MD. Mosaddek Hossain
20-42766-1

Computer Vision and Pattern Recognition
Section: A

Topic: Implementation on different activation function

Activation function

An activation function is a mathematical function that is applied to the output of a neural network node or neuron to introduce non-linearity into the network. Activation functions are used in neural networks to determine the output of a neuron, which is then passed on to other neurons in the network. The activation function essentially transforms the input signal into an output signal that is used to determine whether or not the neuron should be activated, or "fire". The choice of activation function can have a significant impact on the performance of a neural network, and different activation functions are better suited to different types of problems. Common activation functions include sigmoid, ReLU, tanh, softmax, Leaky ReLU, and ELU.

1.Step function

The step function is a simple activation function that maps an input to an output of 1 if the input is greater than or equal to a certain threshold, and 0 otherwise. The threshold is usually set to 0, so the step function outputs 1 for positive inputs and 0 for negative inputs.

$$f(x) = \begin{cases} 1 & \text{if } x > \theta \\ 0 & \text{if } x \leq \theta \end{cases}$$

This equation outputs a value of 1.0 for incoming values of (threshold value)or higher and 0 for all other values. Step functions, also known as threshold functions, only return 1 (true) for values that are above the specified threshold.

Advantage:

- computationally efficient and simple
- Can be used as a binary classifier

Disadvantage:

- It is not appropriate for use in backpropagation-based learning algorithms because it is not differentiable at x = 0.
- Gradient updates are a challenge to improve because they are constant and independent of the input.

2. Sigmoid function:

The sigmoid function is a common activation function used in neural networks. It is a mathematical function that maps any input value to a value between 0 and 1. The sigmoid function is defined as:

sigmoid(x) = 1 / (1 + exp(-x))

where x is the input to the function and exp is the exponential function. The output of the sigmoid function is always between 0 and 1, which makes it useful for modeling probabilities and for binary classification problems.

The sigmoid function has a characteristic S-shaped curve, with a smooth and continuous gradient. The sigmoid function is differentiable, which makes it easy to use in gradient-based optimization algorithms such as backpropagation.

One of the drawbacks of the sigmoid function is that it can saturate, which means that for very large or very small input values, the gradient of the function becomes very small. This can slow down the training of neural networks and make it difficult to optimize the weights of the network.

Advantage:

- Well-suited for use in shallow neural networks with few hidden layers.
- Smooth and different.
- It can be applied to probabilistic modeling, which makes it helpful for binary classification issues.

Disadvantage

- Outputs are not zero-cantered, which can slow down convergence during training.
- vulnerable to the vanishing gradient issue, which makes deep learning challenging.

3. Tanh function

The tanh (hyperbolic tangent) function is another commonly used activation function in neural networks. It is similar to the sigmoid function, but maps input values to a range between -1 and 1, rather than between 0 and 1. The tanh function is defined as:

tanh(x) = (exp(x) - exp(-x)) / (exp(x) + exp(-x))

Like the sigmoid function, the tanh function is also differentiable, which makes it easy to use in gradient-based optimization algorithms. The tanh function has a characteristic S-shaped curve, but its output values range from -1 to 1, which makes it useful for modeling data with negative values.

One of the advantages of the tanh function over the sigmoid function is that it is zero-centered, which means that its outputs have a mean of zero. This can help the optimization process of the neural network and make it easier to converge to the optimal solution.

However, like the sigmoid function, the tanh function can also saturate for very large or very small input values, which can slow down the training process of the neural network. In practice, the tanh function is often used in hidden layers of neural networks, while the sigmoid function is often used in output layers for binary classification problems.

Overall, the tanh function is a useful activation function that can be used in a variety of neural network architectures, particularly for problems where the data has negative values.

Advantage:

- Smooth and unique
- Compared to the sigmoid function, the outputs are zero-centered, making it more useful for teaching deep neural networks.
- Well-suited for use in shallow and deep neural networks

Disadvantage:

- vulnerable to the disappearing gradient issue, which makes training very deep neural networks challenging

4. ReLu function (Rectified Linear Units)

The ReLU (Rectified Linear Unit) function is a popular activation function used in neural networks. It is defined as:

ReLU(x) = max(0, x)

where x is the input to the function. In other words, the ReLU function outputs the input value if it is positive, and 0 otherwise.

One of the main advantages of the ReLU function over other activation functions such as the sigmoid and tanh functions is that it is computationally efficient to compute, which makes it well-suited for large neural networks. Additionally, the ReLU function does not saturate for large input values, which means that it does not suffer from the same optimization problems as the sigmoid and tanh functions.

Another advantage of the ReLU function is that it can help to sparsify the neural network, by setting some of the neurons to 0. This can improve the generalization performance of the network and help to reduce overfitting.

However, the ReLU function does have some limitations. One of the main drawbacks of the ReLU function is that it is not differentiable at x = 0, which can make it difficult to use in some optimization algorithms such as gradient descent. To address this problem, variants of the ReLU function such as the leaky ReLU and ELU (Exponential Linear Units) have been proposed, which have a small non-zero gradient for negative input values or for values close to 0.

Advantage:

- It is a simple function that is computationally efficient to compute.
- It does not suffer from the problem of vanishing gradients for large input values, making it easier to train deep neural networks.
- It can help to sparsity the activation of the network, by setting some of the activations to zero, which can help to prevent overfitting.

Disadvantage:

- Not differentiable at x = 0, which can cause issues during backpropagation.
- vulnerable to the "dying ReLU" problem, in which a significant portion of the network can cease responding and learning.

5. PRELU Function

The parametric rectified linear unit (PReLU) activation function is a variant of the ReLU function, which addresses the problem of "dying ReLU" by allowing for non-zero output for negative input values.

Advantage:

- it can help to prevent the problem of dying ReLU by allowing for non-zero output for negative input values.
- it can help to improve the accuracy of the network, especially for datasets with a high level of noise or variability.
- it is a simple function that is computationally efficient to compute.

Disadvantage:

- Higher processing cost compared to the ReLU function
- May not always improve performance over the ReLU function

6. ELU Function

The Exponential Linear Unit (ELU) is an activation function commonly used in neural networks. It is defined as follows:
ELU(x) = x, if x >= 0
= alpha * (exp(x) - 1), if x <

Advantage:

- The ELU function is smooth everywhere, which can help optimization algorithms converge faster.
- Faster convergence than Relu
- Unlike ReLU, which outputs 0 for negative inputs, the ELU function can output negative values, which can be beneficial for some types of data.

Disadvantage:

- The ELU function requires the use of an exponential function, which can be computationally expensive.
- The ELU function is not as widely used as other activation functions like ReLU, which means that there is less information available on how to tune its hyperparameters.
- The ELU function can lead to overfitting if the alpha parameter is not properly tuned.

7. SeLu Function

The Scaled Exponential Linear Unit (SELU) is an activation function that has gained popularity in neural networks in recent years. It is defined as follows:
SELU(x) = scale * (alpha * (exp(x) - 1)), if x < 0
= scale * x, if x >= 0
where alpha and scale are hyperparameters that are chosen to ensure that the mean and variance of the outputs of each layer are close to 0 and 1, respectively.

Advantage:

- It is self-normalizing, which means that it can maintain a mean activation of 0 and a standard deviation of 1 throughout the layers of a deep neural network, even without batch normalization or other regularization techniques.
- It has been performed on a variety of tasks, including image classification and natural language processing.
- It can induce sparsity in the activation patterns of a neural network, which can reduce the computational cost of training and inference.