Activation Functions

July 23, 2023

1 Activation Function

Activation functions helps to determine the output of a neural network. These type of functions are attached to each neuron in the network, and determines whether it should be activated or not, based on whether each neuron's input is relevant for the model's prediction.

> Activation function also helps to normalize the output of each neuron to a range between 1 and 0 or between -1 and 1.

In a neural network, inputs are fed into the neurons in the input layer. Each neuron has a weight, and multiplying the input number with the weight gives the output of the neuron, which is transferred to the next layer.

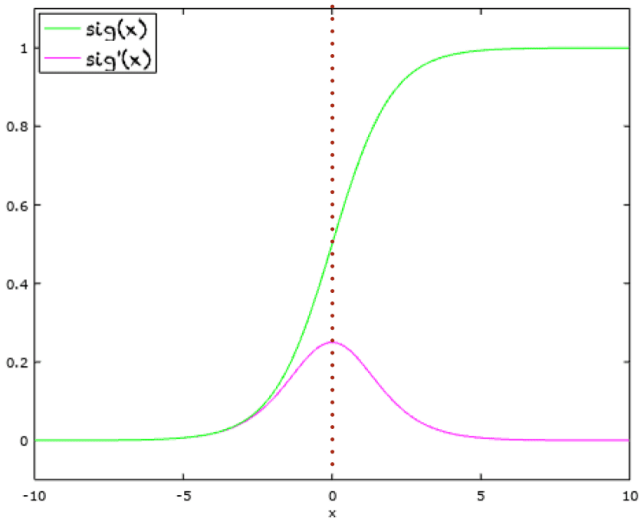
The activation function is a mathematical "gate" in between the input feeding the current neuron and its output going to the next layer. It can be as simple as a step function that turns the neuron output on and off, depending on a rule or threshold.

> Neural networks use non-linear activation functions, which can help the network learn complex data, compute and learn almost any function representing a question, and provide accurate predictions.

1.1 Commonly used activation functions

1.1.1 Sigmod function

The Sigmoid function is the most frequently used activation function in the beginning of deep learning. It is a smoothing function that is easy to derive.



Plot of σ(x) and its derivate σ'(x)

Domain: (-∞, +∞)
Range: (0, +1)
σ(0) = 0.5

Other properties

$$\sigma(x) = 1 - \sigma(-x)$$

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

$$\sigma'(x) = \sigma(x)(1-\sigma(x))$$

In the sigmoid function, we can see that its output is in the open interval (0,1). We can think of probability, but in the strict sense, don't treat it as probability. The sigmoid function was once more popular. It can be thought of as the firing rate of a neuron. In the middle where the slope is relatively large, it is the sensitive area of the neuron. On the sides where the slope is very gentle, it is the neuron's inhibitory area.

The function itself has certain defects.

1) When the input is slightly away from the coordinate origin, the gradient of the function becomes very small, almost zero. In the process of neural network backpropagation, we all use the chain rule of differential to calculate the differential of each weight w. When the backpropagation passes through the sigmod function, the differential on this chain is very small. Moreover, it may pass through many sigmod functions, which will eventually cause the weight w to have little effect on the loss function, which is not conducive to the optimization of the weight. This The problem is called gradient saturation or gradient dispersion.
2) The function output is not centered on 0, which will reduce the efficiency of weight update.
3) The sigmod function performs exponential operations, which is slower for computers.

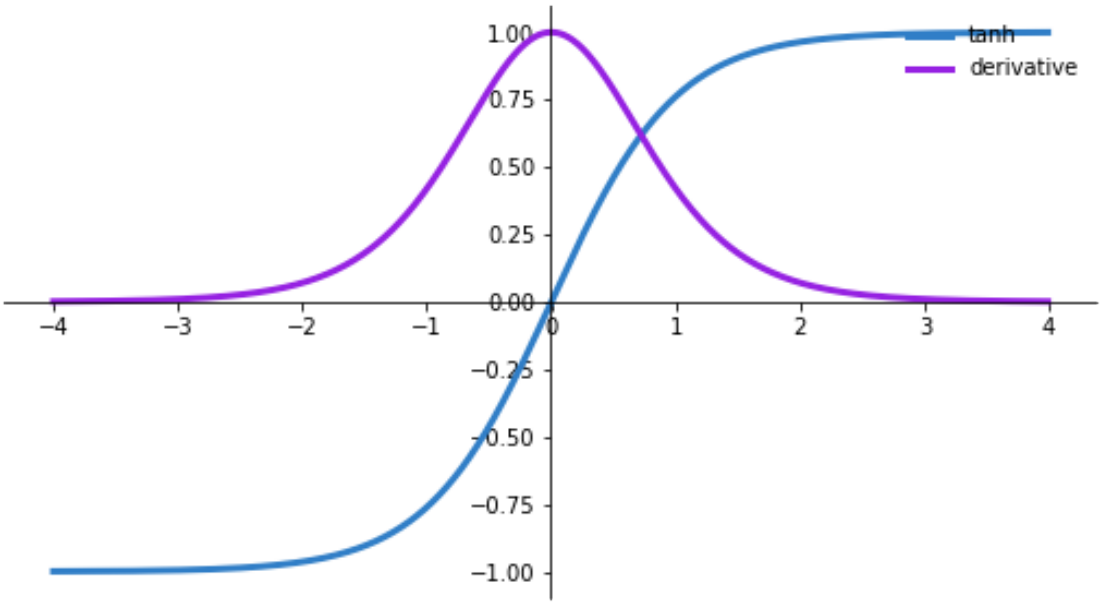
Advantages :-

1. Smooth gradient, preventing "jumps" in output values.
2. Output values bound between 0 and 1, normalizing the output of each neuron.
3. Clear predictions, i.e very close to 1 or 0.

Disadvantages:-

1. Prone to gradient vanishing.
2. Function output is not zero-centered.
3. Power operations are relatively time consuming.

1.1.2 tanh function



Tanh is a hyperbolic tangent function. The curves of tanh function and sigmod function are relatively similar. Let 's compare them. First of all, when the input is large or small, the output is almost smooth and the gradient is small, which is not conducive to weight update. The difference is the output interval.

The output interval of tanh is 1), and the whole function is 0-centric, which is better than sigmod.

In general binary classification problems, the tanh function is used for the hidden layer and the sigmod function is used for the output layer. However, these are not static, and the specific activation function to be used must be analyzed according to the specific problem, or it depends on debugging.

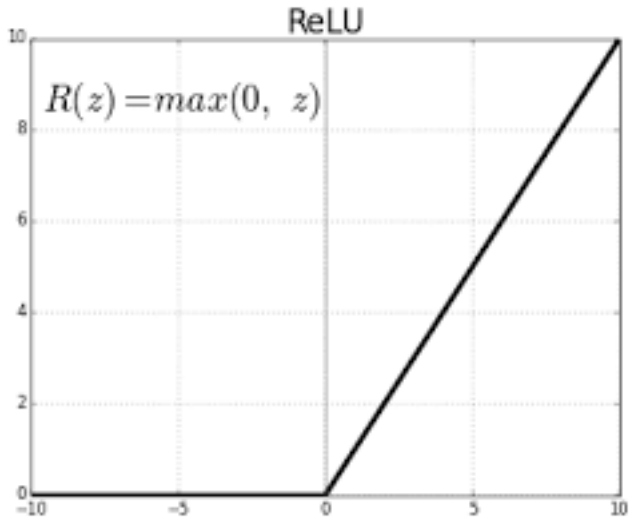
Advantages:-

1. Zero-centered output.
2. Better gradient flow compared to the sigmoid function, the tanh function has steeper gradients in the range of -1 to 1.

Disadvantages:-

1. Vanishing gradients: When the input to the tanh function is very large or very small, the gradient becomes close to zero, leading to slow or ineffective learning in deep networks.
2. Computational cost

1.1.3 ReLU function



ReLU, which stands for Rectified Linear Unit, is an activation function commonly used in neural networks. It is defined as f(x) = max(0, x), where "x" is the input to the function. In other words, ReLU returns the input value if it is positive, and zero otherwise.

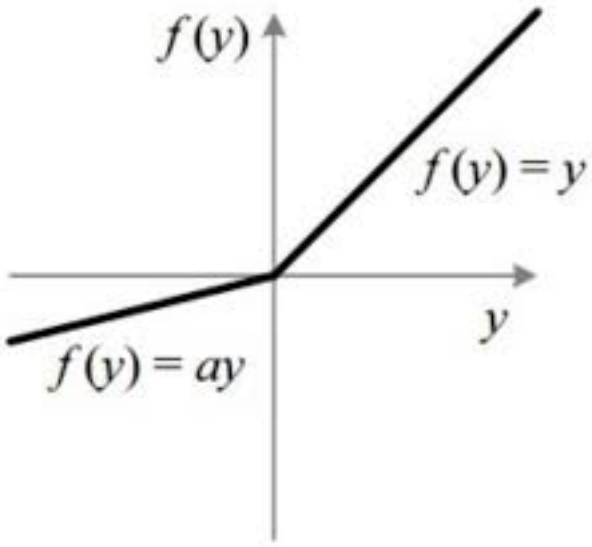
Advantages:

1. Sparsity: ReLU activation can lead to sparsity in the network. When the input is negative, the output is zero, effectively turning off the neuron. This sparsity property reduces the number of active neurons, making the network more efficient in terms of computation and memory usage.
2. Avoiding vanishing gradients.
3. Simplicity and computational efficiency.

Disadvantages:

1. Dead neurons: One issue with ReLU is the possibility of neurons becoming "dead" during training. If a neuron's output is consistently negative (i.e., the weighted sum of inputs is always negative), the gradient for that neuron will always be zero. Consequently, the neuron will not update its weights and will remain inactive for all future data points. Dead neurons hinder the learning capacity of the network and can lead to reduced model performance.
2. We find that the output of the ReLU function is either 0 or a positive number, which means that the ReLU function is not a 0-centric function.

1.1.4 Leaky ReLU function



In order to solve the Dead ReLU Problem, people proposed to set the first half of ReLU 0.01x instead of 0, from graph a=0.01. Another intuitive idea is a parameter-based method, Parametric ReLU : f(x)= max(alpha x,x), which alpha can be learned from back propagation. In theory, Leaky ReLU has all the advantages of ReLU, plus there will be no problems with Dead ReLU, but in actual operation, it has not been fully proved that Leaky ReLU is always better than ReLU.

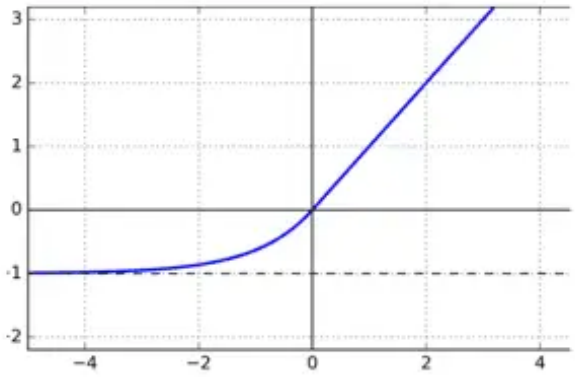
Advantages:-

1. Avoiding dead neurons.
2. No saturation: Unlike sigmoid and tanh activation functions, leaky ReLU does not saturate for positive inputs. This means that it does not suffer from the vanishing gradient problem for positive input values, promoting more stable and faster training in deep networks.

Disadvantages:-

1. Not zero-centered

1.1.5 ELU (Exponential Linear Units) function



ELU

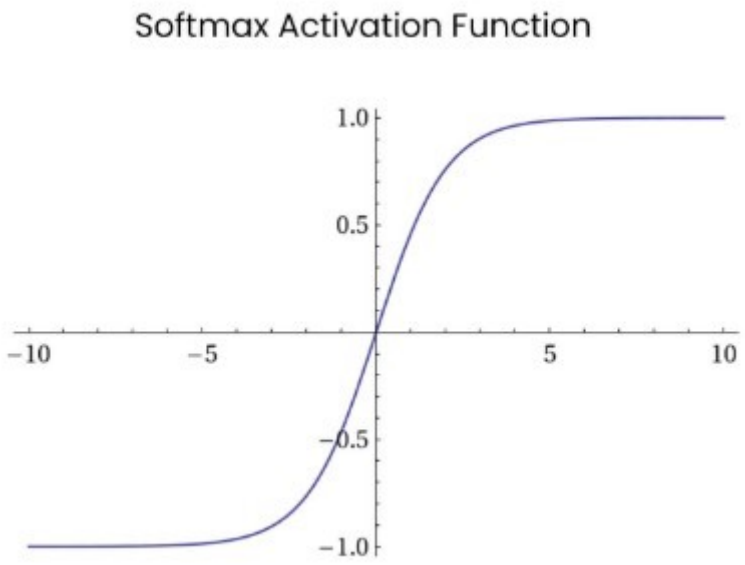
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

ELU is also proposed to solve the problems of ReLU. Obviously, ELU has all the advantages of ReLU, and:

- No Dead ReLU issues
- The mean of the output is close to 0, zero-centered

One small problem is that it is slightly more computationally intensive. Similar to Leaky ReLU, although theoretically better than ReLU, there is currently no good evidence in practice that ELU is always better than ReLU.

1.1.6 Softmax

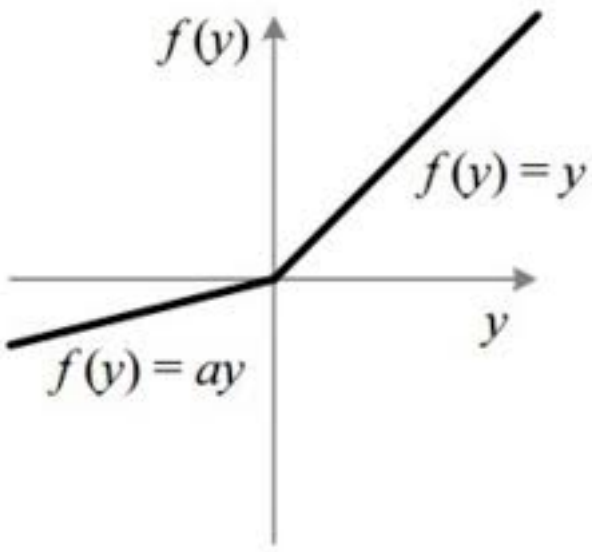


for an arbitrary real vector of length K, Softmax can compress it into a real vector of length K with a value in the range (0, 1), and the sum of the elements in the vector is 1.

It also has many applications in Multiclass Classification and neural networks. Softmax is different from the normal max function: the max function only outputs the largest value, and Softmax ensures that smaller values have a smaller probability and will not be discarded directly. It is a "max" that is "soft".

The denominator of the Softmax function combines all factors of the original output value, which means that the different probabilities obtained by the Softmax function are related to each other. In the case of binary classification, for Sigmoid, there are:

1.1.7 PRelu (Parametric ReLU)

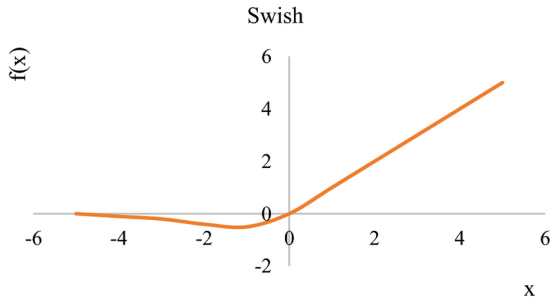


PReLU is also an improved version of ReLU. In the negative region, PReLU has a small slope, which can also avoid the problem of ReLU death. Compared to ELU, PReLU is a linear operation in the negative region. Although the slope is small, it does not tend to 0, which is a certain advantage.

We look at the formula of PReLU. The parameter is generally a number between 0 and 1, and it is generally relatively small, such as a few zeros. When = 0.01, we call PReLU as Leaky Relu , it is regarded as a special case PReLU it.

Above, y is any input on the ith channel and a is the negative slope which is a learnable parameter. * if a =0, f becomes ReLU * if a >0, f becomes leaky ReLU * if a is a learnable parameter, f becomes PReLU

1.1.8 Swish (A Self-Gated) Function



The formula is: **y = x * sigmoid (x)**

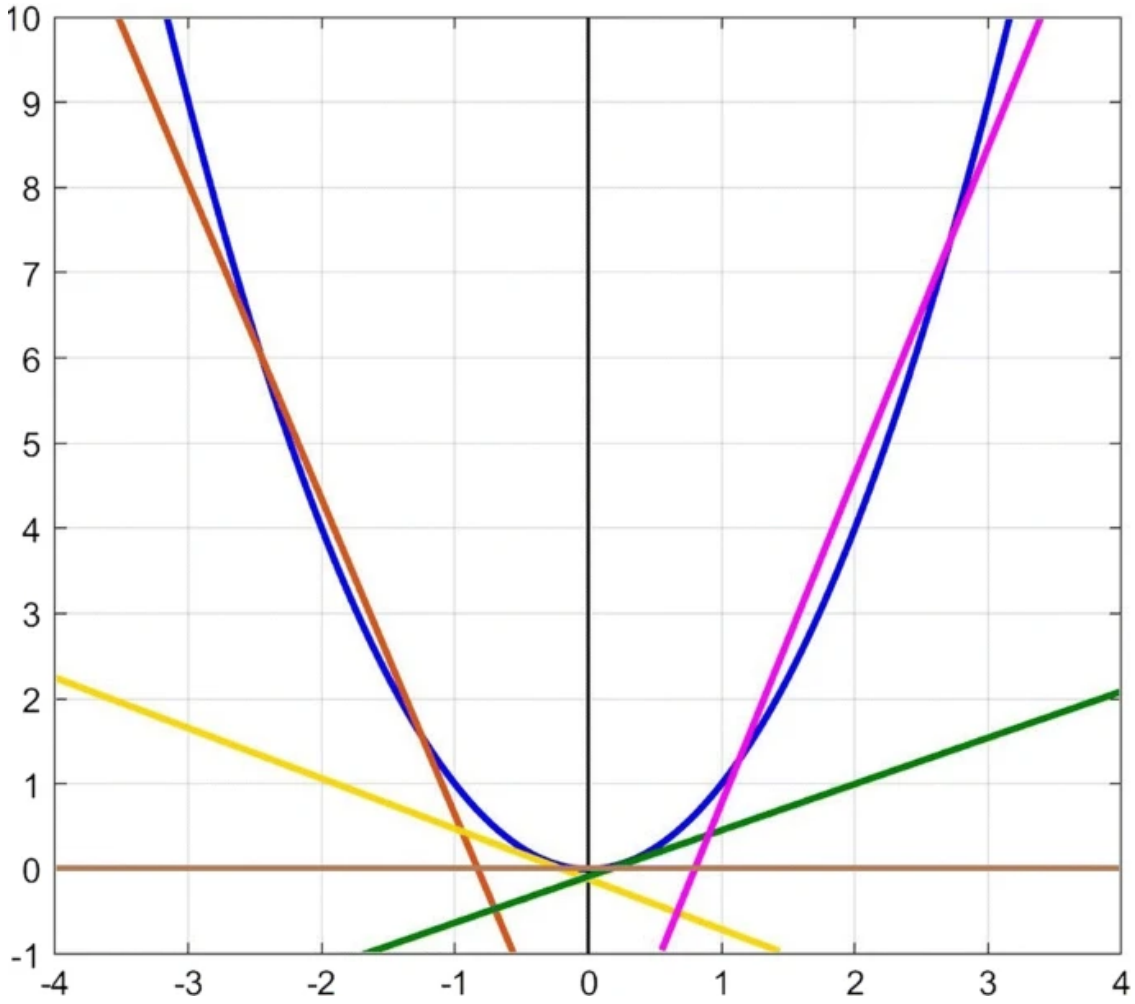
Swish's design was inspired by the use of sigmoid functions for gating in LSTMs and highway networks. We use the same value for gating to simplify the gating mechanism, which is called **self-gating**.

The advantage of self-gating is that it only requires a simple scalar input, while normal gating requires multiple scalar inputs. This feature enables self-gated activation functions such as Swish to easily replace activation functions that take a single scalar as input (such as ReLU) without changing the hidden capacity or number of parameters.

1) Unboundedness (unboundedness) is helpful to prevent gradient from gradually approaching 0 during slow training, causing saturation. At the same time, being bounded has advantages, because bounded active functions can have strong reguairzation, and larger negative inputs will be resolved.

2) At the same time, smoothness also plays an important role in optimization and generalization.

1.1.9 Maxout

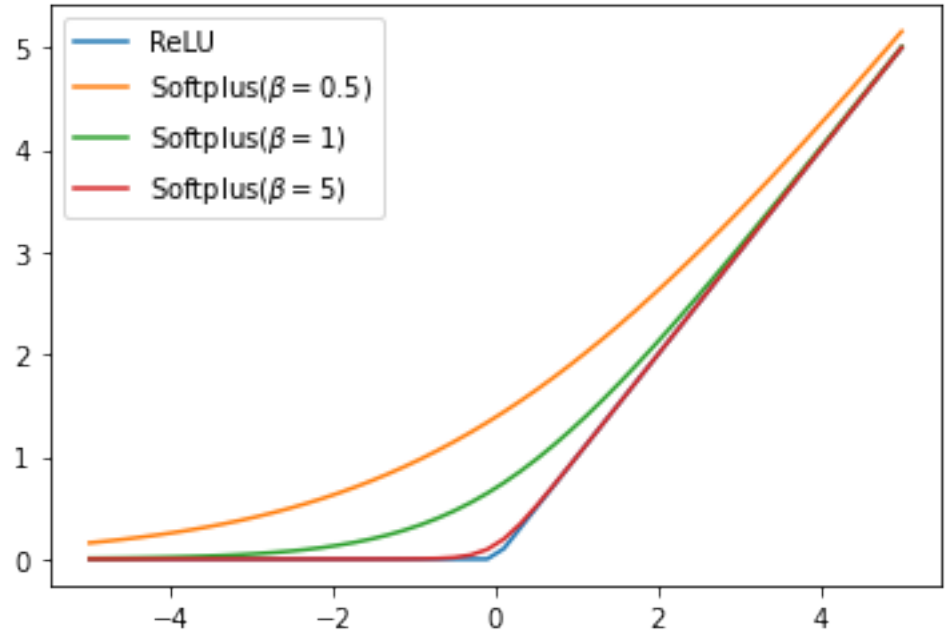


One relatively popular choice is the Maxout neuron (introduced recently by Goodfellow et al.) that generalizes the ReLU and its leaky version. Notice that both ReLU and Leaky ReLU are a special case of this form (for example, for ReLU we have w1,b1 =0).The Maxout neuron therefore enjoys all the benefits of a ReLU unit (linear regime of operation, no saturation) and does not have its drawbacks

The Maxout activation is a generalization of the ReLU and the leaky ReLU functions. It is a learnable activation function.

Maxout can be seen as adding a layer of activation function to the deep learning network, which contains a parameter k. Compared with ReLU, sigmoid, etc., this layer is special in that it adds k neurons and then outputs the largest activation value. value.

1.1.10 Softplus



The softplus function is similar to the ReLU function, but it is relatively smooth.It is unilateral suppression like ReLU.It has a wide acceptance range (0, + inf).

Softplus function: **f(x) = ln(1+exp x)**

Advantages:-

1. Smoothness: Unlike the ReLU and its variants, the softplus activation function is smooth, meaning it has continuous and differentiable derivatives. This smoothness can be beneficial during the optimization process, particularly in gradient-based optimization algorithms like backpropagation, where gradients are used to update the neural network's parameters.
2. No saturation: The softplus function does not suffer from the vanishing gradient problem that can affect sigmoid and tanh activation functions. Its gradient does not approach zero for large positive or negative inputs, ensuring more stable and faster training in deep networks.
3. No dead neurons.

Disadvatages:-

1. Computational cost
2. Not zero-centered

1.2 Questions on activation functions

1) Which activation function is commonly used for binary classification tasks?

=> **Sigmoid**

2) Which activation function is suitable for handling the vanishing gradient problem in deep neural networks?

=> **Leaky ReLU**

3) Which activation function is commonly used for multi-class classification tasks?

=> **Softmax**

4) Which activation function is preferred for most hidden layers in a deep neural network?

=> **ReLU**

5) Which activation function can produce negative values and is centered around zero?

=> **Tanh**

6) Which activation function is a variant of ReLU that allows a small gradient for negative values?

=> **Leaky ReLU**

7) Which activation function can map any real-valued number to a value between 0 and 1?

=> **Sigmoid**

8) Which activation function does not introduce non-linearity to the model?

=> **Linear**

9) Which activation function is used for binary classification tasks where the output ranges from -1 to 1?

=> **Tanh**

10) Which activation function is less prone to the "vanishing gradient" problem compared to the sigmoid function?

=> **ReLU**

1.2.1 *Generally speaking, these activation functions have their own advantages and disadvantages. There is no statement that indicates which ones are not working, and which activation functions are good. All the good and bad must be obtained by experiments.*

[ ]: