A STUDY OF ACTIVATION FUNCTIONS IN NEURAL NETWORK VARIANT TECHNIQUE

OWOJORI O. TUNDE[1], OLABIYISI O. STEPHEN[1], ISMAILA W. OLADIMEJI[1], OMIDIORA E. OLUSAYO[2] ISMAILA FOLASADE. M. [3],

[1]Department of Computer Science, Ladoke Akintola University of Technology, Ogbomoso, Nigeria. [2]Department of Computer Science, Ladoke Akintola University of Technology, Ogbomoso, Nigeria. [3]Department of Computer Science Osun State Polytechnic, Iree, Nigeria.

Abstract

Activation functions are an extremely important part of the artificial neural networks that are used to calculate weighted and biases and also to generate the outputs of neural network. However, selecting an appropriate activation transfer for a classification problem has been task faced by researchers. This paper presented an approach for evaluating the various types of activation functions (sigmoidal, sine, hardlim, and trianglar) of Extreme learning machine (ELM). 997 data samples were collected from an open source on software defection. The dataset were pre-processed to make it suitable for classification. The parameters setting of the ELM network were given.with a fixed learning rate. Sigmoid function gave the highest accuracy of about 71%.

Keywords: Extreme learning machine, activation functions, sigmoidal, sine, hardlim, trianglar

Introduction

Machine Learning is one of ingridients of Artificial Intelligence paradigm which created systems such as computational intelligent, neural network, evolutionary algorithms, automatically learn and improve from experience without being explicitly programmed. Extreme Learning machines, ELM, is a timber move forward neural network model used for clustering, sparse approximation, compression and feature processing [1][2].
Extreme learning machine is a special single hidden layer feed-forward neural network, which was developed by Huang's group in 2004 [2]. The EML consists of layers which

include input layer, the hidden layer and output layer. The hidden layer abores the activation functions. A typical structure of ELM is shown in Figure 1.

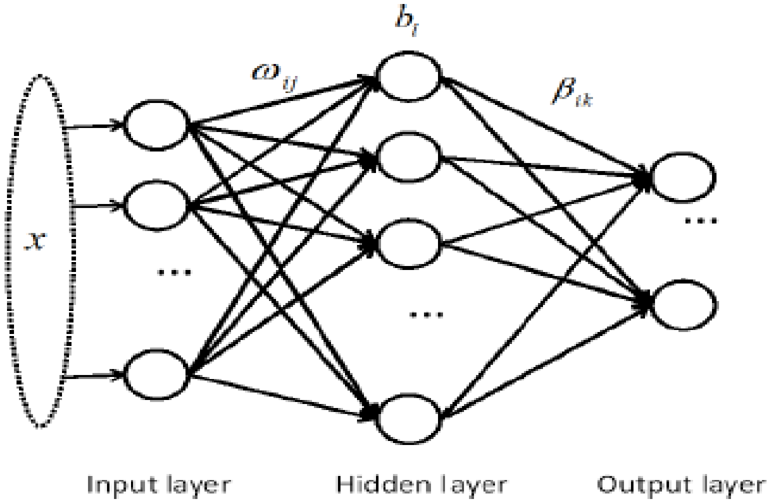

Figure 1. The structure of ELM. [3].

where w denotes the weights between the input layer and hidden layer, b denotes the threshold of hidden nodes, and β denotes the output weights. It is noted that there is an activation function in the hidden layer to perform nonlinear transformation [3]. A typical ELM has L hidden layer nodes, N instances, and an activation function $G(x)$, which can be modeled as:

$$\sum_{i=1}^{L}\beta_i G(\omega_{ij}\cdot x_j+b_i)=o_j, j=1,2,\ldots,N \quad (1)$$

where o_j is the predictive value of ELM model. According to the ELM theory, the optimization goal of ELM is

$$\min_{\|\beta\|}(\min\sum_{i=1}^{N}\| t_i-o_i\|^2), \quad (2)$$

where t is the actual value of each instance, and $||\bullet||$ is the matrix norm in Euclidean space.

Equation (1) and (2) can be rewritten as

$H\beta = T, \quad (3)$

$$H=\begin{pmatrix} G(\omega_1\cdot x_1+b_1) & \cdots & G(\omega_L\cdot x_1+b_L) \\ \vdots & \ddots & \vdots \\ G(\omega_1\cdot x_N+b_1) & \cdots & G(\omega_L\cdot x_N+b_L)\end{pmatrix}_{N\times L}, \beta=\begin{bmatrix}\beta_1^T \\ \vdots \\ \beta_L^T\end{bmatrix}_{L\times m}, T=\begin{bmatrix}t_1^T \\ \vdots \\ t_N^T\end{bmatrix}_{N\times m}$$

Where

H is the output matrix of the hidden layer, and the output weight β can be solved by

$$\beta = H \cdot T \quad (4)$$

where $H\cdot$ is the Moore–Penrose generalized inverse of H. In general, the weights, ω, are generated randomly within the range of [-1, 1] and the thresholds of hidden layer nodes, b , are generated randomly within the range of [0, 1] under a uniform sampling distribution. The output weight β is determined by the Moore-Penrose generalized inverse as shown in equation (4). The activation function maps the resulting values of a neuron in between (0,1) or (-1,1). The activation functions decide if the input information received should be passed or ignored. The activation functions used nowadays in deep neural networks are of three categories viz: monotonic, piece-wise monotonic and sinusoidal [4,5].

Table 1. Categories of Activation Functions [13]

S/No.	Category	Activation Functions	Characteristics
1	Monotonic	Binary Step, Sigmoid, Tanh, ReLU, Leaky ReLU, Hardlim . Swish, Tribasis	non-periodic, quasiconvex
2	piece-wise monotonic	parametric ReLU, Maxout, Hard Hiperbolic, ELSqi, Softmax	non-periodic, two monotonic segments
3	Sinusoidal	Sine, Cosine	periodic

While using random hidden node parameters in ELM brings a fast training speed and easy implementation advantages, using fixed activation function also leads to a deficiency in the ELM. Although ELM and its variants have been widely applied to numerous applications, to the best of our knowledge, there is no comparison on performance of these activation functions using software defection dataset.

Related works

Huang *et al*. [2,6] and authors in [7] did a comprehensive review on Extreme Learning Machine algorithm with current state of the theoretical research and practical advances. Various improvements of ELM that include accuracy, sparsity and stability, under given conditions were discussed. Authors in [8] proposed a modified ELM algorithm which properly selects the input weights and biases before training the output weights of feedforward neural networks with Sigmoidal activation function. The hidden layer output matrix maintains full column rank which improves the speed of training output weights. The experimental results showed good performance of improved ELM algorithm.

The researchers in [4] proposed an extreme learning machine with tunable activation function (TAFELM) learning algorithm, which determines its activation functions dynamically by means of the differential evolution algorithm based on the input data. The main objective is to overcome the problem dependence of fixed slop of the activation function in ELM. Compared with previous learning algorithms with the same network size or compact network configuration, the proposed algorithm has improved generalization performance with good accuracy. Mahmood *et al* [9] presented a new approach to node selection of an ELM based on a 1-norm support vector machine (SVM). In this method, the targets of SVM are derived using the mean or median of ELM training errors as a threshold for separating the training data, which are projected to SVM dimensions. An integrated architecture that exploits the sparseness in solution of SVM to prune out the inactive hidden nodes in ELM was presented.

In 2019, [10] investigated the purposefulness of AF in ELM. The works made use of Sigmoid, Sine, Tanh and Hardlim activation functions in ELM neural network. The effect of the activation functions were verified on different dataset. The test results showed that the learning rate of Sigmoid was highest with accuracy of 86%. The authors in [11] employed the ELM for classifying the active compounds according to its SMILES structure. This experiment uses eleven activation functions. The accuracy and computational time of classification model were depending on the activation function. Based on experimental results, the average maximum accuracy 88.73% on TanHRe.

Materials

This section discusses the materials used in this work which include:

A. Algorithm 1: Extreme Learning Machine

Given a training set with N distinct samples,

$\mathbf{x} = \{(\mathbf{x}_i, \mathbf{y}_i) \mid (\mathbf{x}_i \in \boldsymbol{R}^n, \mathbf{y}_i \in \boldsymbol{R}^m,\ i = 1, 2, \ldots, N\}$, *activation function* $\varphi(x)$, *and the number of hidden neurons L*

1. Randomly generate the input weight vector $\mathbf{w}i$ *and the bias bi,* $i = 1, 2, \ldots, L$.

2. Compute the feature mapping matrix H for given input weights and biases.

$$H(w,x,b) = \begin{bmatrix} \varphi(w_1 \cdot x_1 + b_1) & \varphi(w_2 \cdot x_1 + b_2) & \dots & \varphi(w_L \cdot x_1 + b_L) \\ \varphi(w_1 \cdot x_2 + b_1) & \varphi(w_2 \cdot x_2 + b_2) & \dots & \varphi(w_L \cdot x_2 + b_L) \\ \vdots & \ddots & & \vdots \\ \varphi(w_1 \cdot x_N + b_1) & \varphi(w_2 \cdot x_N + b_2) & \dots & \varphi(w_L \cdot x_N + b_L) \end{bmatrix} = \begin{bmatrix} h(x_1) \\ h(x_2) \\ \vdots \\ h(x_N) \end{bmatrix} = h(x)$$

3. Compute the output weight matrix (vector) $\dot{B}$

$$\dot{B} = H^{+}Y$$

where $\boldsymbol{Y} = [y_1, y_2, \ldots, y_N]^T$

B. EML Activation Functions

i. Sigmoid Activation Function.

The sigmoid function is used for the two-class logistic regression, A sigmoid function is a bounded, differentiable, real function that is defined for all real input values and has a non-negative derivative at each point and exactly one inflection point. Mathematically it can be represented as:

$$f(x) = \frac{1}{1+e^{-1}} \qquad (5)$$

Sigmoid activation function is used for models predict the probability as an output but its output is not symmetric around zero [9].

ii. Sines Function

Sines function is a sinusoidal activation function in neural networks. In contrast to other common activation functions, it has rises and falls. However, the function saturated and its output converges to zero for large positive and negative inputs [12]. The sine function is defined as

$$\sin = \begin{cases} 0, & \text{if } -\pi/2 < x \\ \sin(x), & \text{if } -\pi/2 \le x \le \pi/2 \\ 1, & \text{if } x > \pi/2 \end{cases} \qquad (6)$$

iii. Hard limit function

The hard limit function is essentially a transfer function that allows the output neuron to produce a 1 if the input attains a threshold, otherwise, it outputs a 0. It calculates the output of a layers based on its input. Transfer functions calculate a layer's output from its net input [9]

iv. Triangular basis Activation

This function computes a layer's output in its net input are known as triangle basis functions. In the form of a triangle, it is a mathematical function is defined as follows

$$y = \begin{cases} 1 - |x| & if -1 \leq x \leq 1 \\ 0 & otherwise \end{cases} \quad (7)$$

C. Software Defect Dataset

The software dataset consists of features that have values and a goal field that describe the instance as defect and non-defect. These features describe the program from different sides including the McCabe cyclomatic complexity (which include McCabe's line count of code, McCabe "cyclomatic complexity", McCabe "essential complexity", McCabe "design complexity") and Halstead software metrics. Halstead's indices includes; number of operators, number of operands, sum of operators and sum of operands. For example, the expression "return max(w+x,x+y)" has (i) Number of operators (return, max, +,+) = 4; number of operands (w,x,x,y) = 4; number of unique operators (return, max,+) =3; and number of unique operands (w,x,y) =3.

D. *Performance metrics*

The performance metrics used to evaluate the performance of ELM for the software defect prediction with different activation fucntions were accuracy, precision, recall and F-measure. These metrics were estimated from the confusion matrix shown in table 1.

Table 1: Confusion matrix

		Actual Labels	
		Defect	Non-Defect
Predicted labels	Defect	TruePos	FalsePos
	Non-defect	FalseNeg	TrueNeg

Accuracy is the average of the sum of correctly classified detected and undefected instances.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (8)$$

Precision: The ratio of the correctly classified defected instances among the retrieved instances. It can be calculated by Equation (9):

$$Precision = \frac{TruePos}{TruePos + FalsePos} \quad (9)$$

Recall: The ratio of correctly classified defected instances, as in Equation (10):

$$Recall = \frac{TruePos}{TruePos + FalseNeg} \qquad (10)$$

F-measure is the compound measure of precision and recall, as shown in equation (11).

$$\text{F-measure} = \frac{2 \times (precision \times recall)}{precision + recall} \qquad (11)$$

Methodology

A software defect prediction dataset used was ECLIPSE JDT CORE acquired from http://bug.inf.usi.ch/download php. The dataset was preprocessed using principal component analysis for feature extraction. And then, the dataset were normalized using the min-max approach to a range between 0 and 1. And finally, the extracted features were classified using EML with different activation functions. The performance metrics used to evaluate the performance of different ELM activation functions for the software defect prediction were accuracy, precision, recall and F-measure.

System Implementation

The simulations were performed in MATLAB 7.11.0.584 -64-bit, running on Intel Core i5-2430M, 2.4GHz with 4GB of RAM. The system interface is shown in figure 2. The parameters setting for the EML activation functions (sigmoidal, sine, hardlim, and trianglar) is shown in table 2.

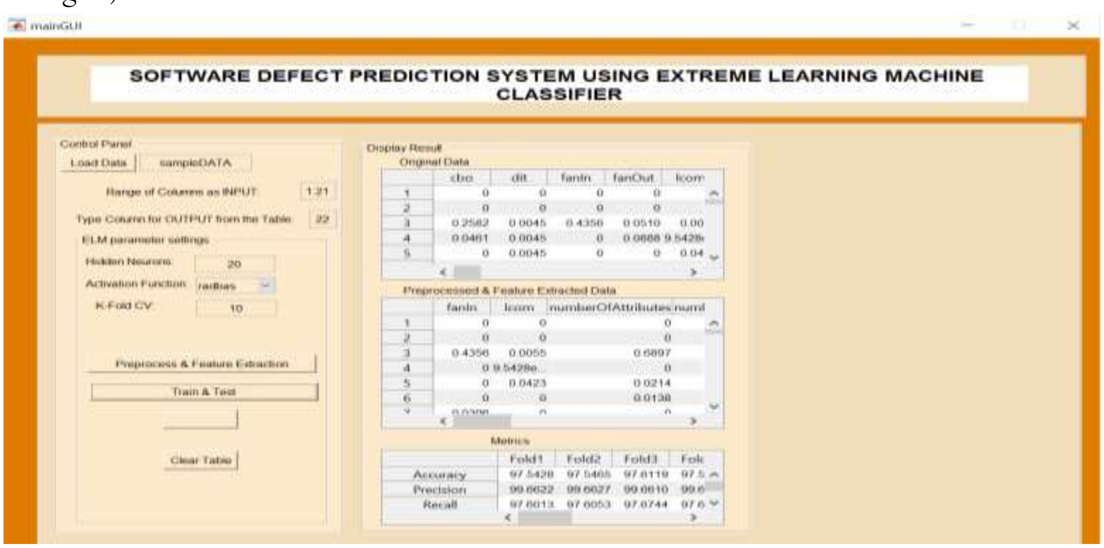

Figure 2: The Software defect system Interface

The dataset was validated using a K-fold cross-validation strategy in which all datasets participated in both the training and testing phases, with K equal to 5.

Parameter Name	**Value**
No of Input Samples	997
No of Inputs/features	22 (5 LoC, 3 McCabe, 4 Halsteads, 8 derived Halsteads, 1- branch-count, and 1 goal field)
Hidden neurons (fixed)	20
No of Outputs Neurons	1
Weights	0.85
Thresholds	0.75
Learning rates	0.01

Results and Discussion

The results of the simulation of the software defect system using EML with different activation fucntions using accuracy, precision, recall, and the F-measure as performance metrics are shown in Table 2.

Table 2: Accuracy prediction, recall, and F-measure of the AFs classifications

Activation function	**Accuracy**	**Precision**	**Recall**	**F-measure**
Sigmoidal	80.25	79.36	78.48	78.92
Sine	72.14	70.48	67.72	69.14
Hardlim	75.46	71.05	68.47	69.75
Trianglar	73.95	70.42	70.49	70.48
Softmax	79.15	77.43	78.50	77.96

The results of experimentations of ELM with the different activation gave average values of accuracy, precision, recall and F-measure of 71.25%, 76.36%, 71.48% and 73.85% respectively for functions sigmoidal function; average values of accuracy, precision, recall and F-measure for sine function were 69.14%, 70.48%, 67.72% and 69.14% respectively; average values usng hardlim function for accuracy, precision, recall and F-measure were 70.46%, 71.05%, 68.47% and 69.75% respectively; and average values using triangular function for accuracy, precision, recall and F-measure were 69.95%, 70.42%, 70.49% and 70.48 % respectively.

Conclusion

This paper presented an approach for evaluating the various types of activation functions of ELM. In practical applications, the selection of activation functions (sigmoidal function, sine function, hardlim fun3ction, and trianglar function) of ELM has been considered crucial. The activation functions embedded inside ELM were subjected to implement a linear problem, software defection. The dataset were pre-processed to make it suitable for classification and principal component analysis was used for feature extraction.

997 data samples were collected from an open source. The dataset consists of 22 inputs/attributes with one output. The parameters setting of the ELM network were given.with a fixed learning rate. Accuracy, precision, recall, and the F-measure were the performance metrics used. Sigmoid function gave the highest accuracy of about 71%. However, there is need to establish the quantitative relationship between the parameters and the performance in the multi-layer and regularization method in ELMs.

References

Kishan Maladkar (2018). Types of Activation Functions in Neural Networks and Rationale Behind It.

Huang G.B, Zhu Q.Y and Siew C.K (2004) Extreme learning machine a new learning scheme of feedforward neural networks, *Proceedings of 2004 IEEE International Joint Conference onNeural Networks*, IEEE, Vol. 2, pp. 985-990.

Weipeng C., Jinzhu G., Zhong M., and Shubin C. (2017) Some Tricks in Parameter Selection for Extreme Learning Machine, IOP Conf. Series: Materials Science and Engineering **261** (2017) 012002

Li B., Li Y., Rong X. (2013).The extreme learning machine learning algorithm with tunable activation function, Neural Comput & Applic (2013) 22:531–539.

Sevinç E. (2018). Activation Functions In Single Hidden Layer Feed-Forward Neural Networks, *Journal Of Selcuk-Technic Special Issue 2018 (Icente'18),Pg 1-13.*

Huang S. G.,.Huang G.B,.You K, (2015). Trends in extreme learning machines: A review, Neural Networks 61, 32–48.

Albadr M. A. and Tiun S. (2017). Extreme Learning Machine: A Review, International Journal of Applied Engineering Research ISSN 0973-4562 Vol. 12, Number 14, 4610-4623

Chen Z., Zhu H. and Wang Y. (2011). A Modified Extreme Learning Machine, *Submitted to ELM2011.*

Mahmood S.F., Marhaban M. H., Rokhani F Z., Samsudin K and Arigbabu O A (2016). SVM–ELM: Pruning of Extreme Learning Machine with Support Vector Machines for Regression, J. Intell. Syst. 2016; 25(4): 555–566.

Kumar R S. and G. Apparao. (2019). Extreme Learning Machine with sigmoid activation function on large data, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-2S11.

Ratnawati D., Marjono ,Widodo, Anam S. (2020). Comparison of Activation Function on Extreme Learning Machine (ELM) Performance for Classifying the Active Compound, AIP Conference Proceedings 2264, 140001;

Parascandolo G., Huttunen H. And Virtanen T. (2017).Taming The Waves: Sine as Activation Function In Deep Neural Networks, Conference Paper at ICLR.

https://analyticsindiamag.com/most-common-activation-functions-in-neural-networks-and-rationale-behind-it/