When this happens, we say that the gradients have saturated. Hence, further multiplication via backpropagation causes the gradient to either vanish or explode; and as a result, the affected neurons become dead and transfer no information across the network, thus negatively affecting training.

Another drawback is that the outputs of the function are not zero-centered. As a consequence, during backpropagation, the gradients can either become all positive or all negative. This has a negative effect in minimizing the function objective (i.e., the cost function).

Hyperbolic Tangent (tanh)

The hyperbolic tangent illustrated in Figure 29-8 improves on the sigmoid function by bordering its output within a range of −1 and 1. So, while it still suffers from the exploding and vanishing gradient problem, its outputs are now zero-centered. From the formula, the reader will observe that tanh is merely a scaled sigmoid function.

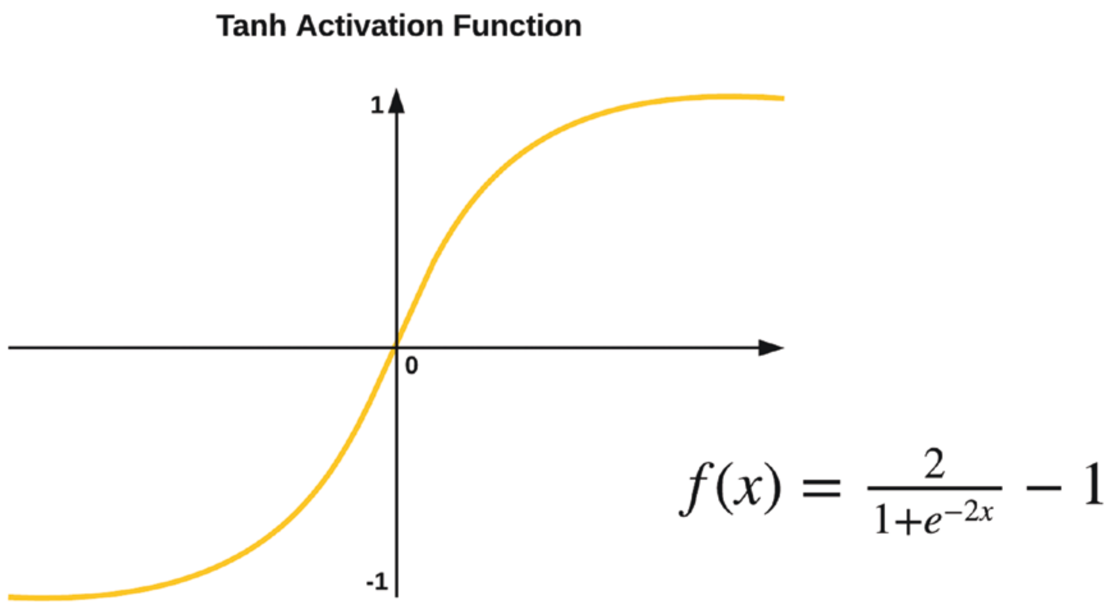

Figure 29-8. *The hyperbolic tangent activation function*

Rectified Linear Unit (ReLU)

The rectified linear unit or ReLU activation function is illustrated in Figure 29-9 and works by setting the activation to 0 for values, x, less than 0 and a linear slope of 1 when values, x, are greater than 0.

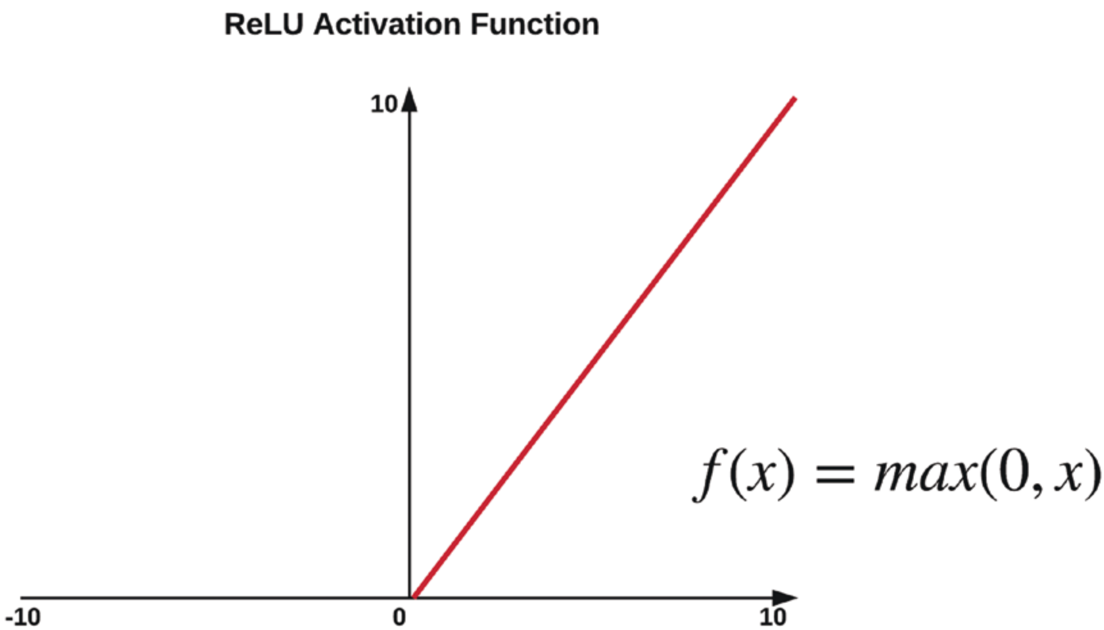

Figure 29-9. *ReLU activation function*

ReLU offers a vast improvement on the tanh and sigmoid activation functions by greatly mitigating the vanishing and exploding gradient problem. However, some gradients can still die out during backpropagation with a large learning rate. However, with a well-defined learning rate, we should not have a problem.

Leaky ReLU

Leaky ReLU is another activation function that is proposed to solve the case of some neurons completely dying out in ReLU by avoiding zero gradients. Leaky ReLU is illustrated in Figure 29-10. The function works by setting the activation to a small negative slope when the value $x < 0$.