Indian Institute of Technology Kanpur

National Programme on Technology Enhanced Learning (NPTEL)

Course Title
Introduction to Experiments in Flight

Lecture-20
Aerodynamic parameter Estimation using Delta Method

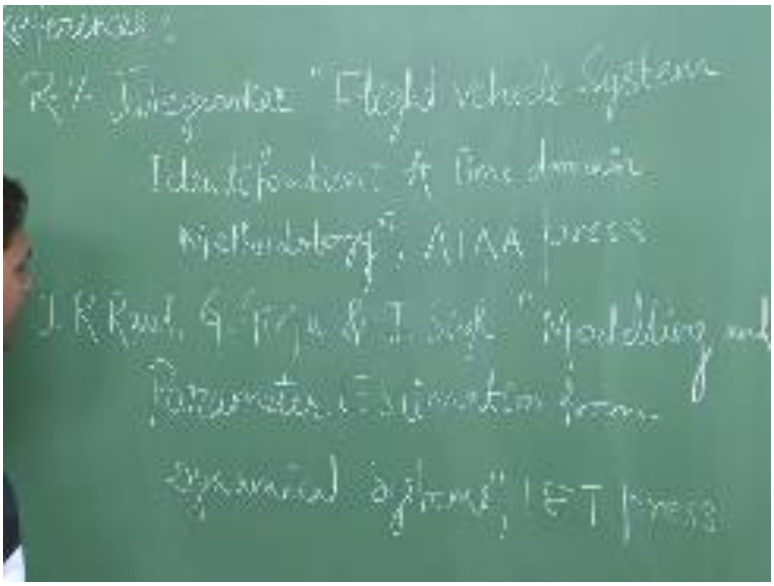
By
Prof. A. K. Ghosh
Dept. of Aerospace Engineering
IIT Kanpur

So hello and welcome to today tutorial, today we will be learning about Aerodynamic Parameter Estimation using delta method which I said I will be discussing right and what we learned so far in this section of lecture is aerodynamic parameter estimation using least square method. So I have shown you with the real flight data how to estimate the parameter switching least square methods rate and before starting today's lecture.

I would like to give you reference of two of my favorite books so you can learn and read more about the content so those reference books are just following so first learning by R.V. Jettagawker right and the book is called "Flight Vehicle System Identification: A Time Domain Methodology" by AIAA press. So this is a very good book you will enjoy reading.

I hope and second one is by J.R.Raol, G.Girija and J. Singh the title, of the book is "Modeling and Parameter Estimation from Dynamical System" by IET press okay.

(Refer Slide Time: 03:20)



So if you read these two books and any other sources you will have a lot more things to learn and these differences also will be useful for today's class like whatever contains I will be covering tonight. So in this part of the tutorial we will be learning about Delta method so what is Delta method and how do you use the delta method to estimate aerodynamic parameters?

I will be discussing that and since I have discussed this is based on the artificial neural network, so I will be giving you the little bit detail about the what is artificial neural network and how it works and then we will see the examples maybe same examples what we have learned during this fiscal estimation and we will learn the process. So I will start with the delta method, so what is delta method? It yes so delta method which proposed by your course instructor and my supervisor professor AK Ghosh and a great researcher in scientist Dr. Rai Singhania, so it is developed by Professor Ghosh and Rai Singhania a way back maybe a almost two decades back in 1998 yeah.

So the philosophy of delta method works like this so it derives or it estimate your aerodynamic derivatives, it estimates aerodynamic derivatives or parameters using feed-forward neural network right. When I you see this term came now a feed-forward neural network, so before

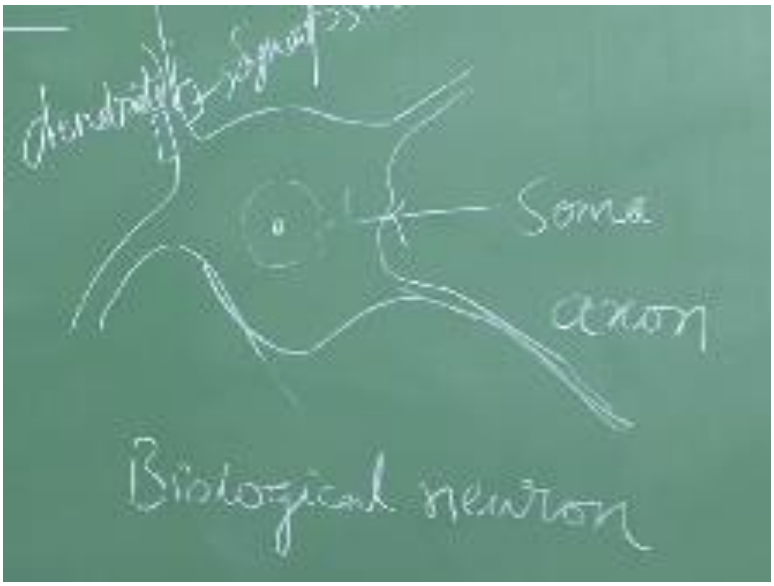
proceeding to the method I would like to discuss about this method first, we will get understanding of this method and then you will be appreciate more to this delta method right.

So let us start with this feed forward neutral network. So I will write FFNN. so Feed Forward Neural Network is a class of neural network, where you have three different layers; input layer, hidden layer, and output layer right and architecture is feed-forward it flow the flow of information just single way in one direction so it is called Feed-Forward Network all right. So basically all the neural network concept works on the our biological neurons right.

So as we know our brain has around 100 billion cells and each cells are called neurons and artificial neural network mimics our biological neurons right. So if you see a biological neurons so it will be something like this yeah, so here you receive the information through dendrite right and it passes the information to other neurons like this singles biologically neuron through axon, this is the output part where you pass this information to other neurons right and then right you will have the special kind of satellite contact they are called synapses right.

So it passes the information to dendrite with this context synapses and where you accumulate all the things so this is part of this full this is nucleus and this full part is corner soma ok. So this is a roughly the structure of your biological neuron.

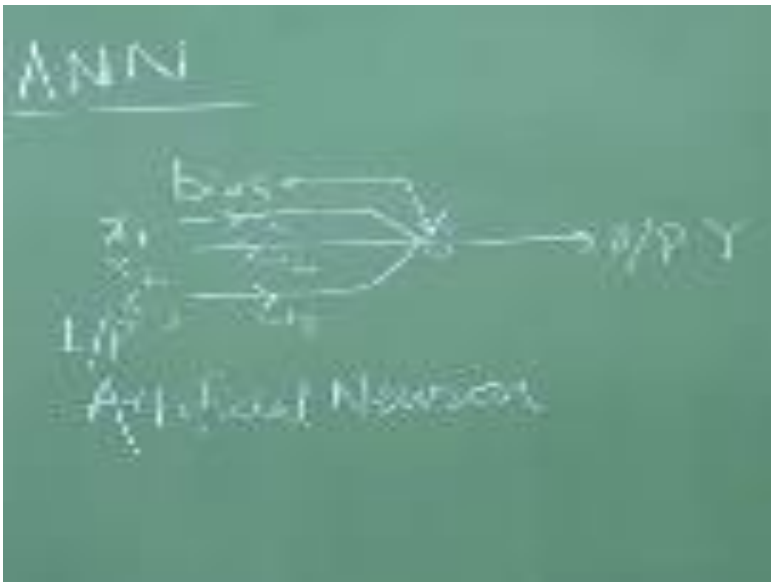
(Refer Slide Time: 08:04)



Now how your artificial internal network works so as I said it tries to be make this process, so I will just show you one of the artificial network neuron, yeah you have bias this is output side a dressage input side, so maybe here you have input like x1, x2, x3 and y is your output so each information will be having some weight age. So I will write weights as w1, w2, w3 so w1 weight corresponds to first input w1 corresponds to first input w2 for second and w3 for third.

It can have the number of n number of neurons and it is the bias so you have two major parameters in this artificial neuron, so this is a single neuron so I will write artificial neuron you know ok, now try to correlate these artificial neuron with this biological neuron.

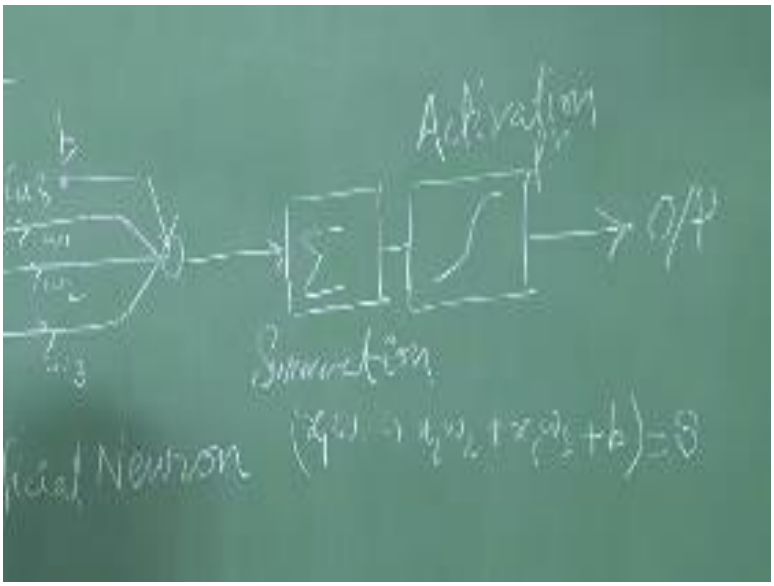
(Refer Slide Time: 09:39)



So here you receive the information to dendrite and here you receive the information in input layer right so this is called the input layer. So these are the inputs attached and all the information has to pass through these certain weighted which is equivalent to synopsis and then you get the output here right which is equivalent to your axons or and then further processing as I discussed about soma.

So how soma is related in this so again this output you need to sum it over with the weight age and have to process through some non linear function right which is called Activation Function and then you receive the output of the neuron right. So this is salvation so what happens here like here you will have x1 plus sorry x1 times w1 plus x2 times w2 plus x3 times w3 if you have three input plus bias, so let us say bias be so now this is a summation actually so you can represent this by some number maybe you can call it as summation.

(Refer Slide Time: 11:10)



Now it should pass through your activation function on linear function so which you can call it f(s) easier at output after passing through activation function. So now you are able to understand what is neuron in this artificial neural network. With the comparison of biological neuron right. So now I will be talking about structure of artificial neural networks which I have mentioned feed-forward neural network and let me tell you also this like this architecture is not the only architecture in artificial neural network.

You have other architectures also they are called so this is as you know Feed Forward Neural Network so let me show in terms of structure so you will be understanding better, so it had a three layers basically so input and maybe I will do one also here so what happens like this layer is called your input layer, and this is your output layer here. So in the problem of identification we basically deal with input and output.

So the input layer is a output layer and in between which in the heart of this network is called hidden layer right, now this whole set of neurons is called the input layer, whole thing hidden layer, this output you have single neuron here. So suppose now I have kept three neurons in this it means we have three different inputs so three inputs are there in this input layer number of

zeros in hidden layer depends on the complexity of the problem right. So if it is more complex you can have more number of neurons in hidden layer and selection of hidden layer also depends on the again complexity of the problem so literature suggests one-hidden layer is good enough to capture the complexity of problem like aero smith application.

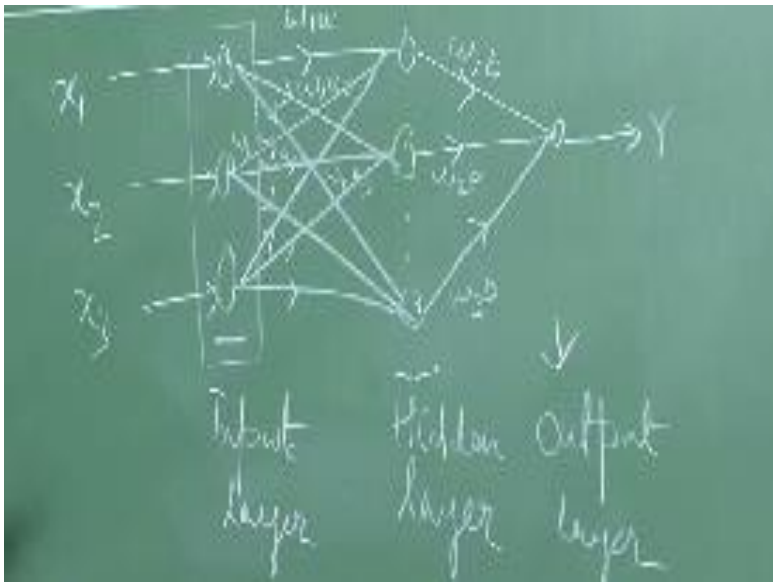
(Refer Slide Time: 13:51)

So that we have so you only one hidden layer with any number of hidden neurons right and then if you have one output then you will have one neuron in output layer, if you have two output you will have two output layer right. So this is the architecture of feed-forward so let me complete this so suppose now you have three different inputs again x1, x2, and x3 right so each neuron should can get connected to the next neuron right like this okay and it always flows in forward direction ok and then hidden layer each one will get connected to your output neuron like this.

As I said neuron are connected with some weight is so information weight is you can write the push I give the name A and B so here it will be w1a you can write or maybe this is a from the first input so you can write w1a right a1 and here you can write w1a2, w1a3 like that w2a1, w2a2 like that. You can assign any symbols does not matter but each neuron will have the weight

age right and further it will now have w1b weighted maybe w2b and w3b and output yes so let us say this is of course Y okay.

(Refer Slide Time: 15:58)

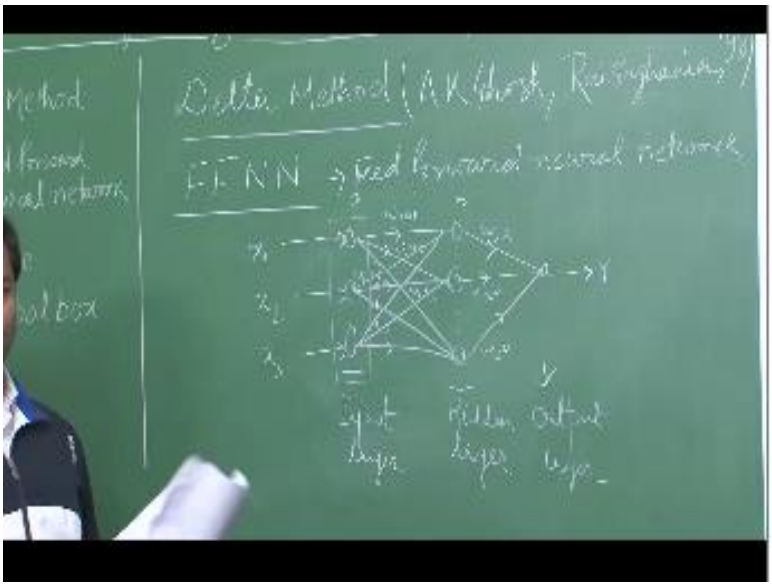


So this is the architecture of feed-forward neural network apart from that you have other architecture also. So I will just write list the name so this is called RNN Recurrent Neural Network, RNN and the Radial Basic Function Neural Network. As you saw in this architecture you have slope information in a single direction in a forward direction in RNL you will have the flow of information in both the direction which is a bi-directional flow of information and here are RBFNN.

(Refer Slide Time: 17:12)

In this kind of structure in hidden layers you will not have a ωβ to the hidden layer so hidden layer will have the radial basis function. So this is how they work, so I will be focusing more on this feed-forward TV network right so let us try to understand is how they work. So now this is the architecture.

(Refer Slide Time: 17:30)



It so let us take any simple example suppose you want to have fun like you have function like y equals to X1 X2 and X3 so this output is simply multiplying all the inputs right so x1 x2 and x2 is so as I said these this kind of approach they, they follow the black box modeling so we know the relationship between X and all important. Why suppose they are multiplier we have the free notion of that but now they are in the form of data and from the data we are trying to extract those informants are using the near electric modeling.

So nearer network does not bother about the multipliers hood or how they are correlated actually this neural network does not have to understand this structure without understanding the structure. It will assign the weights or all the weight edge to the neurons and of course there will be a bias and the bias so these are the biases okay. So they will adjust the weight and bias in such a way that you will be able to map your input and output.

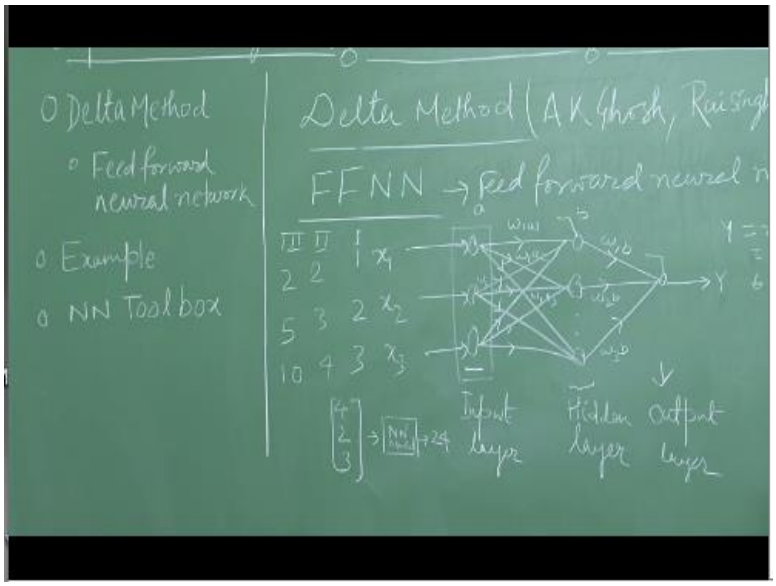
So let us say now you have input like one two three output will become six so this is first set of the data right second set of data you may be you have two three four second it will become 24 third you have 2 5 n^{th} so third output will have this 100 right, yeah. So now you see that you have

three different sets of inputs first set second set and third set and you have corresponding output 6, 24, 100 so these are the data we have available.

Now neural network suppose you want nero network to give you a model with the help of training to achieve that then it will create a trained it means we are trying to train the network with the help of input output data so that it will give you a network where you can if you give some other number right, so whole idea is like if you give 4, 2 and 3. It should give you the 24 right so now if you have designed your an training models and network model.

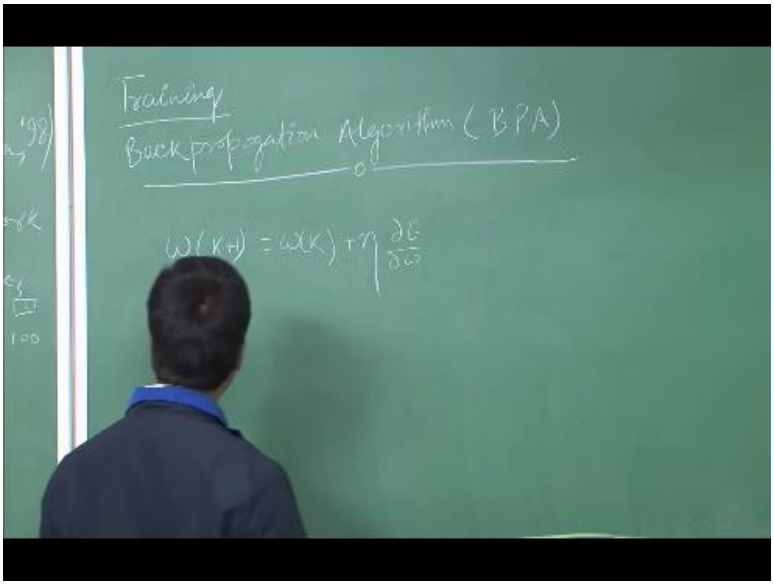
Then for this input it should be able to give you 24 it should predict like that so yeah so this is a whole idea right. Now as I said that you have to train the network and you have to assign the updated weight and the bias such that it can capture the dynamics or capture the relationship between input and output how will you do that. So the most frequently used method is back propagation algorithm so with the help of back propagation algorithm you will be able to train your feed forward neural network.

(Refer Slide Time: 20:59)



So I will just show you in a brief how back propagation algorithm works right as I said now training right training of your even network so we will do the training using back propagation algorithm back, back propagation algorithm Write BPA okay. So it works like this so just I will write the solution, $w(k+1)=w(k)+ \eta \partial E/\partial w$.

(Refer Slide Time: 21:58)

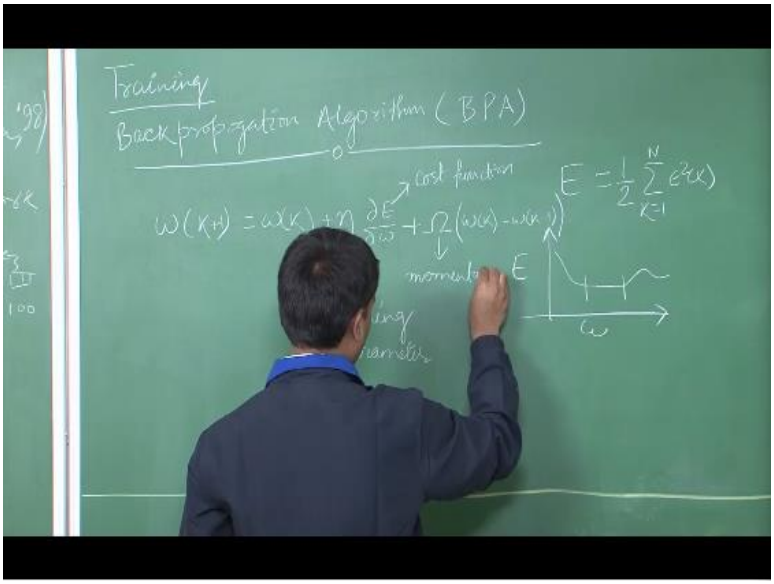


Right so now you see that you are trying to update your weight any weight actually let us take this one w1 a1 that we should pick up $w(k+1)=w(k)+ \eta \partial E/\partial w$ so it will become w1 a1 okay. So what is the inference of this operation like you are trying to update which means you are trying to get the weight of next step with the information of the previous weight right . We initially you will give some weight and then how will you know these weights are correct or not.

It will update your weight on the basis of this so this is basically based on your gradient descent all right so here you see like if you have initially estimated some weight it will be 0 w0 so w one will be w 0 plus ETA times this is a $\partial E/\partial w$. So let me write this term it called the learning parameter yeah and this is easier cause function which we discussed in this square error function or cost function. So what is E basically easier error square $E= ½ \Sigma_{k=1}^{N} E^2(x)$.

Now let us point out one thing here okay, so suppose now if your E versus w sometimes it can be a constant and anything like that so this region if you see you will not be able to update the weight even though you have error if you see error versus wait what de by ∂E/∂w will make this term zero right that time you will not be able to get the weight so you need to add one more term which is called the momentum times momentum parameter times Ω(w(k)-w(K-1))one step back okay, so this is called the momentum parameter right.

(Refer Slide Time: 24:34)

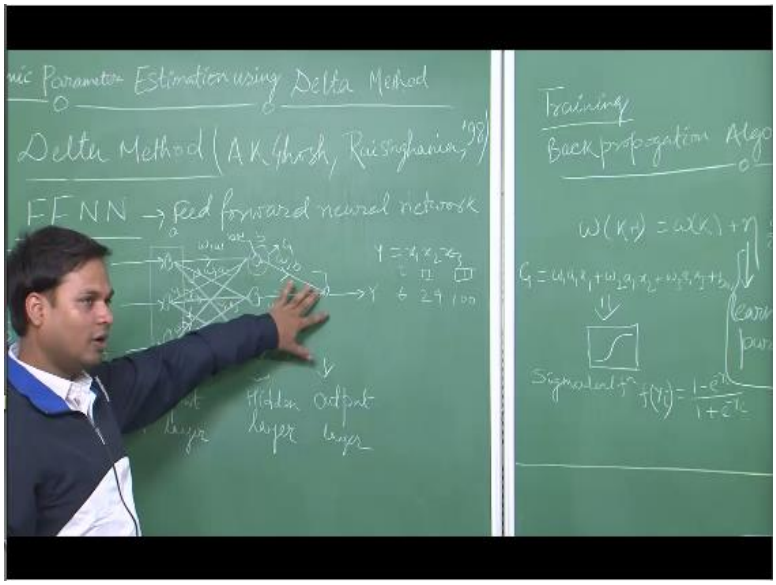


So learning rate learning parameter decides you the learning rate like how fast how slow you are making your network to learn and momentum parameter helps you to improve the performance right. Now each weight you can write like that as I discussed earlier so now here are your inputs like for this hidden layer you will have w1 select right for this in your neuron just I want to show you one neuron expression. So for this let us say the output is suppose x1 this is w2a1 x2 + w3 a1 x3 plus bias you can name with this bias you can take like be a1 so plus be a1 so I am sorry for messing it up here.

So this is your expression right so now this is the summation actually summation of all with all the weight is now it has to pass through some nonlinear function so the very popular popularly

used nonlinear function is called a sigmoidal function so sigmoid function basically they are all your activation function so here Yi $=1-e^{-yi}/1+e-^{yi}$ .One suppose now in this case you have summation at c1 so this is f(Yi) once it will pass with the sigmoidal function then it will become $1-e^{-c1}$ once you got and then with the help of this see one again c2 will come c3 will come from here.

(Refer Slide Time: 26:56)



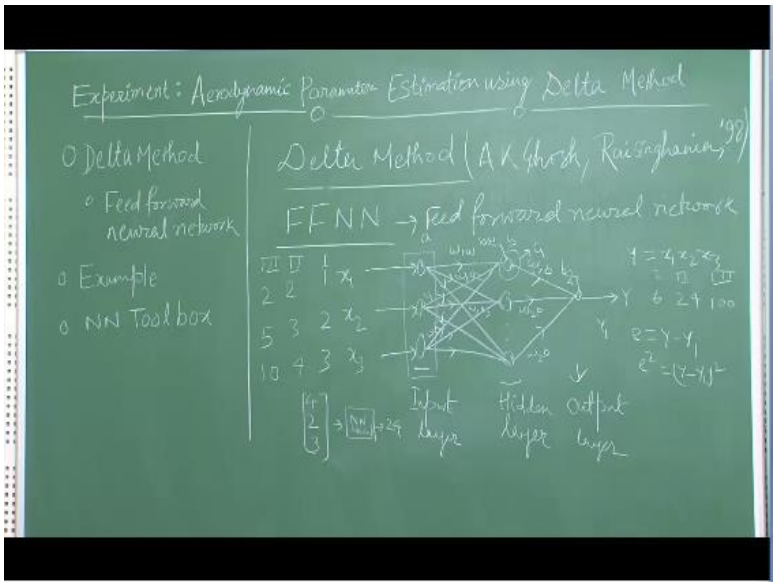
Again your weights are assigned with the output all the output which curve coming from your hidden layer Iran's it will go to output layer again so you can make it like be let us say I would be too okay so with that again you will have the summation of all the inputs with buyers and you will transfer to activation function right and then it has to go through like. This now so suppose here first output for the first hydrogen it came at y1 right so now error is how much error will be your e=Y-Y1 $e^2=(Y-Y1)^2$.

And then you to design your cost function right so this is your house now you see y1 is a function of this weight and this weight or this output either is a function of the previous weight so you have to update first time it will you will give some information like a initial guess it will

go forward you will see the error with the error it will update this weight first and then wait from this backside so the error is propagating back action.

So you are having the connection on the error by going back side so the principle of back propagation algorithm works like that so it we correct the error by propagating back backwards right so that is why it is called the back propagation algorithm. So that with the help of the understanding of back propagation algorithm and the structure of F F and N N will get a network modern right so that is how your N and works.

(Refer Slide Time: 28:34)

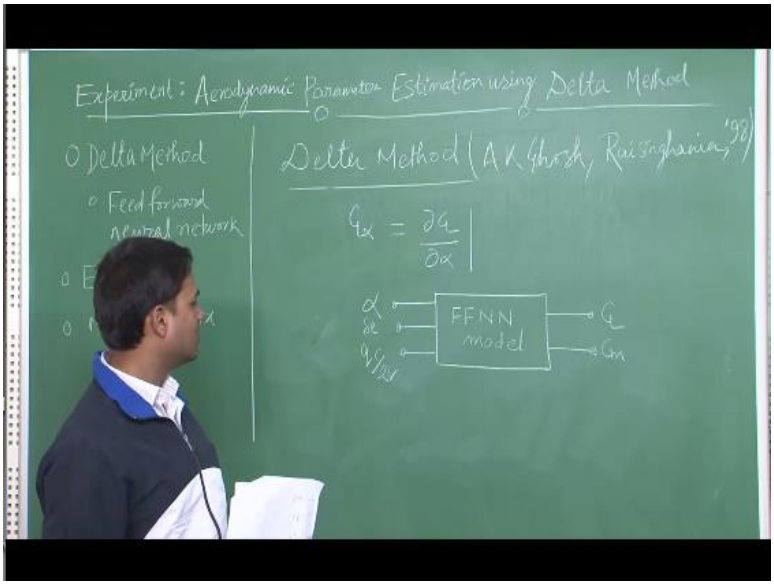


Next we will see go back to the again Delta method how Delta method works now I just ate delta method it estimates your error time dynamic parameters with the help of artificial neural network model and basically feed for one you can network model right so and it is a very prominent method you know and it is quite intuitive also if you see the aerodynamic derivative let us take the example of $C_{L\alpha}$ so what is the physical significance of C_L α it means with the change in α with a change in α how your C_L is going to change. You write $\partial C_L / \partial \alpha$ by keeping other thing constant all if you have other input like δ E and Q .

So by keeping those input constant you will observe the change in C_L because of change in α right angle of attack so this is the basic understanding of the aerodynamic derivative or this parameter $C_L\alpha$ so now how will were late this understanding in δ method now suppose this is the model aerodynamic model actually which is trained by FFN feed-forward neural network model but you know very well.

Now the structure of the model and suppose now I will give you the plane this with the same example what we have used earlier during lists college tuition explanation then we had three input okay, and those inputs where your α δe and QC/2V right and outputs where your force coefficient lift force coefficient C_L and C_M right so this was basically a structure now here we do not have structured aerodynamic model unlike what we used in less square right.

(Refer Slide Time: 30:42)

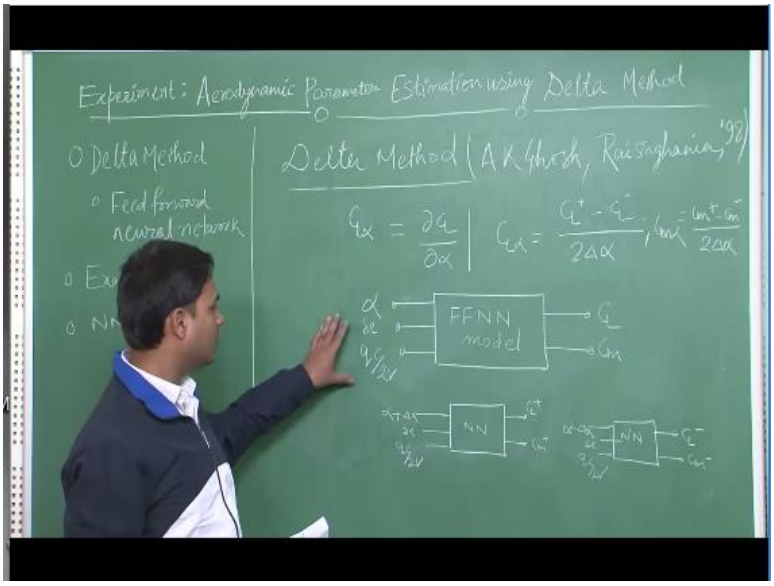


Here but we have the trained network model with the help of a set of input data and set of output data right now Delta method oh! how it works I will just explain you now you perturb this input first let us say first input α you are perturbed by δα right. Let me write here so this is your NN model yes so now you have uttered α with δα by keeping δe and the other input qcv constant then you will see the changes in this.

So I can also write with some different notations CL+ is right and then CM also will see some changes may be and add Cn+ yes I know you perturb in the other direction may be α-δ for this time again your NN and model so here you will have α-δα by keeping this to input constant qc /2V and this representation you can make like C_L^- and Cm^- right so this is what is here. Now if you observe the output C_L^+ and here C_L- the changes in force coefficient live four coefficients by we have made a change in positive direction for δα negative direction.

Also with δα so total changes 2δα . So this is nothing but again a $C_L\alpha$ is 1 now you see you got this aerodynamic derivative the C_L α which the help of perturb method you have perturb perturbed α in both the direction to avoid the biases right so positive perturbation and negative perturbation and then you observe the changes in C_L in both the direction and if you divided by total perturbation then you will get $C_L\alpha$ similarly you will get Cmα like Cm+ - Cm- /2∇α like this you can get all other derivatives also like if you want to get C_L δe.

(Refer Slide Time: 33:29)



Then you perturb in with δy+ δe+ δe and δe- δe by keeping α and q CB constant will get C_L δe and Cm δe like this so using delta method you can estimate all those aerodynamic derivative

with feed-forward neural train model with this approach right so this is the fundamental understanding of delta method. So I will show you with examples maybe with the Matlab simulation how, how did you get the derivative and then you can have the comparison with the least square methods.

Which I have discussed earlier and also for your practice if you are interested then I can show you with a Matlab toolbox which is called NN tool, tool so there you will learn you can learn how to model any network or how can you train the network with a very simplified toolboxes it is very easy to use and I can demonstrate in this tutorial so that you can practice for different kind of problem it is not limited only for aerodynamic parameter estimation problem. We can employ in other application also as per your interest and requirement here thank you so much.

Acknowledgement

Ministry of Human Resource & Development

Prof. Satyaki Roy

Co-ordinator, NPTEL IIT Kanpur

NPTEL Team

Sanjay Pal

Ashish Singh

Badal Pradhan

Tapobrata Das

Ram Chandra

Dilip Tripathi

Manoj Shrivastava

Padam Shukla

Sanjay Mishra

Shubham Rawat

Shikha Gupta

K. K. Mishra

Aradhana Singh

Sweta

Ashutosh Gairola

Dilip Katiyar

Sharwan

Hari Ram

Bhadra Rao

Puneet Kumar Bajpai

Lalty Dutta

Ajay Kanaujia

Shivendra Kumar Tiwari

an IIT Kanpur Production