Impact Of Activation Function In Deep Learning

Md. Safaet (id: 18-38704-3)[a], Abul Kashem Nibir (id: 18-38660-3)[a], Jahid Hasan Shamim (id:18-38425-2)[a]

[a]*Department of Computer Sciences, American International University-Bangladesh*

Abstract

In present neural network models, nonlinear activation functions are used. They allow the model to build complicated mappings between the network's inputs and outputs, which are essential for learning and modeling complex data, such as images and data sets which are non-linear. Signal is communicated through synapses developed between individual neurons in Artificial Neural Networks, which are based after the workings of the human brain. The activation function decides which signal is sent from each Artificial Neural Networks node to the others that are connected, hence the value of the function can be construed as a neuron's "activity." In a neural network, an activation function defines how the weighted sum of the input is converted into an output from a node or nodes in a layer.it gives us a smooth gradient while converging. it gives us a clear prediction(classification) with 1 & 0.it gives us one of the best Normalized functions.

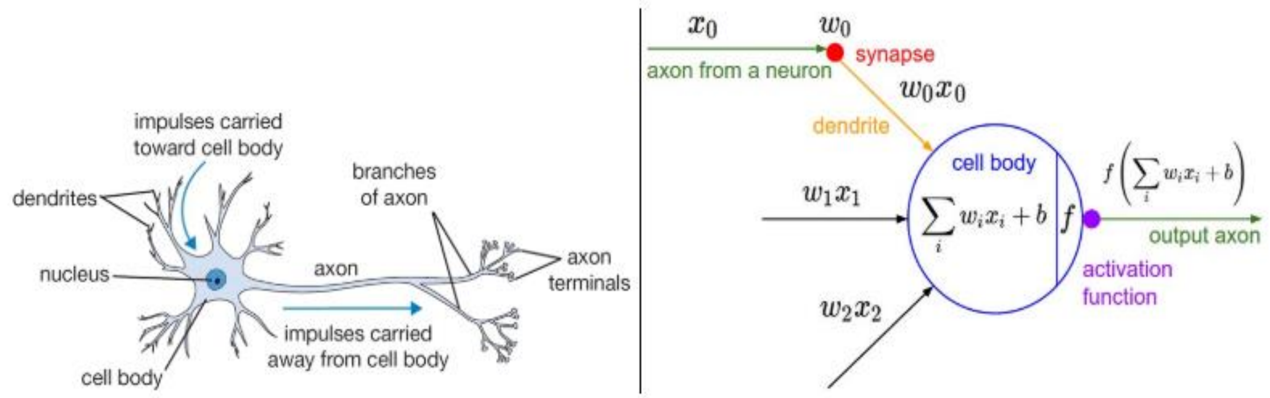
1. Introduction

The representation and recognition performance of neural networks with deep architectures has improved rapidly in recent decades resulting in significant changes in the fields of pattern recognition and machine learning processing. The following essential factors, such as the development of a more powerful model, have resulted in revolutionary improvements. Creating higher-performance technology, inventing more effective regularization techniques, etc. Developments in activation functions, for fact, have played a key role in increasing the performance of deep neural networks, and as a result, more and more effort is also being given to the study of activation functions. Concentrated activation functions, such as sigmoid and tanh, have been replaced by non-saturated peers, such as ReLU, ELU, to fix the so-called vanishing gradient and speed up the computation time in neural networks with deep architectures, such as artificial neural networks and recurrent neural networks, since Nair and Hinton proposed the rectified lin- ear units to improve the performance of Restricted Boltzmann Machines.

A concept, advantages and disadvantages of commonly used activation functions will be reviewed in this study in order to thoroughly understand the status and performance improvement of activation function in deep neural networks. Also shown will be a comparison of experimental outcomes on the MNIST dataset.

2. Literature Review

An activation function is a function that is introduced to an artificial neural network to improve it in learning complex patterns from data. A neural network is a computing system that is based on the human brain's structure. By examining examples, a neural network can "learn" to perform tasks without having task-specific instructions.



A cartoon drawing of a biological neuron (left) and its mathematical model (right).

Step function : A step function is a mathematical function whose graph resembles a sequence of steps because it is made up of a series of horizontal line segments with leaps in between. As a result, it's commonly referred to as a stairwell function.

$$u(t) = \begin{cases} 0 & t < 0 \\ 1 & t > 0 \end{cases}$$

Sigmoid function : A mathematical function with a characteristic S-shaped curve is known as a Sigmoid function. The logistic function, the hyperbolic tangent, and the arctangent are all examples of sigmoid functions.

$$\sigma(x) = \frac{1}{1+e^{(-x)}}$$

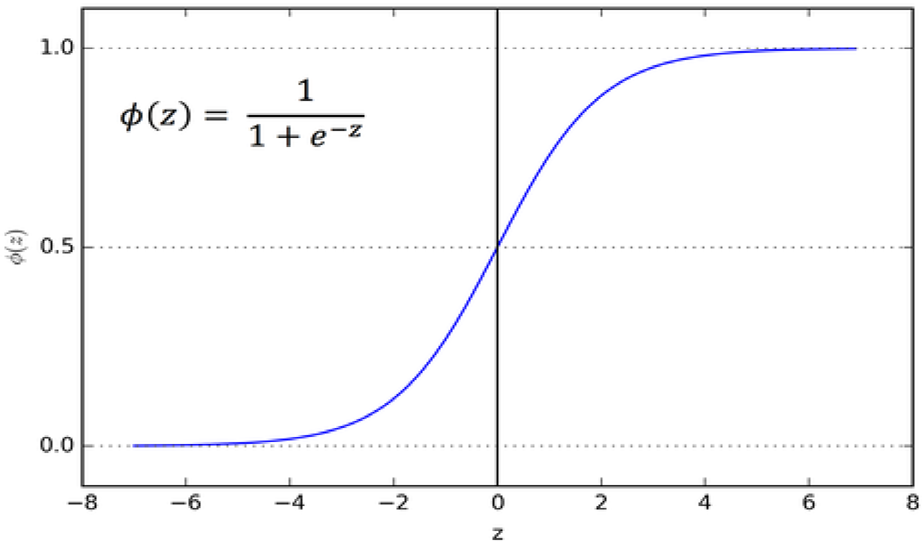
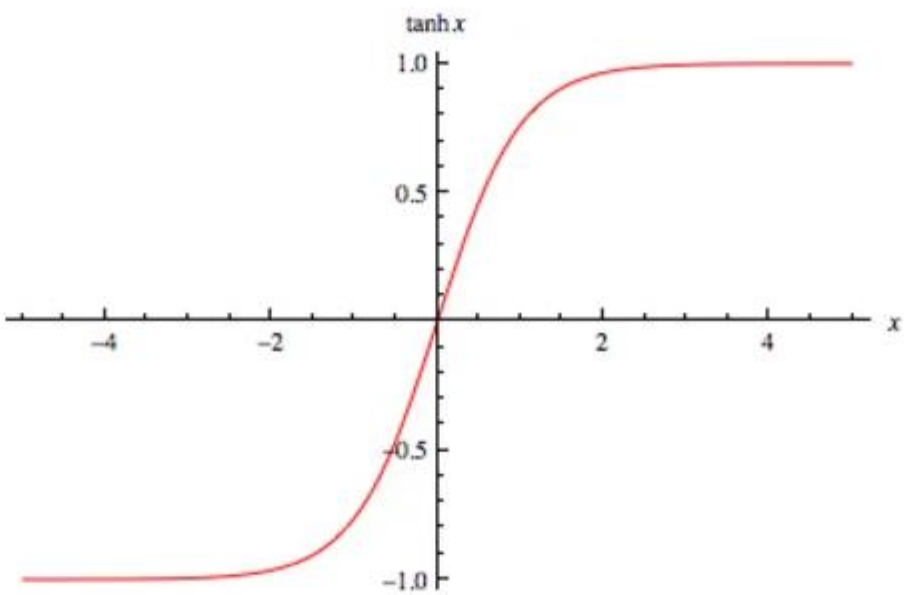


Fig: Sigmoid Function

Tanh : The ratio of the sine and cosine functions can be used to define the hyperbolic tangent function. The Tanh (also "tanh" and "TanH") function is another name for the hyperbolic tangent activation function. It's a just like the sigmoid activation function.



The rectified linear unit has been lauded as a solution to the vanishing gradient problem for neural networks in various places. That is, the activation function is max(0,x). When the activation is positive, it is clear that this is preferable to, instance, the sigmoid activation function, because its derivation is always 1 rather than an arbitrarily tiny number for big x. When x is less than 0, on the other hand, the derivation is exactly 0. In the worst-case scenario, if a unit is never activated, the weights for that unit will never change, and the unit will be permanently worthless a situation that appears to be far worse than even vanishingly small gradient.

3.Discussion

The neural networks perform regression on polynomial datasets of various degrees that were constructed using the approach described above. The neurons that are used in typical ANNs are strongly simplified models of the nerve cell [12]. In order to decrease variance, each neural network normalizes input data and sets its weights and biases to the center of the range of the relevant activation function. The most popular approach is the classical neural network, trained with the use of such algorithms as the Backpropagation or gradient ones [11]. Each neural network has a different optimizer, activation function, and hidden layer design depending on the test. Each neuron's activation would be a linear function if it were calculated just as the weighted sum of its activations. As a result, a linear function may be constructed to model the complete network's output. This is what an activation function is for. An activation function is used in machine learning to add nonlinearities into the neural network. A loss function is a function that is defined on a training set and a neural network. On a training set, a loss function may be used to determine how far off the neural networks' predictions are. The algorithm that is utilized to minimize the loss function is known as an optimizer. Gradient Descent, often known as Stochastic Gradient Descent, is the most frequent optimizer. Gradient descent reduces the loss function in a series of steps by first computing the derivative of the loss function and then altering the weights and biases in the network in the direction of the determined gradient. Stochastic gradient descent operates in the same way, but instead calculates the gradient using a single or a small number of training samples. Another optimizer is Root Mean Square Propagation, which employs momentum to discover the loss function's minimums more quickly than typical optimizers. On single featured polynomial input data, the first test was conducted to examine the performance of several neural network activation functions. First, we used the methods mentioned in II-A to build datasets in NumPy. We next standardized the data and put it into regression-optimized neural networks, as stated in II-B. We used Mean Squared Error (MSE) to assess error and ran the neural networks for 10 epochs with a batch size of 5. In the first test, we used the Adam optimization method on the neural networks, which remained constant throughout the test while the activation functions and network structure were changed. During the test, the average MSE for each activation function can be found in. A final test was conducted to compare different activation functions and optimizer combinations. To test this, we used the best activation function and the worst optimizer to equip a series of neural network designs. Then we compared this set of neural networks to a similar set with the poorest activation function and best optimizer. Future research might focus on generalizing the findings of this experiment beyond artificially manufactured polynomial datasets and comparing other activation functions and optimizers. Understanding activation functions and optimizers in the context of deep learning is vital for developing a comprehensive theory of optimal neural network construction, which is important for both computing efficiency and accuracy in deep learning applications in industry.

4.Conclusion

Deep neural networks have advanced rapidly in recent decades, particularly in computer vision and natural language processing. Along with the network's layers as you go deeper into the study, you'll notice that the training efficiency and accuracy have gotten a lot of attention, which encourages the development of activation functions. Non-saturated activation functions, such as ReLU and ELU, replace saturated activation functions like Sigmoid and hyperbolic tangent. The definition, benefits, and drawbacks of numerous prominent action functions are discussed in this work. The purpose of this work is to contribute to our understanding of development progress, attributions, and proper activation function selection.

5.Refferences

1. https://ieeexplore.ieee.org/document/9005674
2. https://ieeexplore.ieee.org/document/7182774
3. https://ieeexplore.ieee.org/document/8436869
4. https://ieeexplore.ieee.org/document/9207277
5. https://login.research4life.org/tacsgr1opg_optica_org/optica/fulltext.cfm?uri=optica-6-9-1132&id=417261
6. https://www.sciencedirect.com/science/article/abs/pii/S0893608018300078
7. https://www.sciencedirect.com/science/article/abs/pii/S1051200409001948
8. Data-Specific Activation Function Learning for Hyperspectral Image Classification | IEEE Conference Publication | IEEE Xplore
9. Complex backpropagation neural network using elementary transcendental activation functions | IEEE Conference Publication | IEEE Xplore
10. https://www.researchgate.net/publication/331220969_On_the_Impact_of_the_Activation_Function_on_Deep_Neural_Networks_Training

11. LI X., Moussa M., Areibi S., ,,Arithmetic formats for implementing Artificial Neural Networks on FPGAs", IEEE Canada, DOI: 10.1109/CJECE.2006.259201, 26 December 2006.

12. Muthuramalingam A., Himavathi S., Srinivasan E., ,,Neural Network Implementation Using FPGA: Issues and Application", International Journal of Information Technology, vol. 4, nr 2, ISSN 1305-2403, 2007.

Paper writing contribution

Name	**Section Title**
Md. Safaet (id: 18-38704-3)	Abstract,Literature Review,
Abul Kashem Nibir (id: 18- 38660-3)	Introduction, Conclusion, Literature Review,
Jahid Hasan Shamim (id:18-38425-2)	Literature Review, Discussion

References