet another way to reduce the danger of local minima by using a smaller training set to verify the training set. After stopping and restarting ANN training if the error rate rises in comparison to the validation set. ANN training is also suspended if the ANN error against validation training remains consistent across numerous cycles. This prevents over fitting or overtraining of the ANN to the training feature set by pausing and restarting training when the ANN training error deviates from the validation set, the danger of local minima may also be reduced by regularly resetting neural network training with validation. Validation sets may be used to speed up ANN training when the mean squared error is at its lowest point and no further neural network training is required. As a result of this method known as early halting, over fitting is minimized.

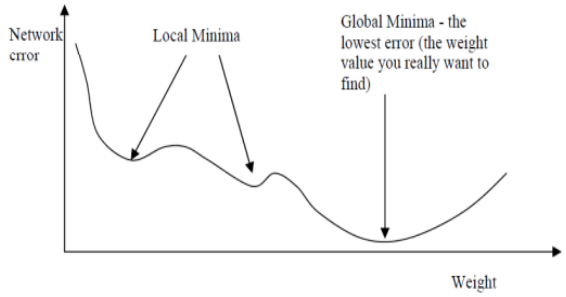

Figure 2.Network Error and Global and Local Minima

FEATURE EXTRACTION

It is not uncommon for people to get feature selection and feature extraction mixed up. The process of producing or learning advantageous modifications based on the original set of input characteristics is referred to as feature extraction or feature induction. Feature extraction may also refer to the process of feature induction. Using rule-based approaches such as decision trees and rule induction, it is possible to create a collection of characteristics that may be interpreted by the user. An example of it may be found in the form of classification and regression trees, abbreviated as CART [7]. There are occasions when the extracted features may express in a straightforward manner the intricate connections that exist between the observed input attributes. At each level of derived characteristics, layered models, such as an artificial neural network (ANN) or deep belief net [8], have a tendency to replicate more abstract conceptions.

First, edges are recognized, followed by visual attributes, and eventually actual objects, which are found in the deepest layer of an artificial neural network (ANN) or deep belief network.

The either-or connections and complexity in the data may be learned by an ANN or a deep belief net, which can then modify itself as the data changes [9]. These are two different approaches to the process of extracting features. ANNs, in their most basic form, are able to comprehend the subtleties of the data and derive useful information from it. On the other hand, feature selection techniques centre their attention either on finding the most accurate predictors or on improving the precision of the classification variables.

FEATURE SUBSET SELECTION

The purpose of the feature extraction process is to demonstrate how ANNs may be used to learn about intricate connections hidden within the data. The process of picking subsets of predictors that, when used in conjunction with one another, have the greatest predictive value is known as feature subset selection. The dimensionality of the feature input space may be reduced by the use of subset selection (refer to Figure 3).

Wrappers and filters are the two basic ways that may be used in the process of selecting subsets of variables that can then be utilised. During the first phase of the processing procedure known as filtering, variable subsets are selected. This step is carried out no matter the categorization technique that is being used. That is to say, Filters are not concerned with the learning approach that is being applied to the data set in order to categorise it; rather, they are only concerned with the data set itself. When utilising Filters, the whole collection of input features is first put through the technique for selecting feature subsets, and then, after the feature subset space has been chosen, the data is put through the classification algorithm. This happens after the feature subset space has been selected.

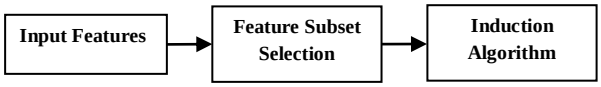

Figure 3.Feature Subset Selection with Filters

Neural Network Architecture

A description was given of the conventional architecture of a fully connected neural network. The GBANN-V system utilises a technique based on dynamic neural networks, which makes advantage of the fully connected ANN architecture. The unique combination of chromosomes that make up a person is responsible for the generation of a one-of-a-kind ANN for every member of the GA population. As a consequence of this, every member of the GA population will have their own set of input nodes, each of which will be linked to a layer of hidden nodes. This is an inevitable consequence. The size of each person's neural network (ANN) differs depending on which chromosome or feature set was chosen for that individual from the population to represent them. For instance, as shown in Figure 4, a person who has just five inputs and a single classifier has a total of four hidden nodes in their network.

In an ANN, the number of hidden nodes will always be one less than the total number of classifiers or features, which is denoted by n. To simplify ANN creation, just one hidden layer and n-minus-1 hidden nodes are needed in the GBANN-V programme design and run-time execution.

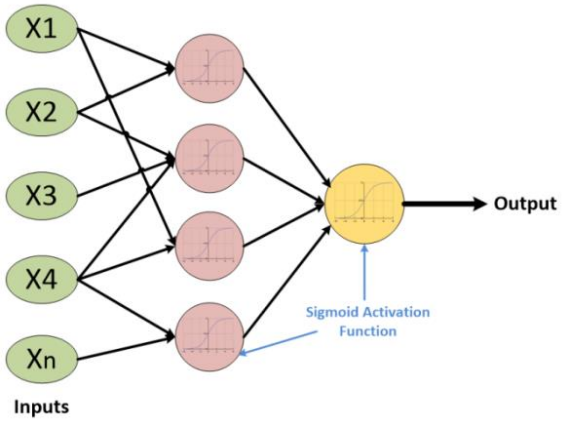

Figure 4.Individual ANN Classifier with one Hidden Layer

GENETIC ALGORITHMS

The Genetic Algorithm (GA) is a search technique that starts its inquiries from a population of chromosomes. It was developed by James Watson and Maurice Wilkins. This is the foundation upon which the Genetic Algorithm (GA) is built (also called individuals). Every chromosome has a collection of coded parameters that are directly related to the issue that needs to be fixed. By "applying GA," we mean the process of creating a population that is "better" than the one that came before it in terms of the average fitness of its chromosomes; in other words, this population improves as a direct result of the application of GA. When we talk about "applying GA," we are referring to this process. The GA is assigned a value that might vary from "better" to "worse" based on the fitness value associated with each chromosome. This value is what is used to determine the GA's value. If a chromosome has a high fitness grade, then we say that the chromosome is "excellent." A function that is referred to as the fitness function is responsible for calculating the fitness score.

The GA places a significant amount of emphasis on mutation and crossover as two of its primary core ideas. If an existing chromosomal bit undergoes a random alteration as a result of the process of mutation, it is possible that other instances of the issue may emerge. When two chromosomes are brought together in this fashion, the characteristics of both chromosomes are combined into a single set of characteristics. The GA will first choose chromosomes to be used in the construction of a new population (chromosomes that are more fit are more likely to be selected for this purpose), it will then modify those chromosomes, and ultimately, it will replicate those chromosomes to the new population.

One method that may be used to estimate the likelihood that a particular chromosome would fare well in the process of natural selection is to divide the chromosome's level of fitness by the average level of fitness found in the population as a whole. Permit the rearrangement of the chromosomes in such a way that the fitness levels of the organism drop from greatest to lowest [10].

RELATED WORK

Within the context of this letter, Li-Ye presented an original method [11] for parametric modelling of the electromagnetic behaviour of thinning arrays. After receiving the element EM responses from the conventional RBFNN, the built HNI–RBFNN works in conjunction with the element spacings of thinning arrays to map the element EM responses to the array ones in the proposed model. This mapping takes place in the proposed model. The mutual coupling between elements and the array environment is what determines the weights that are used to connect the nodes that are found in the hidden layer of the HNI–RBFNN model. To determine whether or not the model is accurate, a numerical example is used. When compared to rival models that use the same data for training and testing, the accuracy of the results produced by the model under consideration is significantly improved. It is possible to extend ANN modelling beyond finite periodic arrays and into the realm of thinning arrays by using the methodologies that have been provided.

CONCLUSION

Numerous applications rely on unsupervised learning, such as system control, pattern recognition (e.g. face recognition), speech and gesture detection, financial applications such as automated trading systems, medical diagnostic handwritten text recognition nosis (MDHTR), data mining (or Knowledge Discovery in databases, "KDD"), visualisation and email spam filtering and diagnosis, and diagnosis and diagnosis.

A critical issue in the design of unsupervised neural networks is determining how many neurons to include in the hidden layer. Random selection of hidden neurons in a network may result in either Underfitting or Overfitting. Using Kohonen's network as an example, this article explains how to select and correct hidden nodes in a Neural Network, as well as the potential of Unsupervised Learning in Artificial Neural Networks. We'll continue to develop and refine these approaches as well as put them to the test with real data in the future.

REFERENCES

[1] D. Jurafsky, J. H. Martin, P. Norvig and S. Russell, Speech and Language Processing, Pearson, 2014.

[2] G. A. Harrison, Neural Networks in Vibration Analysis of Gas Turbines, University of Florida, 1997.

[3] J. Heaton, Programming Neural Networks with Encog3 in Java, 2nd Edition, Heaton Research, Incorporated, 2011.

[4] C. M. Bishop, Neural Networks for Pattern Recognition (Advanced Texts in Econometrics (Paperback)), Clarendon Press, 1996.

[5] S. O. Haykin, Neural Networks and Learning Machines, Pearson, 2011.

[6] Amnesh Goel, 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

[7] L. Breiman, J. Friedman, C. J. Stone and R. A. Olshen, Classification and Regression Trees (Wadsworth Statistics/Probability), Chapman and Hall/CRC, 1984.

[8] Cimenler, O., Kingsley, A., Reeves, A. and Kasti, M., 2021. Testing a Multisource Feedback Instrument in Healthcare Organizatios. International Journal Data Modelling and Knowledge Management, 6(1).

[9] G. E. Hinton, S. Osindero and Y.-W. Teh, "A Fast-Learning Algorithm for Deep Belief Nets," Neural Computation, vol. 18, pp. 1527-1554, jul 2006.

[10] Sadeghi, S. and Saen, R., 2020. Developing an Imprecise Slack-Based Measure for Supplier Selection. Chinese Journal of Decision Sciences, 1(1).

[11] M. Hassoun, Fundamentals of Artificial Neural Networks (MIT Press), A Bradford Book, 2003.

[12] D. Whitley; 'The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best'. In Proceeding of the 3rd International Conference on Genetic Algorithms and their applications (ICGA), 116–121, J.D. Schaffer (Ed.), Morgan Kaufmann, San Mateo CA, 1989.

[13] Li-Ye Xiao " Radial Basis Function Neural Network With Hidden Node Interconnection Scheme for Thinned Array Modeling" IEEE Antennas and Wireless Propagation Letters, Vol. 19, No. 12, December 2020.

[14] Ballova, D., 2021. Analysis of the Development of the Groundwater Level Regime in the lower Rye Island, Slovakia. Stochastic Modelling and Computational Sciences, 1(1), pp.43-51.

[15] Istianingsih et.al., 2022. The Role of Self-Control in the Impact of Artificial Intelligence Innovation on Lending Decisions in Online Fintech. International Journal of Applied Engineering and Technology 4(1)