Survey on Activation Functions: A Comparative Study between state-of-the-art Activation Functions and Oscillatory Activation Functions

Prahitha Movva[1]

[1]Indian Institute of Information Technology, Sri City
[1]*prahitha.movva03@gmail.com*

Abstract

Activation functions are extremely important for constructing a neural network. They assist in determining which neurons in each layer will be triggered. The main function of the activation function is to introduce non-linearity into the network. Because the majority of real-world problems are complex and non-linear, we need activation functions for the network to solve them. As a result, selecting the most appropriate activation function based on the problem statement is crucial. This paper will go over some of the most commonly used activation functions such as ReLU, Sigmoid, Swish, Mish, and so on. It will also compare these traditional activation functions with oscillatory activation functions like GCU, SQU, DSU, etc that have been inspired by biological neurons.

1 Introduction

Deep learning has made tremendous progress in recent years in solving a wide range of problems such as object detection, semantic segmentation, anomaly detection, pattern recognition, and many others. Neural networks, also called artificial neural networks, are a means of achieving deep learning. These neural networks have been inspired by what actually happens in the human brain and each of the aforementioned problems may employ a different type of neural network depending on the features that it has to learn from the data. However, the fundamentals of training a network remain the same: weight initialization, activation function selection, hyperparameter tuning, and optimization. The choice of the activation function at different layers in a neural network is critical (especially when the network is not complex and deep) because it determines how the data is presented to the next layer. It also controls how bounded or unbounded the data being sent to the network's next layer is.

Some common properties of activation functions[4][8] are as follows:

- Nonlinear: Activation functions have to be nonlinear because a combination of any finite number of linear equations is equivalent to a single linear function. As a result, they will be limited to solving linearly separable problems, despite the fact that patterns in real-world problems are not always expected to be linear. Furthermore, adding non-linearity improves the training convergence and adds depth to the network.
- Differentiable: An activation function must be differentiable because the gradient of the loss function, which includes the activation function, is calculated during back-propagation.
- Continuous: In order to be differentiable, a function must be continuous. Because activation functions must be differentiable, it follows that continuity is also a required property.

Preprint. Under review.

- Bounded: If the function is not bounded, the output value may explode after passing through a series of layers.
- Zero-centered: A function is said to be zero-centered when its range contains both positive and negative values. If an activation function is not zero-centered, then the mean is either shifted towards positive or negative values and can lead to saturation and numerical accuracy problems in subsequent layers.
- Low computational cost: The calculation involved in gradient descent itself is very time-consuming. So when the activation functions have low computational cost, it requires less time to get trained.

The fundamental limitation of all networks is that individual neurons can only exhibit linear decision boundaries. Hence there is a need for multilayer neural networks with non-linear activation functions to achieve non-linear decision boundaries.

Because of their biological plausibility, the majority of activation functions used today are non-oscillatory and increase monotonically. However, recent research has discovered biological neurons with oscillating activation functions in layers two and three of the human cortex that are capable of learning the XOR function individually, which earlier required a minimum of 3 neurons.

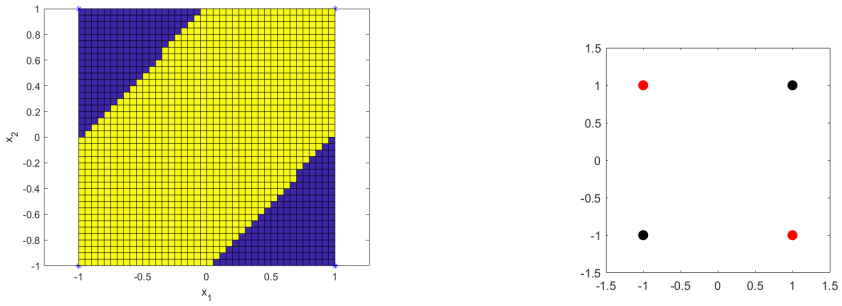

Figure 1: The XOR problem. Points in yellow were assigned a class label of +1 and points in blue were assigned a class label of -1.

This paper talks about the developments in the area of activation functions in neural networks and presents insights on each of them.

2 Literature Review

An artificial neural network is composed of a large number of interconnected working units known as perceptrons or neurons. A perceptron is composed of four components: input node, weight vector, activation function, and an output node. The third component, the activation function, is what helps in determining which neurons in each layer will be triggered.

Activation functions are primarily classified on the possibility of modifying the activation function shape during the training phase[1]. These can broadly be classified into two categories:

- Fixed shape activation functions: All the classic and rectifier-based activation functions used in neural networks such as step function, sigmoid, tanh, ReLU, LReLU, etc fall under this category
- Trainable activation functions: The idea behind these is to search for a good function shape based on the knowledge from the training data. That is, involving the activation function in the learning process along with the other parameters such as weights and biases[3]. However, they also mention that many trainable activation functions can be reduced to classical feed-forward neural sub-networks.

The following sections discuss the advantages and some limitations of the different activation functions based on the above taxonomy.

2.1 Fixed shape activation functions

For many years, bounded activation functions such as sigmoid or hyperbolic tangent have been the most used activation functions for neural networks. These are commonly used when it is useful to restrict the activation value within a certain range. The outputs of biological neurons are observed to saturate for inputs beyond a certain threshold, hence sigmoidal neurons were introduced to capture this behavior. The sigmoidal saturating activation functions are also interpretable as they give a binary yes/no decision. Furthermore, their success in the early years could also be attributed to the excellent results that they gave in shallow network architectures. However, they suffer from the vanishing gradient problem when used with networks that have many layers. Another problem is that they do not follow the rule of zero-centrality[10] which is one of the necessary properties for a good activation function.

ReLU activation functions help avoid the vanishing gradient problem[9] caused by the above saturating activation functions. They also assist in faster training as they do not saturate for a wider range of inputs and compute exponentials and divisions. However, due to its non-differentiability at zero when there is a large negative bias, it leads to stagnation in learning in the subsequent layers.

Leaky ReLU was an attempt to solve the problems of ReLU by allowing information to flow when x < 0. It also suffers from the exploding gradient problem during front propagation if the learning rate is set too high[5]. However, it performs identically to traditional rectifiers, resulting in a negligible impact on network performance.

The softplus activation function is a smooth approximation of ReLU to address the issue of non-differentiability at zero. However, on experimentation, it was observed that ReLU performed better on supervised training than the softplus function.

ELU introduces a variable α to control the value for negative inputs and push the mean closer to zero, which allows for a faster learning phase. However, it comes with the cost of setting an additional hyperparameter.

Mish is one of the recent activation functions and many experiments suggest that Mish works better than ReLU, sigmoid, and even Swish. Being unbounded above, bounded below, and continuously differentiable makes it a good activation function as unbounded above implies it avoids saturation which generally causes training to drastically slow down due to near-zero gradients, and bounded below means it results in strong regularisation effects and reduces overfitting. However, it is computationally very expensive.

2.2 Trainable activation functions

In recent years, there has been a renewed interest in using trainable activation functions to improve the performance of neural networks. One such example is the Swish activation function. It has a parameter β that adjusts the shape of the activation function between the linear and ReLU functions, thereby controlling the non-linearity based on the dataset and network complexity[6]. Its simplicity and resemblance with ReLU has made it popular and it has been replaced by ReLU in many neural networks. However, when β approaches infinity, it behaves as ReLU, and when $\beta = 1$, it is equal to SiLU.

Interestingly, all the above discussed activation functions require a minimum of 3 neurons to solve the XOR problem. As per the universal approximation theorem, a neural network with one hidden layer containing a sufficient but finite number of neurons can approximate any continuous function to a reasonable accuracy, under certain conditions for activation functions (namely, using sigmoids and ReLU). Despite this ability, each neuron in a conventional neural network has a single hyperplane as its decision boundary and hence makes a linear classification.

However, recent studies show that oscillatory activation functions, inspired by the biological neuron, allow single neurons to switch classification within the interior of neuronal hyperplane +ve and -ve half-spaces allowing complex nonlinear decisions with fewer neurons. They are also said to be capable of improving gradient flow and reducing the network size. The proposed oscillatory function, growing cosine function[7], is $C(x) = x \cdot \cos(x)$ solves the XOR problem with just 1 neuron. As we can observe from the function definition, GCU has infinitely many roots, which implies that it has infinitely many hyperplanes in the input space. These hyperplanes will be a set

of uniformly spaced parallel strips where each strip is assigned a different class alternately, thereby solving the XOR problem.

GCU

$$f_1(x) = x \cdot \cos(x) \tag{1}$$

GCU also outperforms sigmoids, Swish, Mish, and ReLU on a variety of architectures and benchmarks[7]. Another property that has been observed to help mimic the biological neurons is that activation functions must closely approximate the linear function for small values. This is good as it has a regularizing effect since the network behaves like a linear classifier when initialized with small weights. Especially for GCU, this helps avoid overfitting effects. It is also computationally cheaper than other SOTA Swish, Mish activation functions.

Sine activation function also has infinitely many hyperplanes and satisfies the XOR property

Sine

$$f_2(x) = \sin(x) \tag{2}$$

Some more oscillating activation functions[8]:

Monotonic Cubic

$$f_3(x) = x^3 + x \tag{3}$$

Decaying Sine Unit (DSU)

$$f_4(x) = \frac{\pi}{2} \cdot (\operatorname{sinc}(x - \pi) - \operatorname{sinc}(x + \pi)) \tag{4}$$

Shifted Quadratic Unit (SQU)

$$f_5(x) = x^2 + x \tag{5}$$

Non-Monotonic Cubic (NCU)

$$f_6(x) = x - x^3 \tag{6}$$

Shifted Sinc Unit (SSU)

$$f_7(x) = \pi \cdot \operatorname{sinc}(x - \pi) \tag{7}$$

Though biologically inspired activation functions have played a significant role in the history of machine learning, they have been largely displaced by ReLU or similar functions in deep learning to mitigate the effects of the vanishing gradient problem during back-propagation. Nonetheless, under physiological settings, the logistic sigmoid does not represent the true input-output relationship in neural cells. This shortcoming has been addressed through bionodal root unit (BRU)[2] with input-output non-linearities that are significantly more physiologically plausible. They are also said to train faster and generalize better than ReLU and ELU with the help of differences in non-linearity across hidden layers. BRU as observed from the equations could likely have oscillatory behaviour, and could likely be in-line with the recently proposed mathematical oscillating activation functions[8] which have been tried and tested in some of the real world standardized and domain-specific datasets. This could pave way for the usage of these functions into neurocomputing, where further research could be possible based on activations inspired or derived from the biological neuron and it's properties in an active or passive environment.

References

[1] Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete. A survey on modern trainable activation functions. *Neural Networks*, 138:14–32, 2021.

[2] Gardave S Bhumbra. Deep learning improved by biological activation functions. *arXiv preprint arXiv:1804.11237*, 2018.

[3] Garrett Bingham, William Macke, and Risto Miikkulainen. Evolutionary optimization of deep learning activation functions. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 289–296, 2020.

[4] Leonid Datta. A survey on activation functions and their relation with xavier and he normal initialization. *arXiv preprint arXiv:2004.06632*, 2020.

[5] Arun Kumar Dubey and Vanita Jain. Comparative study of convolution neural network's relu and leaky-relu activation functions. In *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*, pages 873–880. Springer, 2019.

[6] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. A comprehensive survey and performance analysis of activation functions in deep learning. *arXiv preprint arXiv:2109.14545*, 2021.

[7] Mathew Mithra Noel, Advait Trivedi, Praneet Dutta, et al. Growing cosine unit: A novel oscillatory activation function that can speedup training and reduce parameters in convolutional neural networks. *arXiv preprint arXiv:2108.12943*, 2021.

[8] Matthew Mithra Noel, Shubham Bharadwaj, Venkataraman Muthiah-Nakarajan, Praneet Dutta, and Geraldine Bessie Amali. Biologically inspired oscillating activation functions can bridge the performance gap between biological and artificial neurons. *arXiv preprint arXiv:2111.04020*, 2021.

[9] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018.

[10] Tomasz Szandała. Review and comparison of commonly used activation functions for deep neural networks. In *Bio-inspired neurocomputing*, pages 203–224. Springer, 2021.