The Code Thus Far..

In this lesson, we will take a look at the code we have created so far.

Let's take a breath and pause to check what the code for the neural network class we're building up looks like. It should look something like the following.

```
# neural network class definition
class neuralNetwork:


    # initialise the neural network
    def __init__(self, inputnodes, hiddennodes, outputnodes, learningrate):
        # set number of nodes in each input, hidden, output layer
        self.inodes = inputnodes
        self.hnodes = hiddennodes
        self.onodes = outputnodes

        # link weight matrices, wih and who
        # weights inside the arrays are w_i_j, where link is from node i to node j in the nex
        # w11 w21
        # w12 w22 etc
        self.wih = numpy.random.normal(0.0, pow(self.hnodes, -0.5), (self.hnodes, self.inodes
        self.who = numpy.random.normal(0.0, pow(self.onodes, -0.5), (self.onodes, self.hnodes

        # learning rate
        self.lr = learningrate

        # activation function is the sigmoid function
        self.activation_function = lambda x: scipy.special.expit(x)

        pass


    # train the neural network
    def train():
        pass


    # query the neural network
    def query(self, inputs_list):
        # convert inputs list to 2d array
        inputs = numpy.array(inputs_list, ndmin=2).T

        # calculate signals into hidden layer
        hidden_inputs = numpy.dot(self.wih, inputs)
        # calculate the signals emerging from hidden layer
        hidden_outputs = self.activation_function(hidden_inputs)

        # calculate signals into final output layer
```

```
        # calculate signals into final output layer
        final_inputs = numpy.dot(self.who, hidden_outputs)
        # calculate the signals emerging from final output layer
        final_outputs = self.activation_function(final_inputs)

        return final_outputs
```

That is just the class, aside from that we should be importing the `numpy` and `scipy.special` modules right at the top of the code in the cell below:

```
import numpy
# scipy.special for the sigmoid function expit()
import scipy.special
```

It is worth briefly noting the `query()` function only needs the `input_list`. It doesn't need any other input.

That's good progress, and now we look at the missing piece, the `train()` function. Remember there are two phases to training, the first is calculating the output just as `query()` does it, and the second part is backpropagating the errors to inform how the link weights are refined.