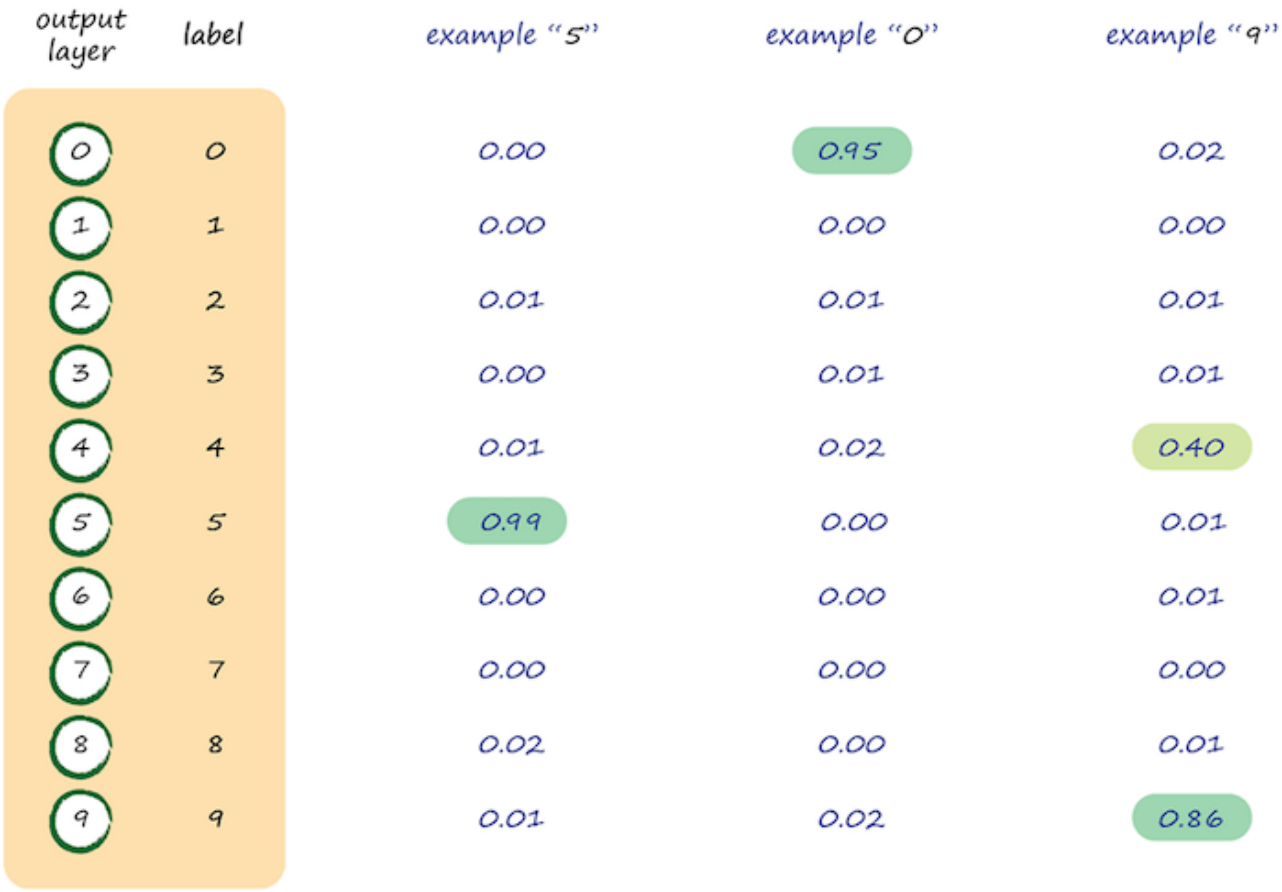
The Need to Rescale the Target Output

Why is it necessary to rescale the input and output values in order to achieve good results? Let's find out.

We now need to think about the neural network's outputs. We saw earlier that the outputs should match the range of values that the activation function can push out. The logistic function we're using can't push out numbers like -2.0 or 255. The range is between 0.0 and 1.0, and in fact, you can't reach 0.0 or 1.0 as the logistic function only approaches these extremes without actually getting there. So it looks like we'll have to scale our target values when training.

But actually, we have a deeper question to ask ourselves. What should the output even be? Should it be an image of the answer? That would mean we have a $28 * 28 = 784$ output nodes.

If we take a step back and think about what we're asking the neural network, we realize we're asking it to classify the image and assign the correct label. That label is one of 10 numbers, from 0 to 9. That means it should be able to have an output layer of 10 nodes, one for each of the possible answers, or labels. If the answer were 0 the first output layer node would fire, and the rest should be silent. If the answer were 9, the last output layer node would fire, and the rest would be silent. The following illustrates this scheme, with some example outputs too.



The first example is where the neural network thinks it has seen the number 5. You can see that the largest signal emerging from the output layer is from the node with label 5. Remember this is the sixth node because we're starting from a label for zero. That's easy enough. The rest of the output nodes produce a small signal very close to zero. Rounding errors might result in an output of zero, but in fact, you'll remember the activation function doesn't produce an actual zero.

The next example shows what might happen if the neural network thinks it has seen a handwritten "zero". Again the largest output by far is from the first output node corresponding to the label 0.

The last example is more interesting. Here the neural network has produced the largest output signal from the last node, corresponding to the label 9. However, it has a moderately big output from the node for 4. Normally we would go with the biggest signal, but you can see how the network partly believed the answer could have been 4. Perhaps the handwriting made it hard to be sure? This sort of uncertainty does happen with neural networks, and rather than see it as a bad thing, we should see it as a useful insight into how another answer was also a contender.