Advances and Applications in Mathematical Sciences
Volume 21, Issue 8, June 2022, Pages 4713-4731
© 2022 Mili Publications, India

IS EXPONENTIAL ACTIVATION FUNCTION ALWAYS BETTER THAN THE NON-EXPONENTIAL ACTIVATION FUNCTION

ARVIND KUMAR and SARTAJ SINGH SODHI

Computer SC and Engineering
GGSIPU Delhi
E-mail: arvind.usict.134164@ipu.ac.in

Abstract

Now-a-day, neural networks are used in many areas such that data or image classification, recognition, and many more other fields. To better achieve the results, we use activation functions (AF) in the neural network. We don't directly apply any AF to the network. The right choice of AF is most necessary for achieving better accuracy. The main aim of this research is to find out if the exponential activation function (AF) is always better than the non-exponential AF. Therefore we have taken elliotsig (non-exponential AF) and compared it with tansig, logsig, and softmax (exponential AFs) with the help of two datasets. We have also presented the limitations and advantages of these AFs. These four AFs are generally used in many types of neural network such as a pattern network. For this analysis, we have made a pattern network. In this network, we have taken a tansig, logsig, elliotsig, or softmax activation function in the hidden layer and a softmax activation function in the output layer. With the help of this network, we have found some advantages of non-exponential AF over exponential AFs. The first advantage is that at the lowest hidden neuron size elliotsig (non-exponential) AF may achieve a better result compared to the tansig and logsig activation function. The second advantage is that when we consider only accuracy parameters with respect to epochs, then also elliotsigmay achieve better results. The third softmax activation function is not acceptable in the hidden layer. We have tested these results on the iris dataset and the cifar-10 dataset. In the case of the iris dataset, elliotsig has achieved100% accuracy at the lowest hidden neuron size (21 hidden neuron size) and 30 epochs. In the case of cifar-10 dataset, elliotsig has achieved 90.3% accuracy at 101 hidden neuron size and 26 epochs. Thus we can say that in some cases non-exponential AFs show better results. But, the major drawback of non-exponential (elliotsig) AF is that it is less useful for the very complex and big datasets

2020 Mathematics Subject Classification: 68T01.

Keywords: Pattern recognition, tansig, logsig, softmax, elliotsig, neural network.

Received October 12, 2021; Accepted January 1, 2022

1. Introduction

Clustering, classification, image recognition, etc. are major areas of Artificial Neural Network (ANN). ANN is inspired by the biological human brain [1, 2]. There are mainly three layers of ANN. They are an input layer, an output layer, and one or more hidden layers. The learning process is an important part of training a network. We train the network with the help of two manners. They are supervised learning, and Unsupervised learning. In the case of the supervised learning process, all the targets are given for recognition and classification after then we train the network. But in the case of the unsupervised learning process, we make only clusters because desired output has not been given. In this paper, we have taken a Pattern Network, which is a type of supervised learning process. The Pattern Network is used for the recognition and classification of images or data.

In the field of the neural network, the Single-layer neural network was given by Rosenblatt (1958). But this network is a limited function such as to the classification of linear patterns only. Then Widrow and Hoff's (1960) had given ADALINE and LMS algorithms. But this algorithm is based on a Single Linear Neuron with adjustable weights, which limits the computing power of the algorithm. To remove the limitations of the perceptron and the LMS algorithm another network comes that is called Multilayer Perceptron, which is given by Minsky and Papert (1996). Of course, multi-layer neural networks are more powerful than Single-layer networks [3]. Training of multilayer perceptron is known as the back-propagation algorithm, which also includes the special case of LMS (provided by Widrow and Hoff). With the help of the back propagation technique, Mrs. A. Sekhon and P. Agarwal achieved 86% accuracy in 28462 iterations [4]. V. Bihday, V. Brygilevych, Y. Hychka, Z. Liubun, N. Pelypets, and V. Rabyk [5] recognized handwritten images using Multilayer Neural Network. The Multilayer Neural Network has also been used for denoising the image [6]. R. Achkar, M. Owayjan, B. Harajli, D. Khazaal, M. Dbouk, and G. Magnifico [7] classify Brain tumors using Back Propagation Algorithm in Multilayer Perceptron (MLP). W. D. N. Pacheco, and F. R. J. López used the K-NN algorithm with the MLP neural network for tomato color classification and got good performance [8]. Some authors used Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), and Multilayer Perceptron Neural Network (MLPNN) for the classification of images. H. I. Dino and M. B. Abdulrazzaq got a recognition rate of 93.53% with SVM, 82.97% with MLP classifier, and 79.97% with KNN classifier [9]. With the help of pattern net, A. Yousaf et al. [10], trained 19422, English alphabets samples and 7720 digits samples that are written through 150 different writers in various styles of handwriting and achieves 95.69% precision without using conventional feature extraction techniques. A. Panyafong, N. Neamsorn, and C. Chaichana developed ANN with the help of tansig and logsig for heat load calculation [11]. T. Hossain, F. S. Shishir, M. Ashraf, M. A. Al Nasim, and F. M. Shah classify the brain using Multilayer Perceptron(MLP)with the help of tansig [12]. Some other applications of ANN are: Early Prediction of Breast Cancer [13], leakage detection and water loss management of urban water supply network [14], Classification of Agricultural Leaf Images [15], Classification of Tobacco Leaf Pests [16], Image compression with VGG16 [17], Garbage Recognition and classification [18], Facial Emotion Recognition [19], Human Action Recognition [20], Prediction of chloride diffusivity in concrete [21], improve salient object detection [22], prediction of covid-19 patient [23, 24, 25], stock price pattern classification and prediction [26, 27, 28], and network traffic classification [29].

2. Methodology

Figure 1(one) is an example of the Pattern Network. Pattern network is also called Multi-layer Feed forward Neural Network (MLFFNN). In this figure number of inputs is 4. Suppose y is input and initial weight is w and b is bias, then local induced output $(v_j(n))$ at j^{th} neuron is found by equation (1).

$$v_j(n) = \sum_{j=0}^{m} w_j y_j + b \tag{1}$$

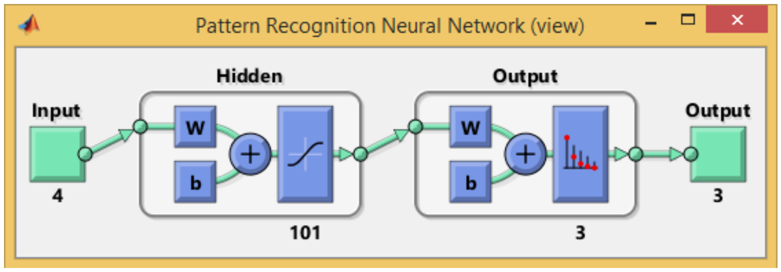

Figure 1. Sample of the pattern neural network.

This network is made with the help of the iris dataset. In this network, the numbers of inputs are 4. Tansig function is used in the hidden layer and softmax function is used in the output layer. The numbers of outputs are 3.

If activation function is $\varphi(\cdot)$, then output is found by equation (2).

$$y_j(n) = \varphi_j(v_j(n)) \tag{2}$$

Activation functions are classified into two parts. They are Linear function and Non-linear function. The linear function is provided by McCulloch and Pitts in 1943 [1, 2, 26]. McCulloch and Pitts provided logical function. The linear activation function is mostly used for linear separability types of problems. For example, in the Hard limit function if input $(n < 0(\text{zero}))$ then output (a) is $0(\text{zero})$ otherwise output (a) is 1(one). In Symmetrical Hard limit function if input $(n < 0(\text{zero}))$ then output (a) is -1(neg. one) otherwise output (a) is 1(one). All the linear function is not more useful for non-separability problems. For non-separability problems, we use logsigmoid, tansigmoid, Elliot symmetric sigmoid transfer function (elliotsig), softmax, and some other non-linear activation functions.

In logsigmoid function, if n is input then output (a) is found by equation (3).

$$a = \frac{1}{(1 + e^{-n})} \tag{3}$$

In tansig function, if n is input then output (a) is found by equation (4).

$$a = \frac{e^n - e^{-n}}{(e^n + e^{-n})} \tag{4}$$

In elliotsig transfer function, if n is input then output (a) is found by equation (5).

$$a = \frac{n}{(1 + |n|)} \tag{5}$$

In softmax function, if n is input then output (a) is found by equation (6).

$$a = \frac{\exp(n)}{sum(\exp(n))} \tag{6}$$

Here, soft means softmax is continuous and differentiable [27]. Softmax function is now mostly used for the output of the classifier [27].

After completing the design of the networks, we train this network. There are two phases of training [2]:

A. Forward phase: In this phase layer-by-layer, the synaptic weights of the network are calculated, and the input signal is propagated through the network until we reach the outputs.

B. Backward phase: Error is calculated with the help of equation (7).

$$\text{error} = (\text{givertarget}) - (\text{inducedoutput}) \tag{7}$$

This error is again propagated through the network [layer by layer] by the backward direction. Again, we calculate the synaptic weight and bias.

In this paper, we have used Scaled Conjugate Gradient (SCG) during optimization of the network, because the SCG algorithm is based on conjugate directions [28]. C. B. Khadse, M. A. Chaudhari, and V. B. Borghate showed that the SCG gave a better result than the Conjugate gradient descent (CGB), Levenberg_Marquardt (LM), Bayesian regularization back propagation (BR), Gradient descent with momentum (GDM), and GDM with adaptive learning rate (GDX) [29]. R. P. F. Amaral, M. V. Ribeiro, and E. P. Aguiar proved that SCG is considerable faster than the BP, CGL and BFGS [30]. N. Aburaed, S. Atalla, H. Mukhtar, M. Al-Saad, and W. Mansoor used SCG in a neural network for optimizing indoor positioning systems [31]. After completing the training process, we update weight and bias with the help of equations (8) and (9):

$$w_{new} = w_{old} + \Delta w_{ji} \tag{8}$$

$$b_{new} = w_{bold} + \eta(error) \tag{9}$$

where, weight correction $(\Delta w_{ji}(n)) = \eta \cdot errer \cdot y_i(n)$. Here, η is the learning rate parameter.

Relu [32], LeakyReLU [33], PReLU [34], swish [35], SReLU [36], PReLU[37]AFs are generally used in CNN and Deep ANN where we use GPU. All of these AFs are less useful for LSTM and some other networks. So, in this paper, we have shown a comparative analysis of only classical AFs.

3. Experiments

We have done this experiment on MATLAB (2021a) and window 11.

3.1. Data Acquisition. We have taken two types of datasets for this research. First of all, we have taken the iris dataset (figure 2). This dataset contains150 data of setosa, vericolor and virginica. Secondly, we have taken cifar-10 dataset. The cifar-10 dataset contains of 60000 32×32 color images in 10 classes [18]. But we have taken three classes. They are airplanes, cars, and cats (figure 3). We have taken 100 images of each class. In this way, we have taken a total of 300 images.

sepal. length	sepal. width	petal. length	petal. width	variety
5.1	3.5	1.4	0.2	Setosa
4.9	3	1.4	0.2	Setosa
.				
.				
.	.	.	.	
7	3.2	4.7	1.4	Versicolor
6.4	3.2	4.5	1.5	Versicolor
.				
.	.	.	.	.
6.3	3.3	6	2.5	Virginica
5.8	2.7	5.1	1.9	Virginica
.				

Figure 2. Sample of the iris dataset.

This dataset contains 150 data of setosa, vericolor and virginica.

Figure 3. Figure (a), (b) and (c) are sample images of cipher_10 dataset.

The cifar-10 dataset contains of 60000 32×32 color images in 10 classes. But we have taken three classes. They are (a) cars, (b) cats, and (c) airplanes. We have taken 100 images of each class. In this way, we have taken a total of 300 images.

3.2. Procedure. We have done the following steps (figure4):

Step 1: We have taken both datasets.

Step 2: In the case of the image dataset (cifar-10), we have done some pre-processing on it. In this process, firstly we convert RGB images into gray images, secondly, we have done enhancing operation on these images and lastly, we have done double operation on these images.

Step 3: After that, we have taken the initial weight and bias value for the network.

Step 4: Then we have divided the datasets into three parts. We have divided all the datasets as training set=80%, validation set=10%, and testing set=10%

Step 5: Now we have taken the target data. In this research, we have taken three targets for the iris and cifar-10 datasets. The entire targets have been put into the output of the layer.

Step 6: In the first layer of the network, we have taken the size of input $s = 4$ for the iris dataset, and the size of input $s = 1024(32 \times 32)$ for the cifar-10 dataset. In the second layer of the network (i.e. hidden layer), we have tested the performance of the network on 21, 41, 61, 81, and 101 hidden neurons size (n). After that, we have used different types of activation functions (here in this research, we have applied logsig, tansig, elliotsig, and softmax functions). In the third layer (i.e. output layer) we have taken three output values for both (iris dataset and cifer-10 dataset), we have initialized the weights and the biases, and then have applied the softmax function.

Step 7: We have taken some iterations (epochs) for proper assigning of weights and bias. In this research, we have taken a maximum of 1000 epochs value.

Step 8: After that, we have trained this network.

Step 9: We have stopped the training when we have found stopping criteria.

After completing the above process, we have gotten the exact weight and bias of the network for both layers. At last, we have achieved the final epochs and accuracy of the network. All these final epochs and accuracies are shown in the tables from 1 to 8.

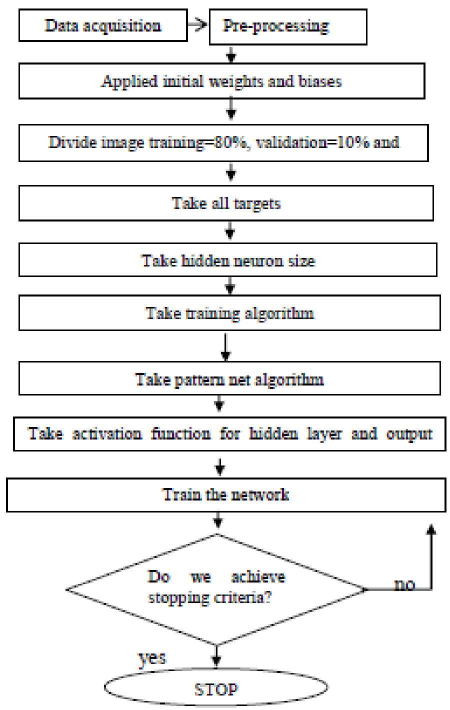

Figure 4. Methodology of this experiment.

4. Results

After completing the training of this network on 21, 41, 61, 81, and 101 hidden neurons size, we have found the following results:

- In the case of the iris dataset (table 3), when the number of hidden neurons is smaller $(n = 21)$, the elliotsig activation function has achieved 100% accuracy. But, at the same number of hidden neurons and at the same dataset, logsig and tansig activation function shave achieved 98% accuracy (table 1 and table 2), and softmax activation function has achieved 96.7% accuracy (table 4).

Similarly, in the case of cifar-10 dataset (table 7), when the number of hidden neurons is smaller $(n = 21)$, the elliotsig activation function has achieved 88.3% accuracy. But, at the same number of hidden neurons and at the same dataset, logsig activation function has achieved 75% accuracy (table 5), the tansig has achieved 81.3% accuracy (table 6) and the softmax activation function has achieved 50.3% accuracy (table 8).

This means we say that if the number of hidden neuron sizes is less, then the elliotsigmay achieve better results. (This result is also shown in figures 5 and 6).

Table 1. Different Epochs and Accuracy value on different hidden neurons size obtained with the help of logsig activation function of iris dataset.

S.N.	Hidden neuron size	Epochs	Accuracy (%)
1	21	21	98.0
2	41	25	98.7
3	61	26	98.7
4	81	19	96.0
5	101	30	98.7

Table 2. Different Epochs and Accuracy value on different hidden neurons size obtained with the help of tansig activation function of iris dataset.

S.N.	Hidden neuron size	Epochs	Accuracy (%)
1	21	19	98.0
2	41	25	98.7
3	61	20	99.3
4	81	17	97.3
5	101	23	99.3

Table 3. Different Epochs and Accuracy value on different hidden neurons size obtained with the help of elliotsig activation function of iris dataset.

S.N.	Hidden neuron size	Epochs	Accuracy (%)
1	21	30	100
2	41	34	98.7
3	61	21	98.7
4	81	15	98.7
5	101	25	99.3

Table 4. Different Epochs and Accuracy value on different hidden neurons size obtained with the help of softmax activation function of iris dataset.

S.N.	Hidden neuron size	Epochs	Accuracy (%)
1	21	20	96.7
2	41	20	97.3
3	61	32	98.7
4	81	23	98.0

5	101	24	98.7

Table 5. Different Epochs and Accuracy value on different hidden neurons size obtained with the help of logsig activation function of cifar-10 dataset.

S.N.	Hidden neuron size	Epochs	Accuracy (%)
1	21	14	75.0
2	41	14	58.7
3	61	15	61.0
4	81	17	84.0
5	101	19	79.0

Table 6. Different Epochs and Accuracy value on different hidden neurons size obtained with the help of tansig activation function of cifar-10 dataset.

S.N.	Hidden neuron size	Epochs	Accuracy (%)
1	21	16	81.3
2	41	15	75.0
3	61	6	36.7
4	81	25	88.0
5	101	23	87.3

Table 7. Different Epochs and Accuracy value on different hidden neurons size obtained with the help of elliotsig activation function of cifar-10 dataset.

S.N.	Hidden neuron size	Epochs	Accuracy (%)
1	21	22	88.3
2	41	22	88.7
3	61	16	72.3

4	81	18	86.3
5	101	26	90.3

Table 8. Different Epochs and Accuracy value on different hidden neurons size obtained with the help of softmax activation function of cifar-10 dataset.

S.N.	Hidden neuron size	Epochs	Accuracy (%)
1	21	9	50.3
2	41	11	49.3
3	61	11	56.3
4	81	8	46.0
5	101	8	53.7

- In the case of the iris dataset (figure 7), elloitsig has achieved 100% accuracy and in the case of the cifer-10 dataset (figure8), elloitsig has achieved 90.3% accuracy. This means we say that when we consider only accuracy parameters with respect to epochs, then the elliotsig activation function may achieve better results.

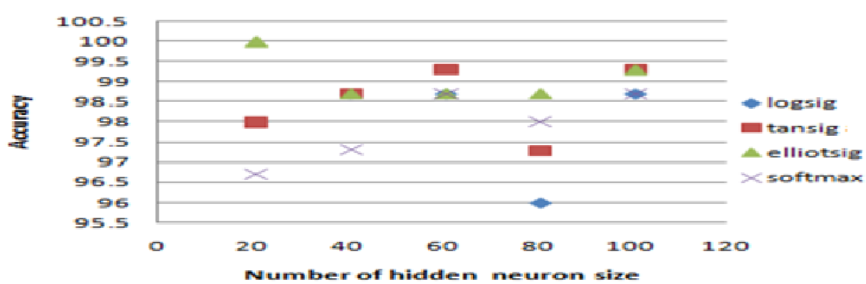

Figure 5. Accuracy on a different number of hidden neuron sizes and activation functions was achieved with the help of iris dataset.

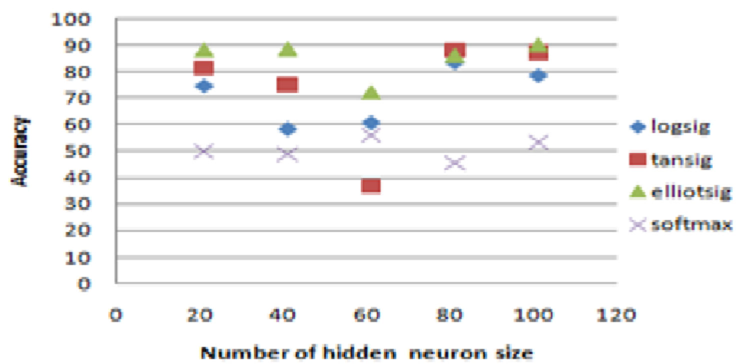

Figure 6. Accuracy on a different number of hidden neuron sizes and activation functions was achieved with the help of cifar-10 dataset.

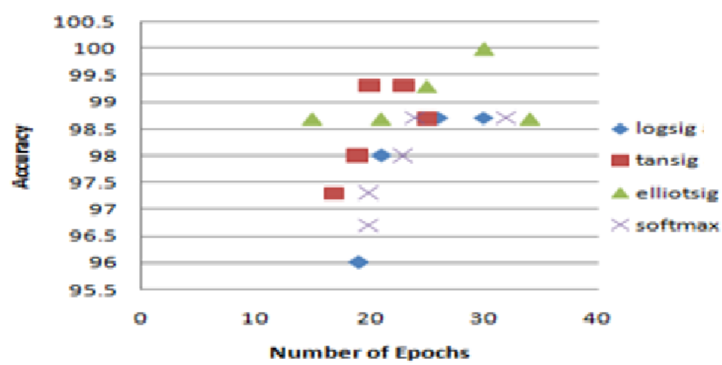

Figure 7. Accuracy on a different number of Epochs and activation functions was achieved with the help of iris dataset.

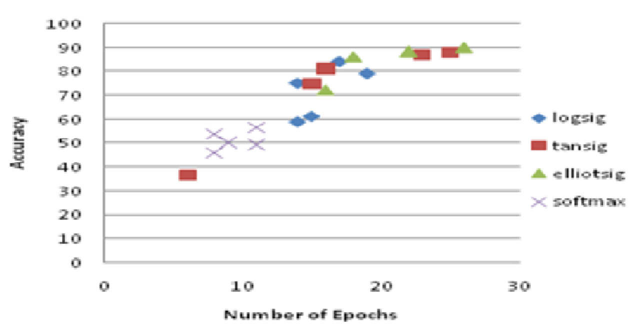

Figure 8. Accuracy on a different number of Epochs and activation functions was achieved with the help of cifar-10 dataset.

In the case of the cifar-10 dataset (table 8), the accuracies are 50.3%, 49.3%, 56.3%, 46.0%, and 53.7% have been achieved with the help of softmax activation function. This accuracy is not acceptable. So, we can say softmax activation function is not acceptable in the hidden layer.

5. Discussion

Every activation functions have some properties. Some of these properties are:

(i) Nonlinear: With the help of equations (3), and (4) we can say that the tansig and logsig A Fsuse the exponential functions. But from equation (5) we can say elliotsig does not use the exponential function. The exponential function is more suitable for a nonlinear problem. But predication speed happens slowly in the case of tansig and logsig AFs due to the use of exponential function and predication speed in elloitsig happens fastly due to not uses the exponential function.

(ii) Range and Shape: The range of tansig and elloitsig AFs are (-1, 1), and the range of logsig AF is $(0, 1)$. All these ranges are finite ranges. So in the case of pattern recognition, these AFs will show more stability. The shape of these activation functions is 'S' (figure 9).

(iii) Continuously Differentiable: These AFs are a continuously differentiable function (figure 10).The differentiation of tansig is equation (10), logsig is equation (11) and elliotsig is equation (12). So, we may use these AF in the gradient-based Optimization method.

$$f'(n) \frac{4e^{2n}}{(1+e^{2n})} \tag{10}$$

$$f'(n) \frac{e^{-n}}{(1+e^{-n})^2} \tag{11}$$

$$f'(n) = \frac{1}{(|n|+1)} - \frac{n * sign(n)}{(|n|+1)} \tag{12}$$

(iv) Zero-Centered: The tansig and elliotsig AFs are a zero-centered activation function. The logsig is not a zero-centered activation function.

Every AF has some limitations. The logsigmoid, elliotsig, and tansig AFs have a finite range, due to this reason for very high or very low value of inputs; there is almost no change to prediction. This problem is also called the vanishing gradient problem. These AFs may show slow convergence.

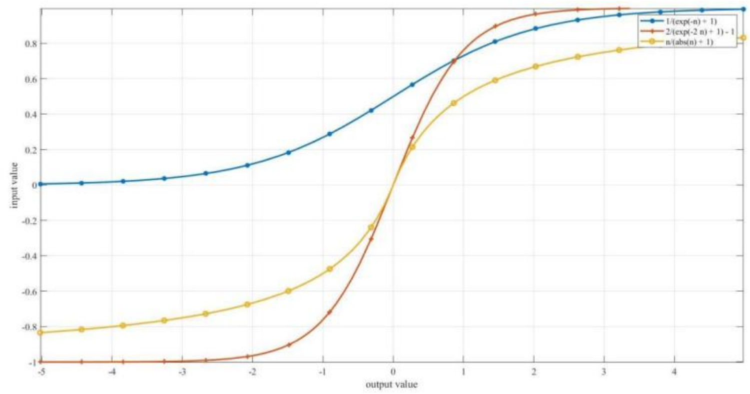

Figure 9. Plot of tansig, elliotsig and logsigmoid activation function.

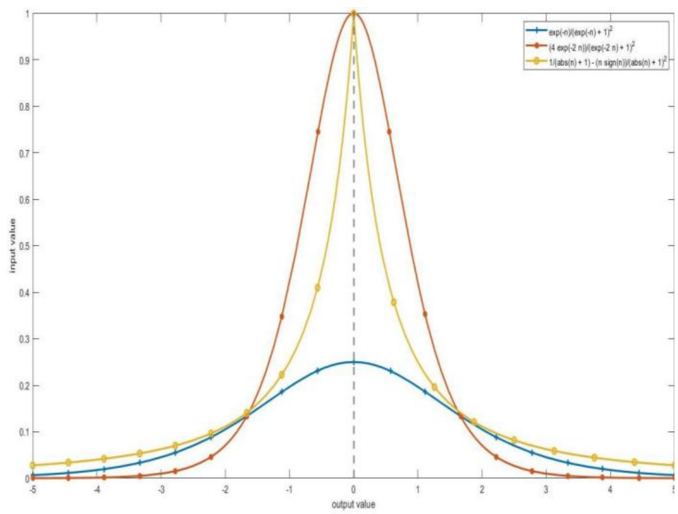

Figure 10. Plot of differentiation of tansig, elloitsig, and logsigmoid activation functions.

6. Conclusion and Future Scope

The main aim of this paper is to find out if the exponential activation function (AF) is always better than the non-exponential AF. We have taken three exponential activation functions (tansig, logsig and softmax) and one non-exponential activation function (elliotsig). So, first of all, we have explained the neural networks and their usage. After that, we have taken a pattern network and explained its working function. After that, we have explained all these AFs and their properties. Then we have donean experiment on our made pattern network. In the discussion part, we have shown that the tansig and elliotsig AFs are a smooth S-shape, bounded range $(-1, 1)$, continuously differentiable, and zero centered function. The logsigmoid AF is a smooth S-shape, bounded range $(0, 1)$, and continuously differentiable function. Vanishing gradient and slow convergence are two limitations of these AFs. We have shown some advantages of elliotsig AF upon the tansig and logsig activation function. The first advantage is that at the lowest hidden neuron size elliotsigmay achieve a better result compare to the tansig and logsig activation function. The second advantage is that when we consider only accuracy parameters with respect to epochs, then also elliotsigmay achieve better results. The third softmax activation function is not acceptable in the hidden layer. We have done this experiment with the help of two datasets. They are the iris dataset and the cifar-10 dataset. In the case of the iris dataset, elliotsig has achieved 100% accuracy at the lowest hidden neuron size (21 hidden neuron size) and 30 epochs. In the case of thecifar-10 dataset, elliotsig has achieved 90.3% accuracy at 101 hidden neuron size and 26 epochs. Thus we can say that in some cases non-exponential AFs show better results. But, the major drawback of non-exponential (elliotsig) AF is that it is less useful for the very complex and big datasets. Future work may be considered for elliotsig or modified elliotsig activation function in the pattern network for better accuracy.

References

[1] T. Hagan, Neural Network Design, 2nd Edition Book, 2014.

[2] S. Haykin, Neural Networks and Learning Machines, 3rd Edition, Pearson Prentice Hall, 2009.

[3] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, A. M. Umar, O. U. Linus and M. U. Kiru, Comprehensive review of artificial neural network applications to pattern recognition, IEEE Access 7 (2019), 158820-158846.

[4] A. Sekhon, A, and P. Agarwal, Face recognition using back propagation neural network technique, In 2015 International Conference on Advances in Computer Engineering and Applications, pp. 226-230, IEEE.

[5] V. Bihday, V. Brygilevych, Y. Hychka, Z. Liubun, N. Pelypets and V. Rabyk, Recognition of Handwritten Images Using Multilayer Neural Networks, in 2019 XIth International Scientific and Practical Conference on Electronics and Information Technologies (ELIT), IEEE pp. 59-62.

[6] A. Rubel, O. Rubel, V. Abramova, G. Proskura and V. Lukin, Improved Noisy Image Quality Assessment Using Multilayer Neural Networks, In IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON), IEEE (2019), pp. 1046-1051,

[7] R. Achkar, M. Owayjan, B. Harajli, D. Khazaal, M. Dbouk and G.Magnifico, Brain Tumor Classification Using Back Propagation Algorithm in MLP, In 2019 Fourth International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), IEEE, pp. 1-5.

[8] W. D. N. Pacheco and F. R. J. López, Tomato classification according to organoleptic maturity (coloration) using machine learning algorithms K-NN, MLP, and K-Means Clustering, In 2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA), IEEE pp. 1-5.

[9] H. I. Dino and M. B. Abdulrazzaq, Facial expression classification based on SVM, KNN and MLP classifiers, In 2019 International Conference on Advanced Science and Engineering (ICOASE) IEEE, pp. 70-75.

[10] A. Yousaf, M. J. Khan, N. Javed, H. Ibrahim, K. Khurshid and K. Khurshid, Size Invariant and written Character Recognition using Single Layer Feed forward Back propagation Neural Networks, In 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), IEEE, pp.1-7.

[11] A. Panyafong, N. Neamsorn and C. Chaichana, Heat load estimation using artificial neural network, Energy Reports 6 (2020), 742-747.

[12] T. Hossain, F. S. Shishir, M. Ashraf, M. A. Al Nasim and F. M. Shah, Brain Tumor Detection Using Convolutional Neural Network, In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), IEEE, pp. 1-6.

[13] K. Mridha, Early Prediction of Breast Cancer by using Artificial Neural Network and Machine Learning Techniques, 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT) IEEE (2021), 582-587.

[14] X. Hu, Y. Han, B. Yu, Z. Geng and Fan, Novel leakage detection and water loss management of urban water supply network using multiscale neural networks, Journal of Cleaner Production 278 (2021), 123611.

[15] S. Abhilasa, A. Srilakshmi and K. Geetha, Classification of Agricultural Leaf Images using Hybrid Combination of Activation Functions, In 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS) IEEE , pp. 785-791.

[16] D. I. Swasono, H. Tjandrasa and C. Fathicah, Classification of Tobacco Leaf Pests Using VGG16 Transfer Learning, 2019 12th International Conference on Information and Communication Technology and System (ICTS), Surabaya, Indonesia (2019), 176-181.

[17] A. Selimovic, B. Meden, P. Peer and A. Hladnik, Analysis of Content-Aware Image Compression with VGG16, 2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI), San Carlos 2018, 1-7.

[18] H. Wang, Garbage Recognition and Classification System Based on Convolutional Neural Network VGG16, 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Shenzhen, China (2020), 252-255.

[19] E. Enkhtaivan, T. A. Adesuyi and S. Kim, Facial Emotion Recognition using Convolutional Neural Network Based on Repetitive Learning Blocks Approach (2020), 512-514.

[20] S. Hanxu, L. Yue, C. Hao, L. Qiongyang, Y. Xiaonan, W. Yongquan and G. Jun, Research on Human Action Recognition Based on Improved Pooling Algorithm, Chinese Control And Decision Conference (CCDC), IEEE, 2020.

[21] Liu, Qing-feng, Muhammad FarjadIqbal, Jian Yang, Xian-yang Lu, Peng Zhang, and MominaRauf, Prediction of chloride diffusivity in concrete using artificial neural network: Modelling and performance evaluation, Construction and Building Materials 268 (2021), 121082.

[22] Kousik, Nalliyanna V, Yuvaraj Natarajan, R. Arshath Raja, Suresh Kallam, Rizwan Patan and Amir H. Gandomi, Improved salient object detection using hybrid Convolution Recurrent Neural Network, Expert Systems with Applications 166 (2021), 114064.

[23] Shorfuzzaman, Mohammad, and M. Shamim Hossain, MetaCOVID: A Siamese neural network framework with contrastive loss for n-shot diagnosis of COVID-19 patients, Pattern Recognition 113 (2021), 107700.

[24] R. Rawal, K. Goel and C. Gupta, COVID-19: Disease Pattern Study based on Semantic-Web Approach using Description Logic, In2020 IEEE International Conference for Innovation in Technology (INOCON) IEEE 6 (2020), 1-5.

[25] K. Goel, C. Gupta, R. Rawal, P. Agrawal and V. Madaan, FaD-CODS Fake News Detection on COVID-19 Using Description Logics and Semantic Reasoning, International Journal of Information Technology and Web Engineering (IJITWE) 16(3) (2021), 1-20.

[26] Zhang, Dehua and Sha Lou, The application research of neural network and BP algorithm in stock price pattern classification and prediction, Future Generation Computer Systems 115 (2021), 872-879.

[27] Ren, Xinming, Huaxi Gu and Wenting Wei, Tree-RNN: Tree structural recurrent neural network for network traffic classification, Expert Systems with Applications 167 (2021), 114363.

[28] B. Yegnanarayana, Artificial Neural Networks, Prentice Hall of India, 2006.

[29] Charu Gupta, Prateek Agrawal, Rohan Ahuja, Kunal Vats, Chirag Pahuja and Tanuj Ahuja, Pragmatic Analysis of Classification Techniques based on Hyperparameter Tuning for Sentiment Analysis, International Semantic Intelligence Conference (ISIC'21), Delhi (2021), 453-459.

[30] D. Tayal, S. Sonawani, G. Ansari and C. Gupta, Fuzzy Time Series Forecasting of Low Dimensional Numerical Data, Proceedings of International Journal of Engineering Research and Applications 2(1) (2011), 132-5.

[31] C. Gupta, A. Jain, DK. Tayal, O. Castillo, ClusFuDE: Forecasting low dimensional numerical data using an improved method based on automatic clustering, fuzzy relationships and differential evolution, Engineering Applications of Artificial Intelligence 1(71) (2018), 175-89.

[32] I. Good Fellow, Y. Bengio, and A. Courvlle, Deep Learning, MIT Press, 2016.

[33] M. F. Møller, A scaled conjugate gradient algorithm for fast supervised learning, Neural networks 6(4) (1993), 525-533.

[34] C. B. Khadse, M. A. Chaudhari and V. B. Borghate, Comparison of seven backpropagation algorithms for three phase power quality assessment, In TENCON 2017-2017 IEEE Region 10 Conference, (2017), 2548-2553.

[35] R. P. F. Amaral, M. V. Ribeiro and E. P. Aguiar, Type-1 and singleton fuzzy logic system trained by a fast scaled conjugate gradient methods for dealing with binary classification problems, Neurocomputing 355 (2019), 57-70.

[36] N. Aburaed, S. Atalla, H. Mukhtar, M. Al-Saad and W. Mansoor, Scaled Conjugate Gradient Neural Network for Optimizing Indoor Positioning System, In 2019 International Symposium on Networks, Computers and Communications (ISNCC), IEEE, pp. 1-4.

[37] Javid, Alireza M., Sandipan Das, Mikael Skoglund and Saikat Chatterjee, A ReLU dense layer to improve the performance of neural networks, In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, (2021), 2810-2814.

[38] Biswas, Koushik, Sandeep Kumar, Shilpak Banerjee and Ashish Kumar Pandey, SAU: Smooth activation function using convolution with approximate identities, arXiv preprint, (2021) arXiv:2109.13210.

[39] Wu, Xian, YanhanJi and Li Xiao, High-accuracy handwriting recognition based on improved CNN algorithm, In 2021 International Conference on Communications, Information System and Computer Engineering (CISCE), IEEE (2021) pp. 344-348.

[40] Zhu, Hegui, Huimin Zeng, Jinhai Liu and Xiangde Zhang, Logish: A new nonlinear nonmonotonic activation function for convolutional neural network, Neurocomputing 458 (2021), 490-499.

[41] Watanabe, Thomio and Denis F. Wolf, Image classification in frequency domain with 2SReLU: a second harmonics superposition activation function, Applied Soft Computing 112 (2021), 107851.

[42] Adu Kwabena, Yongbin Yu, Jingye Cai, Isaac Asare and Jennifer Quahin, The influence of the activation function in a capsule network for brain tumor type classification, International Journal of Imaging Systems and Technology, (2021).

[43] https://www.cs.toronto.edu/~kriz/cifar.html for downloading cifer-10 datasets.