American International University- Bangladesh
Faculty of Science and Technology

Report Title: REPORT ON ACTIVATION FUNCTION

Mid Report
Spring 2022-2023

Course Name: Computer vision and pattern recognition
Section: B
Faculty Name: DR. DEBAJYOTI KARMAKER

Submitted By:

Student Name	Student Id
Aritra Bhowmick	**20-43249-1**

Report overview:

The activation function determines whether to activate the neuron. This means that the prediction process uses simpler mathematical operations to determine if a neuron's input to the network is significant. An activation function is a mathematical function that is applied to a neuron's output in artificial neural networks and machine learning in order to bring nonlinearity into the network's output. Based on information from other neurons in the network, the activation function aids in deciding whether or not a neuron should fire. I'll go through six different activation functions in this study, their mathematical foundations, benefits, and drawbacks.

1. Step function: The step function, also known as the Heaviside step function, is a simple activation function commonly used in neural networks. The output takes the value 0 for negative arguments and 1 for positive arguments. Features include:

f(x) = 0 when x<0

1 when x >= 0

The range is between (0,1) and the output is binary in nature. These types of activation functions are useful for binary schemes. If you want to classify your input model into one of two groups, you can use a binary compiler with a unit step enable function. If the value of Y is above a certain value, declare it valid. If it's less than the threshold, say it's not.

Advantages:

• The step function is simple to implement and can be computed quickly.

• It is a binary function, which makes it useful for binary classification problems.

• It is easy to interpret and understand, which makes it useful for educational

purposes.

Disadvantages:

• The step function is not differentiable at x=0, which can cause problems when

training neural networks using gradient descent algorithms.

• It is not suitable for regression problems or any task that requires the output to

be a continuous value.

• The step function can suffer from the problem of vanishing gradients, where the

gradients of the function become too small to be useful for training the network.

• The slope of the step function is zero. This makes the step function less useful during backward propagation, where the gradient of the activation function is sent for error computation to improve and optimize the result.

• Not available for multiclass classification.

Thus, despite being a helpful activation function in some situations, the step function is

not frequently used in conventional neural networks due to its drawbacks. Due to their smoother gradients and capacity to handle a larger range of issues, other activation functions, such as the sigmoid or ReLU functions, are often favored.

2. Sigmoid function:

The sigmoid function is a mathematical function that maps any input to a value between 0 and 1. The most commonly used sigmoid function is the logistic function, which is defined as:
f(x) = 1 / (1 + e^(-x))

where x is the input to the function. The sigmoid unit of the neural network. If a neuron's activation function is a sigmoid function, this ensures that the output of that unit is always between 0 and 1. Since the sigmoid function is a nonlinear function, the output of this unit will be a nonlinear function of the weighted sum of the inputs.

Advantages:

• Sigmoid functions are easy to work with and can be easily differentiated, which makes them useful in optimization algorithms like gradient descent.

• They are widely used in neural networks to map the output of a neuron to a probability distribution.

• It is a smooth function and continuously differentiable. it is non-linear. Therefore, the output is also non-linear.

• Easy to understand and easy to use. Calculating derivatives is easy

• Sigmoid functions are bounded, meaning that their outputs are always between 0 and 1, which can be useful in certain applications like probability calculations.

Disadvantages:

• The vanishing gradient problem. The sigmoid saturates and kills the gradient.

• Sigmoid functions are prone to saturation, which means that for large values of x, the output of the function becomes very close to 1, making it difficult for the function to learn further.

• The gradient of the sigmoid function becomes very small for large values of x, which can lead to slow convergence in optimization algorithms.

• Sigmoid functions are not symmetric around zero, which means that they can introduce bias into the output of a model.

3.Tanh function:

The main advantage of the tanh function is that it aids the backpropagation process by producing a zero-centered output. The tanh function is primarily used in recurrent neural networks for natural language processing and speech recognition tasks. The mathematical formula for the tanh function is follows:

f(x) = (e^x – e^(-x)) / (e^x + e^(-x))

Most tanh functions are commonly used in the hidden layers of neural networks. This is because their values range from -1 to 1. This is why the hidden layer means 0 or very close to 0. We center the data by bringing the mean closer to 0, allowing for further training.

Advantages:

• It is continuous and differentiable at all points.

• Like the sigmoid function, tanh is also a smooth function that can be easily differentiated, which makes it useful in training neural networks using backpropagation.
• It is a zero-centered function, which means that its outputs are centered around zero. This can help in preventing vanishing gradients during the training of deep neural networks.
• The function as you can see is non-linear so we can easily backpropagate the errors.

Disadvantages:

• Vanishing gradient problem.

• The gradients are low.

• Tanh is not monotonic, which means that its derivative is not always positive or negative. This can make it more difficult to optimize using some optimization techniques.

• Computationally expensive function

4.ReLU function:

A Rectified Linear Activation Function (ReLU for short) is a piecewise linear function that outputs an input directly if it is positive and zero otherwise. Models that use it are easy to train and often perform better, making it the default activation function for many types of neural networks. The mathematical formula for the ReLU function is as follows:

f(x) = max (0,x)

Advantage:

• The ReLU function is nonlinear. This means that errors can be easily backpropagated and multiple layers of neurons are activated by her ReLU function.

• It has been found to significantly speed up the convergence of stochastic gradient descent compared to the sigmoid and tanh functions.

• Not all neurons are activated at the same time. Since some neurons have zero output, few neurons are activated, the network is sparse and efficient, and easy to compute.

Disadvantages:

• It is not differentiable at zero and ReLU is unlimited.

• Negative inputs have zero slope. This means that activations in this region do not update the weights during backpropagation. This can create dead neurons that are never activated. This can be managed by reducing the learning rate and bias.

• The ReLU output is not zero-centered, which degrades neural network performance. The gradients of the weights during backpropagation are either all positive or all negative. This can introduce undesirable zigzag dynamics into the weight gradient updates. This can be handled with batch norm. Batch norm alleviates this problem somewhat by adding these gradients to the entire batch of data, which may change sign with the last update of the weights.

• Average activation is not zero. From ReLU, the network of subsequent layers has a positive bias because the average activation is greater than zero. Its computational simplicity makes it less computationally intensive than Sigmoid and Tanh, but the positive mean shift of the next layer slows down learning.

Always remember that ReLU functions should only be used in hidden layers. In conclusion, the ReLU function is a well-liked activation function since it is computationally effective and useful in many different kinds of neural networks. It does, however, have certain drawbacks, such as the potential for the "dying ReLU" problem and difficulties with optimization techniques.

5.ELU function:

An ELU activation layer performs the identity operation on positive inputs and an exponential nonlinearity on negative inputs. The layer performs the following operation:

$f(x) = \{ x, x \geq 0 \; \alpha(\exp(x) - 1), x < 0.$

The default value of α is 1. Specify a value of α for the layer by setting the Alpha property. where alpha is a hyperparameter that determines the negative saturation value.

Advantages:

• Smoother gradients: The ELU function has a continuous derivative, which helps avoid the vanishing gradient problem that can occur with other activation functions like the sigmoid function.

• Gives smoother convergence for any negative axis value. For any positive output, it behaves like a step function and gives a constant output.

• Better performance: Empirical evidence suggests that ELU activation functions can lead to better performance in deep neural networks compared to other activation functions.

• Faster convergence: The ELU function has been shown to help networks converge faster than other activation functions.

Disadvantages:

• Computationally expensive: The ELU function requires computing the exponential function, which can be computationally expensive.

• Unstable for large negative inputs: The ELU function can be unstable for large negative inputs, which can lead to numerical instability in the network.

• Not as well-known: The ELU function is not as well-known as other activation functions like the sigmoid and ReLU functions, which can make it harder to find resources and support for implementing it in neural networks.

6. SELu function:

The SELu (Scaled Exponential Linear Unit) activation function is a type of activation function used in neural networks. Unlike other activation functions, such as ReLU, which have a hard cutoff at zero, SELu has a smooth, continuous curve. The Scaled Exponential Linear Unit (SELU) activation function is defined as:

if x > 0: return scale * x. if x < 0: return scale * alpha * (exp(x) - 1) where alpha and scale are predefined constants (alpha=1.67326324 and scale=1.05070098).

Advantages:

• Like ReLU, SELU does not have vanishing gradient problem and hence, is used in deep neural networks.

• SELUs learn faster and better than other activation functions without needing further procession. Moreover, other activation functions combined with batch normalization cannot compete with SELUs.

• SELu has been shown to perform better than other activation functions, such as ReLU and tanh, on a variety of deep learning tasks, including image classification and speech recognition

Disadvantages:

• SELU is a relatively new activation function, so it is not yet used widely in practice. ReLU remains the preferred option.

• More research on architectures such as CNNs and RNNs using SELUs is needed for wide-spread industry use

• SELu may not work well on all types of data. While it has been shown to work well on many deep learning tasks, it may not be the best choice for all types of data, and some experimentation may be necessary to determine the best activation function for a particular task