AI and Deep Learning

2-3 Activation Functions

Zhonglei Wang

WISE, SOE, CHOW

XMU

(Version v1)

Contents

1. Sigmoid activation function

2. Tanh activation function

3. ReLU activation function

4. Leaky ReLU activation function

Why activation?

1. Neural networks mainly involve two steps for each neuron
   - Linear transformation
   - Activation (nonlinear transformation)
2. If there is no activation, neural networks are simple linear models!
3. Activation empowers neural networks to learn complex structures
4. Universal approximation theorem (Chen and Chen, 1995): Under certain conditions, neural networks can approximate any continuous function to any desired accuracy

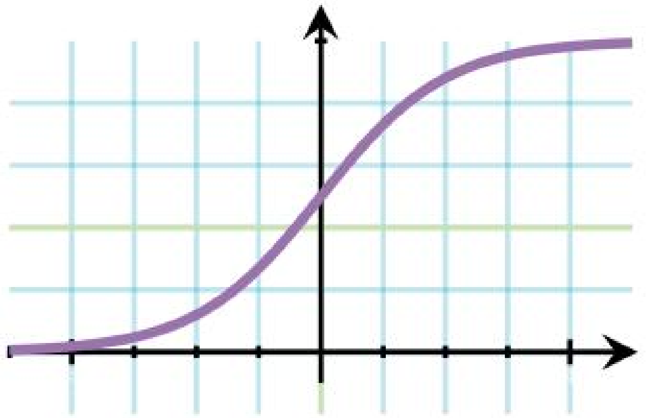
Sigmoid activation function

1. Form

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

- Range: $(0, 1)$
- Symmetric with respect to the point $(0, 1/2)$ (check by yourself)
- Derivative: $\sigma'(z) = \sigma(z)\{1 - \sigma(z)\} \leq 0.25$

Sigmoid activation function



1. Advantages
    - Range is $(0, 1)$, and it is commonly used for binary classification (last layer)
    - Gradient vanishes (actually it is not good) if z is large, so it is robust to outliers
    - Easy to obtain its derivative for backpropagation
2. Disadvantages
    - Gradient vanishing
        - For a neural network with 5 layers, we may have $0.25^5 \approx 0.001$!
    - Heavy computation for the derivative due to the exponential
    - Saturated neurons make it less efficient to update model parameters
    - Output is not symmetric about 0, it is inefficient to update weights

Sigmoid activation function

1. To better understand the disadvantage when the output is not symmetric about 0, consider the case when we only have ONE training example $(\boldsymbol{x}, y)$
2. In Section 2-2, we have shown that $\mathrm{d}\boldsymbol{W}^{[2]} = \mathrm{d}z^{[2]} \cdot (\boldsymbol{a}^{[1]})^{\mathrm{T}}$
   - $\boldsymbol{a}^{[1]} \in \mathbb{R}^{4\times 1}$
   - $\mathrm{d}z^{[2]} = a^{[2]} - y$
3. If a sigmoid activation function is used in the first hidden layer, then evey element of $\boldsymbol{a}^{[1]}$ is positive
4. Then, the signs of elements of $\mathrm{d}\boldsymbol{W}^{[2]}$ is determined by a scale $\mathrm{d}z^{[2]}$
5. In other words, all elements of $\mathrm{d}\boldsymbol{W}^{[2]}$ share the same sign with $\mathrm{d}z^{[2]}$
6. It causes the inefficient "zig-zag optimization path" when updating $\boldsymbol{W}^{[2]}$

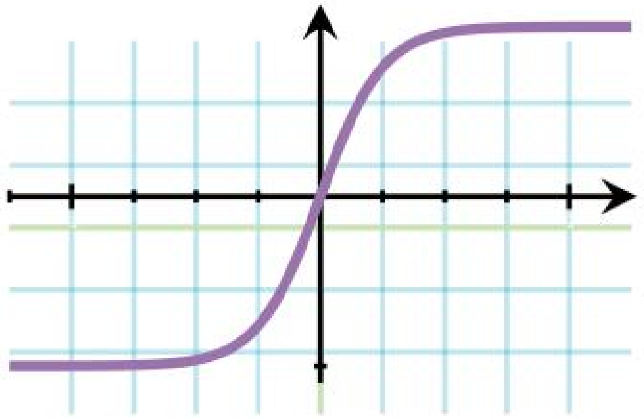
Tanh activation function

1. Form

$$\tanh(z) = \frac{2}{1 + \exp(-2z)} - 1$$

- $\tanh(z) = 2 \cdot \sigma(2z) - 1$
- Range: $(-1, 1)$
- Symmetric with respect to the origin point $(0, 0)$ (check by yourself)
- Derivative: $\tanh'(z) = 1 - \tanh^2(z) < 1$

Tanh activation function



1. Advantages
   - Function is symmetric about 0
   - Compared with sigmoid, the magnitude of gradient is larger near 0
   - Range is finite: $(-1, 1)$
2. Disadvantages
   - Gradient vanishing
   - Heavy computation for the derivative due to the exponential.
   - Saturated neurons make it less efficient to update model parameters

ReLU activation function

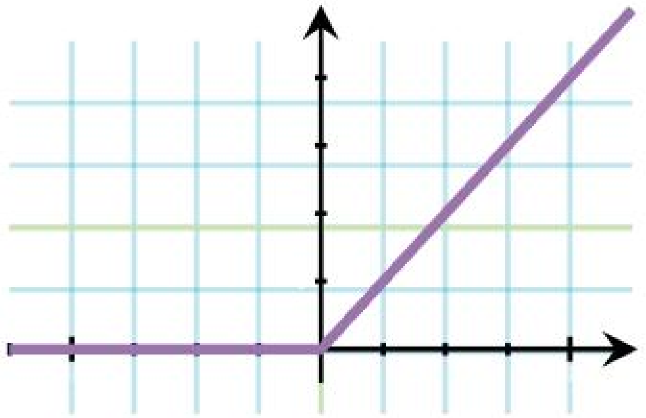
1. Form (Rectified Linear Unit)

$$\text{ReLU}(z) = \max\{0, z\}$$

- Range: $[0, \infty)$
- Derivative:

$$\text{ReLU}'(z) = \begin{cases} 0 & z \leq 0 \\ 1 & z > 0 \end{cases}$$

ReLU activation function



1. Advantages
    - High computation efficiency
    - Alleviate the gradient vanishing problem associated with sigmoid/tanh activation functions
    - Sparse activation
2. Disadvantages
    - Dying ReLU
        - $\mathrm{d}\boldsymbol{W}^{[1]} = \mathrm{d}\boldsymbol{z}^{[1]} \cdot (\boldsymbol{x})^{\mathrm{T}}$
    - Output is not symmetric about 0
    - Not differentiable at z=0, but sub-gradients exist
    - Unbounded for positive values

ReLU activation function

1. Question: If there are 40% neurons are "dead" during the training procedure
   - What are the possible reasons?
   - Is that always a bad thing?
   - How to rivive those "dead neurons" if necessary?
2. Answer:
   - Large learning rate or bad initialization
   - May not always be a bad thing
   - Try small learning rate and different initializations

Leaky ReLU activation function

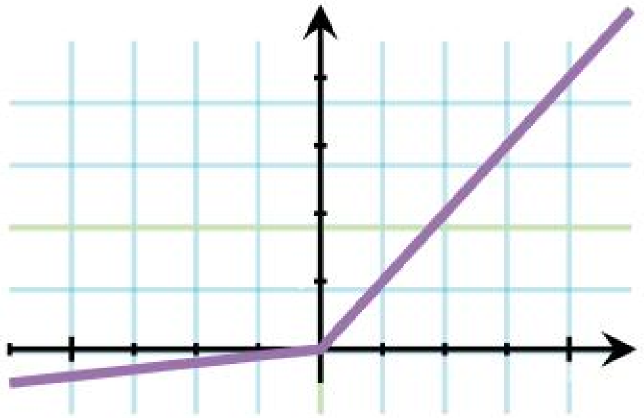
1. Form (Leaky Rectified Linear Unit)

$$\text{LeakyReLu}(z) = \begin{cases} \alpha z & z \leq 0 \\ z & z > 0 \end{cases}$$

- α: Small positive constant. For example, $\alpha = 0.01$
- Range: $(-\infty, \infty)$
- Derivative:

$$\text{LeakyReLu}'(z) = \begin{cases} \alpha & z \leq 0 \\ 1 & z > 0 \end{cases}$$

Leaky ReLU activation function



1. Advantages
   - Alleviate the dying ReLU problem
   - Easy computation
   - Fast convergence rate
   - No saturated region
2. Disadvantages
   - Hard to determine the value for α
   - Output is not symmetric about 0, it is inefficient to update weights
   - Slight decrease in sparsity

Other activation functions

1. Other activation functions:
   - PReLU
   - ELU
   - SELU
   - Swish (SiLU)
   - GELU
   - Mish
   - Softmax
   - ...
2. Check by yourselves

Tips for choosing activation functions

1. ReLU (or its variations) is the default option for hidden layers
2. Consider leaky ReLU or other variations, if ReLU induces problems
3. Consider sigmoid or softmax, if probabilities are required for the output layer
4. Consider a linear (identity) activation function for regression problems
5. Trials are required to solve other possible problems

Summary

1. Four criteria for choosing activation functions
   - Nonlinearity
   - Symmetry about 0
   - Computation efficiency
   - Task orientation for output