International Journal of Civil Engineering and Technology (IJCIET)
Volume 9, Issue 5, May 2018, pp. 1253–1259, Article ID: IJCIET_09_05_140
Available online at http://www.iaeme.com/ijciet/issues.asp?JType=IJCIET&VType=9&IType=5
ISSN Print: 0976-6308 and ISSN Online: 0976-6316

MATHEMATICALLY DESIGNED ARTIFICIAL NEURAL NETWORKS WITH GAUSS NEWTON ALGORITHM

G. Gajendran
Department of Mathematics,
SRM Institute of Science and Technology, India

T. Venugopal
Director of Research and Publication
SCSVMV University, India

ABSTRACT

A simplified representation of some discrete problem is used to predict working of that problem. Mathematical model makes certain resembling on that problem. A multilayer Feedforward neural network is one of the computational mathematical model that can be performed as a linear or nonlinear mapping within the real world problems. Mathematically designed Artificial Neural Networks is applied on Multilayer Feedforward neural network with Backpropagation under Gauss Newton algorithm to overcome the complexity of conventional model. Also proposed model makes a prediction about real world problem which requires a model on classification for clinical findings and obtained results are compared with existing results.

Keywords: Multilayer Feed forward Neural Networks, Backpropagation, learning rate, Gauss Newton algorithm and Diabetic.

Cite this Article: G. Gajendran and T. Venugopal, Mathematically Designed Artificial Neural Networks with Gauss Newton Algorithm, International Journal of Civil Engineering and Technology, 9(5), 2018, pp. 1253–1259.
http://www.iaeme.com/IJCIET/issues.asp?JType=IJCIET&VType=9&IType=5

1. INTRODUCTION

A Multilayer Feedforward neural networks (MLFFN) with Backpropagation (BPN) under Gauss Newton algorithm method is a computational mathematical model that addresses the current need for sophisticated computation in both engineering and sciences [2, 5]. Computational methods can be developed only after a deep analysis of the qualitative properties of a model and of the related mathematical problems. Such computation requires an understanding of parallel and vector capabilities on a range of real world problems [5]. MLFFN expresses learning activities from environment and their corresponding observation relations. MLFFN with BPN resembles human brain in two aspects: BPN helps to learn

MLFFN to acquire the knowledge from environment and acquired knowledge stored in memory of MLFFN as a set of weight components [5].

Most of the researches are using randomly generated learning set and testing set, fixed learning rate and randomly generated initial weights or fixed interval for weight initialization. It affects the accuracy of the result. Researchers still investigate the methods on MLFFN with BPN to improve the accuracy of result. Mathematically designed Artificial neural networks (MDANN) is a set of mathematical techniques to improve the performance of conventional MLFFN with BPN method under Gauss Newton algorithm [2]. Proposed MLFFN with BPN under Gauss Newton algorithm and various combination of MDANN are applied on Pima Indian diabetes data set (PIDD) [6] to classify whether patient is non-diabetic patient or diabetic patient. Section 2 discusses Gauss Newton algorithm and various combination of MDANN. Application of the proposed model on classification of clinical findings on PIDD performed in Section 3.

2. MATHEMATICALLY DESIGNED ANN

In order to improve performance of conventional MLFFN with BPN method, objectives of this section are pre – process for designing real world problem, controlling internal representation of MLFFN and post – process that provides information to BPN whether MLFFN needs to learn or stop the learning. Designing real world problem is a process that designs given real world problem as ductile problem for MLFFN. It consists of normalization and categorization processes which are discussed in previous work [1, 2]. Controlling internal representation of MLFFN consist mapping representation of MLFFN, modified learning rule for batch mode model BPN which are discussed in previous work [1, 2].

2.1. Designing Real World Problem

It is a process that designs given real world problem as a ductile problem for MLFFN. It consists of normalization of real world problem and categorization of normalized examples. Let $X \subseteq [0,1]^m$ be a compact set of normalized input vectors and $D \subseteq [0,1]$ be a set of corresponding normalized target output values. It is must to categorize them into two sets namely training set for training MLFFN and testing set for generalization purpose and it can be proposed [1,2] as follows:

It is possible to choose a positive real number, called as a categorization constant, $c \in R$, for a positive integer $P \in Z$, $\bar{x}^p \in X$ and $d^p \in D$, the training set $\left\{ \left(\bar{x}^p, d^p\right) : p = 1, 2, \cdots, P \right\}$ can be defined as a subset of set of normalized examples such that for $u \neq v$, any pair of elements $\left(\bar{x}^u, d^u\right)$ and $\left(\bar{x}^v, d^v\right)$ in training set satisfies the condition [1, 2]

$$\frac{1}{m+n}\left[\left\| \bar{x}^u - \bar{x}^v \right\|_1 + \left\| d^u - d^v \right\|_1 \right] > c \tag{1}$$

where m is the number of components in $\bar{x}^p \in X$, n = 1 is the number of components in $d^p \in D$ and $\| \, . \, \|_1$ is 1 – norm.

And testing set $\left\{\left(\bar{x}^{p}, d^{p}\right): p = P+1,\ P+2,..., K\right\}$ can be defined as a subset of set of normalized examples such that for any element $\left(\bar{x}^{u}, d^{u}\right)$ in testing set, there exists atleast one element $\left(\bar{x}^{v}, d^{v}\right)$ in training set such that

$$\frac{1}{m+n}\left[\ \left\|\bar{x}^{u}-\bar{x}^{v}\right\|_{1}+\left\|d^{u}-d^{v}\right\|_{1}\right] < c \tag{2}$$

Categorization of two classes expresses that each element of testing set is a neighborhood of at least one element of training set so that trained MLFFN avoids over fitting or under fitting of any testing values.

2.2. Internal representation of MLFFN

Internal representation (IR) of an MLFFN is an activity of formally presenting the accumulated information by BPN from the range of given problem to MLFFN. Controlling the IR consists of mapping representation of MLFFN and improving the performance of BPN [1, 2].

2.2.1. Mapping representation of MLFFN

An analytical presentation to choose MLFFN architecture in the form of mapping or function is known as mapping representation. An MLFFN is a group of artificial neurons where each artificial neuron receives input vector from neurons in the previous layer and transforms them as an output. It implies that initialization of weight components for each neuron in MLFFN is the initialization of MLFFN. At every point of the intervals $(-\infty, 3.4)$ and $(3.4, \infty)$, slope of the tangent of sigmoidal function is at most parallel to axis of input value of neuron and sigmoidal value is very small. MLFFN may strruck with local minima or learning of MLFFN may diverge. It is important to initializing artificial neurons in MLFFN in which input value of neuron leis in $[-3.4, 3.4]$. It is possible when weight components leis in the following derived interval [1, 2]. For a real constant $g > 1.0$ and given set of input vectors $\bar{x} = (x_1, x_2, ..., x_m) \in X$ from training set or derived from training set to any neuron, all weight components w can be chosen from proposed interval

$$\left[-s\sqrt{\frac{g}{(m+1)\Omega}},\ s\sqrt{\frac{g}{(m+1)\Omega}}\right] \text{ if } \sum_{i=1}^{m} w^{2}\sum_{i=1}^{m}\left(x_i^{u}-x_i^{v}\right)^{2} \le s^{2} \tag{3}$$

$$\Omega = Max\left\{\sum_{i=1}^{m}\left(x_i^{u}-x_i^{v}\right)^{2} : \forall u \ne v \in \{1,2,...P\}\right\}$$

where and s = 6.4

for various values of weight initialization constant g, infinitely many neurons can be determined using proposed closed interval (3). This procedure is common for all neurons in MLFFN. Objective to initialize MLFFN is that it can produce output value with lengthy range of output values than randomly generated initial weights. Mathematical model of initial

MLFFN with one hidden layer is defined as a function $N : X^{training} \rightarrow Y$,where constant coefficients are chosen from proposed interval, $X^{training} \subseteq [0,1]^m$ is a proposed training set of input vectors and $Y \subseteq [0,1]$ is a set of corresponding MLFFN output values.

2.2.2. Improving the performance of BPN

Due to nature of initial weight vectors, MLFFN's output value $y^p \in Y$ may or may not equal to target output value $d^p \in D$. An error function $E(\overline{w})$ of weight vector $\overline{w}$ is a sum of Residual Square. BPN aims to minimize $E(\overline{w})$ by updating weight vectors $\overline{w}$ using weight rule:

$$\overline{w}_{t+1} = \overline{w}_t + \Delta\overline{w}_t \tag{4}$$

In general, BPN calculation for MLFFN is less in nature for updating the weight vectors or its components, because BPN knows the direction of optimization only but not its magnitude. Moreover, BPN algorithm is a trial and error model endower. In addition, BPN does not have any updating rule to update all the weight components at a time. It updates either weight vector wise or its component wise. It expresses that optimization of weight vector or its component to disturb the other optimized values of weights. These disadvantages of BPN lead the error function $E(\overline{w})$ to converge slowly or struck with local minima. Gauss–Newton algorithm still exposes convergent problem for complex error space optimization. Researchers still investigate training skill required by BPN to overcome from their longer time of convergence and local minima. In order to improve the performance of BPN, proposed model initiates modified updating value for batch mode Gauss Newton method as follows:

$$\Delta\overline{w_t} \cong \begin{cases} -\eta \left(J^T J\right)^{-1} J^T e & if\ matrix\ norm(J) \geq r \\ 0 & otherwise \end{cases} \tag{5}$$

where matrix norm is the maximum value of sum of each column in the Jacobean matrix J of $E(\overline{w})$, *e* is residual matrix and η is proposed learning rate which given as follows:

For a small real number $\eta(0)$, there exists $0 < q, r < 1$ such that

$$\eta(t+1) = \begin{cases} 0 \ \ if\ (N_t)_{t=0}^{\infty}\ converges\ or\ if\ 3AEB(t) \leq AEA(t) \\ \eta(t) + q + r\dfrac{AEA(t)}{AEB(t)} \quad if\ AEB(t) > AEA(t) \\ \eta(t) - q - r\dfrac{AEA(t)}{AEB(t)} \quad if\ AEB(t) \leq AEA(t) \leq 3AEB(t) \end{cases} \tag{6}$$

where $AEB(t)$ and $AEA(t)$ are defined as sum of absolute error of residues during before and after learning of MLFFN N_t at t^{th} epoch. The proposed learning rate gains information from learning capability of MLFFN N_t and applies the acquired knowledge to the subsequent MLFFN N_{t+1} by updating all weight components involved in MLFFN.

2.3 Post process

During learning process, the sequence of MLFFNs $(N_t)_{t=0}^{\infty}$ has to converge to required MLFFN. But due to its non-monotonic nature, a systematic process is required to obtain required MLFFN. For this, proposed post process at each epoch characterized by renormalization process and stopping rule.

2.2.3. Renormalization

Numerated environment and their corresponding numerated observation are normalized for training and testing purpose for MLFFN. At the end of each epoch, all MLFFN output values, $y^p \in Y \subset [0,1], \forall p$, are re-normalized as follows:

$$Theoretical \quad Output = \begin{cases} Class\ I\ if\ \ y^p < l \\ Class\ II\ if\ \ y^p \geq l \end{cases} \tag{7}$$

where *l* is classification constant and independent of learning. It expresses that *l* can be obtained during learning by comparing best number of classification accuracy.

2.2.4. Stopping Rule

Usually convergence of $(N_t)_{t=0}^{\infty}$ is based on stopping rule. Proposed stopping rule presented for consideration of best approximation. At the end of each epoch, proposed re- normalized output values produced by equation (7) are compared with observation and existing output values of corresponding real world problem. If proposed re-normalized output values of proposed trained MLFFN at some epoch performs best assessment with observation than the existing result, then $(N_t)_{t=0}^{\infty}$ consider to be converge to required MLFFN and proposed learning rate η in equation (6) takes zero so that MLFFN stop the learning.

3. MODEL ON CLINICAL WORKS

Mapping ability of MLFFN with BPN can be estimated if trained MLFFN can provide information from environment of a real world problem sufficiently nearer to corresponding observation of that real world problem. In healthcare industry, it is essential to develop a model on precise data prediction that helps doctors to diagnose diseases. Many researchers involved in health care benefits using artificial neural network to classify clinical diagnosis [3, 4]. In this section, proposed model applied on classification of data from Type 2 Diabetes. Vincent Sigillito, Research Center, RMI Group Leader, Laurel donated Pima Indian diabetic database (PIDD) to UCI Machine Learning Lab. Now PIDD Publicly available in [6] which is a collection of medical diagnostic reports from a population living near Phoenix, Arizona, USA. Pima Indians of Arizona have the highest prevalence and incidence of Type 2 diabetes of any population in the world. All patients are females and at least 21 years old of Pima Indian heritage.

3.1. Description of Proposed Model

PIDD consists of 768 input vectors with eight input components in each input vector and corresponding output value with one component. All 768 examples are collected from various patients. Input components are described as number of times pregnant, 2-Hours OGTT plasma glucose, Diastolic blood pressure, Triceps skin fold thickness, 2-Hours serum insulin, BMI, Diabetes pedigree function, and Age. Corresponding output value is one of two possible outcomes, namely whether patient is tested positive for diabetes or not. Corresponding output value which is non-diabetic numerated as 0 or diabetic as 1. Out of 768 patients, there are 500 (65.1 %) non-diabetic patients and 268 (34.9 %) diabetic patients. Goal of the model is to predict patient is non-diabetic patient or diabetic patient using their corresponding eight input components.

3.2. Proposed Parameters

Experimentally, value of c from equations (1) and (2) is used as c = 0.02242. Outcome of proposed categorization model is described as: out of 768 data, proposed model uses 555 (72.3 %) data for training and rest of 213 (27.7 %) data for testing. According to categorization constant c, empirically thirteen units are used in hidden layer. For proposed weight components between input layer and hidden layer, Ω =1.8606 and g = 3, 3, 4, 4, 3.4, 4.5, 5, 3, 4, 2.8, 3.6, 4.3 and 5.2 are used in equation (3). For proposed weight components between hidden layer and output layer, Ω =2.754 and g = 4 are used in equation (3). For matrix norm, r = 0.249 is used in equations (4) and (5). For initial learning rate $\eta(0)$ = 0.000910031088, there exists learning coefficients q = 0.0010015 and r = 0.00098108 such that proposed learning rate (6) is an automatic review of previous learning and adaptive for present learning or any one of subsequent learning. Classification constant l = 0.53656 is used in the equation (7).

3.4. Performance of Proposed Model

Performance of proposed set of techniques can be verified with their various combinations. They are randomly generated initial weights with other proposed parameters and produced 84.64 % of accuracy on PIDD with 494 epochs, fixed learning rate with other necessary proposed parameters and produced 84.9% of accuracy on PIDD with 423 epochs and Gauss Newton method with proposed techniques and produced 85.6% of accuracy on PIDD with 259 epochs.

Polat [4] used the cascade learning system based on Generalized Discriminant Analysis (GDA) and Least Square Support Vector Machine (LS-SVM) to diagnose PID data set and obtained 78.21% of classification accuracy using LS-SVM with 10-fold cross-validation (10x FC) and 79.16% of classification accuracy using GDA–LS-SVM with 10x FC. Hasan Temurtas [3] used the multilayer neural network structure with Levenberg Marquardt training algorithm with 576 training examples and 192 testing examples, two hidden layers with 50 neurons in each hidden layer, randomly initialized weights and fixed learning rate and reported 82.42% of accuracy on PID data set.

4. CONCLUSIONS

Obtained result from section 3.3 shows that proposed model MDANN on Gauss-Newton method performs well with 85.6% of accuracy on PIDD classification when compare with the existing models on classification of PIDD. Obtained result from section 3.3 shows that categorization model sorts the problem as training set with 555 data and testing set with 213 data and achieves 88.73 percentage of accuracy on testing data whereas 84.72 percentage of accuracy achieved on training data. It concludes that proposed categorization model performs well on generalization of MLFFN. Using proposed weight initialization method, MLFFN with BPN has achieved the accuracy which is better than the accuracy produced by MLFFN with BPN using random initial weight. Using proposed learning rate, MLFFN with BPN has achieved the accuracy which is better than the accuracy produced by MLFFN with BPN using fixed learning rate.

5. FUTURE WORK

Even though MLFFN with BPN performs well on many applications of real world problems, still its mathematical theorizations are less in nature [7]. To improve the mathematical theory behind on MLFFN with BPN, research on MLFFN have to define the number hidden units and BPN have to update weight components without learning rate but analytically so that MLFFN with BPN will converge to required result without stopping rule.

REFERENCES

[1] Gajendran G and Venugopal T, Mathematical approach to design an on line Multilayer Feedforward Neural Network and its model on Laboratory Work, *Journal of International Academic Research for Multidisciplinary*, 3(12), 2016, 143 – 155.

[2] Gajendran G and Venugopal T, Mathematical approach to design a batch mode Multilayer Feedforward Neural Network and its model on type 2 diabetes, *Global Journal of Pure and Applied Mathematics* 12(1), 2016, 105 – 109.

[3] Hasan Temurtas, Nejat Yumusak and Feyzullah Temurtas, A comparative study on diabetes disease diagnosis using neural networks, *Expert Systems with Applications – Elsevier*, 36, 2009, 8610 – 8615.

[4] Polat K, Gunes G and Arslan A, A Cascade Learning System for classification of Diabetes disease: Generalized Discriminant Analysis and Least Square Support Vector Machine, *Expert Systems with Applications*, 34(1), 2008, 482-487.

[5] Sanjeev K kulkarni, Gilbert Harman, An Elementary Introduction to Statistical Learning Theory, *A JOHN WILEY and SONS, INC., PUBLICATION*, 2011.

[6] UCI Machine Learning Repository, *Center for Machine Learning and Intelligent Systems, University of California, Irvine*

[7] Vadamodula Prasad and Srinivasa Rao T, Permissible Thyroid Data sets Assessment through Kernel PC Algorithm and Vapnik Chervonenkis Theory for Categorization and Classification, Springer – Data-Enabled Discovery and Applications, 1, July 2017,1-11.

[8] Sonal Bansal and Dr. Rinku Dixit, ECG Signal Analysis using Artificial Neural Networks - A Review. International Journal of Electronics and Communication Engineering and Technology, 9 (2), 2018 , pp.1 – 6

[9] Dr. Rajeshwari S. Mathad, (2014) " Supervised Learning in Artificial Neural Networks ", International Journal of Advanced Research in Engineering & Technology (IJARET), Volume 5, Issue 3, pp. 208 - 215