

Derivatives of Commonly-Used Activation Functions

Backpropagation

We are interested in the derivatives of activation functions because we need them in the backpropagation algorithm.

Let's start from the final output layer of a neural network, which consists of one or more neurons. Each of the k -th neuron takes M inputs, x_1, \dots, x_M , from the previous layer, and returns

$$g(z_k) = g\left(\sum_{j=1}^M x_j w_{j,k}\right)$$

where g is the activation function.

Let's consider the squared error cost of the final layer, which has K neurons:

$$E = \frac{1}{2} \sum_{k=1}^K (g(z_k) - t_k)^2$$

we have

$$\frac{\partial E}{\partial w_{j,k}} = \frac{1}{2} \frac{\partial}{\partial w_{j,k}} (g(z_k) - t_k)^2 = (g(z_k) - t_k) g'(z_k) \frac{\partial z_k}{\partial w_{j,k}} = (g(z_k) - t_k) g'(z_k) x_j$$

Where $g'(z_k)$ is the derivative of the activation function given its input value.

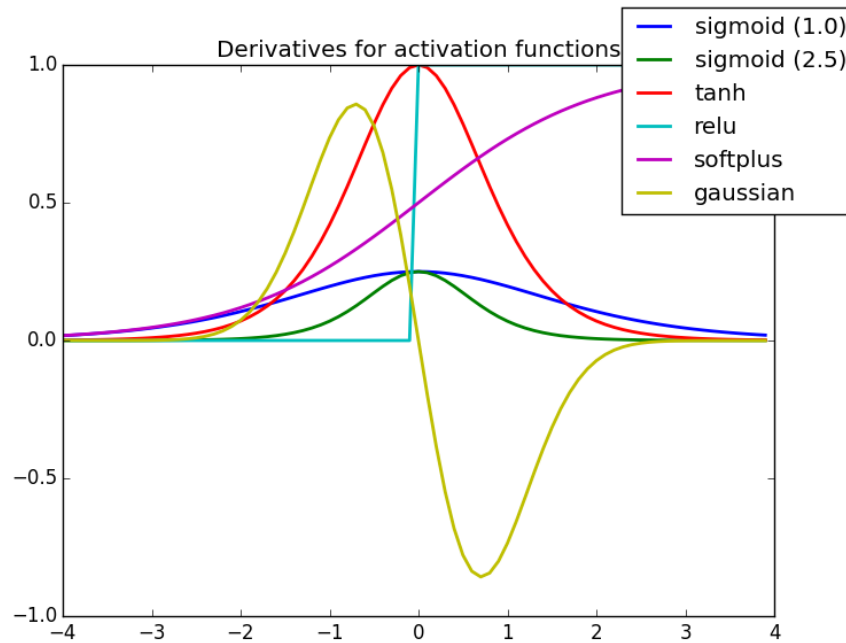
Activation Functions

The shapes of derivatives of commonly-used activation functions are as follows:

Sigmoid

The sigmoid function is the inverse of the logit function, which is the log odd of $P(x)$:

$$z(p) = \ln \frac{p}{1-p}$$



Take the exponential to both sides, we have the sigmoid function

$$p(z) = \frac{e^z}{1 + e^z}$$

According to the quotient rule, the derivative of the sigmoid function is

$$\frac{dp(z)}{dz} = \frac{\frac{de^z}{dz}(1 + e^z) - e^f \frac{d}{dz}(1 + e^z)}{(1 + e^z)^2} = \frac{e^z(1 + e^z) - e^z e^z}{(1 + e^z)^2} = \frac{e^z}{(1 + e^z)^2} = p(1-p)$$

When we're backpropagating the errors in a network through a layer with a sigmoid activation function, p is already computed.

Softmax

The softmax activation is a generalization of the sigmoid activation, where the latter considers two outputs from an experiment and the former considers more than two.

The softmax activation is defined as

$$p(z_k \mid z_{-k}) = \frac{e^{z_k}}{\sum_k^K e^{z_k}}$$

where z_{-k} represents $z_1, \dots, z_{k-1}, z_{k+1}, \dots, z_K$.

The sigmoid function is a specialization where $K = 2$ and $z_1 = 0$:

$$p(z_2 \mid z_1 = 0) = \frac{e^{z_2}}{e^{z_1=0} + e^{z_2}}$$

The derivative of softmax is complex and lengthy because softmax has both multivariate inputs and outputs, so we move the derivation in a [separate document](#).

Hyperbolic Tangent

As explained in [the previous article](#), the tanh function is defined as

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)}$$

and

$$\frac{d}{dx} \tanh(x) = \frac{\frac{d}{dx} \sinh(x) \cosh(x) - \sinh(x) \frac{d}{dx} \cosh(x)}{\cosh^2(x)} = 1 - \tanh^2(x)$$

It is a good property of an activation function because $\tanh(x)$ is already computed when we do backpropagate.