International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT2390287

Performance Analysis of Fully Connected Parametric Activation Functions

Rashmi Awasthi[*1], **Dr. B.K. Sharma**[2]
[*1]Research Scholar, Department of Computer Application, Mandsaur University, Mandsaur, Madhya Pradesh, India
[2]Associate Professor, Department of Computer Application, Mandsaur University, Mandsaur, Madhya Pradesh, India

ARTICLEINFO

Article History:

Accepted: 10 April 2023
Published: 25 April 2023

Publication Issue
Volume 10, Issue 2
March-April-2023

Page Number
613-617

ABSTRACT

Modern studies have shown that the choice of activation function can significantly affect the performance of several learning methods and deep networks. The activation function plays an important role in solving the nonlinear problem, and various nonlinear fully connected activation functions have been studied. In this paper, we propose a combined parametric activation function that can improve the performance of a fully connected Artificial intelligence methods and neural network. Combined parametric activation functions can be created by simply adding parametric activation functions. The parametric activation function is a function that can be optimized in the direction of minimizing the loss function by applying a appropriate parameter that converts the scale and location of the activation function according to the input data. The development of Artificial Neural Networks (ANNs) has achieved a lot of fruitful results so far, and we know that activation function is one of the principal factors which will affect the performance of the networks. In this paper, we discussed about the impact of varying model width and depth on robustness, the impact of using learnable parametric activation functions (PAFs).

Keywords : Combined parametric functions, Artificial Intelligence, Artificial Neural Networks (ANN), Machine Learning, Deep Learning.

I. INTRODUCTION

It has been proved that the growth of computer knowledge and learning has promoted the considerable improvement in many different fields: such as Data mining, Machine learning, Artificial Intelligence, pattern recognition, and other core research areas. The development of computer technology has promoted the considerable progress of many different fields, such as artificial intelligence, pattern classification, machine learning and other research fields. This paper highlights the different types of AFs and their evolution over the years. The AF research and applications in deep architectures, used in different applications has been a core research field. Activation Functions are functions used in neural networks (NNs) to compute the weighted sum of inputs and biases, which decides if a neuron can be fired or not. It manipulates the presented data through some gradient processing usually gradient descent to produce an output for the NN, that contains the parameters in the data. These AFs are often referred to as a transfer function in the literature, with early research results by [5], validating categorically that a proper choice of AF improves results in NN computing.

Activation functions are mathematical equations that define how the weighted sum of the input of a neural node is transformed into an output, and they are key parts of an artificial neural network (ANN) architecture. The successful applications of machine learning techniques rely on the approximation functions that are learned from the underline data of problems. The artificial neural network (ANN) is one of the machine learning techniques that can be used as a universal approximator for complex data [1]. The use of representation learning, which is the technique that allows machines to discover relationships from raw data, needed to perform certain tasks like classification and detection. Between 2013 and 2015, some researchers have proposed improved activation functions based on the phenomenon of "necrosis" that brought by ReLu function, such as: leaky ReLus, ELU, PReLu, tanh-ReLu, and so on. In the practical application of the activation function model, there is a lot of room for improvement due to its complex structure. Many researchers have made a lot of effective ways to improve the recognition results of the CNN model. Some studies have been done on image classification methods.

II. LITERATURE SURVEY

We explore the impact of activation function shape on the robustness of adversarially trained models through PAFs parameterized by a single parameter controlling shape. We choose a set of PAFs which allow us to vary behaviour on negative inputs, inputs near zero, and positive inputs. These PAFs include both pre-existing PAFs such as PReLU and PELU as well as PAFs that we introduce (ReBLU and PReLU+). We will look at the importance of activation functions in the neural network and then we will compare different activation functions. Later, I will show the results I got on implementing different activation functions in image classification problems. Some studies have been done on the design of adaptive learning rate [7]. There are other studies which have been done on the design in the dropout layer.

III. CLASSIFICATIONS OF ACTIVATION FUNCTIONS

Activation Functions can be basically divided into two types, linear activation functions and non-linear activation functions. Where linear activation functions maintain a constant, non-linear activation functions create more variation which utilizes the build of the neural network.

A. Bounded Functions

The bounded activation functions (units) are those functions that are designed based on the properties of universal approximation theorem. Logistic is one of the most common activation functions of this category. Logistic is non-linear bounded and limits the output to be in the range between zero and one [2]. It has a common S-shaped curve defined according to following equation:

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

This function has been widely used in the early work of feed-forward neural networks. It is a differentiable and monotonically increasing function. Its outputs are smoothly continued with positive derivatives everywhere.

B. Rectifier Functions

Rectifier Functions: Rectified Linear units are firstly used within Restricted Boltzmann machines. Then, the Rectified Linear Units (ReLU) was introduced by Glorot et al. in 2011 and became the prevailing activation function used in deep neural nets, especially in Convolutional neural network [10]. ReLU is characterized by its simplicity, computationally cheap, and fast convergence compared to the other activation functions belonging to the bounded groups.

$$f(x) = \begin{cases} x, & x \geq 0 \\ ax, & x < 0 \end{cases} \qquad (2)$$

However, ReLU is not continuously differentiable, which causes some problems in gradient-based optimization.

C. Non Linear below Functions

This category includes those activation functions which have linear and unbounded above units but not rectifier. Also, such functions map the negative inputs with negative values using non-linear functions. An example of such a category is the Exponential linear units (ELU) is defined according to following function:

$$f(x) = \begin{cases} x, & x \geq 0 \\ \alpha e^{x-1}, & x < 0 \end{cases} \qquad (3)$$

In contrast to the Sigmoid family, ELU is a zero-centered activation function.

D. Non Linear and Unbounded Above Functions

In contrast to the previous categories' previous activation functions, this category contains those activation functions that do not have identity transformations. Hence, those functions may need a high computational cost, and voiding the exploding / vanishing problem is not guaranteed [9]. It produces outputs in the range from 0 to ∞ and is defined by following function:

$$f(x) = \log_2(1 + e^x) \qquad (4)$$

The Sigmoid AF is sometimes referred to as the logistic function or squashing function in some literature.

IV.WHY DO WE NEED NON LINEARITY IN NEURAL NETWORK

A non-linear activation function allows the stacking of multiple (2) s of neurons to create a deep neural network, which is required to learn complex data sets with high accuracy [4]. In this section we try to visualize the activation effect of need for non linearity in neural network at each of the given layer. Consider the following network in the diagram without any activation function.

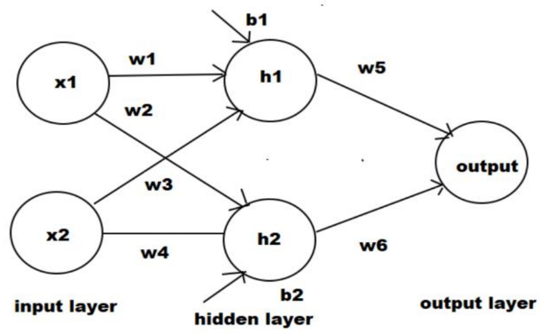

Figure1. Basic Network Diagram

Here , There are two layers in given directed network diagram, and each node is proper labelled. from the first layer we get:

$h1 = w1 \times x1 + w3Xx2 + b1$

&

$h2 = w2Xx1 + w4Xx2 + b2.$

From the second layer, we get:

$output = w5Xh1 + w6Xh2$

or, $output = w5X(w1Xx1 + w3Xx2 + b1) + w6X(w2Xx1 + w$

or, $output = (w5Xw1 + w6Xw2)Xx1 + (w5Xw3 + w6Xw4)X$
$w5Xb1 + w6Xb2$

or, $output = W1Xx1 + W2Xx2 + B,$

where, $W1 = w5Xw1 + w6Xw2, W2 = w5Xw3 + w6Xw4$ & B
$w6Xb2.$

So the output of the above network is similar to some linear regression models. This will not be the case if we use an activation function. Activation functions will introduce some non-linearity in the model. This added non-linearity makes neural network differ from any simple linear regression model [8]. It gives the neural network the ability to solve complex problems.

In a neural network, inputs are fed into the network from the input layer. In the neurons of the next layer, a weighted sum of the inputs is calculated and a bias is added to the sum. This sum is then passed through an activation function. The output of this activation function is the input of the next layer. The basic process carried out by a neuron in a neural network is:

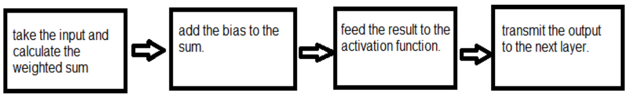

Figure2. Activation Function Pipeline

If we used Sigmoid activation function, It is derivable at every point. This is a desired property for any activation function. That means we can find the gradient of the sigmoid curve at any two points which will help in back propagation of error in the model [3]. The output values are bound between 0 to 1. This means the activations will not be blown up. This functions gives clear predictions at points. This means that for input values greater than 2 or less than -2, the output is brought close to 1 or 0 respectively. In above pipeline the identification of input and its weight is primary task and output transmission is the final task.

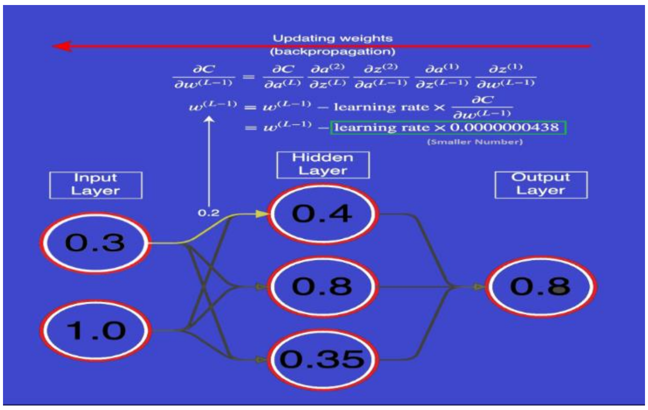

Figure 3. Network Diagram with Updating Weights

One of the major disadvantages of using sigmoid is the problem of vanishing gradient. See in the image, for a very high or very low value of x, the derivative of the sigmoid is very low. The highest value of the derivative is 0.25. The outputs aren't zero centred [6]. The output of this activation function always lies within 0 & 1 i.e. always positive. A zero centred function would be a function where the outputs are sometimes less than 0(negative) and sometimes greater than 0(positive). As a result, it would take a substantially longer time to converge. Whereas zero centred function helps in fast convergence.

V. CONCLUSION

An Activation Function decides whether a neuron should be activated or not. This means that it will decide whether the neuron's input to the network is important or not in the process of prediction using simpler mathematical operations. This paper provides a comprehensive summary of AFs used in neural networks and most importantly, highlights the current trends in the use of these functions in practice for the development of the performance analysis. In this paper we visualize the activation effect of need for non linearity in neural network at each of the given layer. By visualizing the activations, we can have an essence of what's going on inside a neural network. We explore the impact of activation function shape on the robustness of adversarially trained models through PAFs parameterized by a single parameter controlling shape.

VI. REFERENCES

[1] C. Nwankpa, W. Ijomah, A. Gachagan and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning" in arXiv:1811.03378, 2018, Available : http://arxiv.org/abs/1811.03378.

[2] S. Hayou, A. Doucet and J. Rousseau, "On the impact of the activation function on deep neural networks training" in arXiv:1902.06853, 2019, [online] Available : http://arxiv.org/abs/1902.06853.

[3] Bingham, W. Macke, and R. Miikkulainen. Evolutionary optimization of deep learning

activation functions. In Genetic and Evolutionary Computation Conference (GECCO '20), July 8–12, 2020, Cancún, Mexico, 2020.

[4] Elfwing, E. Uchibe, and K. Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural Networks, 107:3–11, 2018.

[5] N. Y. Kong and S. W. Ko, "Performance improvement method of deep neural network using parametric activation functions," Journal of the Korea Contents Association, Vol.21, No.3, pp.616-625, 2021.

[6] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall. Activation functions: Comparison of trends in practice and research for deep learning. arXiv:1811.03378, 2018.

[7] Keras Team. Keras Documentation: Reuters Newswire Classification Dataset. Accessed: Sep. 2, 2020. [Online]. Available: https://keras.io/api/datasets/reuters/.

[8] T. Carneiro, R. V. Medeiros Da Nobrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, and P. P. R. Filho, "Performance analysis of Google colaboratory as a tool for accelerating deep learning applications," IEEE Access, vol. 6, pp. 61677–61685, 2018.

[9] B. Karlik and A. V. Olgac, "Performance analysis of various activa- tion functions in generalized mlp architectures of neural networks," International Journal of Artificial Intelligence and Expert Systems, vol. 1, no. 4, pp. 111–122, 2011.

[10] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," arXiv preprint arXiv:1505.00853, 2015.

Cite this article as :

Rashmi Awasthi, Dr. B.K. Sharma, "Performance Analysis of Fully Connected Parametric Activation Functions", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 2, pp.613-617, March-April-2023. Available at doi : https://doi.org/10.32628/CSEIT2390287

Journal URL : https://ijsrcseit.com/CSEIT2390287