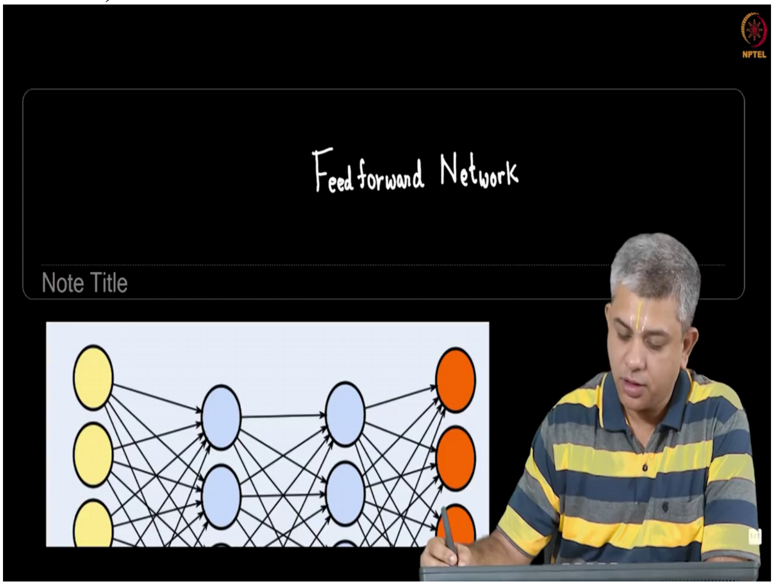
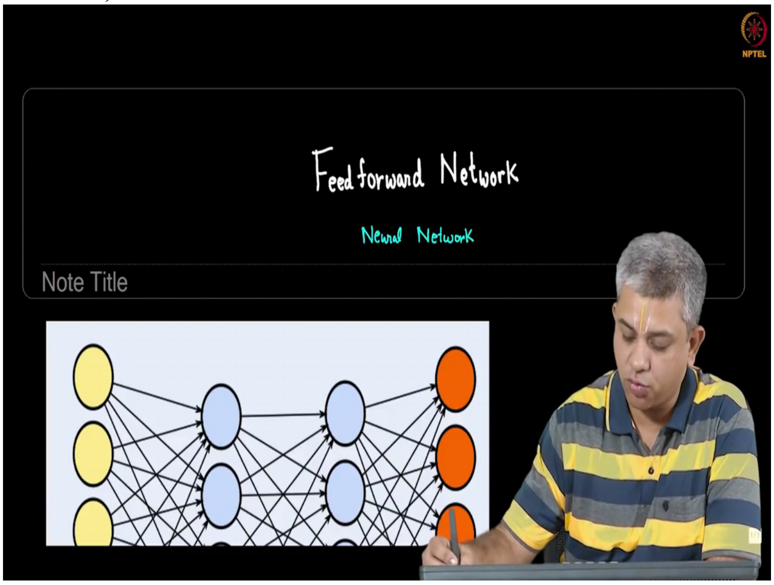
Machine Learning for Engineering and Science Applications
Professor Doctor Balaji Srinivasan
Department of Mechanical Engineering
Indian Institute of Technology Madras
Feedforward Neural Network

(Refer Slide Time: 00:14)



In this video we look at a general version of the neural network called the feedforward network. In the previous videos we had seen what an artificial neuron was.

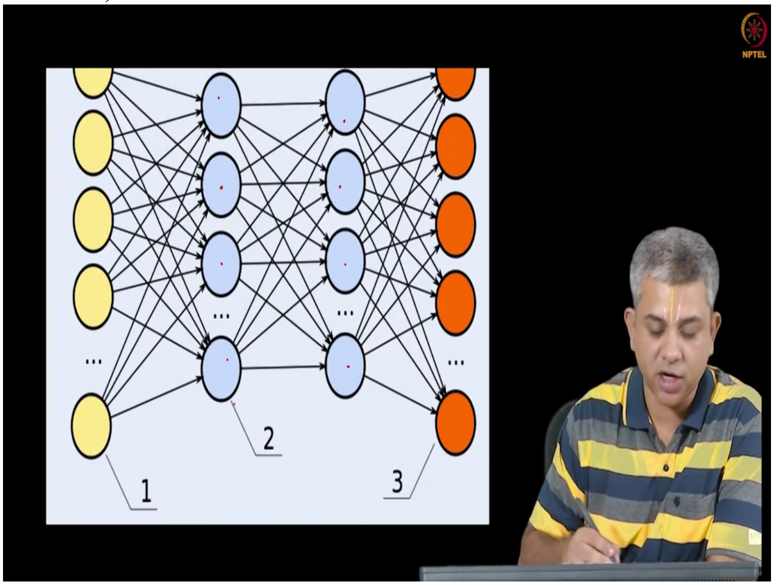
(Refer Slide Time: 00:32)



A feedforward network or a simple neural network, the term that you would have heard most commonly is basically a collection of neurons. Each of these units here is neuron.

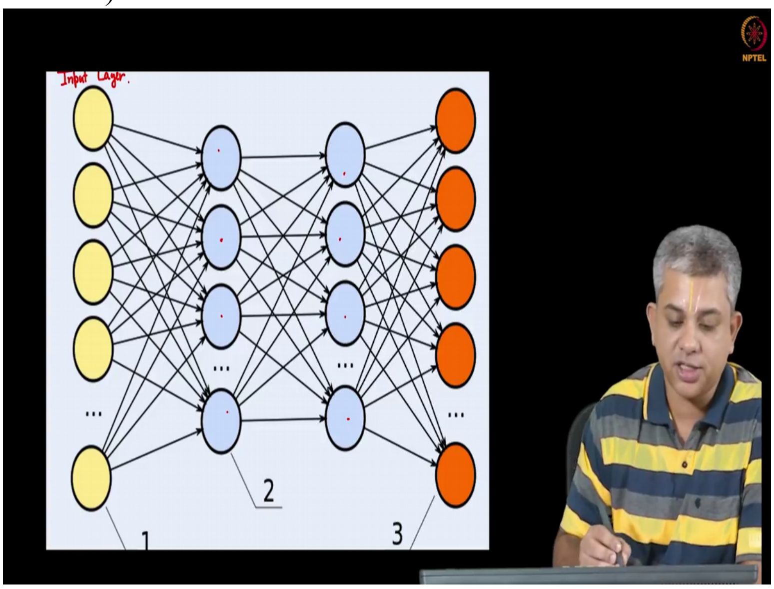
Now each of these neurons or each of these layers

(Refer Slide Time: 00:49)



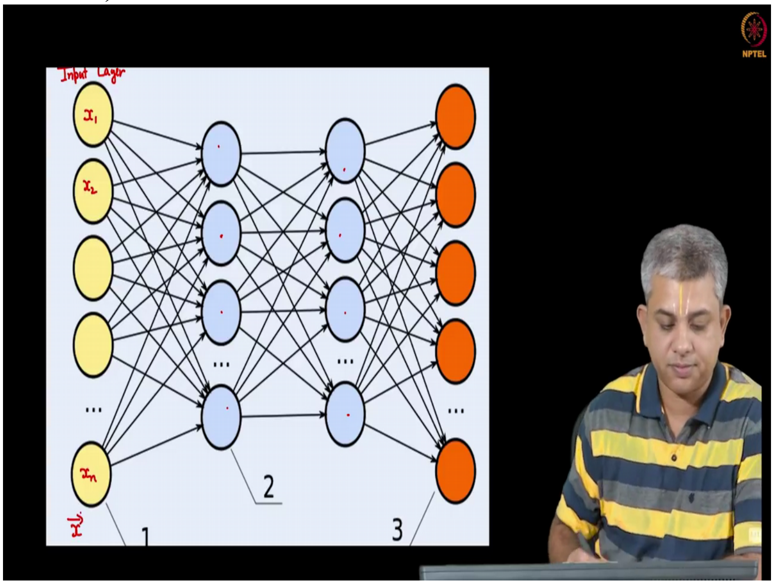
which are vertically concatenated have specific name. The very first layer is called the input layer.

(Refer Slide Time: 01:01)



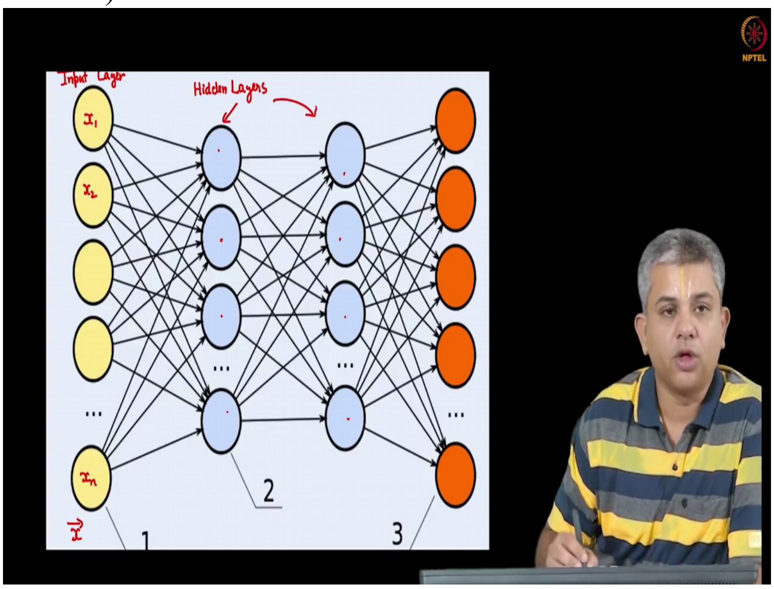
We had seen this during logistic regression even in linear regression. So you will have multiple features. This is the input vector.

(Refer Slide Time: 01:12)



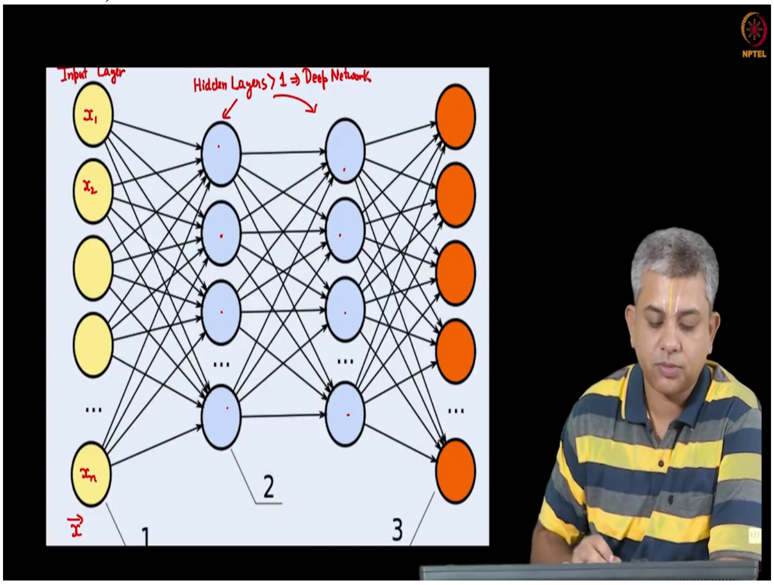
The intermediate layers here are called hidden layers. You could have multiple

(Refer Slide Time: 01:25)



hidden layers. If the number of hidden layers is greater than 1 then it is called the deep network. Hence the name deep learning.

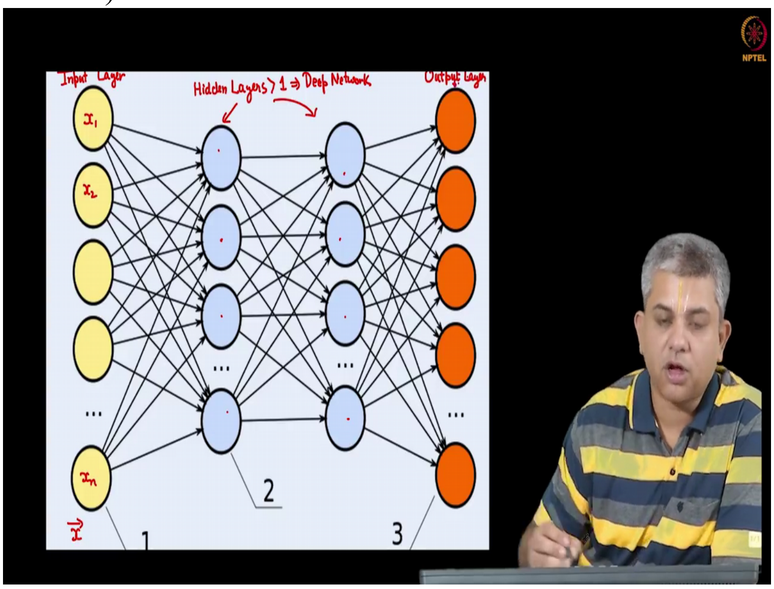
(Refer Slide Time: 01:39)



So deep network is simply a network with the number of hidden layers greater than 1.

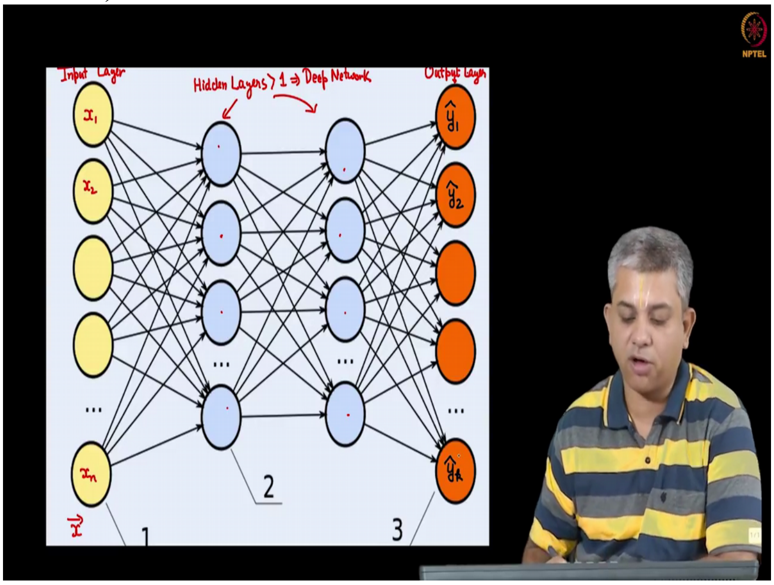
The final layer where you actually get the output you are interested in is called

(Refer Slide Time: 01:52)



output layer. So you have our predictions. We have our predictions here, y 1 hat, remember hat is used for predictions uptil whatever is the number of

(Refer Slide Time: 02:05)

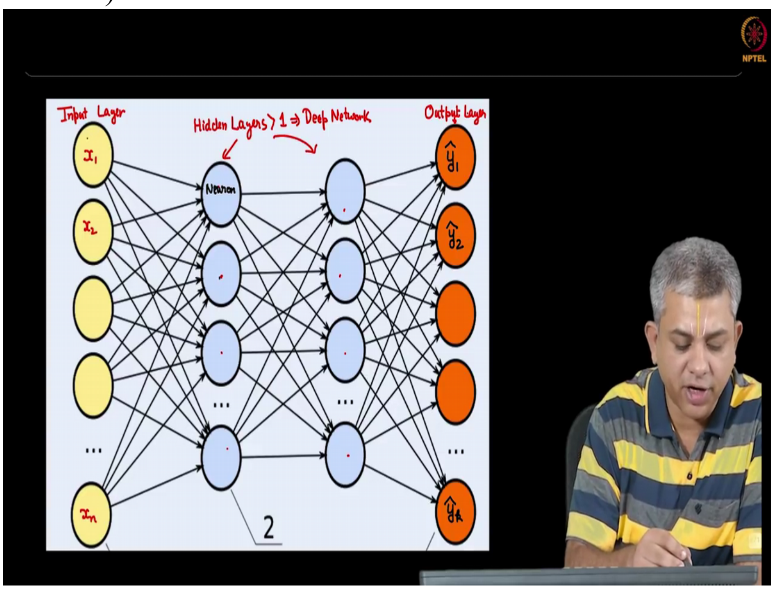


classes that we are predicting for.

Remember k in general need not be equal to n and in general each layer might have a different size. So these are the elements. So each of these elements here is an artificial neuron, Ok.

Technically speaking we can

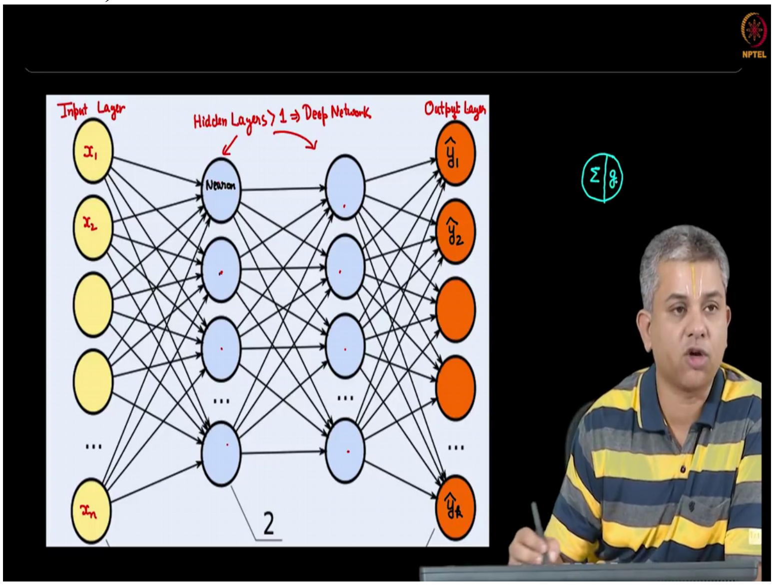
(Refer Slide Time: 02:24)



even treat the input layers as if they were neurons but generally it is only after the input layer that we look at each of these neurons and call them an artificial neuron.

Remember that within each neuron we have 2 portions. We have

(Refer Slide Time: 02:42)

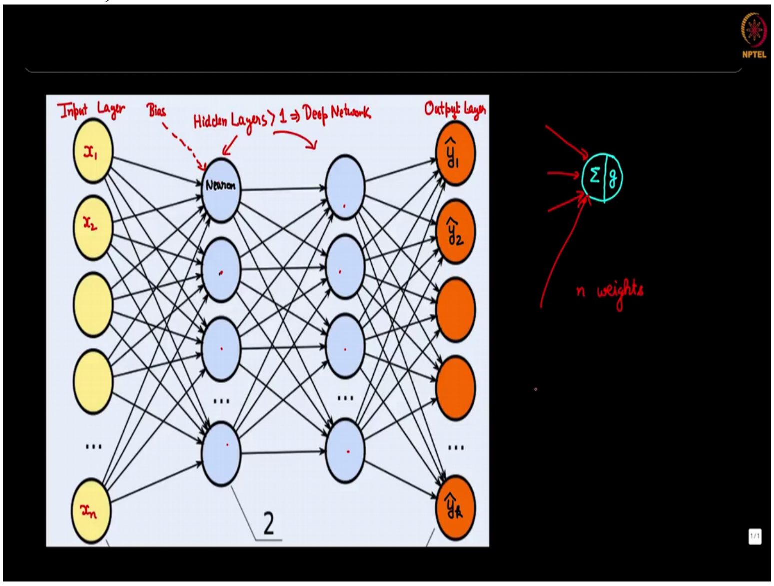


a linear combination and we have nonlinear activation function sitting there.

Now if I look at any neuron here, so for example this neuron it has inputs coming from all the previous entities in the input layer, Ok. So for example, so this neuron here has n inputs plus, even though not explicitly shown here, you will have a bias unit which will be coming in here.

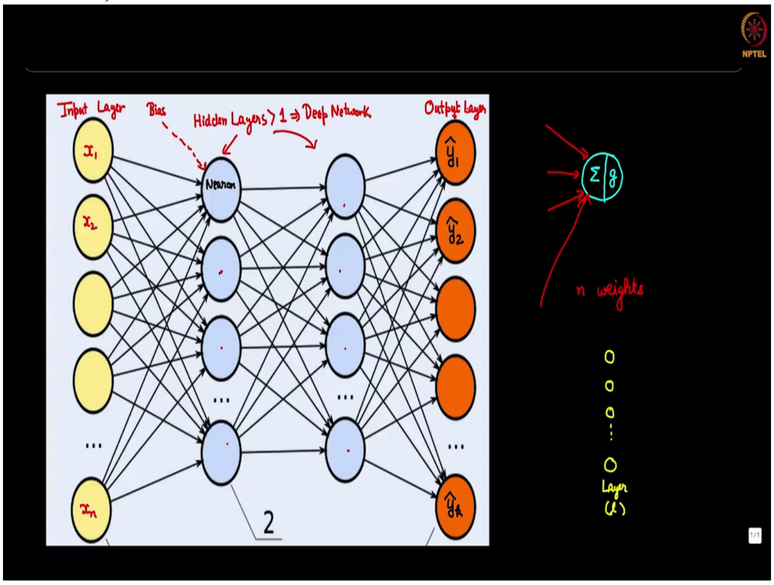
So for this neuron you have n weights from the input layer. So let us

(Refer Slide Time: 03:32)



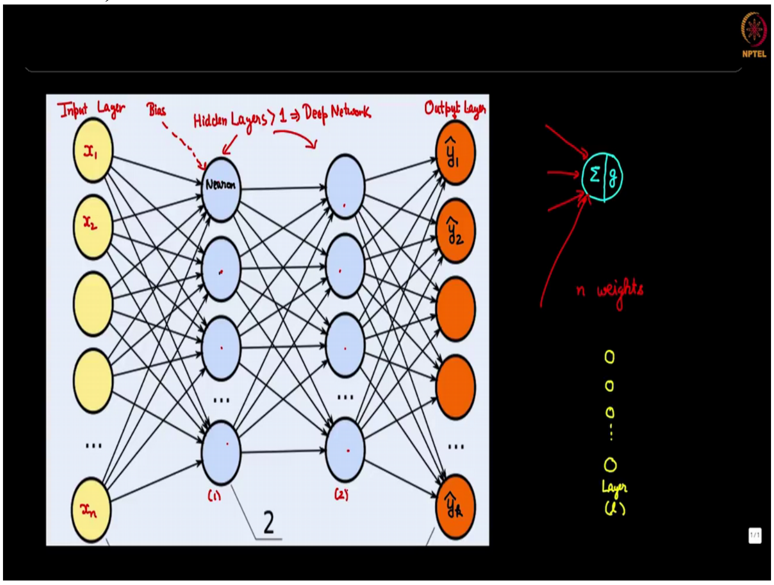
take a general neuron or a general set of neurons in some hidden layer. So let us say this is layer l, Ok for example

(Refer Slide Time: 03:47)



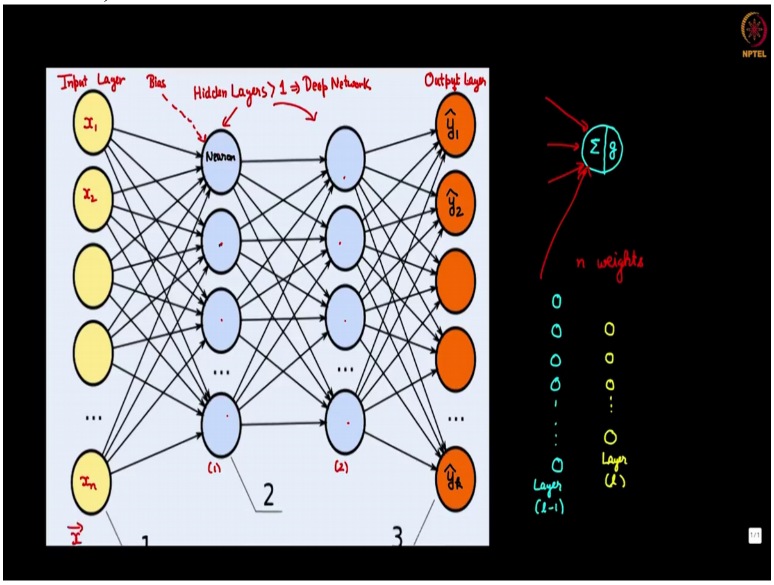
this one would be layer 1. This one would be layer 2, hidden layer 1, hidden layer 2.

(Refer Slide Time: 03:53)



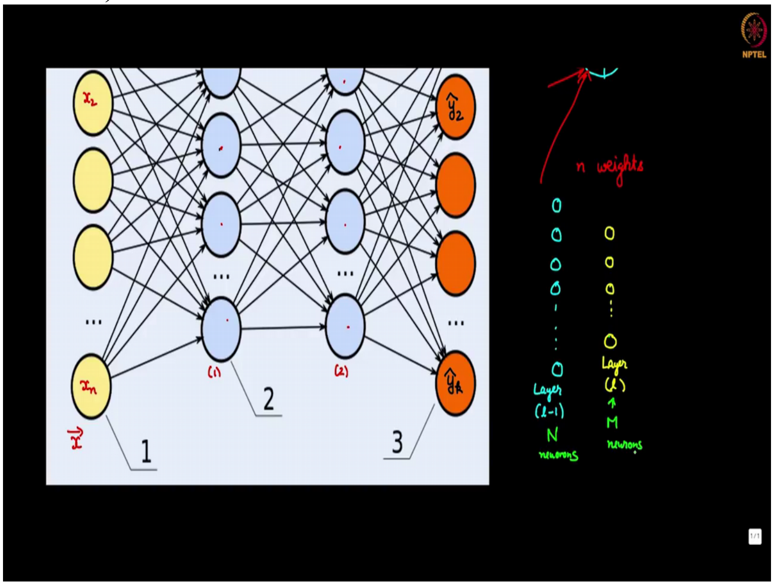
Let us say we have layer l where all these neurons are there. And you have a prior layer. This is layer l minus 1.

(Refer Slide Time: 04:11)



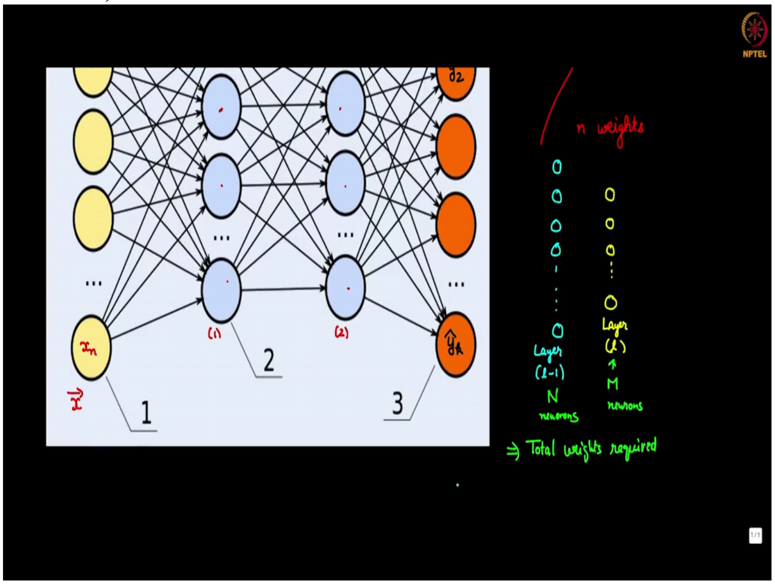
Let us say further that this layer had n neurons and this layer m, sorry l has N neurons.

(Refer Slide Time: 04:28)



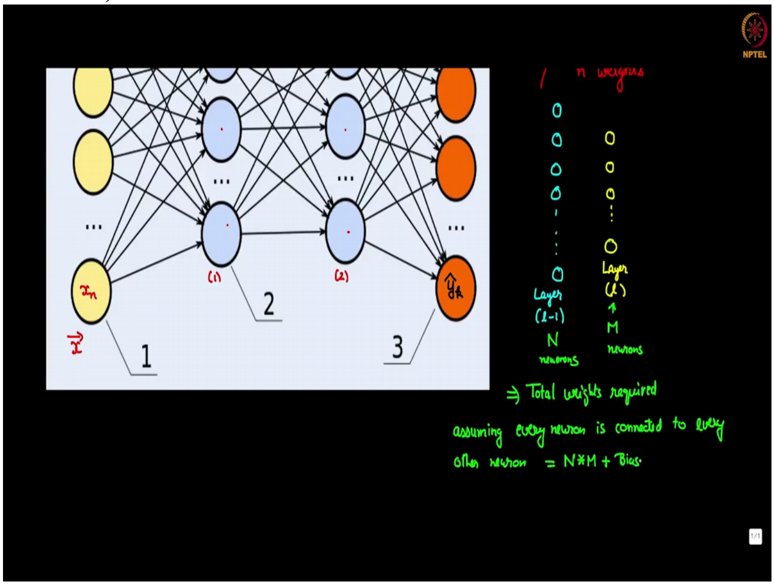
So this means the total number of weights required, assuming

(Refer Slide Time: 04:43)



every neuron is connected to every other neuron would be N times M plus the bias units.

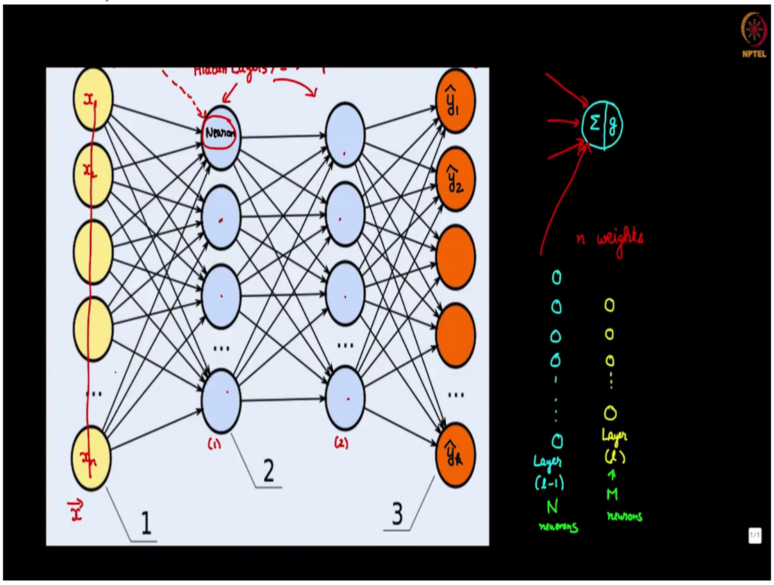
(Refer Slide Time: 05:13)



Now how many bias units do we have in such a case, Ok?

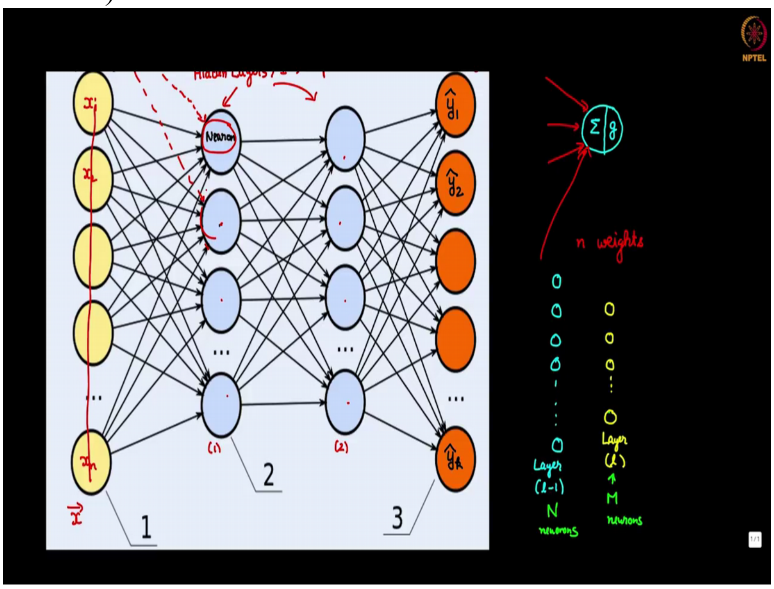
Now if I consider this neuron for example, this takes input from all these n,

(Refer Slide Time: 05:28)



plus one bias. Now if I take this one, it also takes all these n, plus a different bias,

(Refer Slide Time: 05:40)

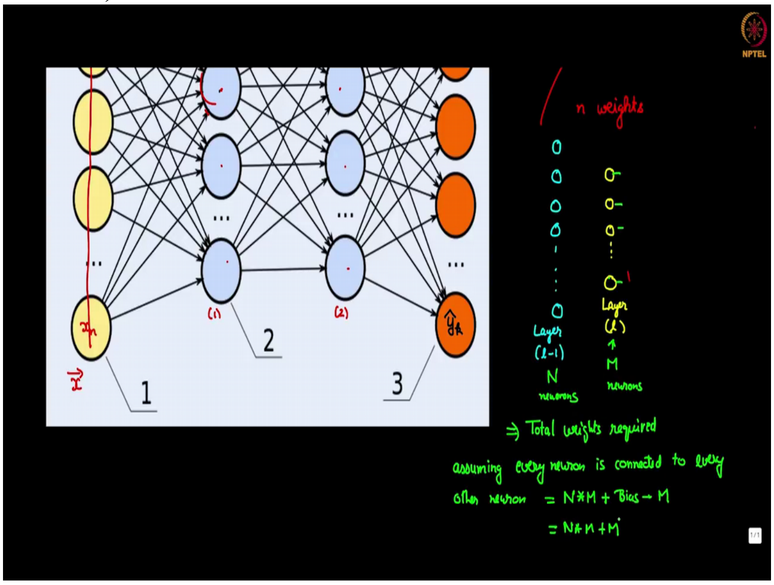


Ok.

So each neuron gets a different bias. So in this case if we look at these M neurons you have N times M which are normal weights, the number of bias weights will be equal to M, because each of these neurons has a different bias.

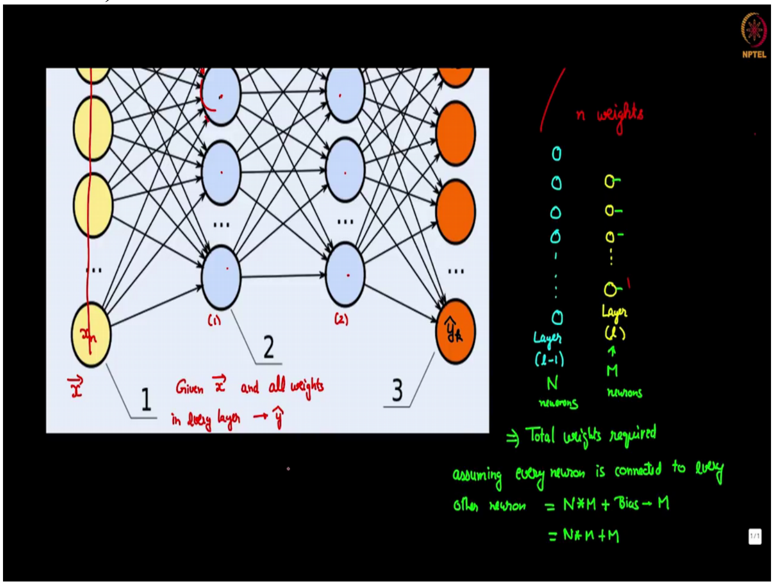
So the total number of weights in such a case is N M plus M.

(Refer Slide Time: 06:06)



In a feedforward network, all you need to do is you give all these x is, if you give X vector and all the weights in every layer we can find out y hat vector,

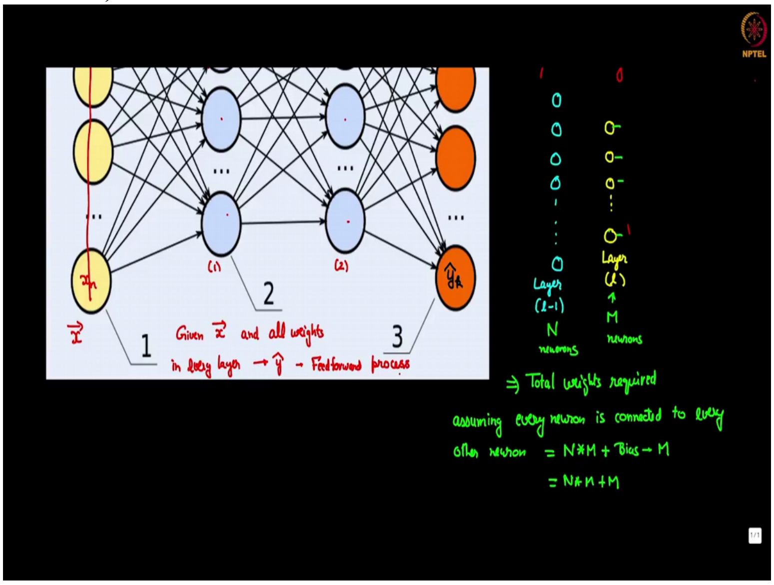
(Refer Slide Time: 06:34)



Ok.

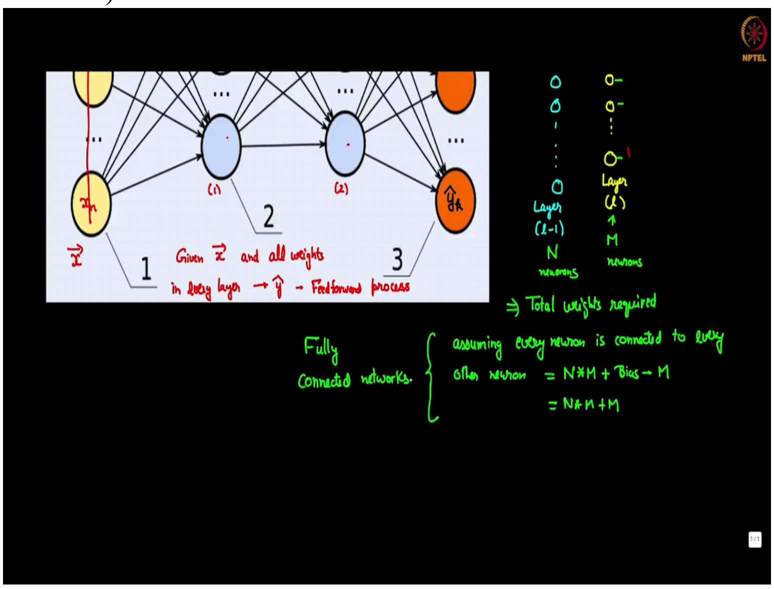
So this is called the feedforward process. Basically you are feeding the x, you also give all the ws of every single layer here and simply by taking a linear combination, nonlinearity; linear combination, nonlinearity; linear combination, nonlinearity you can predict all the ys. This is called the feedforward process.

(Refer Slide Time: 07:06)



Such a network where all neurons are connected to every other neuron are called fully connected networks.

(Refer Slide Time: 07:24)



In the general case you need not have all connections active.

In fact we will see later in convolutional neural networks that we have only some of these weights which are non-zero and most of them are 0 which means each neuron is connected only to a few other neurons in the previous layer.

So that would be the special case but in the most general case you can think of a fully connected network, sometimes simply called F C network. The assumption behind the feedforward process is you know all the weights. And you know all the input neurons.

Later on we will see when we come to back propagation how to actually determine these weights.