Initializing the Network

Let's begin with the initialization of the Neural Network.

We know we need to set the number of input, hidden and output layer nodes. That defines the shape and size of the neural network. Rather than set these in stone, we'll let them be set when a new neural network object is created by using parameters. That way we retain the choice to create new neural networks of different sizes with ease.

There's an important point underneath what we've just decided. Good programmers, computer scientists, and mathematicians try to create general code rather than specific code whenever they can. It is a good habit because it forces us to think about solving problems in a deeper more widely applicable way. If we do this, then our solutions can be applied to different scenarios. What this means here is that we will try to develop code for a neural network which tries to keep as many useful options open, and assumptions to a minimum, so that the code can easily be used for different needs. We want the same class to be able to create a small neural network as well as a very large one — simply by passing the desired size as parameters.

Don't forget the learning rate too. That's a useful parameter to set when we create a new neural network. So let's see what the `__init__()` function might look like:

```
# initialise the neural networ
    def __init__(self, inputnod
        # set number of nodes 
        self.inodes = inputnod
        self.hnodes = hiddenno
        self.onodes = outputno

        # learning rate
        self.lr = learningrate
        pass
```

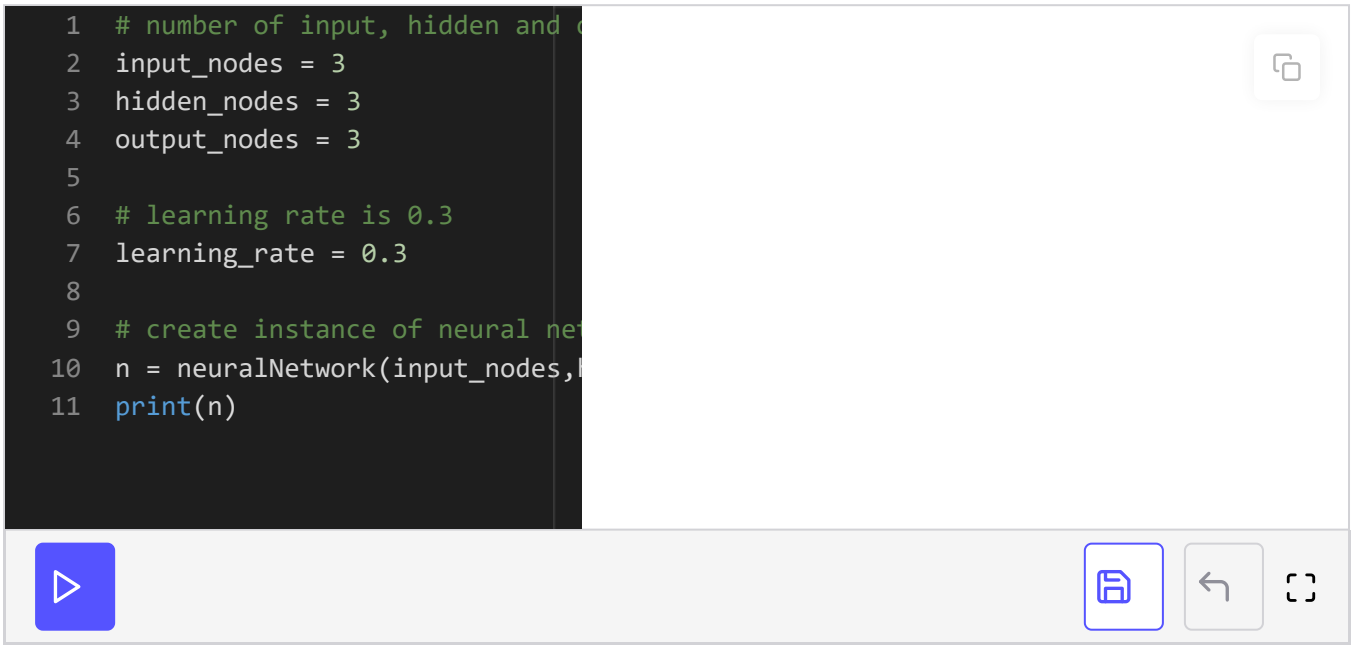
Let's add that to our class definition of a neural network, and try creating a

small neural network object with three nodes in each layer, and a learning rate of 0.5.

```
# number of input, hidden and
input_nodes = 3
hidden_nodes = 3
output_nodes = 3

# learning rate is 0.3
learning_rate = 0.3

# create instance of neural ne
n = neuralNetwork(input_nodes,
print(n)
```



That would give us an object, sure, but it wouldn't be very useful yet as we haven't coded any functions that do useful work yet. That's ok; it is a good technique to start small and grow code, finding and fixing problems along the way.

What do we do next? Well, we've told the neural network object how many inputs, hidden and output layer nodes we want, but nothing has really been done about it.