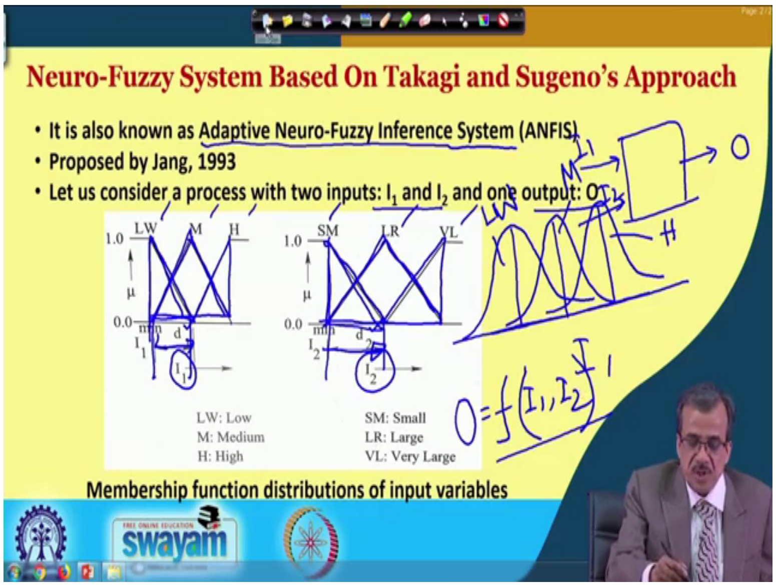
Fuzzy Logic and Neural Networks

Prof. Dilip Kumar Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 35
Neuro - Fuzzy System (Contd.)

(Refer Slide Time: 00:15)



Neuro-Fuzzy System Based On Takagi and Sugeno's Approach

- It is also known as Adaptive Neuro-Fuzzy Inference System (ANFIS)
- Proposed by Jang, 1993
- Let us consider a process with two inputs: I_1 and I_2 and one output: O

Membership function distributions of input variables

We have discussed the working principle of Takagi and Sugenos approach of fuzzy reasoning tool. And this particular approach is nothing, but a precise fuzzy reasoning tool; that means, we will be able to establish the input output relationship in a very accurate way. Now today we are going to discuss how to represent this Takagi and Sugenos approach a fuzzy reasoning tool using the structure of a neural network so that we can trained, we can optimize the performance of this particular the fuzzy reasoning tool.

Now, the title of your today's lecture is your neuro fuzzy system based on Takagi and Sugenos approach. Now let us see how to model this Takagi and Sugenos approach or fuzzy reasoning tool and how to develop the neuro fuzzy system. Now this particular neuro fuzzy system is very popularly known as Adaptive Neuro Fuzzy Inference System that is nothing but your ANFIS. So, this is nothing, but the ANFIS. So, an in short this is known as A N F I S Adaptive Neuro-Fuzzy Inference System. Now the first proposal came in the year 1993 by Jang and after that so, this particular ANFIS has been modified in a number of ways.

Now, to explain the working principle let me consider a system having 2 inputs like I 1 and I 2 and there is one output that is O. So, this is a very simple system having two inputs and one output. So, we have got I 1 and I 2 and we have got the output O. Now let us see how to model using the principle of your this ANFIS. Now the 2 inputs are represented using triangular membership function distribution, now here the first input that is I 1 that is represented using three linguistic terms like your low, medium and high.

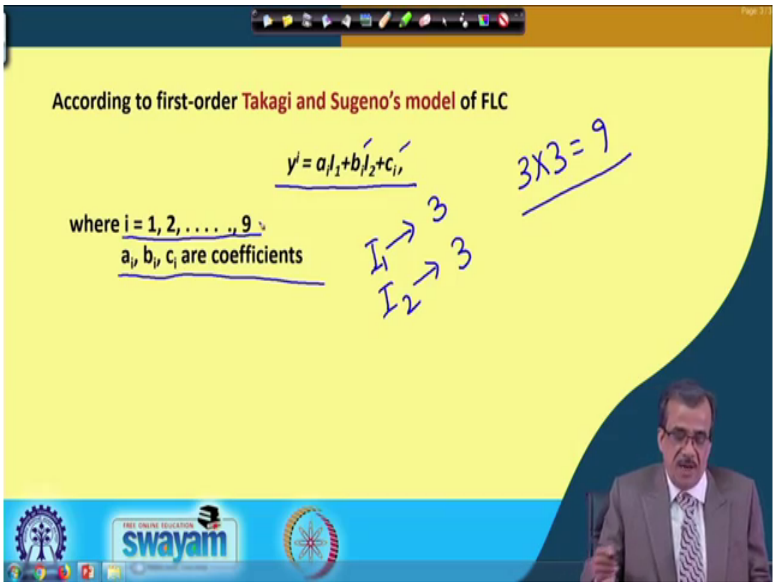
And for simplicity we have considered the triangular membership function distribution. So, this type of triangular membership function distribution we have consider. Now truly speaking actually if you want to model more accurately so might be we will have to go for some sort of non-linear distribution also. For example, I can also take some sort of Gaussian distribution for example, if I consider the low, medium and high and if I consider the Gaussian distribution; the Gaussian distribution will look like this and there will be some overlapping of the regions in this Gaussian distribution. So, if I consider the Gaussian distribution for this particular I 1 so, this could be low, this could be your medium and that could be the high.

So, this type of non-linear distribution also we can consider. But for simplicity as I told we have considered the linear membership function distribution that is your the triangular membership function distribution. Now the size of this particular distribution are depends on so, these particular the d 1.

Now, d 1 indicates the base width for this right angle triangle or half base width of these particular the isosceles triangle. Now similarly to represent the second variable that is your I 2, we are going to use three linguistic terms like your small, large and very large. And once again for small we are going to consider this type of right angle triangle, for large we are going to consider this type of isosceles triangle and for very large we are going to consider this type of your the right angle triangle. And your the size of these particular distribution so, that is decided by actually your this d 2. So, d 2 is going to represent the base width of your this right angle triangle, and the half base width of these isosceles triangle.

Now, during the optimization and during the training actually what will happen we will try to vary the values for this particular d 1 and d 2 and accordingly we will be getting the modified membership function distribution for I 1 and I 2. Now here one thing we will have to mention that, in Takagi and Sugenos approach we do not consider any membership function distribution for the output. And here as we discuss the output is expressed as the function of the input parameters, that is O is nothing, but f of your I 1 and I 2, now we can consider the linear function of the input parameters, we can also consider some sort of non-linear function of these particular the input parameters. But most of the time we generally use only the linear function of the input parameters. So, these are the membership function distribution for the 2 inputs and as I told that for this output there is no membership function distribution used.

(Refer Slide Time: 06:17)



According to first-order Takagi and Sugeno's model of FLC

$$y^i = a_i I_1 + b_i I_2 + c_i,$$

where i = 1, 2, ..., 9

a_i, b_i, c_i are coefficients

$3 \times 3 = 9$

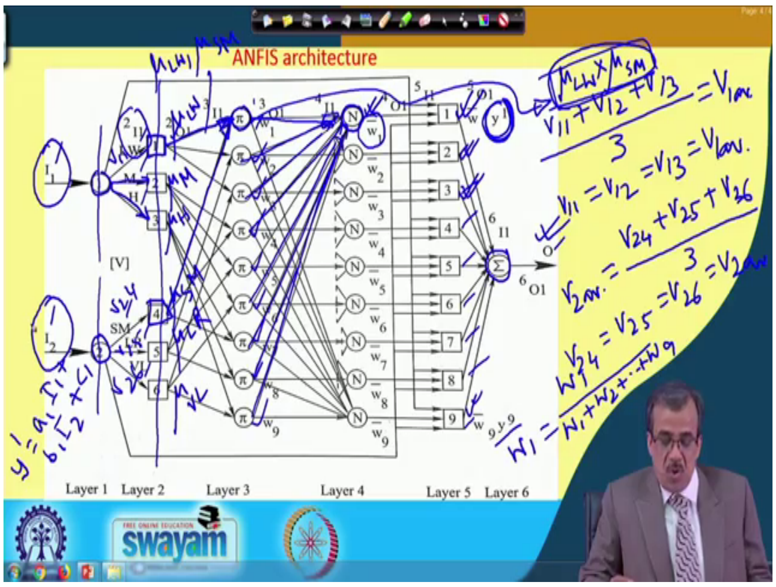
$I_1 \to 3$

$I_2 \to 3$

And here as we discussed while discussing the Takagi and Sugenos approach, that output here is expressed as the function of the input parameters. Now, we can see that the output of the ith rule that is denoted by y i is nothing, but a i I 1 plus b i I 2 plus c i now I 1 and I 2 are the input parameters and this a i b i c i are the coefficients. Now this values for the coefficients are to be determined, now using some optimizer, now we can use the least square error technique to find out what should be the values for this a i b i and c i, we can also use some sort of nature inspired optimization tools like genetic algorithm and others, like what should be the optimal values for this particular the a i b i and c i. Now if you remember for the first input that is I 1. So, we have considered three linguistic terms and for the second input that is I 2 we have considered three linguistic term. So, we have got 3 multiplied by 3. So, 9 possible combination of the input parameters and that is why so, here this y i is a i I 1 plus b i I 2 plus c i where i varies from 1 2 up to n.

So, we have got 9 combination of the input parameter; that means, we have got 9 rules. Now let us see how to determine the output for a set of input here.

(Refer Slide Time: 08:09)



ANFIS architecture
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
[V]
$\frac{V_{11}+V_{12}+V_{13}}{3} = V_{1av}$
$V_{11} = V_{12} = V_{13} = V_{av}$
$V_{2av} = \frac{V_{24}+V_{25}+V_{26}}{3} = V_{2av}$
$V_{24} = V_{25} = V_{26}$
$W = W_1 + W_2 + \dots + W_9$
$y_i = a_i I_1 + b_i I_2 + c_i$
swayam

Now, this shows actually the architecture of this ANFIS, now let me explain this particular architecture. Now here as I consider that there are 2 inputs like your I 1 and I 2 and we have got only one output that is O. And this ANFIS architecture can see stop actually 6 layers now this layer 1. So, this is known as actually the input layer and here we use linear transfer function on the input layer. Then layer 2 is nothing, but the fuzzification layer; that means, corresponding to the real values of this I 1 and I 2.

So, we try to find out what should be the membership function distribution. For example, say I 1 is represented using three linguistic terms, that is your low medium and high and the connecting weights between your the neurons lying on the first layer and the neurons lying on the second layer are going to represent what should be the base width of the triangular membership function distribution or half base width of the membership function distribution. Now for example, the connecting weights between so, this particular neuron and this particular the node is nothing, but your say V 11 similarly we have got the connecting weight V 12 we have got V 13.

Now, here so this V 11 V 12 and V 13 may have different numerical values and once again what we consider is your we consider the average of this three that is nothing, but is your V 11 plus V 12 plus V13 divided by 3 and this is nothing, but is your say V 1 average and we assign that V 11 equals to V 12 is equals to V 13 is nothing, but is your v average. Now this we do just to consider the symmetrical membership function distribution for these particular I 1. Similarly I 2 is represented using three linguistic terms like your small large and very large and we try to find out the connecting weights between. So, this particular neuron and that particular node that is nothing, but V 24. Similarly we have got V 25 and here we have got V 26.

And once again we follow the same method so, we try to find out the V 2 average. So, V 2 average is nothing, but is your V 24 then comes V 2 5 and then comes your V 26 divided by 3 and after that we assign that V 24 equals to V 25 then V 26 and that is nothing, but V 2 average and once again we do just to maintain the symmetrical membership function distribution for the different linguistic terms.

Now, if you see the input of the second neuron second layer. So, that is nothing, but your 2 I 1. So, 2 I 1 means that is the input of the first neuron or the first node lying on the second layer and so, these particular inputs are nothing, but is your the values the real values of these particular parameters like I 1 and I 2, I 2 proper here and here actually we will have to carry out some sort of the fuzzification. And for this particular fuzzification so we can get your the mu value.

So, this O12 that is 2 O1 is nothing, but the output of the first neuron lying on the second layer. So, here actually I will be getting some sort of mu value that is the membership function value that is your mu low. Similarly here we will be getting the mu medium and here you will be getting the mu high and if you see here also you will be getting the outputs like mu small, then comes your mu large and here will be getting your mu very large.

So, all such the membership function values will be getting as the output of the second layer, now let us concentrate on the third layer. Now on the third layer we have got all 9 possible combinations of the input parameters. So, all the 9 rules we are going to consider here and what should be the input of a particular neuron lying on the third layer? For example, say 3 I 1 that is the input of the first neuron lying on the third layer so the inputs are nothing, but your say mu low.

So, this is the inputs are nothing, but mu low and we have got another input and that is nothing, but is your mu small. So, mu low and mu small will be used as actually the input for example, say one input is coming from here and another input is coming from here. So, there are 2 mu values and these 2 mu values will be multiplied just to find out the firing strength or the output of this particular the third layer. So, output of this that is denoted by pi.

So, that can be written here that is nothing, but is your mu low multiplied by is your mu that your the small. So, we can find out like what should be the output of this particular the third layer. Now following the same principle so I can find out the output of each of these particular neurons lying on the third layer. So, I will be able to find out the firing strength that is denoted by is your the w. Now here actually we try to we multiply the 2 mu values, and unlike the Mamdani approach we do not consider the minimum.

So, here we simply multiply the 2 mu values, now each of the mu value is going to lie between 0 and 1. So, its multiplied form or the multiplication. So, this is also going to lie from 0 to 1. So, we will be getting some the firing strength that is w 1 w 2 up to your w 9 and the values of the firing strength we lie in the range of 0 to 1. Now then we concentrate on this particular your the layer 4. Now if I concentrate on a particular the neuron on layer 4 for example, I am going to concentrate here. Now if you see the inputs so, all such w values from the different neurons of the previous layer as coming as input to these particular the neuron.

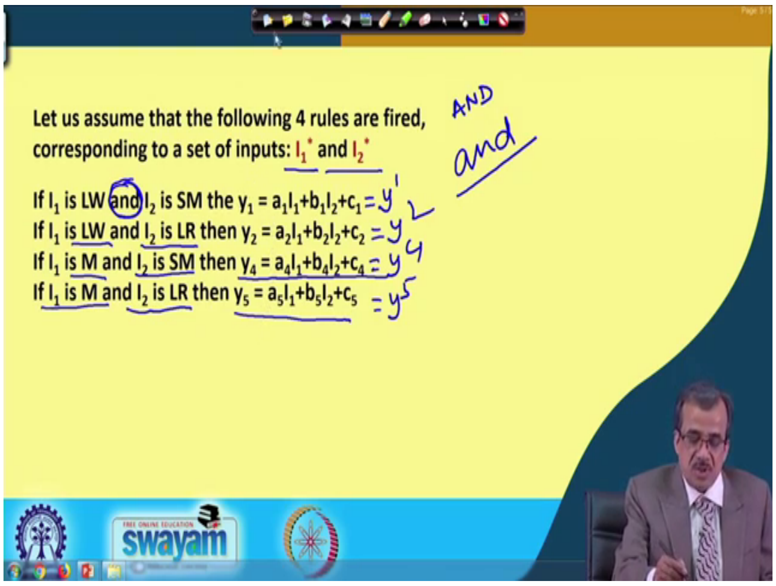
So, all such w values are coming as input and here you will be getting all your the w value; that means, all 9 w values and then we try to find out the normalized value for this particular w 1 as the output and your w 1 is nothing, but w 1 bar is nothing, but w 1 divided by w 1 w 2 up to your w 9. So, we try to find out what should be the normalized value for this particular your the firing strength.

Now, once you have got it as a output of the fourth layer, now we are proceeding to the fifth layer. Now on fifth layer you will find that say we are going to find out what should be the output of each of these particular the rule. For example, say we are trying to find out or we are trying to calculate what should be the output of your the first rule. The output of the first rule is nothing, but your a 1 I 1 plus your b 1 I 2 plus your c 1.

Now, if I get the values of this a 1 b 1 and c1 and if I supply the numerical values for I 1 and I 2. So, very easily I can calculate what is your y 1. Now similarly here we try to find out y 2 y 3 y 4 y 5 y 6 y 7 y 8 and y 9 and once you have get that. So, we simply multiply. So, this your w bar, that is a normalized value for the firing strength and this particular y one. So, and that will be the output of your. So, this particular neuron lying on the layer fifth, that is the fifth layer.

So, by following the same procedure so, I can find out the output here I can find out the output I can find out for all the nodes lying on the fifth layer. And once you have got that particular thing now on 6 layer, we sum them up just to find out what should be the final output of this particular the network for one set of input parameters that is your I 1 and I 2; now whatever I discussed here. So, those things actually are written a stepwise in the next slide.

(Refer Slide Time: 18:33)



Let us assume that the following 4 rules are fired, corresponding to a set of inputs: I_1^* and I_2^*

If I_1 is LW and I_2 is SM the $y_1 = a_1I_1+b_1I_2+c_1 = y^1$

If I_1 is LW and I_2 is LR then $y_2 = a_2I_1+b_2I_2+c_2 = y^2$

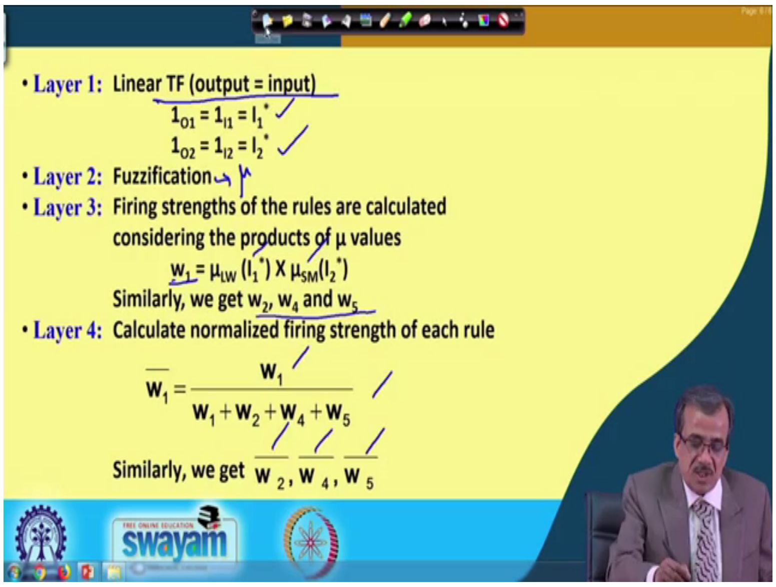
If I_1 is M and I_2 is SM then $y_4 = a_4I_1+b_4I_2+c_4 = y^4$

If I_1 is M and I_2 is LR then $y_5 = a_5I_1+b_5I_2+c_5 = y^5$

Now, before that let me tell you that corresponding to the set of inputs that is I 1 star and your I 2 star. So, only 4 rules are going to be fired out of 9 and those fired rules are nothing, but if I 1 is low and I 2 is small then y 1 is nothing, but a 1 I 1 plus b 1 I 2 plus c 1. Now this particular actually y one I can also write the notations which I am following. So, in this particular format; now here you can see so, I am using one and now this particular and is actually not the AND operator which are used in Mamdani. So, in Mamdani approach we use this type of AND operator, but in Takagi and Sugeno actually we do not use the AND operator instead we use the conjunction and that is nothing, but a n d small letter.

So, here in Takagi and Sugenos approach. So, use this type of and conjunction and. Now similarly if you see the second fired rule which states if I 1 is low and I 2 is large, then y 2 is nothing, but a 2 I 1 plus b 2 I 2 plus c 2 and this is nothing, but y 2. Now similarly we can also find out the output of the third rule and that is nothing, but y 4 and the rule is as follows, if I 1 is medium and I 2 is small then y 4 is nothing, but this that is a 4 I 1 plus b 4 I 2 plus c 4 and the fourth fired rule is as follows if I 1 is medium and I 2 is large, then y 5 is nothing, but is your a 5 I 1 plus b 5 I 2 plus c 5 and this is nothing, but is your y 5. So, these are the outputs of these 4 fired rules and let us see how to how to proceed all such things and how to explain step wise.

(Refer Slide Time: 20:55)



- **Layer 1:** Linear TF (output = input)
  $1_{O1} = 1_{I1} = I_1^*$
  $1_{O2} = 1_{I2} = I_2^*$
- **Layer 2:** Fuzzification
- **Layer 3:** Firing strengths of the rules are calculated considering the products of μ values
  $w_1 = \mu_{LW}(I_1^*) \times \mu_{SM}(I_2^*)$
  Similarly, we get w_2, w_4 and w_5
- **Layer 4:** Calculate normalized firing strength of each rule

$$\overline{w}_1 = \frac{w_1}{w_1 + w_2 + w_4 + w_5}$$

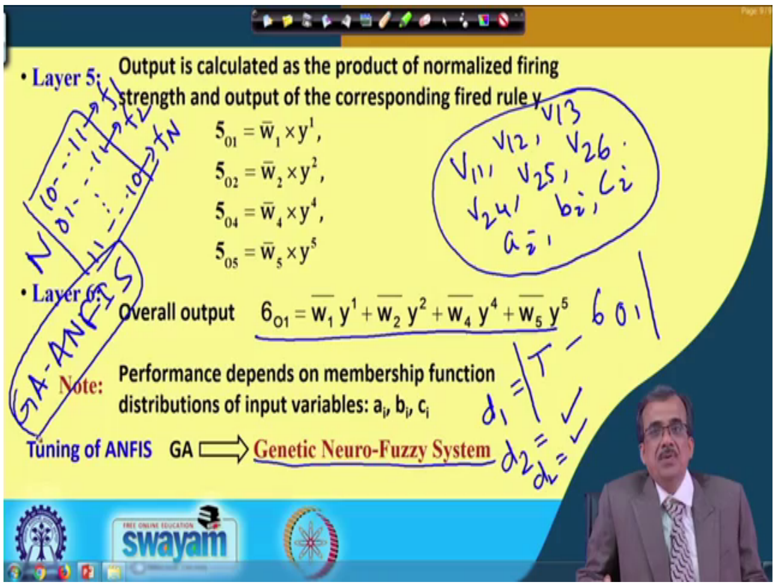
Similarly, we get $\overline{w}_2, \overline{w}_4, \overline{w}_5$

Now, whatever I discussed I am just going to write here step wise, now layer one we use or your the linear transfer function and that is why the output is nothing, but the input. So, this 1 o 1 is nothing, but 1 I 1 and that is nothing, but I 1 star. Similarly 1 o 2 is nothing, but 1 I 2 and that is equal to your I 2 star because we have used the linear transfer function and y equals to x. So, output is nothing, but input.

Now, layer 2 is the fuzzification and as I told we try to find out the membership function value that is mu. Layer 3 we try to find out the firing strength for each of the rules. Now to determine the firing strength for each of the rules. So, what you do is, the mu values we multiply. So, corresponding to the first fired rule the firing strength is nothing, but is your mu low corresponding to I 1 star multiplied by mu s m corresponding to your I 2 star. So, we will be getting the firing strength by following the similar procedure, we can find out what is y 2 what is y 4 and what is your y 5.

Now, one step got that firing strength, now you can find out your the normalized firing strength that is your w 1 bar is nothing, but w 1 divided by w 1 plus w 2 plus w 4 plus w 5. And by following the same procedure so, we can find out the normalized firing strength like your w 2 bar, w 4 bar, then w 5 bar. So, all such normalized firing strength values we can calculate.

(Refer Slide Time: 23:03)



- **Layer 5:** Output is calculated as the product of normalized firing strength and output of the corresponding fired rule y

$5_{01} = \bar{w}_1 \times y^1,$

$5_{02} = \bar{w}_2 \times y^2,$

$5_{04} = \bar{w}_4 \times y^4,$

$5_{05} = \bar{w}_5 \times y^5$

- **Layer 6:** Overall output $\underline{6_{01} = \overline{w_1}\, y^1 + \overline{w_2}\, y^2 + \overline{w_4}\, y^4 + \overline{w_5}\, y^5}$ 6_{01} |

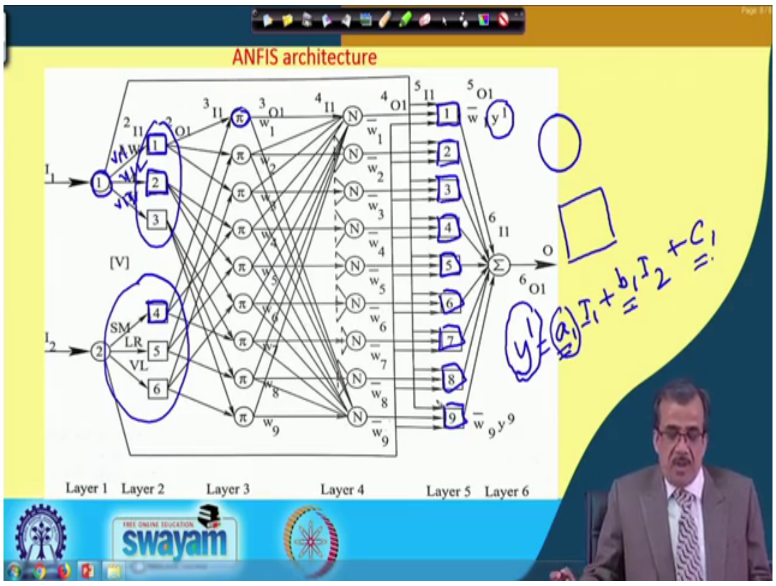
Note: Performance depends on membership function distributions of input variables: a_i, b_i, c_i

Tuning of ANFIS GA $\Longrightarrow$ Genetic Neuro-Fuzzy System

And in layer 5 so, what we do is we try to find out the output that is the output of the first neuron lying on the fifth layer is nothing, but w 1 bar multiplied by y 1. Then 5 o 2 that is the output of the second neuron lying on the fifth layer is nothing, but w 2 bar multiplied by y 2. Then comes here 5 o 4 that is w 4 bar multiplied by your y 4 then comes here 5 o 5 that is the output of the fifth neuron lying on the fifth layer is nothing, but w 5 bar multiplied by is your y 5.

So, we can find out the output and once you have got those outputs, now we are in a position to find out the final output or the overall output and that is nothing, but your 6 o 1 is nothing, but w 1 bar y 1 plus w 2 bar y 2 plus w 4 bar y 4 plus w 5 bar y 5. So, we can find out the output of this particular the set of inputs. Now here we should mention that the performance of this particular ANFIS depends on the coefficient of transfer functions that is a i b i and c i and of course, it depends on the membership function distribution or these, the 2 inputs that is your I 1 and I 2.

(Refer Slide Time: 25:01)



Now, here actually what you do is like if we just go back to the architecture if we look into the architecture once again now, this particular architecture you can see there are a few the neurons which are denoted by circle, on the other hand we have got a few other neurons which are denoted by the square. So, this is nothing, but a square. So, we use 2 types of symbols in this particular the ANFIS, one is your the circle and another is your the square. Now wherever we use square there is a chance of further improvement, there is a chance of optimization. So, if you see. So, here I am putting your the square; that means, we can optimize the values for this particular the connecting weights, that is your V 11 V 12 and V 13 and that is why we have put square here.

So, there is a chance of improvement, there is a chance of optimization. Now for the same reason actually we have put here square; that means, we can also optimize your the membership function distribution for these linguistic terms that is your small large and very large. Now similarly if you see here also you have put actually your the square symbol in place of the circle; that means, here also there is a chance of improvement and that improvement actually lies on the value for this particular your y. Because if you write down the expression for this particular y 1 you will see that we have got your a 1 I 1 plus your b 1 I 2 plus your c 1.

So, this particular a 1 b 1 and c 1 are actually going to control the value for this particular your y 1. So, there is a chance of further improvement of these particular the performance by selecting the proper values for this particular a 1 b 1 and c 1. Now as I have already mentioned that these values for the coefficient that is your a i b i and c i can be determined using some optimizer.

For example, we can use the traditional tools for optimization also, we can use some sort of the error minimization algorithm. So, that we can find out this coefficient like a i b i and c i. So, that it can predict the output very accurately at the error in prediction should be as minimum as possible. That means, your here there is a chance of improvement of the performance of this particular network and that is why we put the square here, in place of this particular your the circle.

So, this is the way actually your so, this particular the network works and ultimately for the set of inputs will be getting that particular the output. And as I told that we can improve the performance using some optimizer some traditional optimization tool we can use, we can also use some sort of nature inspired optimization tool, now if use genetic algorithm to tune this particular ANFIS. So, what you can do is on the GA string we can put all such the information like your connecting weights, that is your the v values. So, the connecting weights we can put that is the V values like your V 11 V 12 V 13 then we can also put your V like 2 4 then comes V 2 5 then V 2 6.

All such things you can put there and we can also encode the values of the coefficients like your a i b i and c i then 0 will take the responsibility to find out what should be the optimal values of these particular parameters so that this ANFIS work in the optimal sense. So, that the ANFIS should be able to make the prediction of the output or the set of inputs as accurately as possible, now here actually what you do is as you have already discussed. So, this is nothing, but the calculated output, at this calculated output will be compared with the target output just to find out the deviation for the first training scenario that is nothing, but is your d 1 and we try to find out the target output and your this calculated output and we consider the mod value.

Now, a corresponding to a particular GA string the way I discussed, we pass all the training scenarios one after another. So, if I pass the second training scenario. So, I will be getting d 2, similarly if I pass the lth training scenario. So, I will be getting like d L, you sum them up find out the average and that particular average value will be the fitness of the GA. So, if I consider. So, this type of population of the GA if it is a binary coded GA. So, I will be getting f 1, similarly for the second I will be getting f 2 as the fitness and for the nth one.

So, I will be getting f N as the fitness and using the fitness information and with the help of its operator like reproduction, crossover and mutation GA we will try to find out or try to evolve like what should be the optimal values of these particular parameters so, that this ANFIS can make the prediction as accurately as possible.

Now, if I use genetic algorithm to optimize on or train this particular ANFIS network. So, this will be known as actually genetic neuro fuzzy system and a more specifically. So, if I use genetic algorithm to tune this ANFIS so, this is very popularly known as your GA ANFIS. So, this GA ANFIS means I am just going to optimize the ANFIS parameters the variables for the ANFIS with the help of a genetic algorithm, and GA ANFIS will be able to evolve a very optimal and you are the sweet able ANFIS network which will be able to make the prediction for the input output relationship as accurately as possible.

Thank you.