

“Analysis Of Activation Function In The Convolution Neural Network Model” (Best Activation Function Sigmoid Or Tanh)

Archana Tomar¹, Harish Patidar²

¹Department of Computer Science, Mandsaur University, India

E-mail: archana.simmy@gmail.com

²Department of Computer Science, Mandsaur University, India

Abstract

The activation functions, which are mathematical formulae, govern how a neural network responds. A particular sort of function is called an activation function. In ANN, the Activation function performs the conversion task of the input signal into an output signal. The succeeding layer of the stack then uses this output signal as input. Every neuron in the network is connected to the function, and it controls whether or not a neuron fires according to how crucial its input is to the prediction the model is making. In this paper, an explanation of the most used Activation Function which is sigmoid and Tanh. With the help of the activation functions, which also serve a secondary function in the process, Each neuron's output can be normalized to fall into one of two ranges: either 0 to 1 or -1 to 1. For prediction best activation in sigmoid and tanh applied in multiple dataset like MANIST, FER etc data set with different epochs and identify the result of both Activation functions. The sigmoid, tanh, and Relu activation functions—the three most common activation functions—have all been subjected” to analysis.

Keywords:

Activation functions, Neurons, CNN, RELU, Sigmoid, Tanh

1. INTRODUCTION

In Artificial Neural networks we need more computation to train large size image dataset and it is also not suitable for fetching all details from images, CNN is a model in NN which is used for fetching some special dependent information and it is also used in classification of data.

In the CNN model perform multiple filtration and in this filtration process gain different types of extraction it may be size or shape or anything which provides information for determining the input or picture. The main properties of CNN is how to identify pictures correctly and perform classification. so in picture identification multiple layers work together. So, the logic behind the CNN is, Firstly, provide some images as an input and pass them to the input layer. After that in this stage, feature extraction or filtration is applied in input which covers the convolution layer and Activation function.

This function controls how a neuron reacts to the information it has received from the layer beneath it. The construction of neural networks has involved the use of several activation functions of various types. The three functions that make up the classical activation functions are the step function, sigmoid function, and tanh function. The sigmoid functions smoothness makes it easier to create learning algorithms

After receiving the result of activation function application some pooling layers are applied in this model and the result of this layer is optimized feature map and this is used for preventing and optimizing the time. Now here the classifier works by applying the fully connected layer which is performed as multilayer processing and the use of a fully connected layer is to classify the picture. and at the last perform an output layer for providing accurate prediction. This

process is done again and again in CNN model to train our data for predicting accurate results.

The majority of experts agree that the CNN networks activation

function has an impact on classification accuracy as well as how long it takes to train the network. Although the performance of CNN could be impacted by the findings of this study, that possibility is not being fully explored. To address a vacuum in the pertinent research, we examine how activation functions affect the CNN model[4] datasets prediction accuracy. During this phase, a convolutional neural network (CNN) is constructed and its layers are each given a different activation function.

2. LITERATURE SURVEY

In this article Author Albawi[1], explained in detail about the CNN and their issues with all parameters or elements and also explained about their working which are used in CNN.

convolution layer plays an important role in the CNN network model and this layer work like time taken so whole CNN process will focus on convolution layer working here whole network or their performance dependent on their working so in CNN multilayer operation perform for prediction accurate result and for that multiple layer work together but due to multilayering working time also increases.

Author R. Chauhan[2] implements CNN on MNIST dataset for image detection and recognition. After evaluation, CNN model accuracy calculated in MNIST dataset is about 99 percent and in CIFAR-10 dataset it is around 80 percent. The author also explains that for calculating the accuracy of the CIFAR-10 dataset it uses real-time data and dropout on CPU units. and for final conclusion author evaluate the better result of both dataset like MNIST and CIFAR-10 and he analyze that MNIST provide better performer result than the CIFAR-10 and if we trained our model with GPU unit so we will definitely improved their performance

Author Ajit A.[3] explained about the feature extraction from the given data and he also defined that nowadays for any object detection or any image recognition CNN has been used. He also defines the working of algorithms in a step like in this model including multiple procedure which involve

1. Backpropagation concept
2. Convolutional Layers
3. Feature formation
4. Pooling.

In this article the author also explains about the different types of architecture of CNN[23] like LeNet, AlexNet, VGG Net, Google Net and Microsoft ResNet with detailed explanation about how CNN works in all these architectures.

Author Karam A. -F.[5], involve a supervised learning concept in this article. During research the author started with techniques which are generally involved in model construction which provide the high accuracy and finalize the best technique for image classification. author also include optimization technique in this research which involve

Learning Rate Reduction

Training Generator:

Preparing data:

Fitting the Model:

Here the author concluded that

If you are working with zoomed pictures, here add more parameters for predicting accurate results and in this to ignore or avoid the over-fitting model data augmentation technique are used. so here author predict the accuracy of ADAM is 0.95 percent and this is better than SGD and RMSprop.

Feng et al[6] author analyzes the performance of multiple activation functions in ANN. In this paper author define the part of various sorts of activation functions, just as their individual preferences and detriments and applicable fields are additionally talked about, so that individuals can pick the most suitable activation functions to get the superior performance of ANNs. The author analyzes various activation function attributes and clarifies their focal points and impediments. Every activation function has its own qualities, so the author can only with significant effort state which one is the more suitable, the only thing it can do is picking an appropriate activation function dependent on our point and explicit organization structure.0

Rasamoelina. et al[7] Review of Activation Functions for Artificial Neural Networks In this paper, the authors present the most commonly used state-of-the-art trend activation functions commonly used in hidden layers of artificial neural networks. Shows a review of the function. Basically, the author described her two classes of activation functions: derivatives, zero-centered saturated monotonicity. Next, the author also confirmed her 8 major activation functions. According to the author, the SELU, Swish, and Mish experimental results landscapes provide smoother output. Note also that an automatic search gives satisfactory results for finding the activation function.

et al[8] author proposed and designed a new AF (activation function) called RELu. So in this function the author introduces two parameters, one is constant and The second threshold parameter smoothes the function, makes it non-monotonic, and introduces non-linearity into the network. The results of the experiments show that the Relu-Memristor-Like Activation Function for Deep Learning outperforms ReLU and other activation functions on deeper models and across a variety of challenging datasets. As in this paper, the author conducted practical experiments by training and classifying a multi-layer perceptron (MLP) on benchmark data like the Wisconsin breast cancer, MNIST, Iris and Car evaluation.

The RMAF is based on the properties of the memristive window function, which can adapt to deep networks. and the function's threshold parameter p allows negative representations to flow through the network during forward propagation and is capable of scaling to any given network. These RMAF properties allow networks to benefit from negative representation.

Sharma. et al[9] ACTIVATION FUNCTIONS IN NEURAL NETWORKS In this article, briefly describe various activation functions used in the field of deep learning and discuss the importance of activation functions in developing effective and efficient deep learning models. It also describes gender. Performance of artificial neural networks. In this article, the authors emphasize the need for activation functions and the need for nonlinearity in neural networks. First, the author describes activation functions, then briefly discusses the need for activation functions and the need for nonlinearity in neural networks. Next, we discuss different types of activation functions commonly used in neural networks. Therefore, the authors focus here on the use of activation functions in the computational (hidden) layers of neural networks and their actual placement in different domains.

Relu AF (activation function) and Tanh AF have certain correlated characteristics, according to Li. et al[10] (activation function). The Tanh Activation function's results might increase the qualities Relu units activate and decrease the qualities they cut. The author also demonstrates how the networks could significantly improve. by converting the Tanh activation function into the weighted sum of the Relu activation function. So the author ran a series of experiments on some datasets, and the results show that this method could improve ResNet's accuracy. When the Relu unit's output is positive, the yield of the TF(Tanh function) is also certain, which could improve the Relu unit's outcome. When the result of the Relu unit is zero, the result of the Tanh function is negative, which may introduce the information discarded by Relu. The weighted sum of the Relu and Tanh activation functions could improve the popular deep convolutional neural networks.

3. CNN MODEL

In the CNN[22] model the first layer works as an input layer. In this layer we collect all the images and input in the form of RGB or gray color scale.

3.1 TYPES OF LAYERS:

INPUT LAYER:

Firstly collect all images and after that arrange in a pixel format and the range of this format is 0-255 and before calculation in a machine this range is represented or transformed into 0 or 1 format du eto machine calculation.

this is an example of 4*4 image with three color channel

12	24	12	2		
10	2	21	12	6	
1	12	25	19	10	5
5	5	10	12	5	12
	9	6	8	14	6
		4	8	2	1

Figure 1: 4*4 image color channel

CONVOLUTION LAYER:

For extracting the features of images in this layer filtration is applied and extract information from this. and this filtration is perform many times in convolution layer

1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	0

-1	0	1
-2	0	2
-1	0	1

0	-4	-4	0
0	-4	-4	0
0	-4	-4	0
0	-4	-4	0

Figure 2: Filtration in 6*6 image set

This is 1 step of filtration but in this model the same process is applied multiple times.in this process filtration is applied like image value set and filtered value like 1*0+ and in the same perform calculation and design a final map.

POOLING LAYER :

after detecting a final map its converted to reduced or optimized feature map or it can be a max or average pooling here we select avg or max value from a final feature map like from set of 4 values 0,-4,-4,0 so the max value is 0,0,0,0 and the average value like -2,-2,-2,-2. and this is a final optimized image set and now apply final connected layer or we can say here perform classification using this layer and retrieve output based on Activation function[16].

3.2 ACTIVATION FUNCTIONS

SIGMOID ACTIVATION FUNCTION

The “activation function[14] is the one that is most usually used since it is a non-linear function. Values between 0 and 1 can be changed with the Sigmoid function. It may be” characterised as follows:

$$f(x) = 1/e^{-x}$$

The “sigmoid function[12] is continuously differentiable and has the shape of an S-curve. The equation for the derivative of” the function,

$$f'(x) = 1-\text{sigmoid}(x)$$

Additionally, because “the sigmoid function is not symmetric around zero, all of the values generated by neurons will have the same signs. The sigmoid function can be scaled as one solution to this problem.

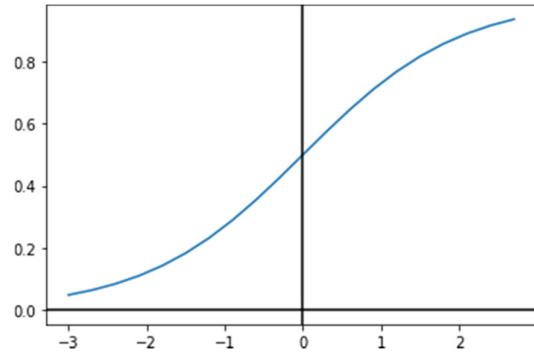


Figure 3: Sigmoid Function

TANH FUNCTION

The appropriate function is the hyperbolic tangent. The Tanh function is symmetric to the origin unlike the sigmoid function, which is similar to it. This alters the signs of the outputs of the layers that came before it, which are then utilized as inputs for the layer that follows. It might be stated” as follows:

$$f(x) = 2\text{sigmoid}(2x)-1$$

The “Tanh[11]” function may take on any value between -1 and 1, and it is continuous and differentiable. The gradient of the Tanh function is more abrupt when compared to the gradient of the sigmoid function. Tanh is preferred over the sigmoid function because it is zero-centered and has gradients that can alter in any direction.

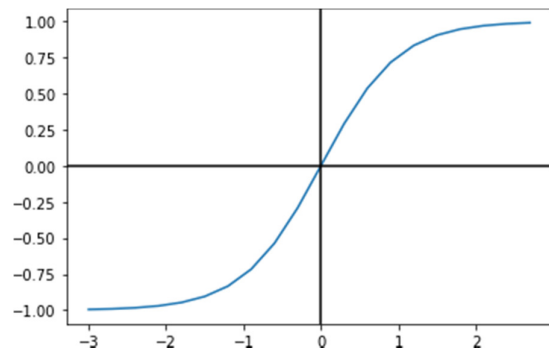


Figure 4: Tanh Function

ACTIVATION FUNCTIONS BASED ON LOGISTIC SIGMOID AND TANH

In “the early phases of neural networks, conventional AFs[19] like Logistic Sigmoid and Tanh were often employed. Despite their enormous output, these AFs made deep neural network training difficult. There have also been other efforts to enhance these AFs for other networks. The parametric, monotonic, smooth, and limited

features of the Tanh based and Logistic Sigmoid AFs are contrasted. The limitations of Tanh's[13] output range and zero gradient are addressed by using sTanh, a scaled version of the hyperbolic” tangent.

$$sTanh(x) = A \times Tanh(B \times x) \dots\dots\dots(1)$$

comprising “the [A, A] output range. The parametric sigmoid function (PSF) has been defined as a continuous, differentiable, and bounded” function.[17]

$$PSF(x) = \frac{1}{(1 + e^{-x})^m} \dots\dots\dots(2)$$

The “gradient flow is shown to get better when m is increased to a larger value, which is a hyperparameterThe symmetry of the recently formed features is preserved by investigating the sum of the shifted log sigmoid for use as an AF

SOFTMAX FUNCTION

Another type of Activation function is softmax. It’s basically a mathematical function which is used in neural networks output layer to calculate and evaluate the probability distribution. And it also used in classification problem resolutions. And The output of this function is in the range of 0 and 1. This function is applied in every class neuron. The softmax function is in the form of:

$$f(x) = \frac{\exp(x)}{\sum \exp(x)}$$

MISH ACTIVATION FUNCTION

The Mish activation function, an unique self-regularized non-monotonic activation function that has the mathematical formula

$$f(x) = x \tanh(\text{softplus}(x)).$$

As a result, this function has many similarities to Swish and GELU, including the unbounded positive domain, bounded negative domain, non-monotonic shape, and smooth derivative. According to the author[24] of this study, Mish therefore yields better empirical results than Swish, ReLU, and Leaky ReLU under the majority of experimental conditions.

SWISH

Swish[15] activation functions apply the properties of one sided boundedness at zero with smoothness and also apply non-monotonicity, which may assume a role in the noticed adequacy of Swish and comparable AF(activation capacities). In the experimental result the author[15] shows that Swish will in general work in a way that is better than ReLU on more profound models across various testing datasets.

$$f(x) = x.\text{sigmoid}(x)$$

RECTIFIED LINEAR UNIT (RELU)FUNCTION

The rectified linear measure (ReLU) function, one of the most popular AFs in DL models, is a fast-learning AF that promises to deliver cutting-edge performance with stellar results. In deep learning, ReLU functions outperform other AFs such as the sigmoid and tanh functions in terms of performance and generalization.

For making it easier to optimize by using gradient descent method we use Relu function and it is a linear function that holds the properties of linear models.

The ReLU function applies in each input element and the output of this function is the same as the input values if it is greater than zero otherwise it is set as 0. So, the ReLU is

$$f(x_j) = \begin{cases} \max(0, x_j) \\ x_j, \text{ if } x_j \geq 0 \\ 0, \text{ if } x_j < 0 \end{cases}$$

Table.1.The following table provides a summary of the activation functions that are based on the Rectified Linear Unit

Name	Parametric	Monotonic	Smooth	Bounded
Rectified Linear Unit (ReLU), 2010	Not present	Present	Not present	For Negative Inputs
Leaky ReLU (LReLU), 2013	Not present	Present	Not present	Not present
Parametric ReLU (PReLU), 2015	**	Present	Not present	Not present
Randomized ReLU (RReLU), 2015	Not present	Present	Not present	Not present
Concatenated ReLU (CReLU), 2016	Not present	Present	Not present	For Negative Inputs
Bounded ReLU (BReLU), 2016	Not present	Present	Not present	Yes
Parametric Tanh Linear Unit (PTELU), 2017	Present	Present	Present	For Negative Inputs
Flexible ReLU (FReLU), 2018	Present	Present	Not present	For Negative Inputs
Elastic ReLU (EReLU), 2018	Not present	Present	Not present	For Negative Inputs
Randomly Translation ReLU (RTRReLU), 2018	Not present	Present	Not present	For Negative Inputs
Dual ReLU (DualReLU), 2018	Not present	Present	Not present	Not present
Paired ReLU (PairedReLU), 2018	Present	Present	Not present	Not present
Average Biased ReLU (ABReLU), 2018	Not present	Present	Not present	For Negative Inputs
Natural-Logarithm ReLU (NLReLU), 2019	Not present	Present	Not present	For Negative Inputs
Multi-bin Trainable Linear Units (MTLU),	Present	Not present	Not present	Not present

2019				
Lipschitz ReLU (L-ReLU), 2020	Present	Depends upon ϕ and η	Depends upon ϕ and η	Depends upon ϕ and η

Relu 2010 Not even a smooth graph predictor and parametric but it is monotonic similarly to the remaining Functions detail.

There are no. of Relu transformation functions also like dynamic Rectified Linear Unit[18],integral Relu[20], ReS[21]. As per the analysis report Relu Function provides better result for positive values but still a problem with negative values so in this analysis firstly detail focus on sigmoid and tanh function for evaluating their performance.

DIFFERENCES BETWEEN SIGMOID AND THE TANH ACTIVATION FUNCTION

The “way the two functions' gradients behave is a key distinction between them.

The “gradients of the sigmoid (red) and tanh (blue) activation functions are plotted” below:

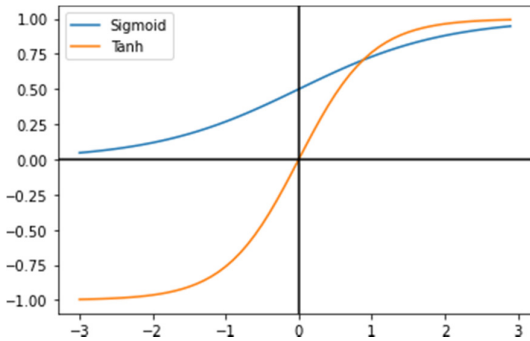


Figure 5. The sigmoid and the tanh activation function

The data tend to cluster around the number zero when we apply these activation functions to a neural network. This is so because the network's origin is zero. As a result, we must concentrate the vast majority of our efforts on how each gradient behaves when it is close” to zero.

It “has been pointed out to us that the gradient of the tanh function is around four times bigger than the gradient of the sigmoid function. Throughout the whole training phase, using the tanh activation function produces higher gradient values and larger updates to the network weights. Therefore, the tanh activation function is the one we should employ if we want our gradients to be reliable and our learning steps to be significant. Another distinction between these two methods is the output of tanh, which is symmetric about zero and leads to a faster convergence. Another distinction between the two methods” is this.

4. COMPARATIVE PARAMETERIZED RESULTS OF SIGMOID AND TANH DATA SET

For implementing here we used numeric or mnist dataset. which are in the form of images some sample images shown below in figure 6. The dataset is grouped together to identify numbers.

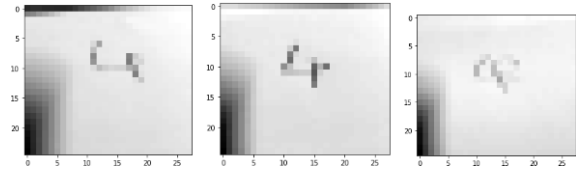


Figure 6: Sample image of data set

In the FEB data set, 48 by 48 pixels is the resolution of each image. The effectiveness of the model will be affected by the comparison of several optimizers in the test's subsequent stage. It assists in identifying the optimizer with the best all-around performance. The validation accuracy and validation loss figures are calculated to assess the efficiency of the different optimizer and hyper-parameter settings. In order to assess how effective our proposed constrained CNN is at spotting instances of picture tampering, we ran a number of experiments and ran some” analyses.



Figure 7: Sample images from FER data set

4.1 RESULTS AND DISCUSSION

The “technical feasibility of the proposed mixed activation function was investigated through an experimental inquiry.

Firstly discuss about MNIST dataset result so as a resultant in CNN first convolution layer use Tanh Activation function and check the accuracy of classification over this model and after that us Relu and tanh in fully connected and output layer and check the accuracy of classification here input shape is used as 28*28 also uses the same optimizer as Adam in this model .

and then compared with sigmoid with relu Activation in CNN model on same input shape

so the result of this experiment is :

Table 2: The findings obtained after analyzing the relative effectiveness of the activation functions

Activation Function	Accuracy	Validation loss
Sigmoid	0.98	0.04

Tanh	0.11	2.3
------	------	-----

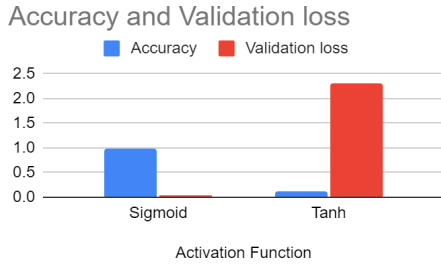


Figure 8: Graph of both activation function

In this result different Activation shows the validation accuracy on changing activation over the 5 training epochs for the model.

Table 3:resulting table of validation accuracy on changing activation over the course of 10 training epochs for the model

Activation Function	Accuracy	Validation loss
Sigmoid	0.98	0.05
Tanh	0.23	2.3

as the result shows that comparare with tanh and sigmoid function better result provide sigmoid with minimum loss with high accuracy

Here accuracy and loss checked on different epochs first checked with 2 epochs and after that checked on 10 epochs and its shows like minor changes are discovered here in the accuracy of tanh while increasing the epochs.

Secondly discuss about FEB dataset result

The CNN “layer’s activation functions were alternated in the first stage of the experiment while the remaining hyper-parameters were left at their default values.

Table.4. The findings obtained after analyzing the relative effectiveness of the activation functions

Activat ion Functi on	Train Accuracy(%)	Validation Accuracy (%)	Validati on loss
Sigmoid	43.12	51.34	2.23
Tanh	33.36	42.01	2.111

Using ten boards for different activation functions, we displayed three separate graphs for validation accuracy, validation loss, and training accuracy. From looking at these graphs, we can see that the Sigmoid activation function had the highest accuracy among the Tanh activation functions. But the problem arises with validation loss.

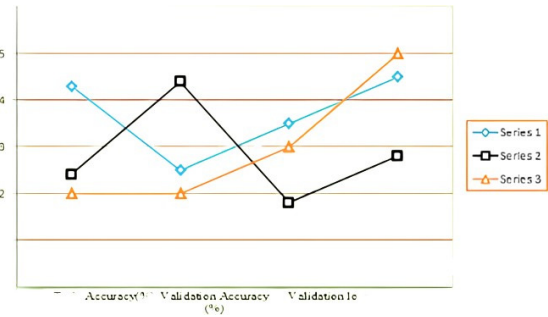


Figure 5 shows the resulting graph of validation accuracy on changing activation over the course of 30 training epochs for the model.

These “variances have a significant impact on the model’s accuracy and the value it loses due to variations in activation functions. If we pay attention to the Tanh and Sigmoid activation functions, we might be able to practically detect the difference in accuracy” and loss value.

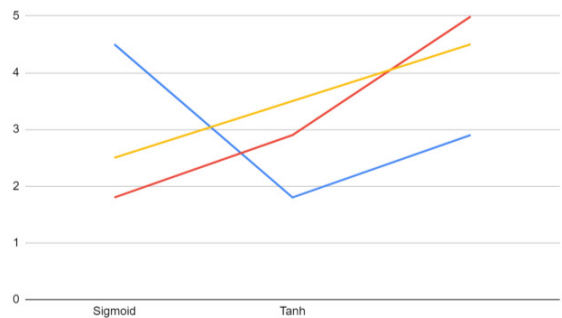


Figure 6 shows the resulting graph of validation loss as a function of changing activation when the model was being trained for 30 epochs.

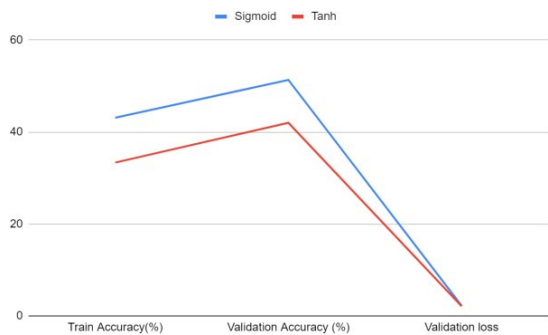


Figure 7 shows the resulting graph of training accuracy on changing activation over the course of 30 training epochs for the model.

The “optimizer’s hyper-parameter must be changed in the next step while keeping the activation level constant. Three different optimizers—Adam, RMSprop, and SGD—are used in this test, and they are each applied to one of three possible activation functions. Table 3 shows the accuracy as well as the loss value that was seen while keeping the activation function Swish and changing” the optimizers.

Table.3.The results that were obtained after comparing several optimizers while maintaining the Sigmoid activation function for a total of 30 epochs.

Optimizers	Train Accuracy (%)	Validation Accuracy (%)	Validation loss
Adam	72.64	81.90	1.112
RM Sprop	72.46	81.30	1.034
SGD	38.94	43.29	2.234

In conclusion, table 4 below shows the accuracy and loss value that were seen while keeping the Sigmoid activation function and changing the” optimizers.

Table.4.The results obtained during the comparison of several optimizers while maintaining the tanh activation function for a total of 30 epochs

Optimizers	Train Accuracy(%)	Validation Accuracy (%)	Validation loss
Adam	75.43	78.51	1.104
RMSprop	70.99	73.72	1.757
SGD	35.28	46.22	2.215

The data in the three tables above show that, depending on the activation function they are connected to, various optimizers have varying consequences. The findings were enhanced when RM Sprop and the Sigmoid activation function were combined, and the validation loss was estimated to be 1.034. However, if Adam is factored into the tanh computation, the loss value is considerably better, coming out” at 1.104.

4. CONCLUSION

This article provides a comprehensive overview of the CNN model, which is often used for image classification applications. The complexity of this model leaves a lot of room for improvement, and a number of researchers have put up a variety of ideas to boost the accuracy of different CNN models. the capacity of the activation functions to enhance pattern learning in the data justifies their use in the hidden layers of neural networks, where automated feature recognition is performed. The activation functions can also be used for categorization in a number of machine learning fields. The resultant observed that the accuracy of Sigmoid is better than the Tanh in both MNIST and FEB dataset with different epochs like 10 and 30 training epochs. The activation function is a crucial element of convolutional neural networks, which can map out non-linear” properties

REFERENCES

1. S. Albawi, et al., “Zawi, S,” Understanding of a Convolutional Neural Network International Conference on Engineering and Technology (ICE T), vol. 2017, 2017.
2. R. Chauhan et al., “Convolutional neural network (CNN) for image detection and recognition” First International Conference

- on Secure Cyber Computing and Communication (ICSCCC), vol. 2018, 2018 [doi:10.1109/ICSCCC.2018.8703316].
3. A. Ajit et al., “A review of convolutional neural networks” International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), vol. 2020, 2020 [doi:10.1109/ic-ETITE47903.2020.049].
4. D. Dai, “An introduction of CNN: Models and training on neural network models,” International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR), vol. 2021, 2021 [doi:10.1109/ICBAR55169.2021.00037].
5. J. Feng and S. Lu, “Performance analysis of various activation functions in artificial neural networks,” J. Phys. Conf. Ser., vol. 1237, no. 2, p. 022030, 2019 [doi:10.1088/1742-6596/1237/2/022030].
6. A.-F. Karam et al., “Applying convolutional neural networks for image detection” International Conference on Smart Applications, Communications and Networking (SmartNets), vol. 2019, 2019 [doi:10.1109/SmartNets48225.2019.9069750].
7. A. Rasamoelina et al., “A review of activation function for artificial neural network,SAMI” 18th World Symposium on Applied Machine Intelligence and Informatics, vol. 2020. IEEE, 2020.
8. Y. Yu, et al., “Rmaf: Relu-memristor-like activation function for deep learning,” IEEE Access, vol. 8, pp. 72727-72741, 2020 [doi:10.1109/ACCESS.2020.2987829]
9. S. Sharma et al., “Activation functions in neural networks,” Towards Data Sci., vol. 6, no. 12, pp. 310-316, 2017.
10. X. Li et al., Combine ReLU With Tanh, vol. 1. IEEE. IEEE,2020[doi:10.1109/ITNEC48623.2020.9084659].
11. B. L. Kalman and S. C. Kwasny, “Why tanh: Choosing a sigmoidal function.” [Proceedings, 1992], IJCNN International Joint Conference on Neural Networks, vol. 4. IEEE, 1992 [doi:10.1109/IJCNN.1992.227257].
12. H. K. Kwan, “Simple sigmoid-like activation function suitable for digital hardware implementation,” Electron. Lett., vol. 28, no. 15, pp. 1379-1380, 1992 [doi:10.1049/el:19920877].
13. S. Marra et al., “Tanh-like activation function implementation for high-performance digital neural systems,” Physiol. IEEE, 237-240, 2006 [doi:10.1109/RME.2006.1689940].
14. B. Karlik and A. Vehbi Olgac, “Performance analysis of various activation functions in generalized MLP architectures of neural networks,” Int. J. Artif. Intell. Expert Syst., vol. 1, no. 4, pp. 111-122, 2011.
15. Ramachandran, Prajit, B. Zoph, and Quoc v. Le, ‘Swish: a self-gated activation function.’ arXiv Preprint ArXiv:1710.05941 7.1, vol. 5, 2017.
16. B. Ding et al., “Activation functions and their characteristics in deep neural networks,” 2018, Chinese Control and Decision Conference (CCDC). IEEE [doi:10.1109/CCDC.2018.8407425].
17. C. Nwankpa, et al., ‘Activation functions: Comparison of trends in practice and research for deep learning.’ arXiv Preprint ArXiv:1811.03378, 2018.
18. Si et al., “A dynamic ReLU on neural network” 13th Dallas Circuits and Systems Conference (DCAS). IEEE. IEEE, 2018.
19. M. M. Lau and K. H. Hann Lim, “Review of adaptive activation function in deep neural network.” IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES). IEEE, 2018 [doi:10.1109/IECBES.2018.8626714].

20. K. C. Kirana, et al., "Improved neural network using integral-RELU based prevention activation for face detection" International Conference on Electrical, Electronics and Information Engineering (ICEEIE), vol. 6. IEEE, 2019 [doi:10.1109/ICEEIE47180.2019.8981443].
21. O. Sharma, 'A new activation function for deep neural network.' 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon). IEEE, 2019.
22. Y. Guo, et al., "Algorithm research on improving activation function of convolutional neural networks," 2019, Chinese Control and Decision Conference (CCDC). IEEE.
23. Z. Qiumei et al., "Improved convolutional neural network based on fast exponentially linear unit activation function," IEEE Access, vol. 7, pp. 151359-151367, 2019 [doi:10.1109/ACCESS.2019.2948112].
24. D. Misra, 'Mish: A self regularized non-monotonic activation function.' arXiv Preprint ArXiv:1908.08681, 2019.