Advances and Applications in Mathematical Sciences
Volume 21, Issue 9, July 2022, Pages 5417-5429
© 2022 Mili Publications, India

ROOT2SIGMOID ACTIVATION FUNCTION IN NEURAL NETWORK

ARVIND KUMAR and SARTAJ SINGH SODHI

Computer Sc. and Engg.
USICT GGSIPU, Delhi, India
E-mail: arvind.usict.134164@ipu.ac.in

Abstract

Activation function (AF) acts an important role in Convolutional Neural Network (CNN). But there is still requires some improvement in the Activation Function. So, in this paper, first of all we have proposed our root2igmoid activation function in the CNN, after that we compare this AF with tanh, relu and Swish AF. We found that root2Sigmoid activation function gives better accuracy with compare to these activation functions. Our activation function is a smooth shape, bounded (-0.25, 0.25) or unbound depends on some conditions, zero centered and continuously differentiable function. Vanishing gradient and slow convergence may or may not happen depend upon choosing of bound or unbounded value. For testing our result, we make a small CNN and run our AF and other these three AF on this CNN. We compare our sigmoid with the help of adam and rmsprop training algorithms. We take digit, cipher-10 and mathworkcap datasets for this purpose.

1. Introduction

Artificial neural network (ANN) is a branch of machine learning [1]. Clustering, classification, recognition, etc. are major areas of ANN [1-4]. Classification of Tobacco Leaf Pests [5], Image compression with VGG16 [6], Garbage Recognition and classification [7], Facial Emotion Recognition [8], Human Action Recognition [9], Prediction of chloride diffusivity in concrete [10], improve salient object detection [11], prediction of covid-19 patient [12], stock price pattern classification and prediction [13], and networktraffic classification [14] are some applications where ANN are used. Convolutional Neural Network (CNN) is a part of ANN. CNN is made by many convolutional layers and fully connected layers. For example: Alexnet [15],

2020 Mathematics Subject Classification: 68T07.
Keywords: Convolutional Neural Network, activation function, tanh, relu, swish.
Received September 2, 2021; Accepted January 1, 2022

VGG [6], etc. So, CNN has two parts. First part is made by many multiple layers, such as convolutional layers, activation layers, pooling layers and normalization layers, and second part is made by fully connected layer, softmax layer and classification layer. In this paper, we made three CNN. Because CNN takes more time for running a network (some times more days), so we make a small network (maximum of 14-layer CNN). We mention details of our made CNN in table 3. Optimization algorithms and activation functions (AF) do many important works during training of ANN. Losses are reduced with the help of these optimization algorithms. We check performance of our root2sigmoid AF on adam [16] and rmsprop [17] optimization algorithms. Output is calculated with the help of Activation Function (AF). Generally, there are two types of AFs are used in ANN. They are: Linear activation function and nonlinear activation function. The linear activation function is useful for linear separability types of problems. Hard limit, Symmetrical Hard limit, and Linear functions are examples of the linear activation function. Non-linear function is more useful for non-separability problems. The log sigmoid, hyperbolic tansigmoid (tanh), Elliot symmetric sigmoid transfer function (elliotsig), softmax, etc are example of non-linear activation function. Aranda, Bi-sig1, Bi-sig2, Bi-tanh1, Bi-tanh2, cloglog, cloglogm, Elliott, Gaussian, logarithmic, loglog, logsigm, modified Elliott, rootsig, saturated, sech, sigmoidalm, sigmoidalm2, sigt, skewed − sig, softsign, wave, etc are some other activation function [18]. Every sigmoid function has some properties. These properties are Nonlinear, Range, Continuously Differentiable, and Shape.

1.1. Some Activation Functions used in CNN and their properties

There is a number of activation functions used in CNN. Out of them we take three AF for comparison of our root2sigmoid AF. They are tanh [19], relu [20] and swish [21]. In Hyperbolic Tangent sigmoid (tanh) function, if n is input then equation (1) is output value $(f(n))$ of tanh.

$$f(n) = (e^{n} - e^{-n})/(e^{n} + e^{-n}) \tag{1}$$

The derivation of tanh AF is equation (2).

$$f(n) = 4e^{2n}/(1 + e^{2n})^{2} \tag{2}$$

This AF is bound AF and its range is (-1, 1) [1-4]. There are two limitations of the tanh AF. This AF has finite range, due to this reason for vary high or very low value of input; there is almost no change to prediction. This problem is also called vanishing gradient problems. This AF may be performing slow convergence.

In rectified linear unit (relu) function, if n is input then equation (3) is output value $(f(n))$ of relu.

$$f(n) = x(\text{if } x > 0), \text{ otherwise } 0 \tag{3}$$

The derivation of relu AF is

$$f(n) = 1(\text{if } x > 0), \tag{4}$$

Derivation of equation (4) is constant. So, this is not more useful for recurrent neural network or LSTM. This AF is unbound above and bounded below.

In swish function, if n is input then equation (5) is output value $(f(n))$ of swish.

$$f(n) = n \cdot \alpha(n) \tag{5}$$

where $\alpha(n) = 1/(1 + e^{-n})$.

The derivation of swish AF is

$$f(n) = f(n) + \alpha(n)(1 - f(n)) \tag{6}$$

This AF is unbound above and bounded below.

2. Proposed root2sigmoid Activation Function and its properties

Equation (7) is our proposed root2sigmoid activation function (AF).

$$f(n) = \frac{(\sqrt{2})^{n} - (\sqrt{2})^{n}}{2\sqrt{2}(2(\sqrt{2})^{2n} + 2(\sqrt{2})^{-2n})^{\frac{1}{2}}} \tag{7}$$

Where n is input value and $f(n)$ is output value. We take $\sqrt{2} = 1.414$ for this research.

Root2sigmoid AF has following three properties:

Range and Shape. With the help of table 1, if we take output value upto four decimal points, then we get bounded range and its range becomes [-0.25, 0.25]. If we don't take output value upto four decimal points, then we get unbounded range. Vanishing gradient and slow convergence happens when we take bounded value. If we take unbounded value of this function, then Vanishing gradient and slow convergence problem not happens. With the help of figure (1), we can say that shape of equation (7) is smooth.

Continuously differentiable. Root2sigmoid is a continuously differentiable function. Its differentiation is found by equation (8). So, we may use root2sigmoid AF into gradient-based optimization method.

Table 1. Output value of root2sigmoid and tanh activation function.

Sr. No.	root2sigmoid	tanh
1	0.111779751	0.7615942
2	0.18188608	0.9640276
3	0.217062425	0.9950548
4	0.233933946	0.9993293
5	0.242093731	0.9999092
6	0.246093684	0.9999877
7	0.248072602	0.9999983
8	0.249056776	0.9999998
9	0.249547596	1
10	0.249792723	1
11	0.249915233	1
12	0.249976485	1
13	0.250007114	1
14	0.250022432	1

15	0.250030093	1
16	0.250033925	1
17	0.250035841	1
18	0.2500368	1
19	0.250037279	1
20	0.250037519	1

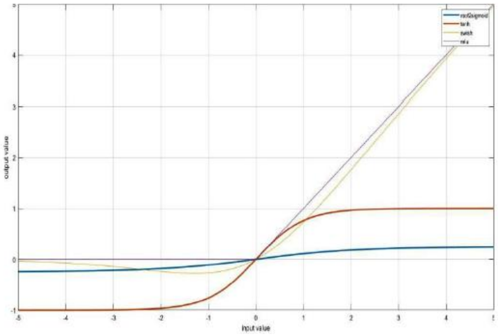

Figure 1. Plot of root2sigmoid, tanh, swish and relu activation function.

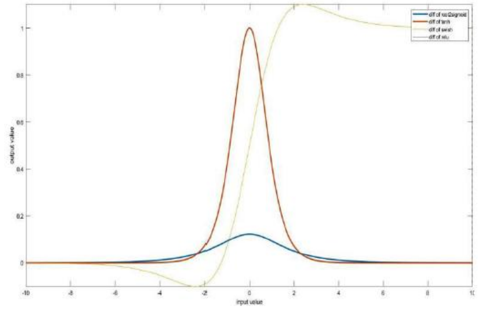

Figure 2. Plot of differentiation of root2sigmoid, tanh, swish and relu activation function. This graph clear shows that the differentiation of root2sigmoid (f') is continuous at 0.

$$f(n)=\frac{(\sqrt{2})^{n}\cdot\log(\sqrt{2})+(\sqrt{2})^{n}\cdot\log(\sqrt{2})}{2\sqrt{2}(2(\sqrt{2})^{2n}+2(\sqrt{2})^{-2n})^{\frac{1}{2}}}$$

$$-\frac{[4(\sqrt{2})^{-2n} \cdot \log(\sqrt{2}) - 4(\sqrt{2})^{2n} \cdot \log(\sqrt{2})] * [(\sqrt{2})^{-n} - (\sqrt{2})^{n} - (\sqrt{2})^{n}}{4\sqrt{2}(2(\sqrt{2})^{-2n} + 2(\sqrt{2})^{2n})^{\frac{1}{2}}} \tag{8}$$

Figure 2 shows the graph of the derivation of the equation (8).

Suppose $n = 0$, then value of equation (7) is as under:

$$f(0) = 0 \tag{9}$$

and value of equation (8) is as under:

$$f(0) = 0.1225 \tag{10}$$

with the help of equation (9) and (10) we can say f' is continuous at 0. As like logsigmoid and tansigmoid this AF also allows backpropagation, because this function is differential function. During optimization process this activation function may give high accuracy, due to give more variations as compare to tanh (see table 1). So this activation function is more useful for LSTM or recurrent neural network.

Zero centered. Root2sigmoid is a zero-centered activation function, because it show symmetrical on both side of zero. So as like tanh AF, this function works better than logsigmoid. The logsigmoid is not symmetric around zero, but root2sigmoid is symmetric around zero (as like tansigmoid). So, due to this reason we may use root2sigmoid for solving very complex problems (Non-linear problems); such as audio, images or any high dimensionality problems.

3. Experiments

We did this experiment on intel core i7, window 10, and MATALB 2020b [22]. We did the following four steps for this research:

Step 1 (Datasets)**.** We had taken 3-datasets. These datasets were digit (figure 3.a), merch (figure 3.b) and cipher-10 (figure 3.c). As per table 2, we take number of samples for training and testing purpose. Output value for digit dataset is 10 (digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9). Output value for merch dataset is 5. The cifar-10 dataset consists of $60000, 32 \times 32$ color images in 10 classes. But we only took three classes. They were frog, truck and cats. We took 5000 images of each class for training purpose. In this way, we took a total of $15000(= 5000 * 3)$ images for training. Similarly, we took 1000 images of each class for testing purpose. In this way, we took a total of $3000(= 1000 * 3)$ images for testing.

Step 2 (Network Selection)**.** In all the above three datasets input and corresponding output values are given. For testing our root2sigmoid activation function, we had taken different layer of CNN for different dataset. As per table 3, we had taken 14-layer CNN for digit dataset, 14-layer CNN for cipher-10 dataset and 12-layer CNN for merch dataset.

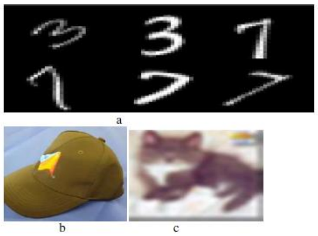

Figure 3. '*a*' is a figure of digit dataset, *b* is a figure of merch dataset and *c* is a figure of cipher-10 dataset.

Table 2. Training and testing samples of different datasets.

Datasets	Number of samples for training purpose	Number of samples for testing purpose
Digit	5000	5000
Merch	60	15
Cipher-10	15000	3000

Table 3. Different CNN layer for different datasets.

Sr. No.	Layer of CNN	Size(S) and number(n) of CNN		
		Digit dataset (14-layer CNN)	Merch dataset (12-layer CNN)	Cipher-10 dataset (14-layer CNN)
1	ImageInput layer	S=[28 28 1]	S=[128 128 3]	S=[32 32 3]
2	Convultion2d layer	S=[5 5] and n=8	S=[5 5] and n=32	S=[3 3] and n=128
3	Relu	Relu	Relu	Relu

4	Convultion2d layer	S=[5 5] and n=16	S=[5 5] and n=16	S=[3 3] and n=64
5	Relu	Relu	Relu	Relu
6	Convultion2d layer	S=[5 5] and n=32	S=[5 5] and n=8	S=[3 3] and n=32
7	Relu	Relu	Relu	Relu
8	Convultion2d layer	S=[5 5] and n=64	not taken	S=[3 3] and n=16
9	Relu	Relu		Relu
10	batchnormalization	batchnormalization	batchnormalization	Batchnormalization ion
11	Root2sigmoid /relu/tanh/swish	Root2sigmoid /relu/tanh/swish	Root2sigmoid /relu/tanh/swish	Root2sigmoid /relu/tanh/swish
12	Fully connected	Output value=10	Output value=5	Output value=3
13	Softmax	Softmax	Softmax	Softmax
14	Classification	Classification	Classification	Classification

Step 3 (Selecting activation function)**.** As per table 3, we compare our root2sigmoid AF at the layer-11 of 14-layer CNN and at the layer-9 of 12-layer CNN. One-by-one we took tanh, relu, swish and root2Sigmoid activation function at sr. no. 11. Softmax activation function is used for multi-classification model, but logistic function is used for binary classification model, so we took softmax activation function at layer sr. no 13 of table 3.

Step 4 (Run the network)**.** Here we had train/run the network with adam and rmsprop algorithm. We had taken maximum number of epochs was 30, so our network stops training/running; when the number of epochs reach upto 30. We kept default value of other parameters, because in this paper, we want to show the performance of our activation function with compare to the other activation function.

After completing the above process we calculated accuracy. In table 4, we showed accuracy result.

Table 4. Achieved accuracy with the help of different activation function and adam/rmsprop algorithm on different datasets.

Sr. No.	Activation Function	Digit dataset		Merch dataset		Cipher-10 dataset	
		Accuracy on adam algorithm	Accuracy on rmsprop algorithm	Accuracy on adam algorithm	Accuracy on rmsprop algorithm	Accuracy on adam algorithm	Accuracy on rmsprop algorithm
1	Tanh	99.92	99.86	66.67	66.67	87.70	87.83
2	Swish	99.82	99.26	80.00	66.67	85.40	85.57
3	Root2sigmoid	99.98	99.98	86.67	80.00	90.33	89.63
4	Relu	99.72	99.86	60.00	20.00	86.83	84.57

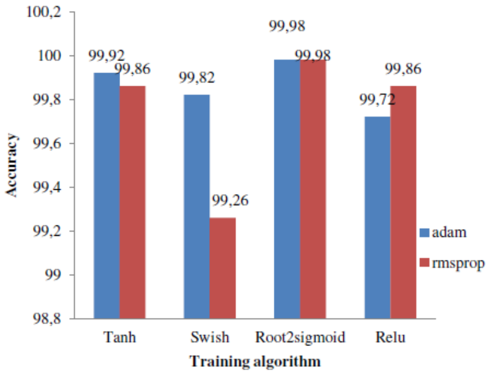

Figure 4. Accuracy on different activation function with help of adam and rmsprop training algorithm in digit datasets.

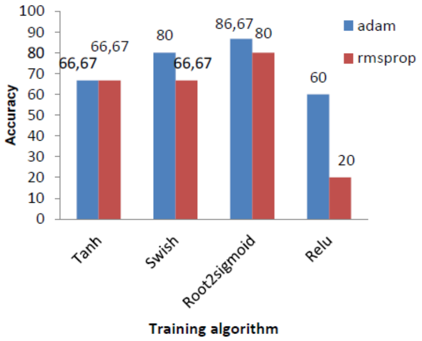

Figure 5. Accuracy on different activation function with help of adam and rmsprop training algorithm in merch datasets.

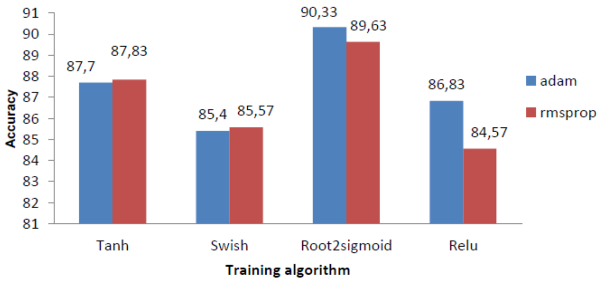

Figure 6. Accuracy on different activation function with help of adam and rmsprop training algorithm in cipher-10 datasets.

4. Results

After completing the training process of this network, we found following three results: In the case of the digits dataset (table 4): With the help of adam training algorithms, 99.92% accuracy was found by tanh, 99.82% accuracy was found by swish, 99.98% accuracy was found by root2sigmoid (highest), and 99.72% accuracy was found by relu. This means root2igmoid gives better accuracy in this case. With the help of rmsprop training algorithms, 99.86% accuracy was found by tanh, 99.26% accuracy was found by swish, 99.98% accuracy was found by root2sigmoid (highest), and 99.86% accuracy was found by relu. This means root2igmoid gives better accuracy in this case (Figure 4).

In the case of merch dataset (table 4): With the help of adam training algorithms, 66.67% accuracy was found by tanh, 80.00% accuracy was found by swish, 86.67% accuracy was found by root2sigmoid (highest), and 60.00% accuracy was found by relu. This means root2igmoid gives better average accuracy in this case. With the help of rmsprop training algorithms, 66.67% accuracy was found by tanh and swish, 80.00% accuracy was found by root2sigmoid (highest), and 20.00% accuracy was found by relu. This means root2igmoid gives better accuracy in this case (Figure 5).

In the case of cipher-10 dataset (table 4): With the help of adam training algorithms, 87.70% accuracy was found by tanh, 85.40% accuracy was found by swish, 90.33% accuracy was found by root2sigmoid (highest), and 86.83% accuracy was found by relu. This means root2igmoid gives better accuracy in this case. With the help of rmsprop training algorithms, 87.83% accuracy was found by tanh, 85.57% accuracy was found by swish, 89.63% accuracy was found by root2sigmoid (highest), and 84.57% accuracy was found by relu. This means root2igmoid gives better accuracy in this case (Figure 6).

5. Conclusion and Future Scope

In this paper, we have explained about our proposed root2sigmoid Activation Function (AF). First of all, we have explained the convolutional neural networks and their usages. After then we have explained some very famous AF which are used in CNN. Then we have presented our root2Sigmoid AF. We have presented its properties. We have also shown that this sigmoid AF is a smooth shape, bounded range (-0.25, 0.25) or unbounded range depend upon some conditions, continuously differentiable, and zero centered function. Vanishing gradient and slow convergence may or may not happen depend upon choosing of bound or unbounded value. We had taken 3-datasets. These datasets were digit, merch and cipher-10. We have shown that in all these datasets, root2igmoid gives better accuracy as compared with tanh, swish and relu for the same CNN. For testing result of our root2sigmoid activation function, we had taken different layer of CNN for different dataset. We had taken 14-layer CNN for digit dataset, 14-layer CNN for cipher-10 dataset and 12-layer CNN for merch dataset. We used adam and rmsprop algorithms during the training the CNN.

Future work of root2igmoid is that we may consider root2igmoid AF in the CNN, LSTM or any neural network for getting better accuracy in place of sigmoid activation function.

References

[1] I. Goodfellow, Y. Bengio and A. Courville, Deep learning, MIT press, 2016.

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

[3] S. Haykin, Neural networks and learning machines, 3rd edition, Pearson Prentice Hall, 2009.

[4] Charu C. Aggarwal, Neural networks and deep learning: A textbook, Springer publication, 2018.

[5] 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.

[6] 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.

[7] H. Wang, Garbage recognition and classification system based on convolutional neural network VGG16, 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Shenzhen, China, (2020), 252-255.

[8] E. Enkhtaivan, T. A. Adesuyi and S. Kim, Facial emotion recognition using convolutional neural network based on repetitive learning blocks approach (2020), 512-514.

[9] 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, (2020) Chinese control and decision conference (CCDC), IEEE, 2020.

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

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

[12] 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.

[13] 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.

[14] 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.

[15] Alom Md Zahangir, Tarek M. Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C. Van Esesn, Abdul A. S. Awwal and Vijayan K. Asari, The history began from alexnet: A comprehensive survey on deep learning approaches, arXiv preprint arXiv:1803.01164, (2018).

[16] Kingma P. Diederik and Jimmy Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, (2014).

[17] R. Vijava Kumar Reddy, B. Srinivasa Rao and K. Prudvi Raju, Handwritten Hindi digits recognition using convolutional neural network with RMSprop optimization, In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE (2018), 45-51.

[18] Farzad Amir, Hoda Mashayekhi and Hamid Hassanpour, A comparative performance analysis of different activation functions in LSTM networks for classification, Neural Computing and Applications 31(7) (2019), 2507-2521.

[19] Szandała Tomasz, Review and comparison of commonly used activation functions for deep neural networks, In Bio-inspired Neurocomputing, Springer, Singapore, (2021), 203-224.

[20] Nair Vinod and Geoffrey E. Hinton, Rectified linear units improve restricted boltzmann machines, In Icml, 2010.

[21] Prajit, Ramachandran, Barret Zoph and V. Le Quoc, Swish: a self-gated activation function, arXiv preprint arXiv:1710.059417, (2017).

[22] https://www.mathworks.com, Website of MATLAB program.