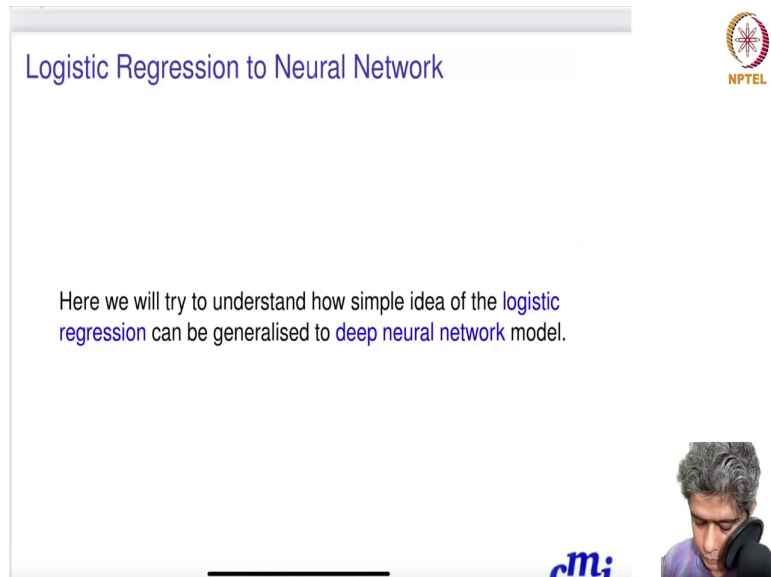


**Predictive Analytics - Regression and Classification**  
**Prof. Sourish Das**  
**Department of Mathematics**  
**Chennai Mathematical Institute**

**Lecture - 45**  
**Logistic Regression to Deep Learning Neural Network**

Hello all, welcome back to the second video of lecture 30. In this lecture, we are going to understand how a simple idea of Logistic Regression can be generalized to Deep Neural Network Model.

(Refer Slide Time: 00:33)



The screenshot shows a video slide with the following content:

- Title:** Logistic Regression to Neural Network
- Text:** Here we will try to understand how simple idea of the **logistic regression** can be generalised to **deep neural network** model.
- Logos:** NPTEL logo in the top right corner, and CMi logo in the bottom right corner.
- Image:** A small inset image of Prof. Sourish Das in the bottom right corner, looking down.

(Refer Slide Time: 00:35)

**Logistic Regression**

mathematical representation of logistic Regression




Consider data set  $\mathcal{D} = (y_i, \mathbf{x}_i | i = 1, 2, \dots, n)$

$z_i = \mathbf{x}_i^T \beta_1 = \omega_1$

$p_i(1) = \frac{\exp(z_i)}{1 + \exp(z_i)}$  and  $p_i(0) = \frac{1}{1 + \exp(z_i)}$

$y_i = \begin{cases} 1 & \text{with } \mathbb{P}(y_i = 1) = p_i(1) \\ 0 & \text{with } \mathbb{P}(y_i = 0) = p_i(0) = 1 - p_i(1) \end{cases}$

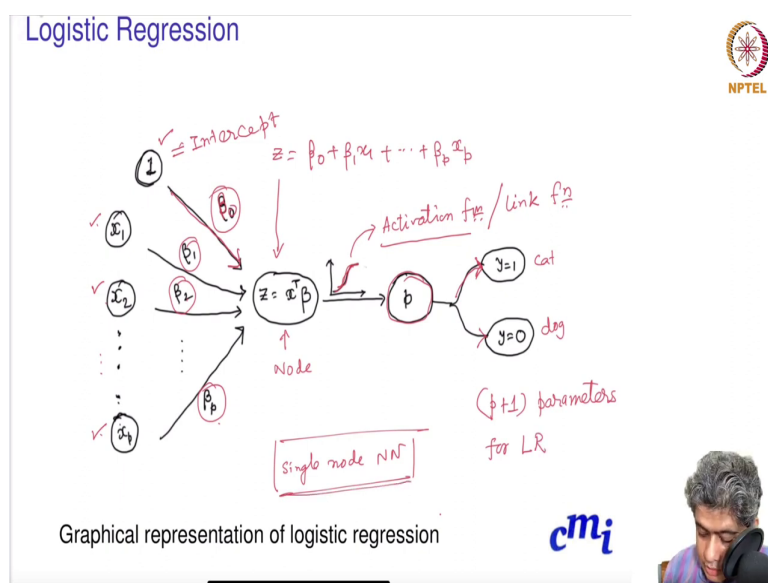
►  $\omega_1$  is the  $p \times 1$  matrix,  
 $\beta_1$



So, let us start with the basic logistic regression model. So, we have this data dataset which is  $y_i$  and  $x_i$ 's. There are  $n$  samples  $i$ 's from 1 to up to  $n$  and then  $z_i$  equal to  $x_i$  transpose beta, ok. So,  $x_i$  transpose beta is  $z_i$  and then you have  $p_i(1)$  equal to this sigmoid function. Then with probability  $p$ , you have  $y$  equal to 1 and probability  $1 - p$ , you have  $y_i$  equal to 0.

Now,  $\omega_1$  is my  $\beta_1$ . So, this  $\beta_1$  is also  $\omega_1$ . Why I am using  $\omega_1$ ? Because in machine learning literature generally, we use  $\omega_1$  as coefficient. In statistics literature, we use  $\beta$  as the notation for coefficient. So, this is mathematical representation of logistic regression. This is mathematical representation of logistic regression ok.

(Refer Slide Time: 02:17)



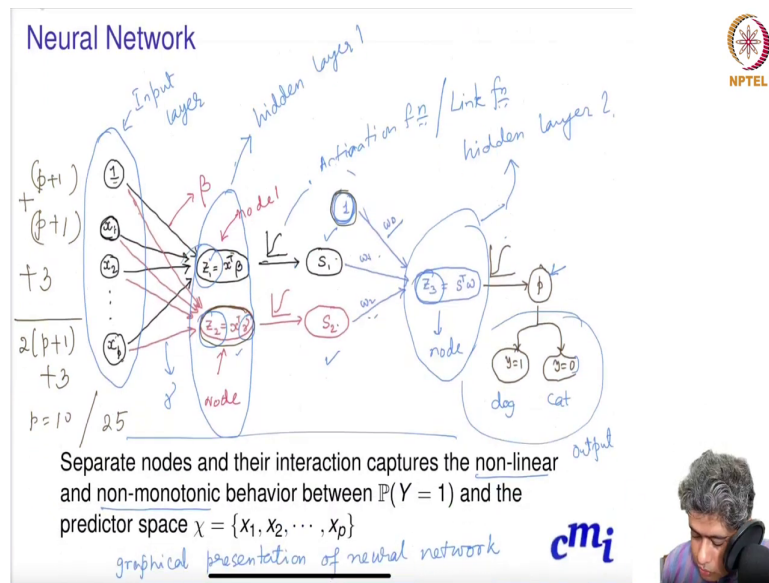
Now, what we do we graphically represent this logistic regression using this graphical nodes and arrows. So, we have  $p$  features  $x_1, x_2, \dots, x_p$  and we have a intercept parameter. This is intercept and these arrows represents the coefficients  $\beta_0, \beta_1, \beta_2$  and  $\beta_p$ . So, what you have is whatever the coefficient times the features.

So,  $\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$  that is our  $z$  that is what is our node. This is often time we call it node, ok. Then this node goes through a sigmoid transformation called activation function. We call it typically activation function in statistics we call it link function, ok. In statistics what we call it link function, in machine learning we call it activation function.

Same thing boss no difference, it is just ok and then once we have typically its sigmoid type behavior and then you get a  $p$  and with certain probability  $p$  you say I am observing  $y$  equal to

1 or y equal to 0 in typical machine learning literature they will say cat and dog, ok (Refer Time: 04:15) So, this is the graphical representation of logistic regression or single node neural network.

(Refer Slide Time: 04:31)



Now, can I generalize this situation? So, I had single node this is my node 1, I do not have to just stick to a node 1, I can create a new node this is my node 2, ok. Now, this where coefficient betas, but what I am thinking this is a different nodes. So, we are going to use a different color these are gammas.

So, these coefficients are called gammas. So, different weights that puts me two different nodes  $z_1$  and  $z_2$  and then they put go through a activation function again each of them will go through a activation function or in statistics literature we call it link function ok. And then now we have  $s_1$  and  $s_2$ .

And then again, I can consider a intercept parameter then  $\omega_0$   $\omega_1$   $\omega_2$  that gives me the  $z_3$  another node then from  $z_3$  this is a node and then from  $z_3$  again another activation function gives me the  $p$  and from  $p$  probability I get dog or cat ok.

So, this is my neural network. So, I have this is typically called input layer this is typically called input layer and this is typically called output layer this is called output layer, this is hidden layer 1, hidden layer 1 this is typically hidden layer 2. So, this how these things get sorry this is hidden layer 2 this is activation function.

So, I have to. So, this is not hidden layer, this is my hidden layer hidden layer 2 this is my hidden layer 2 and eventually and this is the activation variables after activation. So, this is hidden layer 1 is with 2 nodes and hidden layer 2 is with 1 node and from there we get the final output layer. So, why in the neural network we put separate hidden layers and we add more hidden layers and more nodes in each layers and all these things because you see this puts a new engineered features.

So,  $z_1$ ,  $z_2$ ,  $z_3$  this is  $z_3$  this is  $z_2$ ,  $z_1$  these are all engineered features you can see them as an engineered feature these nodes are nothing but engineered features, they are giving lot of interactions between  $x_1$ ,  $x_2$  and  $x_p$  and these interaction effects create a engineered feature in a higher dimension as a result. It is able to model non monotonic behavior non-linear non-monotonic behavior non-linear non-monotonic behavior. So, that is the purpose of neural network.

(Refer Slide Time: 08:51)

### Mathematical Presentation of Neural Network

x<sup>T</sup> Input Layer

Consider data set  $\mathcal{D} = (y_i, \mathbf{x}_i | i = 1, 2, \dots, n)$

}

[

$z_{1i} = \mathbf{x}_i^T \omega_1$

]

→ hidden layer 1 with two nodes

}

$s_{(1)i}^T = \frac{\exp(z_{1i})}{1 + \exp(z_{1i})}$

→ Activation fn / Link fn

}

$z_{2i} = s_{(1)i}^T \omega_2$

→ hidden layer 2 with single node

$p_i(1) = \frac{\exp(z_{2i})}{1 + \exp(z_{2i})}$

and




$p_i(0) = \frac{1}{1 + \exp(z_{2i})}$

/ Link fn

$y_i =$

$$\begin{cases} 1 & \text{with } \mathbb{P}(y_i = 1) = p_i(1) \\ 0 & \text{with } \mathbb{P}(y_i = 0) = p_i(0) = 1 - p_i(1) \end{cases}$$

output layer

So, this is the mathematical representation of the neural network that we just presented above this is the graphical presentation of; graphical presentation of neural network graphical presentation of neural network and this is mathematical representation this of the same neural network with 2 hidden layer and in the layer 1 there will be 2 nodes and. So, this is my first hidden layer ok.

So, this is my first. So, x transposes are my input layer you can think of input layer this is my hidden layer hidden layer 1 with 2 nodes with 2 nodes and this is hidden layer 2 hidden layer 2 with single node this is the activation function are known as link function in statistics ok.

So, this is link function and this is output layer this is output layer ok. So, you have a now graphical representation of neural network this is mathematical representation of the neural




network and these interactions gives you in engineered interaction this if you carefully look into it there will be a lot of interactions.

(Refer Slide Time: 11:11)

Explosion of the parameter space

- ▶ If we add nodes without thinking that may lead to the explosion of parameter space.
- ▶ Check how many parameters do we have in logistic regression, which is single node NN?
- ▶ Check how many parameters do we have in double node NN, which we considered?

- 1 ▶ Shall we add more nodes?
- 2 ▶ Shall we add more layers?



So, if we add nodes the one of the major problem with the neural network is explosion of parameter space this is you have to be very very super careful you have to be super careful that neural network very easily the parameter space can super explode if you add nodes and layers without thinking that may lead to the explosion of the parameter space.

Check how many parameters do we have in logistic regression which is single node neural network? So, stop the video for 5 minutes go back to the logistic regression. So, let me just go back here and how many parameters you see in a single node neural network ok how many parameters you see?

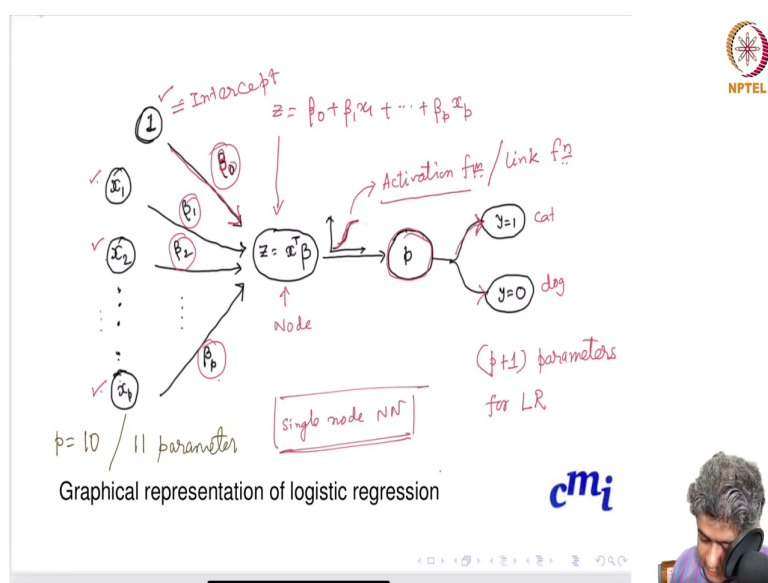
I hope you have figured it out by now. So, let us check how many parameters you have. So, you have  $x_1 \times x_2 \times p$ . So, you have  $p + 1$  to  $p$  many parameters and one more parameters for intercept. So, you have  $p + 1$  parameters for logistic regression which is single node neural network ok.

Let us now go and check how many parameters you have for neural network this neural network with two hidden layers and three nodes can you identify it. So, stop your stop this video pause this video ok and check yourself try for take 2 to 3 minutes try yourself and identify how many parameters you have in this neural network.

So, I hope you figured it out let us let me tell you. So, here you have two nodes from node 1 you have  $p + 1$  many parameters because you have  $p$  features and intercept. So, from node 1 you have  $p + 1$  from node 2 you have  $p + 1$  ok. So,  $p + 1$  plus  $p + 1$  and then these two nodes activating and you are getting  $s_1$  and  $s_2$ . And so, this is like  $2 \times 1 \times 2$  and then it is intercept so,  $1 \times 2 \times 3$  plus 3 ok. So,  $2$  into  $p + 1$  plus 3.



(Refer Slide Time: 14:29)



So, if you have if you use a logistic regression with  $p$  equal to 10 a single node neural network means you have total 11 parameters. Now, if you have a simple neural network with only two hidden layers one in the first hidden layer you have 2 nodes and the second hidden layer only 1 node.

Then you have  $p$  equal to 10. So, you have 22 plus 325 parameters immediately 1.5 times a jump in the number of parameters you can see boom, your number of parameters just more than doubled it just it previously it was just only 11 and now it is 25 just you add one more layer with one node extra.

And that is it you are done. So, in neural network your number of parameters can your parameters space can explode. So, how many parameters do we have in a double node neighbour. So, question is then come two first question is shall we add more nodes? Second

question shall we add more layers? And which will be helpful and which will be more dangerous.

(Refer Slide Time: 16:06)

Fat Single Layer Neural Network

shallow network.

How many parameters do you have in this model?

cmj

So, the first thing we can try out there we will not add too many layers we will add only single layer, but only we will add as many nodes as possible in the single network this is called typically shallow network because you know you do not have much breadth ok, shallow, but fat, fat single layer shallow network.

(Refer Slide Time: 16:34)

$5 \times 3 + 5 = 20$

shallow network.

How many parameters do you have in this model?

cmu

NPTel

So, what happens? So, here is an example you have sort of three features. So,  $x_1$   $x_2$   $x_3$  and you have single layer network single layer single hidden layer network. But how many  $z_1$ ,  $z_2$ ,  $z_3$ ,  $z_4$ ,  $z_5$  let us consider there is no intercept thing. So, even for each node you have three parameters. So, over 5 you have 5, 5 cross 3 15 plus here you have 1 2 3 4 5 so, plus 5. So, 20 parameters you are modelling.

(Refer Slide Time: 17:27)




Thin but Deep Neural Network

$3 + 3 + 2 + 2 + 2 = 12$

deep but thin network

How many parameters do you have in this model?

cmj




Now, with the same three features you just decide like ok I will go for 2 nodes, but 2 layers with each layer will have only 2 nodes then how many? So, you have  $x_1 \times x_2 \times x_3$  you have  $z_1$  and  $z_2$ . So,  $z_1$  has 3 parameters  $z_2$  has 3 parameters and then here you have  $u_1$   $u_2$ ;  $u_1$  has two  $u_2$  has 2 and then finally,  $p$  will have 2.



So, 3 plus 3 6, 6 plus 2 4 is 10 plus 2 12. So, deep network with more layer with each layer with a less number of node like only two nodes is giving you less number of parameter than a single layer with too many nodes.

(Refer Slide Time: 18:37)

Summary



- ▶ The purpose of neural network to discover the engineered feature automatically.
- ▶ We can use Lasso penalty (or L1 penalty) to drop the features or node that are not useful. *sparse network model*
- ▶ It is better to use deep neural network with couple of nodes in each layer, so that we avoid the explosion of parameter space.



And so, that is the reason that deep learning neural network is more popular than your fat shallow network because if you have a fat shallow network with too many nodes your parameter may space may explore, but you have a deep network with quite a few layer. But each layer will have maybe two or three nodes then what happens that it the number of parameters stay under control.

And that makes deep learning neural network very useful it can capture the non monotonic non-linear behavior ok it can capture those behavior, but at the same time your parameter space is somewhat controlled that is a very useful feature of neural network so, the purpose of the neural network to discover the engineered feature automatically.

So, that is the main purpose, but as I say even then my network can grow very fast and my parameter might can explode. So, if my parameter space explore then what we can do? We

can put a very tough L1 penalty to drop the features or loss of penalty to drop the features push them towards 0 and make it a sparse network. So, make it a sparse network model ok.

And then it is better to use deep network with couple of nodes in each layer couple of nodes with each layer. So, that we avoid the explosion of the parameter space. So, with this I will stop here. So, deep learning neural network is not really part of this course this is a really advanced topic, but I just give you a sort of a glimpse of how deep learning neural network are being developed and logistic regression or single node artificial neural network or single perceptron network models are the sort of a key building block; key building block for deep neural network models.

So, I am not going to do deep neural network models in this course you will get a lot of much better course you can you should go and take a full course on deep learning neural network. So, I, but I just this video I just try to show you what is the relationship between simple logistic regression and deep learning neural network for binary classification model.

Thank you very much, see you in the next video.