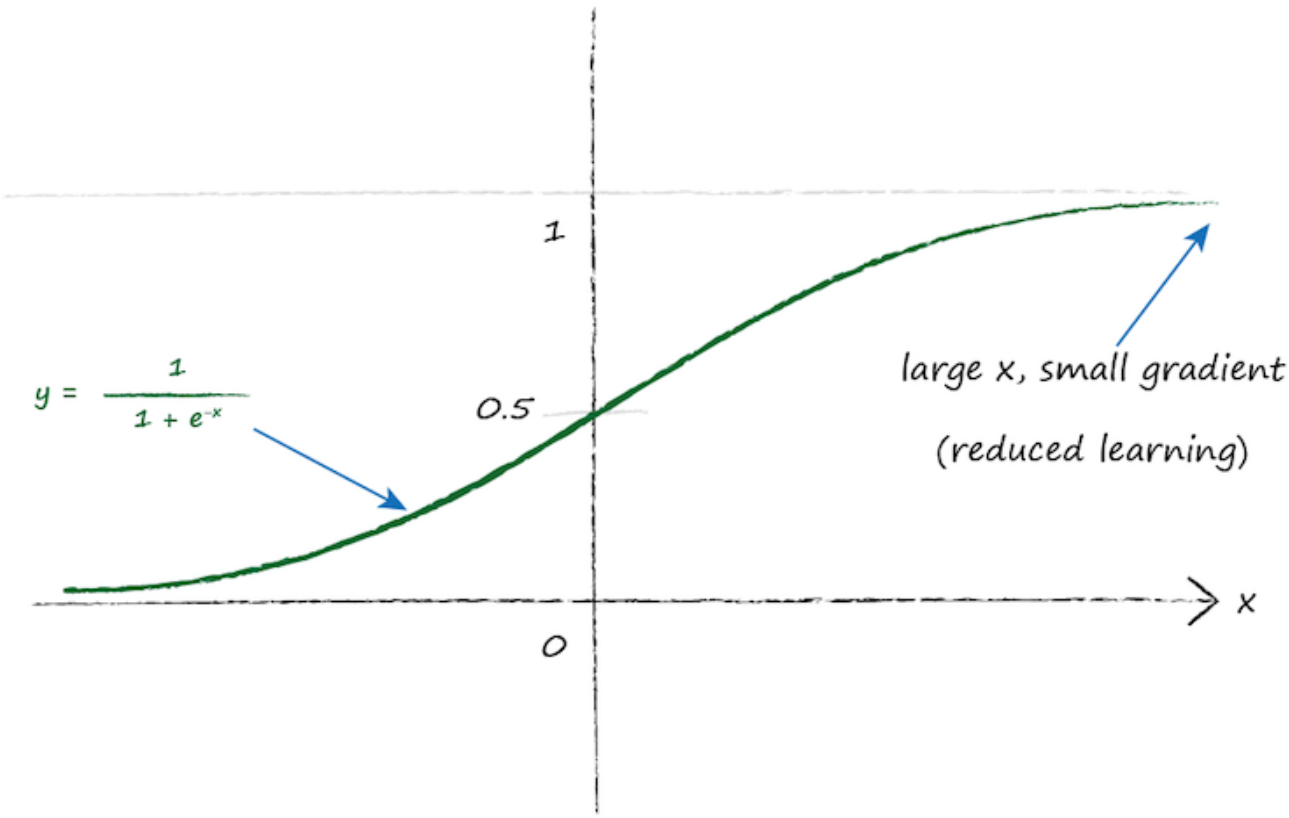
Preparing Data: Inputs & Outputs

In this section, we are going to consider how we might best prepare the training data and even design the outputs to give the training process a good chance of working.

Yes, you read that right! Not all attempts at using neural networks will work well, for many reasons. Some of those reasons can be addressed by thinking about the training data, the initial weights, and designing a good output scheme. Let's look at each in turn.

Preparing Inputs

Have a look at the diagram below of the sigmoid activation function. You can see that if the inputs are large, the activation function gets very flat.



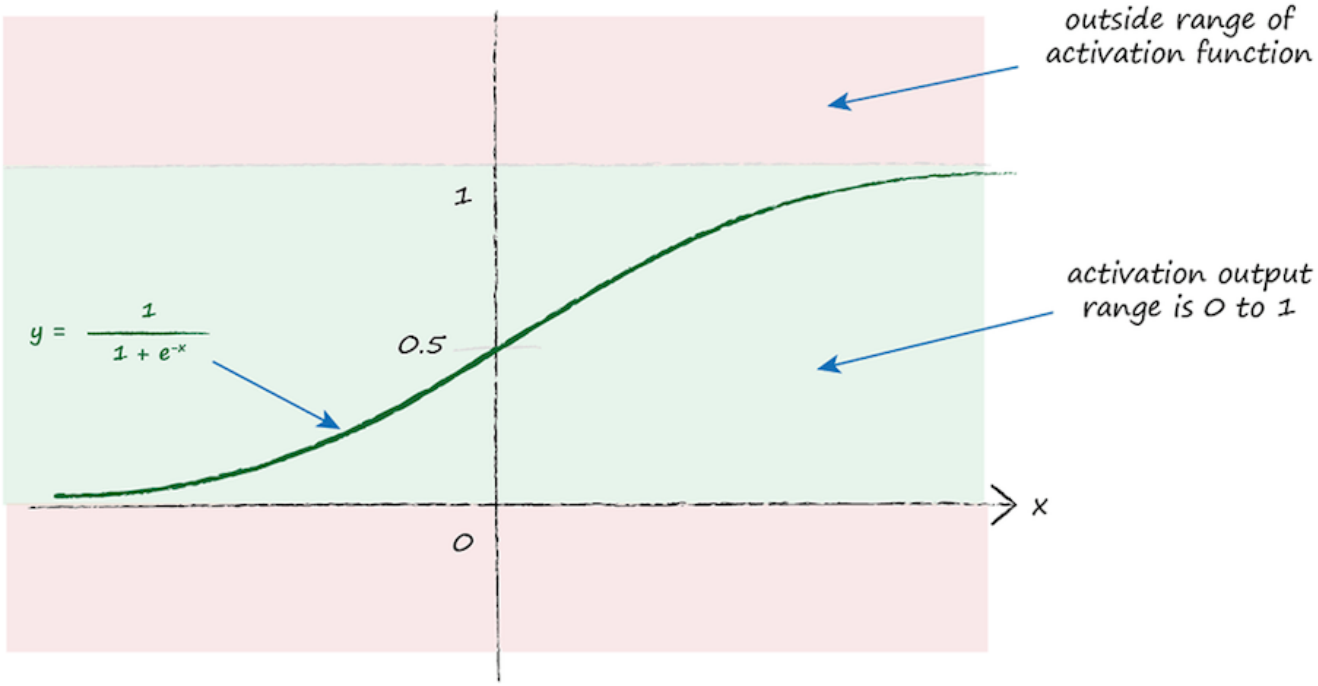
A very flat activation function is problematic because we use the gradient to learn new weights. Look back at that expression for the weight changes. It

depends on the gradient of the activation function. A tiny gradient means we have limited the ability to learn. This is called *saturating* a neural network. That means we should try to keep the inputs small.

Interestingly, that expression also depends on the incoming signal (o_j) so we shouldn't make it too small either. Very very tiny values can be problematic too because computers can lose accuracy when dealing very very small or very very large numbers. A good recommendation is to rescale inputs into the range 0.0 to 1.0. Some will add a small offset to the inputs, like 0.01, just to avoid having zero inputs which are troublesome because they kill the learning ability by zeroing the weight update expression by setting that $o_j = 0$.

Preparing Outputs

The outputs of a neural network are the signals that pop out of the last layer of nodes. If we're using an activation function that can't produce a value above 1.0 then it would be silly to try to set larger values as training targets. Remember that the logistic function doesn't even get to 1.0, it just gets ever closer to it. Mathematicians call this *asymptotically* approaching 1.0. The following diagram makes clear that output values larger than 1.0 and below zero are simply not possible from the logistic activation function.



If we do set target values in these inaccessible forbidden ranges, the network training will drive ever larger weights in an attempt to produce larger and larger outputs which can never actually be produced by the activation function. We know that's bad as that saturates the network.

So we should rescale our target values to match the outputs possible from the activation function, taking care to avoid the values which are never really reached. It is common to use a range of 0.0 to 1.0, but some do use a range of 0.01 to 0.99 because both 0.0 and 1.0 are impossible targets and risk driving overly large weights.