

# Network Initialization

Your first task will be to define a matlab structure to store the parameters and activations/hidden nodes of a fully connected neural network. As explained in the neural network lectures, every layer in the fully connected network can be parameterized as a weight matrix. For instance, a network with two fully connected layers will have two matrices storing the parameters of such a network. The dimensions of these matrices will be determined by the number of hidden nodes in the adjacent layers of the network. For instance, if a layer  $l$  has  $k$  hidden nodes, and a layer  $l + 1$  has  $p$  hidden nodes, then the parameter matrix  $W^{(l)}$  between these two layers will have the dimensions  $p \times k$ . The reason for this is that every fully connected layer implements the following operation:

$$z^{(l+1)} = W^{(l)} a^{(l)}$$

Therefore, in this example,  $z^{(l+1)}$  is the  $p$  dimensional vector that stores hidden nodes in layer  $l + 1$  (before the non-linear function is applied). Then,  $a^{(l)}$  is the  $k$  dimensional vector storing the hidden nodes in layer  $l$  (after a non-linear function has been applied). Finally, as mentioned above,  $W^{(l)}$  is a  $p \times k$  matrix that stores fully connected layer parameters in layer  $l$ .

As a first step, you will initialize the parameters of a network to random values. We will be considering a network that takes 336 dimensional features as its input, projects them to 100 hidden nodes in layer 1, and then projects them to 20 output nodes in layer 2, where each of 20 nodes represents a probability for one of the 20 object classes. The entire network will be stored in a matlab variable nn. The parameters of a fully connected layer will be stored into variables nn.W{1}, nn.W{2} for layers 1, and 2 respectively. Your job will be to correctly figure out the dimensions of each of these variables.

## Your Function

 Save  Reset  MATLAB Documentation (<https://www.mathworks.com/help/>)

```
1 function nn = InitializeNetwork(architecture)
2     % initialize the network
3     %
4     % Input:
5     % - architecture: an 1 x 1 dimensional vector indicating a number of nodes at each layer of a network. For
6     % Output:
7     % - nn: a neural network structure storing randomly initialized parameters of a network in the variables nn
8
9     rand('state',0)
10    for i = 2 : size(architecture,2)
11        rows = architecture(i); %specificy the number of rows for the parameter matrix;
12        cols = architecture(i-1); %specify the number of columns in the parameter matrix;
13        nn.W{i-1}= (rand(rows, cols) - 0.5) * 2 * 4 * sqrt(6 / (rows + cols));
14    end
15 end
16
```

## Code to call your function

 Reset

```
1 architecture=[336 100 20];
2 nn = InitializeNetwork(architecture);
```