Formalize Transition Functions

Up to this point, we have explained neural networks as a combination of piecewise linear functions combined by "kinks." This is because we have been using the ReLU function as our transition function. A transition function is a component of artificial neural networks that gives them the ability to switch linear components on or off and, thus, to express nonlinear functions. Without these transition functions, a neural network is only a linear model.

☆ Key Points

A transition function is the part in artificial neural networks that gives them the ability to express nonlinear functions.

Rectified linear unit, sigmoidal unit, and hyperbolic tangent are examples of transition functions.

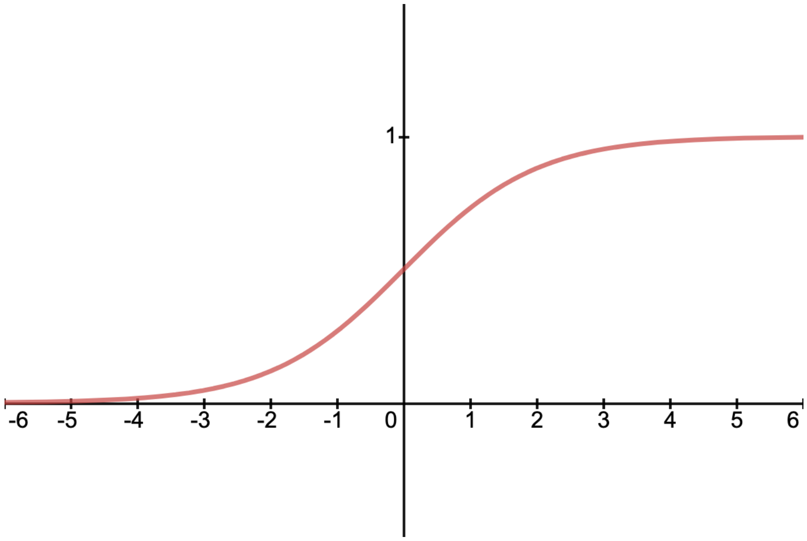
Consider a neural network with 1 hidden layers as $\mathbf{h}(\mathbf{x}) = \mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{x})$.

If we remove the ReLU function σ, we essentially obtain a linear function $\mathbf{h}(\mathbf{x}) = \mathbf{W}_2(\mathbf{W}_1\mathbf{x}) = \mathbf{W}\mathbf{x}$, where $\mathbf{W} = \mathbf{W}_2\mathbf{W}_1$.

Historically, people have used the sigmoid, hyperbolic tangent, and ReLU functions as their transition functions.

Sigmoid Function

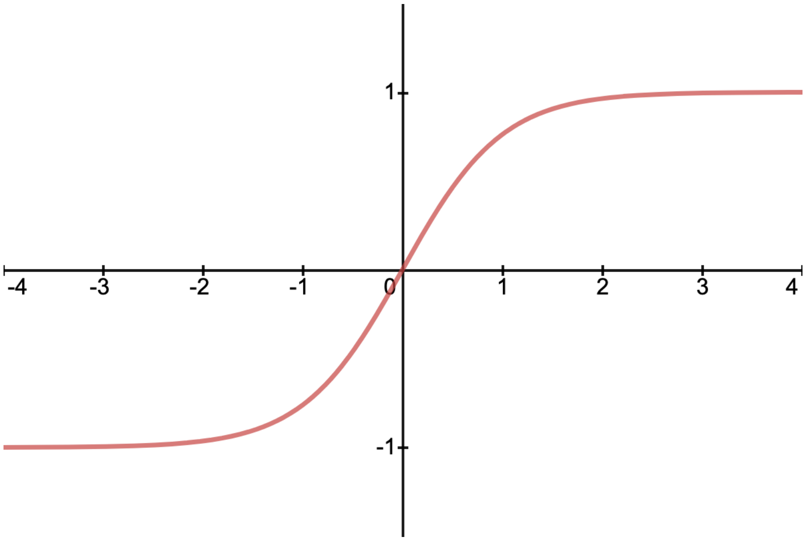
Sigmoidal Unit: $\sigma(z) = \frac{1}{1+e^{-z}}$



As shown in the graph and from its equation, the sigmoid function outputs values between 0 and 1. It is therefore popular when one wants to produce probabilities (e.g., in the very last layer). It also has a nice interpretation of a linear function being switched off (if the transition is close to 0) or switched on (if it is close to 1). However, the two ends of a sigmoid tangent are too flat to provide a good gradient for stochastic gradient descent to learn. After a while, the values of the internal layers become too large and the weights **saturate** because the transition values are stuck in the flat regions. Note that the problem here is that the first derivative for either very large positive or negative weights is close to 0, resulting in a tiny gradient.

Hyperbolic Tangent/ tanh Function

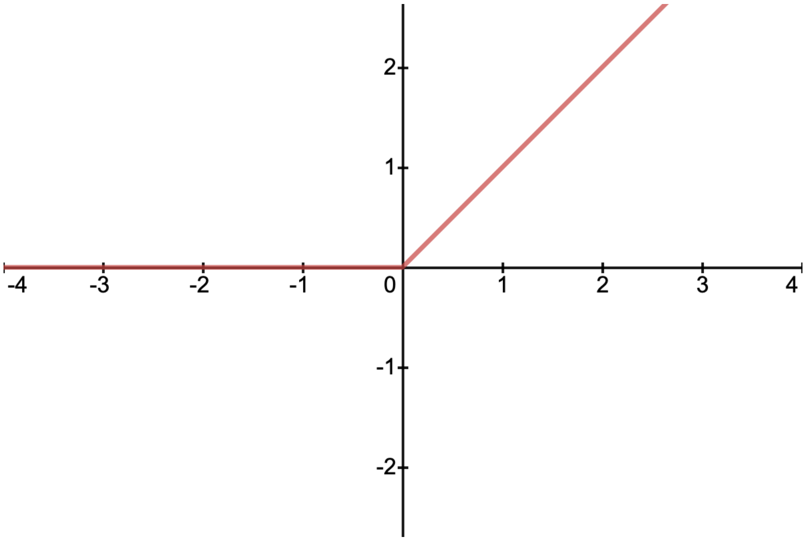
tanh: $\sigma(z) = \tanh(z)$



The hyperbolic tangent function has a similar shape to the sigmoid function. However, its range is between -1 and 1. This allows negative weights to become negative. It is useful for binary classification. However, this transition function runs into the same problem as sigmoid functions due to their similar shapes.

ReLU Function

ReLU: $\sigma(z) = \max(z, 0)$



The ReLU function is currently the most popular transition function as it doesn't completely fall into the same pitfall of the other two functions: having ends that are too flat and thus not easy to optimize. Although the left part of the ReLU is flat, it is not symmetric, so when the sign of the activation changes, the gradient is very strong.