

Python Code to Create and Rescale the Output Array

Given below is the Python code to rescale the output values.

Now we need to turn these ideas into target arrays for the neural network training. You can see that if the label for a training example is 5, we need to create a target array for the output node where all the elements are small except the one corresponding to the label 5. That could look like the following `[0, 0, 0, 0, 0, 1, 0, 0, 0, 0]`.

In fact, we need to rescale those numbers because we've already seen how trying to get the neural network to create outputs of 0 and 1, which are impossible for the activation function, will drive large weights and a saturated network. So we'll use the values 0.01 and 0.99 instead, so the target for the label 5 should be `[0.01, 0.01, 0.01, 0.01, 0.01, 0.99, 0.01, 0.01, 0.01, 0.01]`.

Have a look at the following Python code which constructs the target matrix:

```
#output nodes is 10 (example)

all_values = data_list[1].split(',')

onodes = 10
targets = numpy.zeros(onodes) + 0.01
targets[int(all_values[0])] = 0.99
print(targets)
```



The first line, other than the comment, simply sets the number of output nodes to 10, which is right for our example with ten labels.

The second line simple uses a convenient *numpy* function called `numpy.zeros()` to create an array filled with zeros. The parameter it takes is the size and shape of the array we want. Here we just want a simple one of length `onodes`, which is the number of nodes on the final output layer. We add 0.01 to fix the problem with zeros we just talked about.

0.01 to fix the problem with zeros we just talked about.

The next line takes the first element of the MNIST dataset record, which is the training target label and converts that string into an integer. Remember that record is read from the source files as a text string, not a number. Once that conversion is done, that target label is used to set the right element of the targets list to 0.99. It looks neat because a label of 0 will be converted to integer 0, which is the correct index into the `targets[]` array for that label. Similarly, a label of 9 will be converted to integer 9, and `targets[9]` is indeed the last element of that array.