Determine the Number of Neurons and Layers

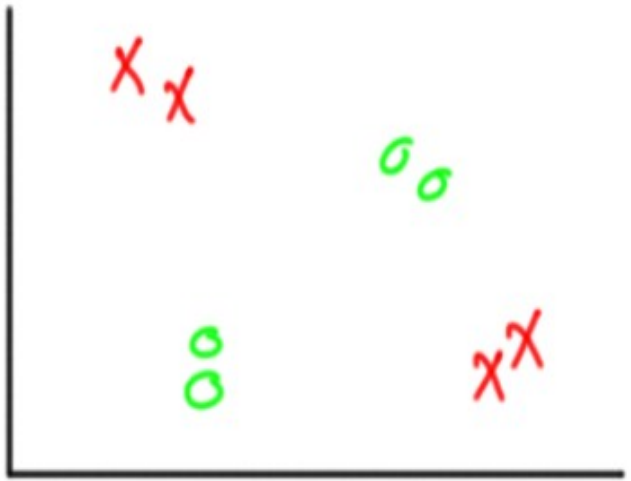
https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e

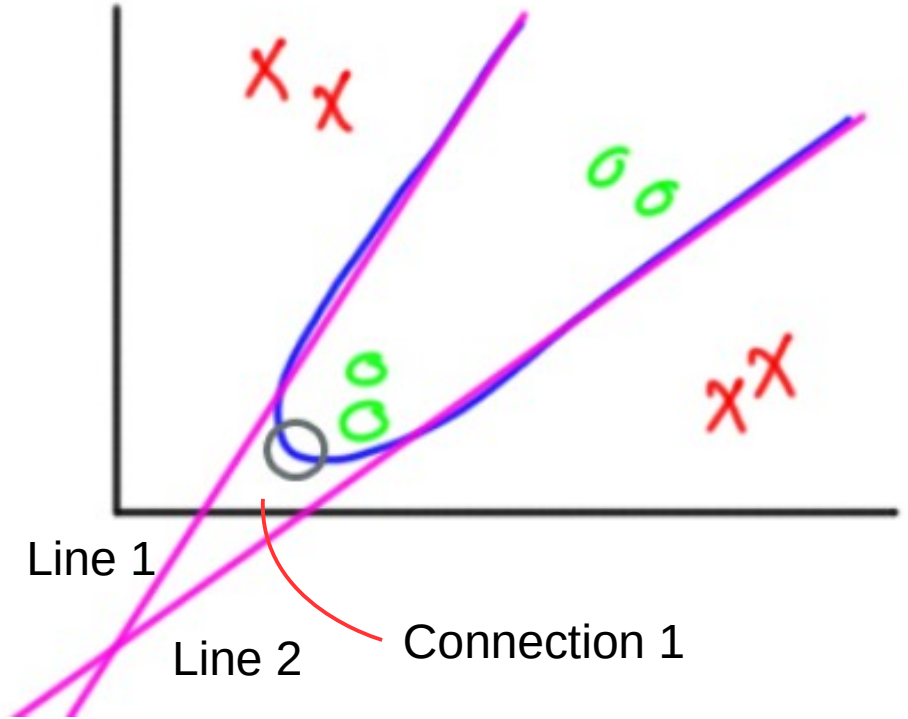
In artificial neural networks, hidden layers are required if and only if the data must be separated non-linearly.

Step 1. Get the data set first.
Step 2. Draw decision boundary lines N to separate the classes.
Step 3. Select number of the hidden neurons = N for the first hidden layer.
Step 4. Count the intersections M of the boundary lines.
Step 5. Add M new hidden layers.

Inspect Data Set and Draw Decision Lines

https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e





Decision making function is composed of lines, each line is a linear function from a single neuron:

$y = x1 * w1 + x2 * w2 + b$ … (1)

Boundary Lines and Number of Neurons

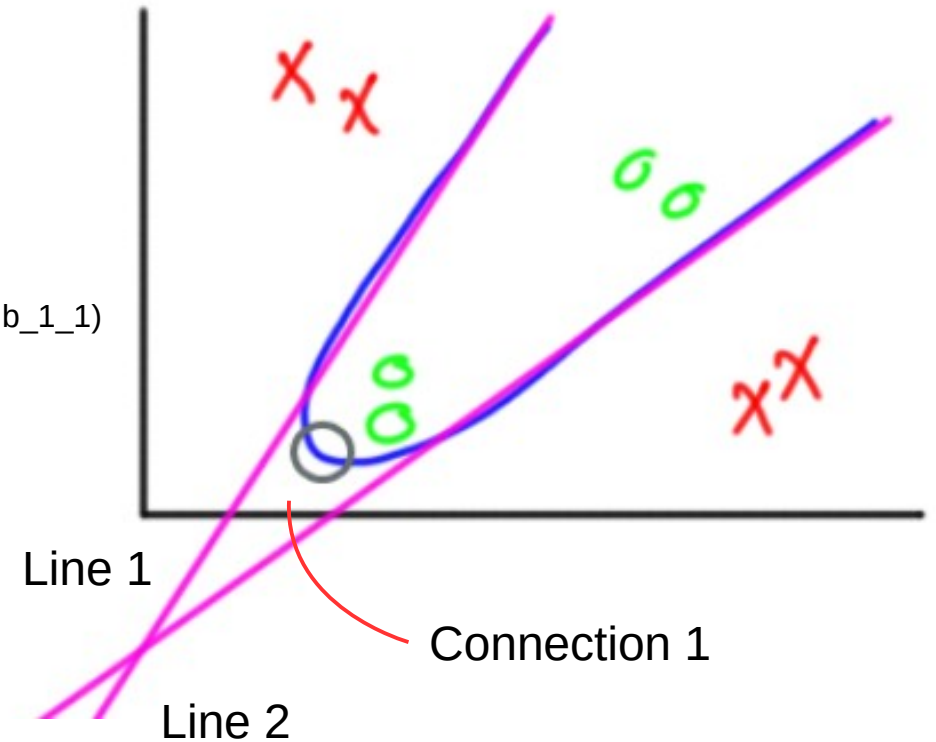
https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e

Step 3. Count the number N of these decision boundary as a set of lines.
Step 4. The number N of selected lines equal to the number of hidden neurons in the first hidden layer.

2 lines leads to 2 neurons in the first hidden layer

```
self.w_1_1=s[0]   # weight

h_1_1 = sigmoid(self.w_1_1 * x[0] + self.w_1_2 * x[1] + self.b_1_1)
# Neuron 1 at hidden layer 1
```



Connecting Lines (Intersection) for Hidden Layers

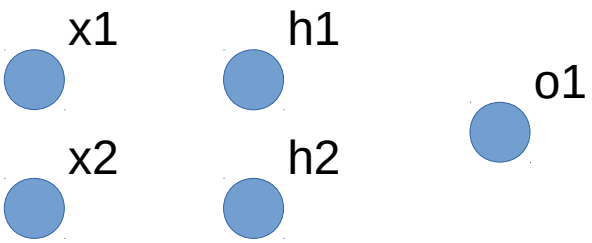
https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e

Step 5. To connect the lines created by the previous layer, a new hidden layer is added. Note that a new hidden layer is added each time you need to create connections among the lines in the previous hidden layer. The number of hidden neurons in each new hidden layer equals the number of connections to be made.

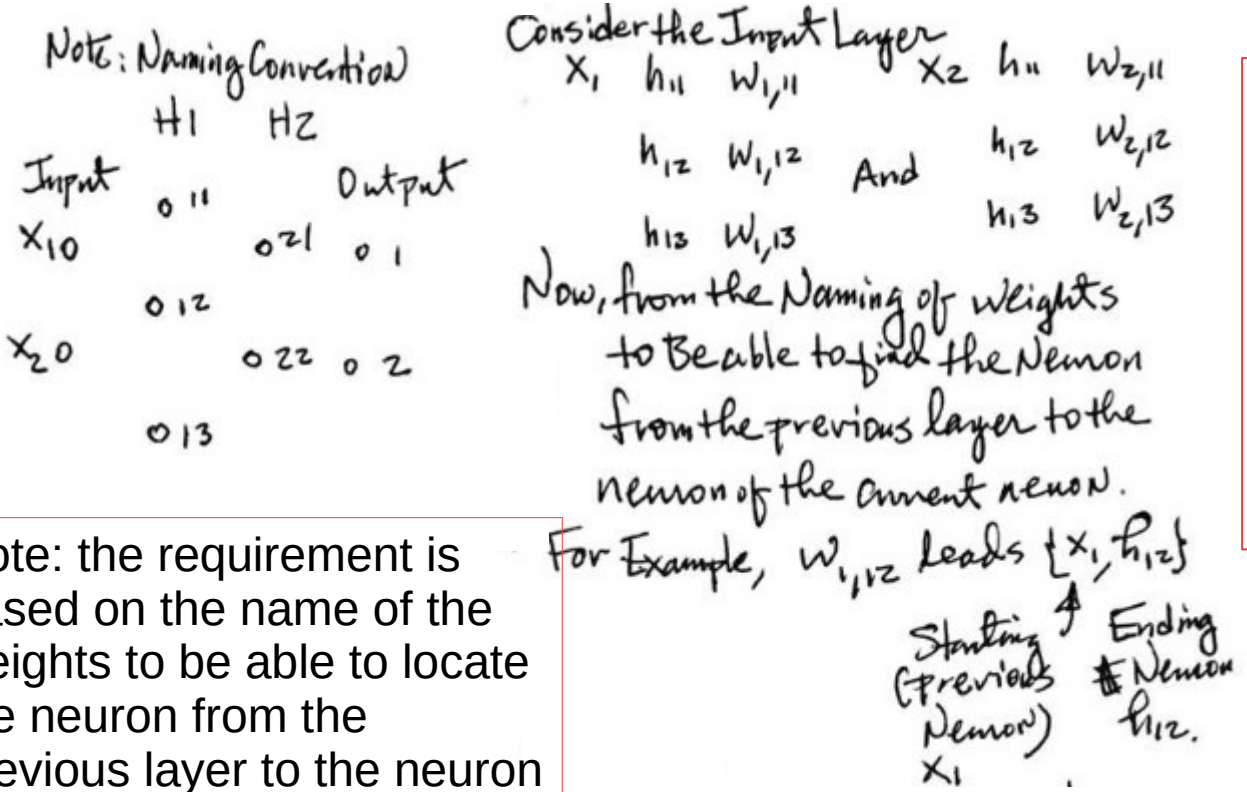
1. Need to connect Line 1 and Line 2 > add new hidden layer (or output layer);

2. One connection to join Line 1 and 2, so (a) add another layer, and (b) since it is one connection for both lines, so one neuron at that layer .

So the architecture of the NN is given in the right



Naming Convention for Coding



Note: the requirement is based on the name of the weights to be able to locate the neuron from the previous layer to the neuron of the current layer.

Example:

```
def __init__(self):
  # load prior weights & biases from a text file
  s=[]
  with open('trained_weights_bias.txt','r') as f:
      for line in f:
          for num in line.split(','):
              s.append(float(num))
  self.w_1_1=s[0]
  self.w_1_2=s[1]
  self.b_1_1=s[2]
```

Additional Example

https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e