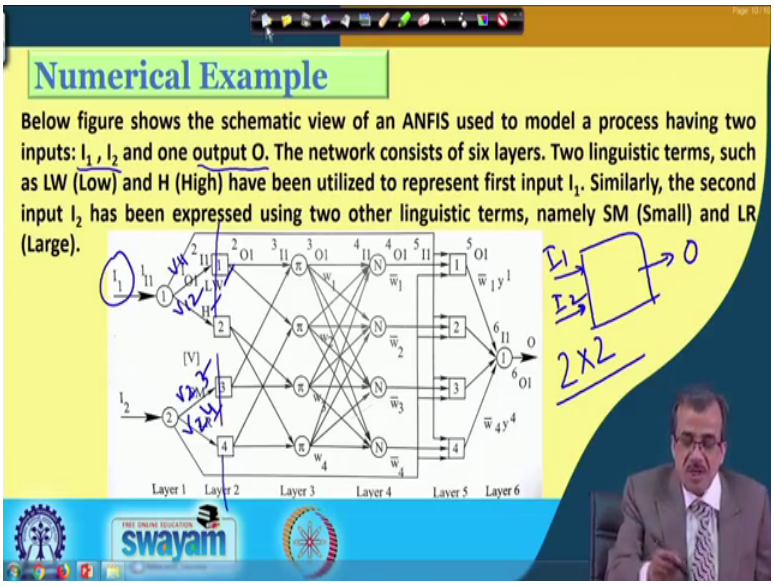
Fuzzy Logic and Neural Networks
Prof. Dilip Kumar Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 36
Neuro - Fuzzy System (Contd.)

(Refer Slide Time: 00:15)

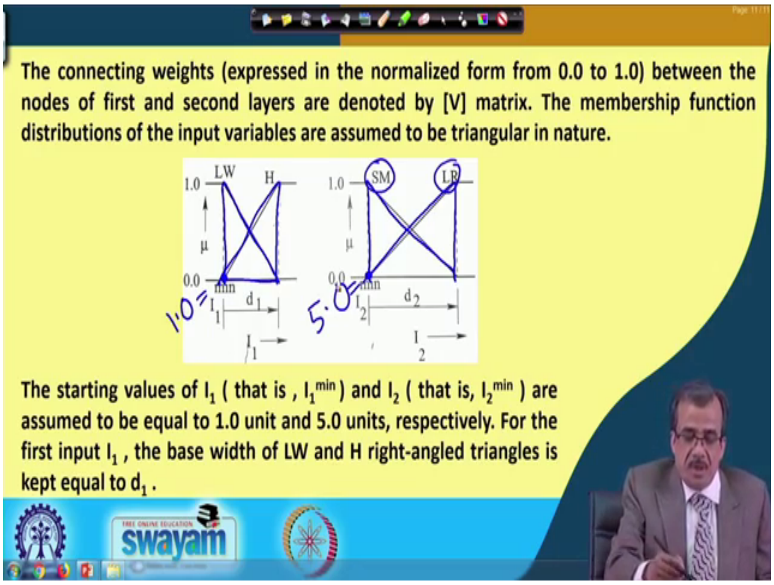


Now, we are going to solve a numerical example just to explain the working principle of this particular the ANFIS. Now here I am just going to take one example numerical example for a system, it is a very simple system having 2 inputs and 1 output. So, this is something like this we have got I 1 I 2 and I have got only 1 output. Now the statement of this particular the numerical example is as follows. So, we will have to model, we will have to design 1 ANFIS, and the purpose is to model a process having 2 inputs I 1 and I 2 and 1 output that is O.

Now, in this particular ANFIS there are six layers, and we use 2 linguistic term to represent each of the input parameters. For simplicity for example, say to represent I 1 so, we have used like your low and your high. similarly to represent this I 2 the second input parameters. So, we have used two linguistic term 1 is called the small another is your the large. So, this is a very simple the ANFIS and the 2 inputs each of the 2 inputs is represented using 2 linguistic term.

So, we have got like 2 multiplied by 2 so, only 4 possible rules. And we can see that so, these shows actually the second layer, now in the second layer. In fact, we have got 2 neurons here and we have got two more neurons here and the connecting weights between your the first layer and the this particular neuron of the second layer. So, this is nothing, but is your v 11 this is your v 12 and here the connecting weight is V 23 and this is your V 24. And is a very simple network now, let us see the other part of the statement of this particular problem.

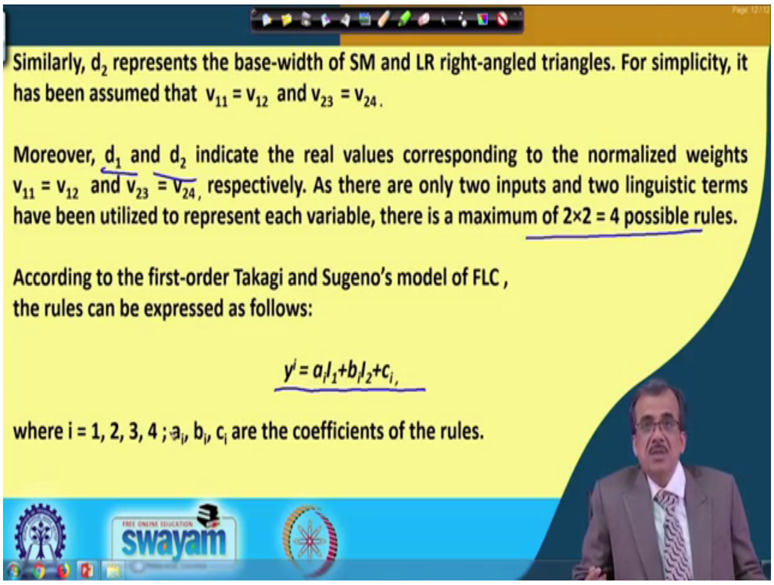
(Refer Slide Time: 02:33)



The connecting weights are lying in the range of 0 to 1 and if you see the membership function distribution which we have considered for the two inputs like I 1 and I 2 are as follows. Now for simplicity we have considered like the low I 1 is nothing, but this. So, this type of membership function distribution right angle triangle and for the high another right angle triangle and there is overlapping also.

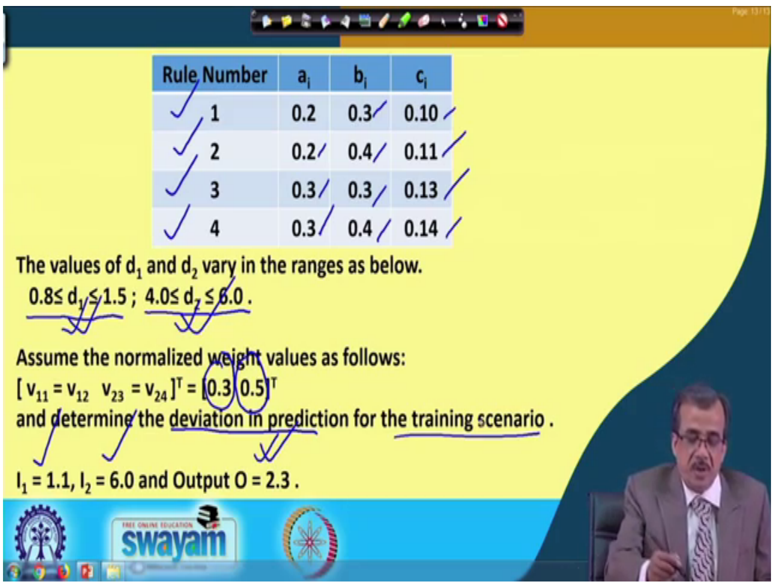
Now, similarly for this I 2 there are two linguistic term, one is the small another is a large and for simplicity we have considered. So, this type of your right angle triangle, its a very simple representation of the membership function and here the starting value of this particular I 1 and I 2. Now you can see that the starting value is nothing, but I 1 minimum and here it is I 2 minimum. So, this I 1 minimum you have considered that is equals to 1.0 and your I 2 minimum we have considered that this is equals to your the 5.0

(Refer Slide Time: 03:55)



Now, the rest of the statement of this particular problem, like we use the connecting weights like V 11 is equals to V 12 and that is denoted by is your the d 1 similarly your. So, this particular V 23 is nothing, but is your v 24 and that is going to represent actually your d 2. Now this values for the d 1 and d 2 should lie in a particular the range and there should be some well defined range also and as I told there are 4 linguist 4 possible rules to multiplied by 2 and the output of a particular rule that is the output of the ith rule that is y i is nothing, but a i I 1 plus b i I 2 plus c i and here. So, this particular i is nothing, but 1 2 3 4 and a i b i c i are the coefficients of this particular the rules.

(Refer Slide Time: 05:03)

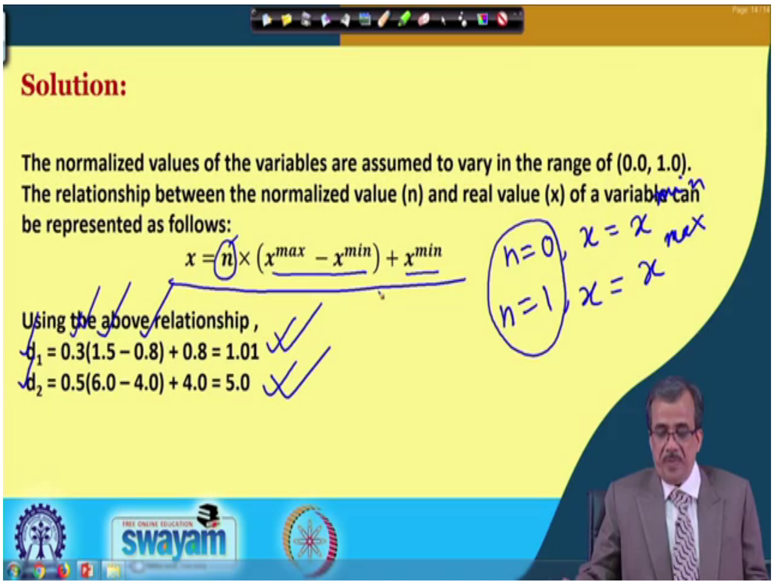


Now here to solve this numerical example so, we are going to consider some numerical values, the predetermined numerical values for this particular the coefficient the values of the coefficients are as follows. Now for the first rule the coefficient the values of the coefficients are nothing, but a i that is a 1 equals to 0.2, b 1 equals to 0.3 and c 1 is 0.10 Similarly, for the second rule a 2 is 0.2 and this b 2 is 0.4 and c 2 is 0.11. Similarly for the third rule your a 3 is 0.3, b 3 is 0.3 and c 3 is 0.13 and for the fourth rule your a 4 is 0.3, then b 4 is 0.4 and c 4 is 0.14.

Now, this d 1 and d 2 so they are varying in this particular range. So, d 1 is varying in the range of 0.8 to 1.5. Now similarly d 2 is going to vary in the range of 4.0 to 6.0. And to carry out this numerical example actually what we do is, we assume some the values for the connecting weights we assume that V 11 equals to V 12 which is going to represent the membership function distribution for the first input that is nothing, but 0.3. Similarly V 23 equals to V 24 this is nothing, but 0.5 and these are in the normalized scale.

So, will be discussing that using these normalized values so, will have to find out the real values lying in the range for this particular d 1 and d 2. And we are going to determine what is the deviation in prediction for a particular training scenario where I 1 the first input is nothing, but 1.0 and I 2 that is the second input is 6.0 and the output O is nothing, but is your 2.3. Now let us see how to determine the output corresponding to so, these set of inputs. And how to determine the deviation in prediction corresponding to this particular your the training scenario. So, those things actually I am just going to calculate.

(Refer Slide Time: 07:45)



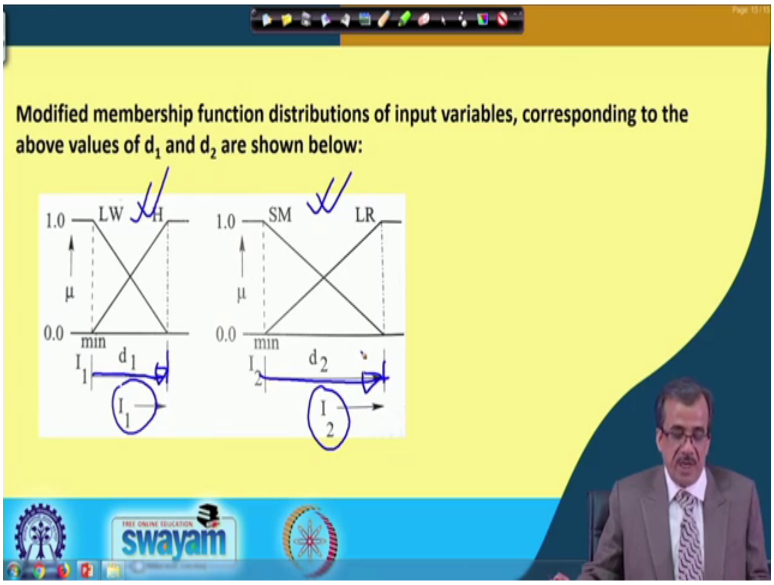
Now, let us concentrate on the solution of this numerical example, now as I told the first thing will have to do is corresponding to the normalized values of the connecting weights. So, will have to find out the real values; that means, will have to find out the real values for this particular d 1 and d 2. Now to find out the real values from the normalized value that is denoted by n we are going to use this particular the formula. Is very simple like x is equals to n multiplied by x max minus x minimum plus x minimum.

Now, if I put n equals to 0. So, if I put n is equals to 0. So, x will become equals to how much? That is nothing, but x minimum. On the other hand if I put n is equals to 1 what will happen to the value of x? So, if I put n equals to 1. So, this is x max minus x minimum plus x minimum. So, this will become your x maximum; that means, so here. So, this particular n varies in the range of 0 to 1, and accordingly I can find out what should be the real values for d 1 and d 2 lying within their respective ranges.

Now, here so d 1 is nothing, but your 0.3 multiplied by 1.5 minus 0.8 plus 0.8 because n is equals to 0.3, the maximum value of d is 1.5, the minimum value is 0.8 and if we calculate will be getting that d 1 is equals to 1.0 1. Now similarly we can find out d 2 is equals to 0.5 that is the value of n, multiplied by 6.0 minus 4.0 plus 4.0.

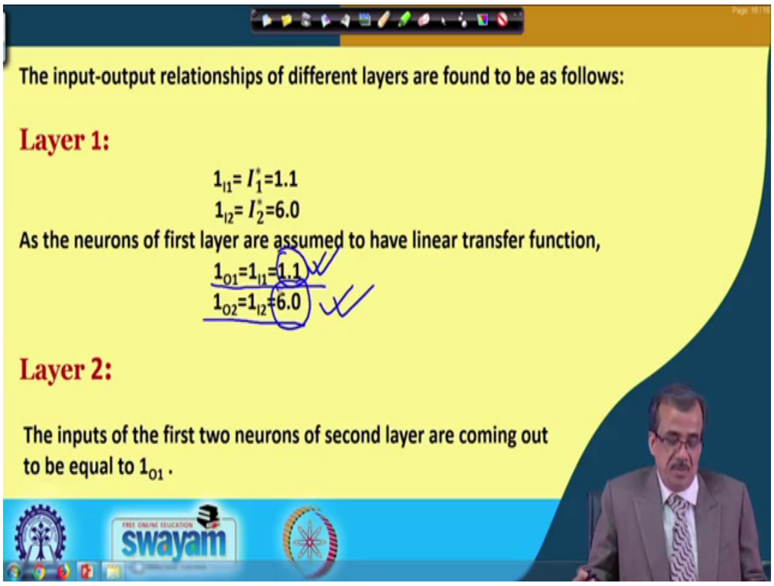
Now, here the d 2 minimum is 4.0, d 2 maximum is 6.0 and value of small n is equals to 0.5 and if you substitute we will be getting the value for this d 2 as 5.0. Now once you have got the real values for this particular d 1 and d 2.

(Refer Slide Time: 10:19)



So, now we are in a position to draw the modified membership function distribution for this particular I 1 and I 2. So, for this particular I 1 so, this is the modified membership function distribution and for this I 2 so, this is the modified membership function distribution; where d 1 is nothing, but this and d 2 is nothing but is your this. So, we can find out the modified membership function distribution for the two inputs.

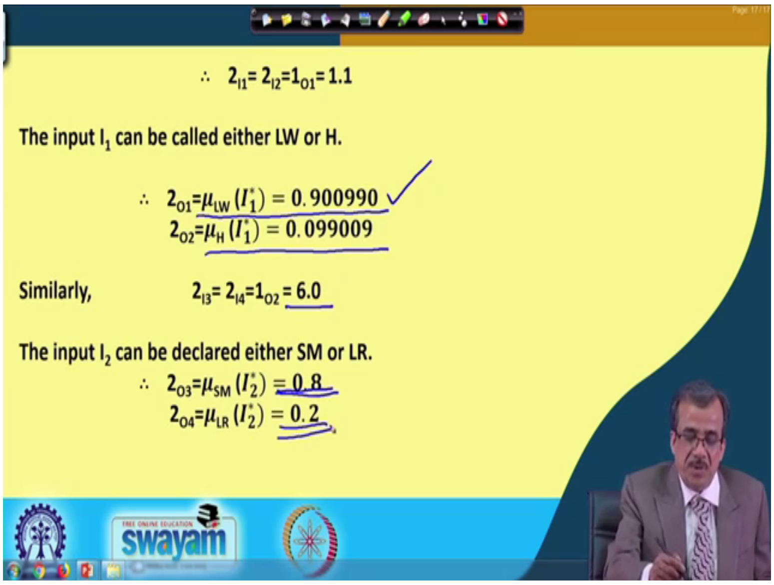
(Refer Slide Time: 10:53)



And now the layer wise I am just going to carry out the calculations. Now on the first layer as I mentioned that we use linear transfer function. So, this output is nothing, but the input. So, the output of the first neuron lying on the first layer is nothing, but the input of the first neuron lying on the first layer and that is nothing, but 1.1.

Now, similarly the output of the second neuron lying on the first layer is same as the input of the second neuron lying on the first layer and that is nothing, but 6.0. Now layer 2 is nothing, but the fuzzification layer. So, what we do is. So, these values for your input that is 1.1 and 10.0 we are going to pass to the corresponding the membership function distribution just to calculate what should be the membership function values.

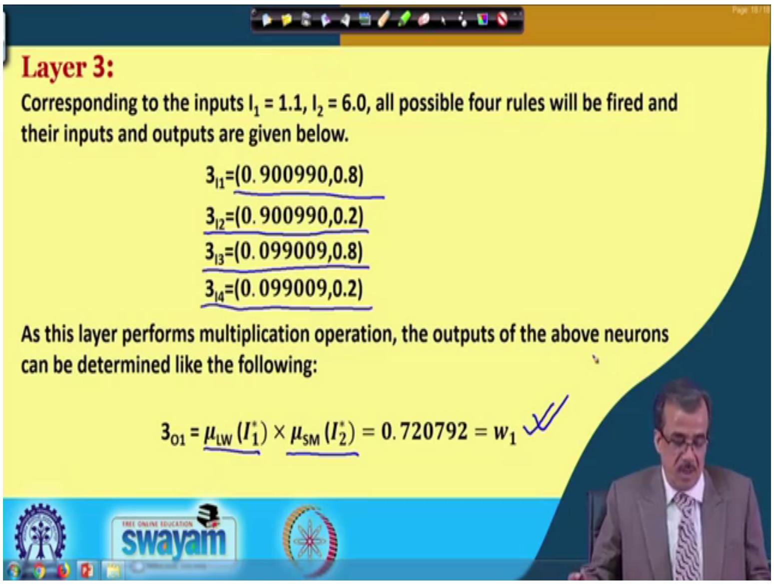
(Refer Slide Time: 12:01)



Now if you calculate the membership function values so, you will be getting that your corresponding to the low I 1. So, the mu low corresponding to I 1 star. So, this will become equal to your 0.900990.

Then corresponding to your I 1 star the mu H will become 0.09009, similarly corresponding to the second input that is equals to 6.0. So, we can find out the values like mu SM corresponding to I 2 star is nothing, but 0.8, then comes your mu large corresponding to I 2 star is nothing, but is your 0.2 and how to determine those things we have discussed several times in details.

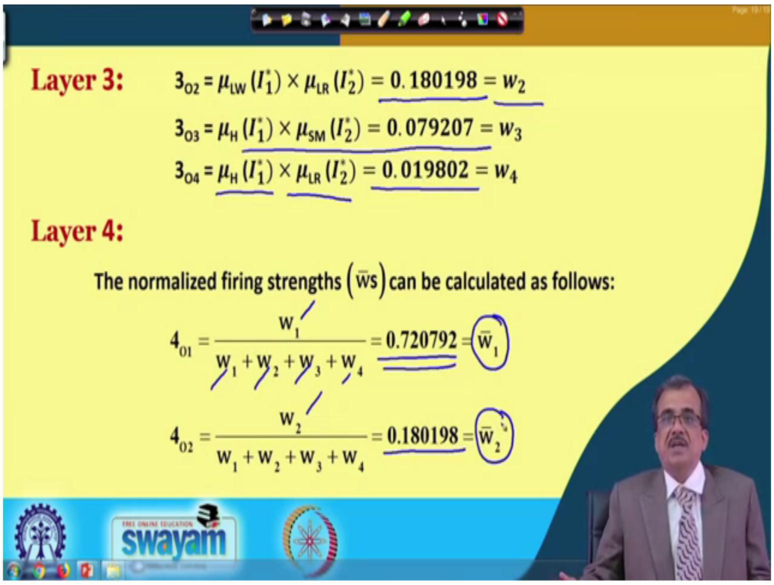
(Refer Slide Time: 12:57)



Now, then comes your layer 3. Now this layer 3 is actually is going to represent all the 4 possible rules for this particular the reasoning tool and here you can see, that the input of the first neuron lying on the third layer is nothing, but the 2 values of the membership. Similarly the input of the second neuron lying on the third layer are nothing, but the 2 values of the membership, then your the input of third neuron lying on the third layer is nothing, but this then the input of the fourth neuron lying on third layer is nothing, but this. And what you will to do is now to determine the output of third layer.

So, we will have to multiply your the mu values. For example, say output of the first neuron lying on the third layer is nothing, but mu low corresponding to I 1 star multiplied by your mu SM corresponding to this I 2 star. And if you substitute the numerical values and multiply. So, you will be getting w 1 as actually your the firing strength for the first fired rule.

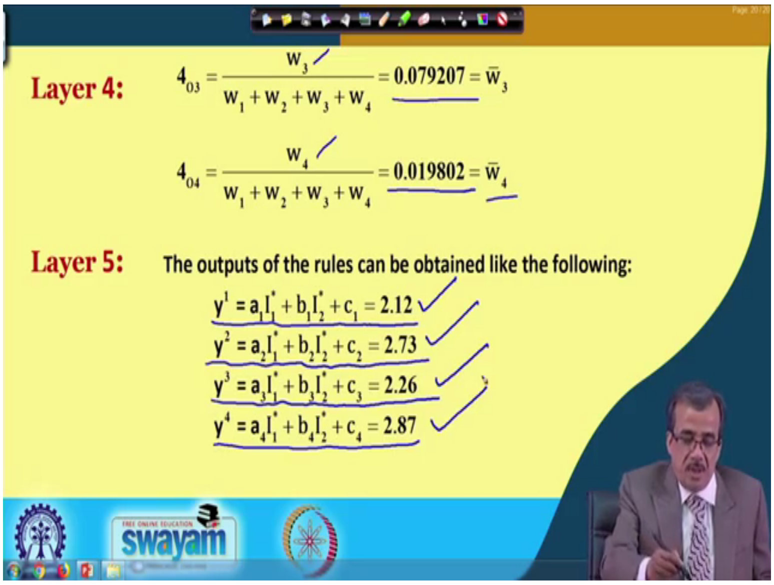
(Refer Slide Time: 14:23)



Now, by following the same procedure so, we can find out the firing strength of the other fired rules. For example, for the second rule the firing strength we can find out and this will become 0.180198 s w 2 and similarly we can find out the firing strength of the third rule that is your w 3 and that is nothing, but is your 0.079207. For the fourth rule so, we can also find out the firing strength and that is nothing, but mu H corresponding to I 1 star multiplied by mu LR corresponding to I 2 star.

And if you substitute the numerical values and carry out the multiplication so, you will be getting w 4 is nothing, but 0.019802. And once you have determined all the firing strength values now we can start with layer 4. And in layer 4 actually we try to find out the normalized values of these particular the firing strength. Now here so, w 1 bar that is the normalized value for the first or the fired rule that is nothing, but w 1 divided by w 1 plus w 2 plus w 3 plus w 4 and if we substitute all the numerical values. So, you will be getting w 1 bar as 0.720792. Similarly this w 2 bar is nothing, but is your w 2 divided by the sum of all w values and if you substitute the numerical values you will be getting w 2 bar is nothing, but 0 point 0 0.180198.

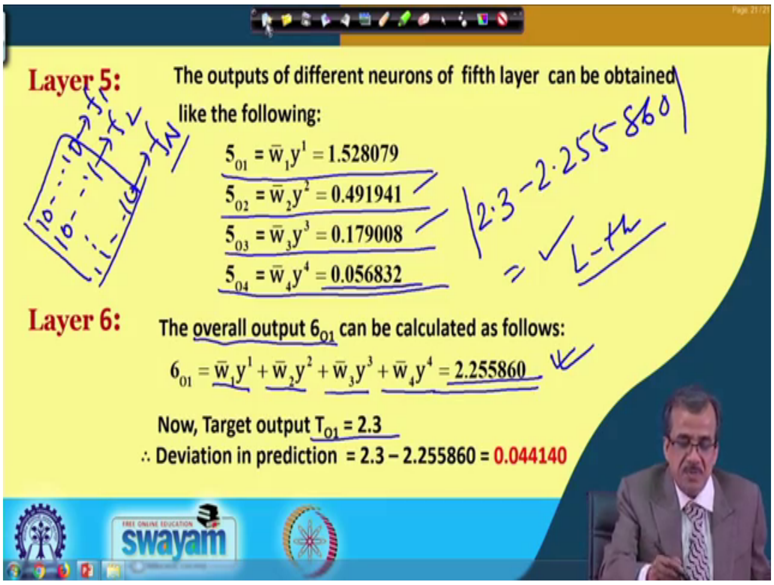
(Refer Slide Time: 16:25)



Now, by following the same procedure so, we can also find out like what should be your w 3 bar. And w 3 bar is nothing, but w 3 divided by the sum of all the w values and you will be getting 0.079207 as your w 3 bar. Similarly w 4 bar is nothing, but w 4 divided by the sum of all w values and after substituting the numerical values, we can calculate the w 4 bar is nothing, but 0.019802. Now we have got all the normalized firing strength values. Now, we go to layer 5.

Now in layer 5 actually what we will have to do is so, will have to find out the output of each of this particular the fired rule; for example, for the first fired rule. So, this w 1 is nothing, but a 1 I 1 star plus b 1 I 2 star plus c 1. So, you will be getting 2.12. Similarly this y 2 is nothing, but a 2 I 1 star plus b 2 I 2 star plus c 2 and this will become equal to 2.73. Then y 3 is nothing, but a 3 I 1 star plus b 3 I 2 star plus c 3 and if you substitute the numerical values, you will be getting 2.26. Then a y 4 is nothing, but a 4 I 1 star plus b 4 I 2 star plus c 4 So, if you substitute all the numerical values you will be getting 2.87 as your y 4.

(Refer Slide Time: 18:23)



Now, once you have completed the layer 5. So, layer 5 actually you will be getting we multiply. So, this w 1 bar by your y 1 and will be getting this as the output of the first neuron lying on the fifth layer. Similarly the output of the second neuron lying on the fifth layer is nothing, but w 2 bar multiplied by y 2 and you will be getting 0.491941. Similarly the output of the third neuron lying on the fifth layer is nothing, but w 3 bar multiplied by y 3 and that is nothing, but 0.179008 and similarly your the output of the fourth neuron lying on the fifth layer is nothing, but w 4 bar y 4 and if you substitute the numerical values you will be getting 0.056832.

And once you have got the outputs of the fifth layer, now in sixth layer actually what we do is, we sum them up just to find out what is the overall output that is your 6 o 1. So, 6 o 1 is nothing, but w 1 bar y 1 plus w 2 bar y 2 plus w 3 bar y 3 plus w 4 bar y 4 and if we substitute the numerical values, we will be getting 2.25586 and this is nothing, but the final calculated output for the set of inputs.

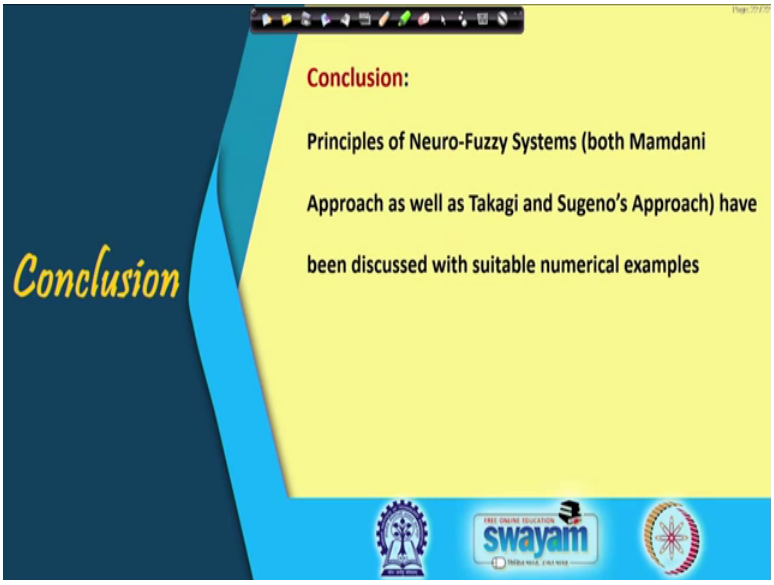
Now, this calculated output will be compared with this particular target output and that is denoted by T o 1 that is 2.3 and very easily we can find out the deviation in prediction and we can also find out the mod value of the deviation prediction that is a 2.3 minus 255860. Now here fortunately we are getting positive, but if we get negative so, we will have to consider the mod value and the way I discussed. So, this is the prediction corresponding to your the first training scenario, then we go for the second training scenario third training scenario up to all the training scenario say capital Lth training scenario and try to find out all the deviation values. You add the mod values of all the deviations find out the average that particular average will be the fitness of the GA as I discuss.

Now, if I use the GA string and if I use a binary coded GA say if I use a binary coded G for each of the GA string, I will be able to find out the fitness and then we use actually your the operators, the GA operators just to modify. And GA through a large number of iterations we try to evolve that particular ANFIS like which will ensure a very good prediction accuracy.

(Refer Slide Time: 21:57)

Now this is the way actually this ANFIS works and now, regarding the reference you can see that textbook for this particular course, that is soft computing fundamentals and applications if you want to get more details regarding this the neuro fuzzy systems.

(Refer Slide Time: 22:15)



Now, to conclude actually in this lecture we discuss 2 neuro fuzzy systems. The first one is based on the Mamdani approach and the second one is based on the Takagi and Sugenos approach. Now the basic the aim of these particular the neuro fuzzy system is to design and develop the fuzzy reasoning tool and we take the structure of this particular network so that we can represent the fuzzy reasoning tool and we can carry out the training or the optimization of this particular network. But truly speaking we are going to design and develop very accurate fuzzy reasoning tool, and we have discussed both Mamdani approach and Takagi and Sugenos approach of fuzzy reasoning tool and we have seen how to represent them using the structure of a network and how to optimize.

So, how to evolve the optimal fuzzy reasoning tool based on both Mamdani approach as well as Takagi and Sugenos approach by taking the help of the structure of a network neural network and with the help of one optimization tool. Now so, this is the way actually we can develop the neuro fuzzy system and these neuro fuzzy systems have got a large number of practical applications. So, these neuro fuzzy systems have been used widely to solve a variety of real world problems.

Thank you.