Extreme Learning Machine based on Rectified Nonlinear Units

Jingtao PENG [a], Liang CHEN [b], Iqbal Muhammad Ather, Ao YU

College of Information Science and Technology, Donghua University, Shanghai 201620, China

[a]email: 1445964736@qq.com , [b]email: chenliang@dhu.edu.cn

Keywords: Extreme Learning machine; Over-saturation; Rectified Linear Units; Rectified Non-Linear Units

Abstract. Traditional Extreme Learning Machine (ELM) networks generally used S-shaped activation function, such as Sigmoid function and Tangent function. However, the problems of slow convergence speed and over-saturation exist. In order to solve the above problems and improve the performance of ELM algorithm, the method of Rectified Non-Linear Units (ReNLUs), combining rectified linear units (ReLUs) with Softplus function method, was proposed. And the ReLUs has the ability of sparse expression and the Softplus possesses smooth and unsaturated features. Experimental results show that the ELM with the method of ReNLUs activation function, the accuracy and time of training and testing have been significantly improved and saved.

1. Introduction

In recent years, artificial neural networks have been a commonly method used in the field of machine learning for the ability of self-learning, self-organizing and adaptive [1]. For its simple structure and the capability of approaching complicated nonlinear functions, Single Hidden Layer Feed forward Neural Network (SLFN) is paid more attention in the research of neural network. However, traditional SLFN training method is generally based on gradient descent algorithm, which is easy to obtain low-rate convergence, trap into local minima value and achieve optimization with a lot of iteration, so that the use of SLFN has been restricted. In order to solve the above problems, Huang etc. [2] proposed a novel SLFN algorithm--Extreme Learning Machine (ELM). The input weights (linking the input layer to the hidden layer) and hidden layer bias of ELM are randomly assigned, and the output weights (linking the hidden layer to the input layer) are calculated by the least square method, and the iteration of all parameters is not needed. Compared with the Back Propagation (BP) neural network, ELM is better in learning speed and generalization performance [3] [4].

The convergence speed and learning accuracy are affected by the activation function in neural network, thus the activation function plays an important role in neural network. Although traditional ELM has overcome some defects of low convergence rate, multi-iterations and local minima, there are still some problems that the neural network will be slower convergence speed and lead to non-convergence owing to over-saturated and gradient diffusion.

The general activation functions include Tanh, Sigmoid and other saturated nonlinear functions. The use of these functions can effectively avoid the insufficient expression ability of neural network caused by linear mapping, but the problems of slower convergence speed and lower learning accuracy are still not solved owing to over-saturation, especially, the negative value in Tanh function will make the application performance poor [5]. ReLUs function has the characteristics of sparse expression and linear unsaturation [6], but it is linearly correct, and its expression ability is weak. Softplus function has slow convergence speed and no ability of sparse expression, but it has the advantage of nonlinear unsaturation, and Softplus function is the smooth approximation expression of ReLUs function [7], so we propose ReNLUs. And ReNLUs can speed up the convergence rate on the basis of the original ELM algorithm and enhance the generalization ability without over-saturation. Finally the simulations used validate our results.

The remainder of the paper is organized as follows. In section 2, we briefly describe the ELM algorithm. In section 3, we present all kinds of activation function and its properties. ReLUs

activation function in section 3.1, Softplus activation function in section 3.2 and our improved activation function in section 3.3. In section 4, we report on computational results and compare them with original ELM algorithm. In section 5, we conclude with a discussion of our findings.

2. ELM

Given N samples $[x_1, x_2, \cdots, x_N]$ and their corresponding ground truth $[t_1, t_2, \cdots, t_N]$, where $x_i = [x_{i1}, x_{i2}, \cdots, x_{iN}]^T \epsilon R^n$ and $t_i = [t_{i1}, t_{i2}, \cdots, t_{im}]^T \epsilon R^m$, n and m denote the number of input and output neurons, respectively. The output weights between the hidden layer and the output layer being learned are denoted as $\beta \epsilon R^{L \times m}$. A model of SLFN with L hidden nodes is described as follows:

$$f_L(x_j) = \sum_{i=1}^{L} \beta_i g(\alpha_i \bullet x_j + b_i), \quad j = 1, 2, \cdots, N \tag{1}$$

where $\alpha_i = (\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{in})$ is input-weight connecting with the ith hidden node; $b_i \in R$ is the bias of the ith hidden node; $\beta_i = (\beta_{i1}, \beta_{i2}, \cdots, \beta_{im})$ is the output weight connecting with the ith hidden node; $\alpha_i \cdot x_j$ is the inner product of α_i and x_j; L is the number of hidden nodes, $g(\cdot)$ is the activation function (e.g., Sigmoid function and Tanh function). Approximating the samples with zero error means the proper selection of β_i, α_i and b_i such that

$$\left\| f_L(x_j) - t_j \right\| = 0 \quad or \quad f_L(x_j) = t_j, \quad (j = 1, 2, \cdots, N) \tag{2}$$

that is,

$$\mathrm{H}\beta = T \tag{3}$$

$$H = \begin{pmatrix} h(x_1) \\ h(x_2) \\ \vdots \\ h(x_n) \end{pmatrix} = \begin{pmatrix} G(\alpha_1, b_1, x_1) & \cdots & G(\alpha_L, b_L, x_1) \\ G(\alpha_1, b_1, x_2) & \cdots & G(\alpha_L, b_L, x_2) \\ G(\alpha_1, b_1, x_N) & \cdots & G(\alpha_L, b_L, x_N) \end{pmatrix}_{N \times L}, \beta = \begin{pmatrix} \beta_1^T \\ \beta_2^T \\ \vdots \\ \beta_L^T \end{pmatrix}_{L \times m}, T = \begin{pmatrix} t_1^T \\ t_2^T \\ \vdots \\ t_N^T \end{pmatrix}_{N \times m} \tag{4}$$

where H is denoted by the hidden layer output matrix. Let $E(W)$ be the square of error between the expected output value and the actual output value. In SLFN frame work, the problem is to find the optimal weight α, b and β, which makes the cost function $E(W)$ least. The mathematical expression is described as:

$$\begin{aligned} &\arg \min_{W=(\alpha,b,\beta)} E(W) = \arg \min_{W=(\alpha,b,\beta)} \frac{1}{N} \sum_{j=1}^{N} \frac{1}{2} \left\| \varepsilon_j \right\|_2^2, \\ &\text{s.t.} \sum_{i=1}^{L} \beta_i g\left(\alpha_i \cdot x_j + b_i\right) - t_j = \varepsilon_j, \quad j = 1, 2, \ldots, N. \end{aligned} \tag{5}$$

where $\varepsilon_j = [\varepsilon_{j1}, \varepsilon_{j2}, \cdots, \varepsilon_{jm}]$ is the error of the jth sample.

Huang etc. have proved that if (α, b) are randomly preselected, the hidden layer output matrix H must be determined. Then, the formula (5) is seen to be a standard least squares problem of $H\beta = T$ and its solution can be explicitly given by

$$\hat{\beta} = \mathrm{H}^{+} T \tag{6}$$

where $H^{\dagger}$ is a Moore-Penrose generalized inverse of H. We direct the interested readers to [2] for more details on ELM theory and the algorithms.

3. Activation function and its properties

In traditional ELM network model, there are some general activation functions that are over-saturation and non-linear, such as Sigmoid and Tanh function. General equation as follows:

$$\begin{cases} Sigmoid: g(x) = \dfrac{1}{1+\exp(-x)} \\ \text{Tanh}: g(x) = \dfrac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)} \end{cases} \tag{7}$$

The graph [Figure.1] is obtained by equation (7). The function will be saturated when the value of the function g (x) is near its critical value, and seriously affect the convergence speed of neural networks, which gradually attracted the attention of domestic and foreign researchers in recent years [8].

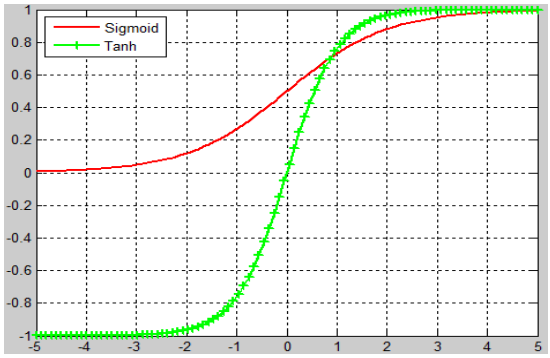

Fig.1. Non-linear saturation function

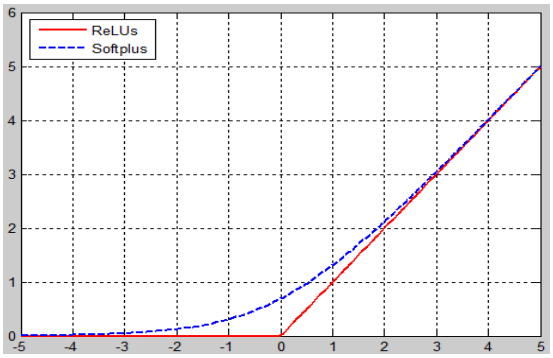

Fig.2. ReLUs and Softplus function

3.1 ReLUs activation function

In recent years, ReLUs function is widely used in various neural networks [6] [8-10], and gradually replaced Sigmoid activation function. ReLUs activation function is defined as:

$$g(x) = \max(0, x) \tag{8}$$

In the type (8): if $x < 0$, then g(x) function value is zero, if $x > 0$, then the function value is x. In other words, it is a piecewise linear function which prunes the negative part to zero, and retains the positive part. The ReLUs activation function allows a network to easily obtain sparse representations. For example, after uniform initialization of the weights, around 50% of hidden units continuous output values are real zeros and this fraction can easily increase with sparsity-inducing regularization. Apart from being more biologically plausible, sparsity also leads to mathematical advantages. Because of this linear, gradients flow well on the active paths of neurons (there is a gradient vanishing effect due to activation over-saturated of Sigmoid or Tanh), and mathematical investigation is easier. Computations are also cheaper: there is no need for computing the exponential function in activations, and sparse can be exploited [10].

Although ReLUs function has many of the above advantages, but due to the use of a linear mapping, reducing the expression of the network ability. Studies have shown that nonlinear mapping is closer to the biology of the active model, and the expression of the network ability will be significantly enhanced. Thus, a smooth version of the rectified nonlinear function is put out, and it is Softplus activation function.

3.2 Softplus activation function

Softplus function is a nonlinear mapping of all the data and an unsaturated. Softplus activation function is defined as:

$$g(x) = \log(1+\exp(x)) \tag{9}$$

Softplus function [Figure.2] is the smooth approximation expression of ReLUs function. Softplus function does not have the ability of sparse expression, and the convergence speed is much slower than ReLUs function. But this function is closer to the biological characteristics of activation than the ReLUs function, while it solves the false saturation phenomenon of Sigmoid function, and improve the learning accuracy and generalization performance of the network, but ELM network learning speed becomes slower.

3.3 Improved activation function

Since ReLUs function has the ability to sparse expression, and Softplus function has a smooth nonlinear and unsaturated features. Based on the advantage of ReLUs and Softplus, by combining ReLUs' sparsity with Softplus' smoothness, a new method of rectified nonlinear units function is proposed, so as to optimize ELM neural network. In order to enhance the activation function of the model controllable, on the Softplus function before adding a parameter a, you can adjust the convergence speed and learning accuracy network through the control parameter a (a = 0.7-0.8).

ReNLUs function can be further optimized the learning accuracy, convergence speed and improve the generalization performance of neural networks, based on the original ELM algorithm. The activation function is continuous and smooth when x is greater than zero, and its derivative shows increasing trend. Thus, ReNLUs activation function can make it easier convergence for network model. ReNLUs activation function is defined as:

$$g(x) = \max(0, a \bullet \log(\frac{1+\exp(x)}{2})) \tag{10}$$

Figure3 is ReNLUs activation function graph, where ReNLUs0 is denoted as a=1, ReNLUs1 is denoted as 0<a<1; ReNLUs2 is denoted as a>1.So you can set a different value depending on the issue, to adjust the performance of the network to make it the best.

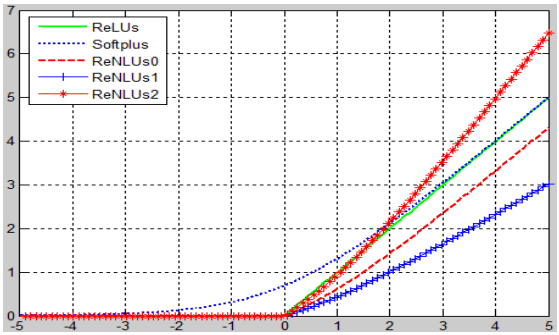

Fig.3. ReNLUs function

4. Test results

The convergence speed of traditional ELM algorithm with S-shaped function is too slow and easy to be over-saturation; our paper applies an improved ReNLUs activation function. On Intel Corei5 3.20GHz, memory 4.00G, 32-bit operating system, matlab R2012a, we carry out numerical simulation. And different activation function was performed with ELM on MNIST database. In addition, the hidden layer nodes as 50 and measuring 50 times was used to obtain averaged results as shown in Table 1.

Tab.1 At each activation function of performance indicators ELM

Performance / Activation function	training accuracy	testing accuracy	training time	testing time	Standard Deviation of Training Accuracy	Standard Deviation of Testing Accuracy
Tanh	0.9333	0.9278	0.1214	0.00718	0.0086	0.0068
Sigmoid	0.9374	0.9352	0.0853	0.0291	0.0077	0.0072
ReLUs	0.9439	0.9404	**0.0614**	**0.0083**	0.0066	0.0072
Softplus	0.9472	0.9426	0.0894	0.0406	0.0055	0.0066
ReNLUs	**0.9481**	**0.9473**	0.078	0.0254	**0.0048**	**0.0059**

By comparison, it is easy to find that ReNLUs ELM algorithm proposed in our paper obtain better results in terms of training and testing accuracy than the results of other ELM activation function, and up to 2% of training and testing accuracy is improved compared with the traditional ELM algorithm. For training and testing time, the training time of ReLUs and ReNLUs is similar, but the testing time is longer than ReLUs'. Finally, as we know that the standard deviation of training and testing accuracy are smaller and the network is more stable. Contrast to the standard deviation of training and testing accuracy, we find that the ELM with the ReNLUs activation

function is most stable.

In order to more efficiently and intuitively reflect the superiority of the algorithm in our paper. The different hidden layer node is measured for 50 times to obtain the average results. The average training and testing accuracy are shown in Figure.4-5 and the average training and testing time that are shown in Figure.6-7. it is clear that the algorithm proposed in our paper are better than traditional activation function in the training and testing accuracy, which is also slightly better than traditional ELM algorithms on training and testing time.

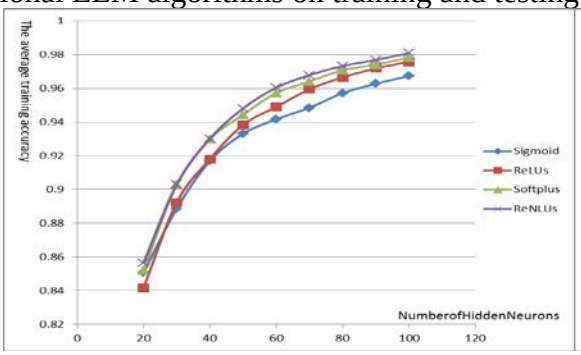

Fig.4.The average training accuracy

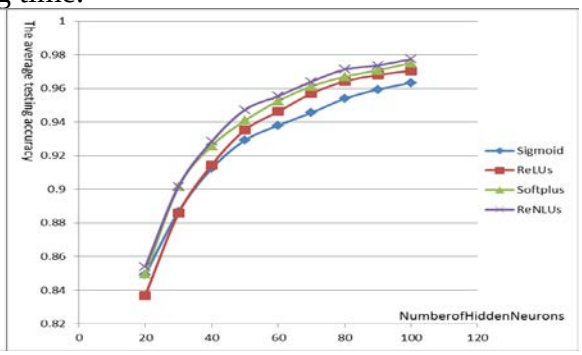

Fig.5.The average testing accuracy

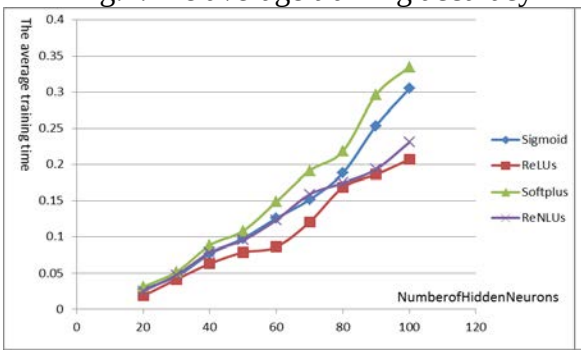

Fig.6.The average training time

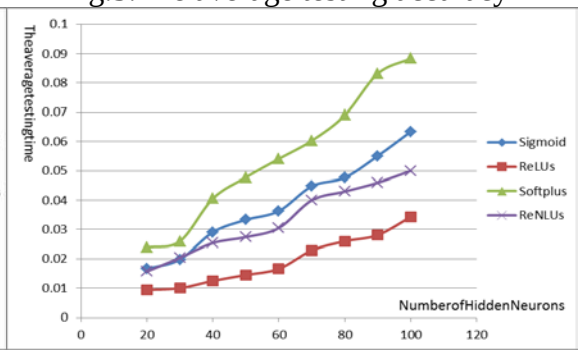

Fig.7.The average testing time

To better illustrate the original ELM and improve ELM algorithm results, 50 times trials is performed respectively on the UCI, ORL and AR database, then the average value of simulations are shown in table 2. The experimental results show that our algorithm both on MNIST database and other databases has significantly improvement in recognition performance.

Tab.2 ELM and the improved algorithm on different databases recognition rate

Database	Performance \ Activation function	Tanh	Sigmoid	ReLUs	Softplus	ReNLUs
UCI	training accuracy	0.7726	0.7725	0.7813	0.7899	0.7917
	testing accuracy	0.7708	0.7656	0.7708	0.776	0.7813
ORL	training accuracy	0.9353	0.9582	0.9625	0.9725	0.9886
	testing accuracy	0.8342	0.8738	0.8802	0.9076	0.9254
AR	training accuracy	0.9542	0.9632	0.9744	0.9802	0.9903
	testing accuracy	0.8797	0.9358	0.9432	0.9536	0.9686

5. Conclusion

In this paper, we try to some problems, such as slow convergence speed, easily being over-saturated in traditional ELM, boosting learning speed and improving learning accuracy. We proposed a ReNLUs ELM algorithm and make effective improvements, which combined the advantages of sparse expression of ReLUs with nonlinear unsaturation of Softplus function. Compared with previous ELM algorithm, the experimental results show the advantages of this novel algorithm: 1) The training accuracy and testing accuracy have increased significantly; 2) Generalization performance is better, and the network is more steady; 3) The time of training and testing is reduced. Compared with BP algorithm, ELM algorithm has obvious advantage on learning speed and accuracy. In addition, compared with previous ELM algorithm, the ReNLUs ELM

algorithm in this paper has better performance.

Then we will focus on the structure of ELM network and try to use dropout algorithm to optimize ELM and achieve the purpose of further sparse ELM. Researchers show that dropout model is more conformable to characteristics of biological neurons. When weights are updating, hidden layer nodes is activated at some probability, so that the updating of weights don't rely on the hidden layer nodes, and this prevents the situation that some characteristics only perform based on other characteristics. In addition, the hidden layer parameters are randomly assigned, which will affect the generalization performance of networks and are all worthy of further research.

Acknowledgement

In this paper, the research was sponsored by the Nature Science Foundation of Shanghai City (Project No. 14ZR1400700).

References

[1] S GE, C HANG, T H LEE, et al. Adaptive neural network control [M]. Berlin: Spring Publish Company, Incorporated, 2010.

[2] G B Huang, Q Y Zhu, C K Siew. Extreme learning machine: a new learning scheme of feed-forward neural network[c].IEEE International Joint Conference on Neural Network. Budapest: IEEE, 2004:985-990.

[3] A LENDASSE, Q HE, Y MICHE, et al. Advances in extreme learning machines [J]. Neurocomputing, 2014, 128(5): 1-3.

[4] M Han, X Liu. An extreme learning machine algorithm based on mutual information variable selection [J]. Control and Decision, 2014, 29(9): 1576-1580.

[5] X GLOROT, Y BENGIO. Understanding the difficulty of training deep feed forward neural networks. Proceeding of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) ， Sardinia: Microtome Publishing, 2010: 249-256.

[6] A KRIZHEVSKY, I SUTSKEVER, G E HINTON. Image net classification with deep convolutional neural networks[C].Advances in Neural Information Processing Systems. Red Hook, NY: Curran Associates, 2012:1097-1105.

[7] X GLOROT, A BORDES, Y BENGIO. Deep sparse networks with random hidden nodes [J]. Journal of Machine Learning Research, 2011, 15: 315-323.

[8] V NAIR, G E HINTON. Rectified linear units improve restricted Boltzmann Machines[C]. Proceedings of the 27th International Conference on Machine Learning (ICML-10).Madison, WI: Omni press, 2010:807-814.

[9] G E DAHL, T N SAINATH, G E HINTON. Improving deep neural networks for LVCSR using rectified linear units and dropout[C].Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. Piscataway, NJ: IEEE, 2013:8609-8613.

[10] X GLOROT, A BORDES, Y BENGIO. Deep sparse rectifier networks[C]. JMLR Workshop and Conference Proceedings Volume 15: AISTATS 2011.Brookline, MA: Microtome Publishing, 2011:315-32.