iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** IV **Month of publication:** April 2019

DOI: https://doi.org/10.22214/ijraset.2019.4103

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue IV, Apr 2019- Available at www.ijraset.com

Associative Neural Network

Mr. Rishabh Singh Rathore[1], Dr. Anshuman Tyagi[2], Mr. Brijesh Kumar Dubey[3]
[1]*Scholar, Department of Electrical and Electronics Engineering, PSIT, Kanpur, Uttar Pradesh*
[2]*Head of Department, Department of Electrical and Electronics Engineering, PSIT, Kanpur, Uttar Pradesh*
[3]*Assistant Professor, Department of Electrical and Electronics Engineering, PSIT, Kanpur, Uttar Pradesh*

Abstract: ***A collaborative neural network (ASN) is a combination of feed-forward neural networks and a group of closest neighboring techniques. The network offered represents a correlation between the responses collected as the measure of distance between the cases analyzed for the nearest neighbor technology and provides a better prediction by the bias improvement of the neural network ensemble. A collaborative neural network has a memory that can be found with the training set. If new data becomes available, the network improves further forecasting ability and can often provide a reasonable estimate of the unknown function without the need to stop the piece of neural network.***

I. INTRODUCTION

The traditional synthetic feed-forward neural network (ANN) is a bleak vision. This means that after completing the training, all the information about the input pattern is stored in neural network weight and input data is not required, i.e. there is no clear storage of any presented instance in the system. On the contrary, the closest-to-neighbors (KNN) (for example, Dasherty, 1991), perjon-window regressions (e.g., Hurdle, 1990), etc., represent the memory-based approach. These approaches keep the whole database of memory in memory and their predictions are based on some local projections of stored examples. Neural networks can be considered as a global model, while the other two approaches are generally considered to be local models (Lawrence et al., 1996).

For example, consider the problem of multiplexing function approximation, i.e. finding mapping RM => RN from given set of sample points. For simplicity, we assume that n = 1 A global model input provides a good estimate of the data space RM's global metric. However, if the work analyzed, F, is very complex, then there is no guarantee that all details of F, i.e., will be represented in its good structure.

Thus, the global model can be insufficient because it does not mainly describe the entire state's place due to the high bias of the global model in certain areas of space. ANN variation can also contribute to poor performance of this method (Gemman et al., 1992). However, variation can be reduced by analyzing the large number of variation networks, i.e. using the artificial neural network ensemble (ANNE), and for example, taking a general average of all networks in the form of the last model.

The problem of ANN's bias can not be easily addressed, for example, using such a large nervous network, such networks can fall into the local minimum and thus there may still be enough bias.

Local models are based on some neighborhood relationships, and analyzing these methods are more relevant to the discovery of a good structure of analysis tasks, i.e. they can get less bias than the global model. However, while applying these methods, the difficult question is how to properly determine the neighborhood relations in the analysis area? Input data analyzed, especially in practical applications, there can be a large number of dimensions and for the final representation, the actual importance of each input parameter and contribution is generally not known. Example 1. Consider the example of the sign function y = sin (x) (1) With dimensions of 1 equals vector x. The training and test sets consisted of N = 100 and 1000 cases respectively, and the input values were evenly distributed over parallel (0, π).

The KNN method was used Z (2) JNN K (X) where z (x) is the estimated value for case X, using the NK (E) Euclidean metric, the training set {xi} Ni = 1 is a collection of the closest neighbors of X between the input vectors x x, xi ||. Note that the memory of KNN was shown by the entire training set {xi} Ni = 1. Number K = 1 was selected to provide minimum leave-one-out error (LOO) for the training set. KNN calculated the basic mean square error, RMS = 0.016 for the test set. Similar results, RMS = 0.022 was calculated by a group of M = 100 ANN

According to the Levenburg-Marquart Algorithm (press et al., 1994), trained with 2 hidden neurons (a hidden layer) trained. Input and output values were generalized for normal (0.1.0.9) intervals and the signogyd activation function was used for all neurons. In this and all the analyzes 50% of the cases were given opportunity and each neural network (TETCO et al., 1995) was used as a training set. The rest of the cases were initially used as a verification set Stopping Method (Bishop, 1995). Thus, each neural network had its own training and verification set. To predict the test network, all the networks were used after learning a simple

average. The selected number of networks in the wired network provided a small difference in forecasting and its standard average error was less than 0.001, that is, the main error of marble prediction was due to the bias of the neural network.

ANNE had a major bias and low forecasting capacity, RMS = 0.20, was calculated when only one hidden neuron was used.

Note that this specific example was based on the KNN metric that matched the learned work (for example, in which case the minimum distance in the input space was also the minimum distance in the output space) and its performance was quite good. However, when the change in the input metric, the performance of this method is greatly reduced for the same method, because it is displayed by the next example.

Example 2. In Example 1 we use two-dimensional input vector x = {x1, x2}, such as y = sin (x) = sin (x1 + x2) and both x1 and x2 equal variables are distributed.

The best KNN performance for this data was about the order of magnitude compared to RMS = 0.13, K = 2, Example 1. Thus, the use of Euclidean metric was not conducive to the KN method and this method was not able to properly determine the nearest neighbors in place of new variables

Conversely, both neural networks with one and two hidden neurons provided results, which were similar to the result 1, respectively, with RMS = 0.22 and RMS = 0.021, Figure 1A). Thus, both types of networks correctly learned the internal metric of the example, i.e. x = x1 + x2 The following simple ideas can be used in combination with each other to improve the nervous network and KNN's prediction potential.Consider

In our previous study (Tetco and Villa, 1997) we evaluate the values of equality of equality (proximity) as the linear correlation coefficient of Pearson between the zi and zj vectors of rij2 (press et al., 1994) Had proposed to use the class. Case in place of ANN Assume that rij = 0 rij <0 for negative valueser a group of M Neural Networks

KNN results, RMS = 0.060, K = 1, for this example, a lot of improvements were made when its metric was based on rij2, estimated by a group of ANE with a hidden neuron. Better results (RMS = 0.020, K = 1) were achieved when the distance between the cases was estimated using neural networks with two hidden neurons.

The early stopping technique was an important factor contributing to this improvement. The neural networks with one and two hidden neurons trained to convergence with all available training cases predicted test set cases with rms=0.25 and rms=0.020, respectively. The KNN method provided rms=0.082 (k=1) and only rms=0.26 (k=55) when the distance between cases was estimated using the converged networks with one and two hidden neurons, respectively. Thus, the observed improvement for the networks with one hidden neuron was lower than in the case of the early stopping while the converged networks with two hidden neurons were unable to correctly represent the metric of the investigated function.

A use of KNN method discarded the value $\bar{z}_i$ calculated by ANNE. Thus the joint performance of KNN and neural networks for this task could not be better than KNN results calculated in the Example 1, i.e. rms=0.016. A better approach consisted of correcting the value of $\bar{z}_i$ according to formula

$$\bar{z}_i' = \bar{z}_i + \frac{1}{k}\sum_{j \in N_k(\mathbf{x})}(y_j - \bar{z}_j) \qquad (5)$$

Where y_i were the experimental values and the summation was over the k-nearest neighbor cases determined using Pearson's r_{ij}, as described above. Since the variance of ensemble prediction $\bar{z}_i$ was small, the difference $(y_i - \bar{z}_i)$ mainly corresponded to the bias of the ANNE for the case **x**i.

Thus this formula explicitly corrected the bias of the analyzed case according to the observed biases calculated for the neighboring cases.

A use of equation (5) provided an improvement of the results, and rms=0.035 and rms=0.011 were calculated following analysis of ANNE with one and two hidden neurons respectively (Figure 1B). The increase of the number of hidden neurons continuously improved the generalization of the network for useless data (Table 1), but the error calculated in all cases was about two times smaller than ANNE. However, when a random noise with normal distribution, N (0,0.01) was added to the target function, using the equation (5) with more than 3 hidden neurons did not improve the expected capacity of the method, RMS = 0.012, which was very close to theoretical value RMS = 0.01.

We see the proposed method as Associate Nervous Network (ASNN), because the final prediction of new data is done according to the case, i.e. the analysis examples or prototypes of associations found in the "memory" of the neural network. It is believed that the memory of ASNN was the same, like in the input training set in case of KNN. However, there is no requirement that the ASNN memory should always be matched with the training set.

This provides many interesting features of this network illustrated by the following examples.

II. DISCUSSION

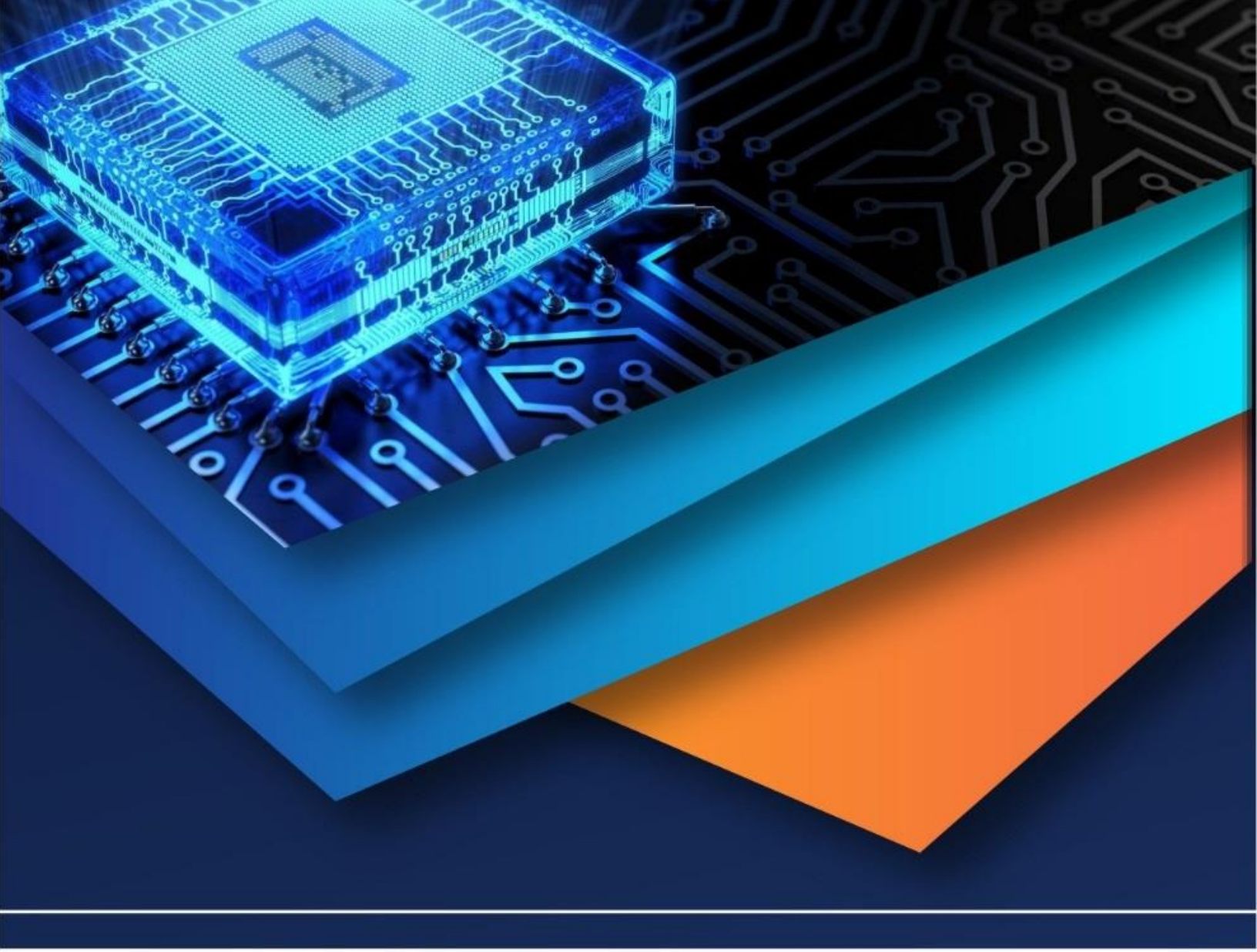
When predicting the property of an object, a person uses the known properties of some other worldly knowledge as well as other similar objects (local knowledge). The final prediction represents some weighted yoga of both of these contributions, and (5) provides a simple model of such a process. This equation also proposes how to add memory-based and memory-based methods to improve their performance.

The current study emphasizes the importance of correlation for processing information. It is an agreement with experimental analysis of the neuronal activity recorded in the mind of these animals, which indicates the presence of fast transient synchronization (Decorks and Zador, 2000; Wadia et al., 1995; Villa et al., 1998). Such transient synchronization (Hopfeld and Brody, 2001) can be used to recover from the memory stored pattern. Recovered patterns can be used for bias improvement.

ASNN has two phases in the learning process. In the first stage, training of neural network fragments involves training of space topology accurately. This is a difficult task and there is a long training time in it, for example it may be in accordance with the acquisition of a driving experience by a person. According to the second stage (5), bias provides correction. This step is used to optimize the number of neighbours to improve local characteristics of the analysis phase. Even if the new car has some special features, only a little customization of local rules is necessary according to the local bias correction to achieve its optimum performance for the driver. To this extent, Example 3 describes a situation when a UK driver (driving on the left) comes into France (driving on the right), that is, when he has to abandon the driving experience on his left side. Separating global and local improvements makes it possible to change the driving experience on the "old" left to the left with new knowledge and to get a better performance in the changed environment. Such changes require the presence of some meeting network, which decides whether old or new knowledge should be used in every particular case. The Getting Network can also use some organizations and a combination of global / local improvements to create their own organization. any algorithms including counterpropagration (Hechat-Nielsen, 1990), Vector Quantization (Kohonan, 2001), Madeline (Widrow and Lehr, 1990), Vector Vector Machine (VapiCal, 1995), SMAC (Albus, 1975) and other methods contain such characteristics that combine closest neighbor techniques with supervised learning. While the exact comparison of the performance of these algorithms with ASNN without any doubt is fully studied, all these algorithms are very different in their designs with the proposed method. For example, they can not improve their generalization or can rapidly guess the new function without training, because it was shown in Examples 2 and 3. It would be interesting to see if other non-linear global approximation methods can be used instead of feed-forward neural networks. Another possibility is to check whether other local regression techniques, for example using weighted sum (ji -yi) w (rij) / w (rij) can further increase ASNN's estimated assets.The number of neighbors in the current study, which is the lubricant parameter of the algorithm, was similar to the whole work. This may be appropriate for those who have almost the same level of noise and have the same distribution of cases on the places examined. However, if the delivery of noise and sampling is not uniform, then selection of lubricating parameters based on local topology could possibly improve ASNN's predicted performance.

REFERENCE

[1] Albus, J.S. (1975) A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC). Trans. ASME J. Dyn. Syst. Meas. Control, **97**, 220-227.
[2] Bishop, M. (1995) Neural Networks for Pattern Recognition. Oxford University Press, Oxford. Dasarthy, B. (1991) Nearest Neighbor (NN) Norms. IEEE Computer Society Press, Washington, DC.
[3] deCharms, R.C. and Zador, A. (2000) Neural representation and the cortical code. Annual Review of Neuroscience, **23**, 613-47.
[4] Geman, S., Bienenstock, E. and Doursat, R. (1992) Neural Networks and the Bias/Variance Dilemma. Neural Computation, **4**, 1-58.
[5] Härdle, W. (1990) Smoothing Techniques with Implementation in S. Springer-Verlag, New York. Hecht-Nielsen, R. (1990) Neurocomputing. Addison-Wesley.
[6] Hopfield, J.J. and Brody, C.D. (2001) What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration. Proceedings of the National Academy of Sciences of the United States of America, **98**, 1282-1287.
[7] Kohonen, T. (2001) Self-Organizing Maps. Springer, Berlin.
[8] Lawrence, S., Tsoi, A.C. and Back, A.D. (1996) Function Approximation with Neural Networks and Local Methods: Bias, Variance and Smoothness. In Bartlett, P., Burkitt, A. and Williamson, R. (eds.), Australian Conference on Neural Networks. Australian National University, Australian National University, pp. 16-21.
[9] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1994) Numerical Recipes in C. Cambridge University Press, New York.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY