Fuzzy Logic and Neural Networks

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

Lecture - 33
Neuro - Fuzzy System

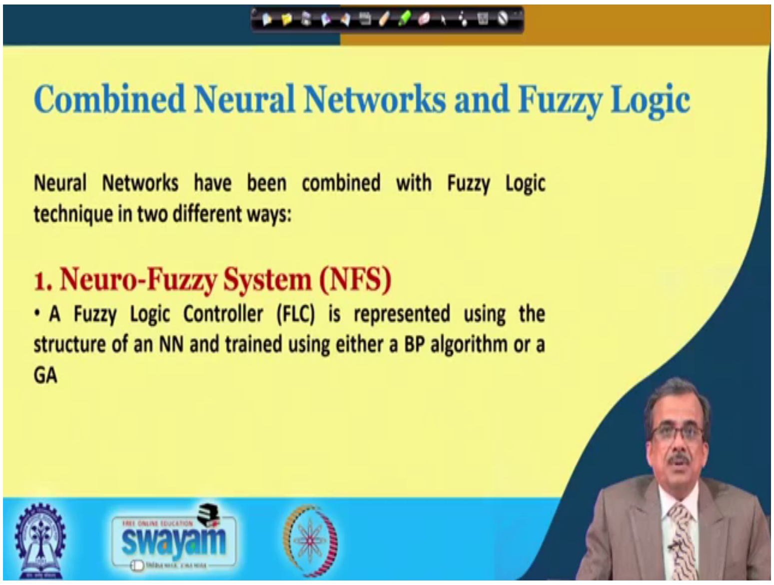
Now, we are going to discuss how to combine Neural Networks with Fuzzy Logic, just to develop a Neuro-Fuzzy System. Now this particular neuro-fuzzy system has got a number of practical applications. Now here actually what we are going to do?

(Refer Slide Time: 00:37)

We are going to concentrate on 2 Neuro-Fuzzy systems; one is based on the Mamdani approach and another is based on the Takagi and Sugeno's approach. Now both Mamdani approach and Takagi and Sugeno's approach we have discussed in details and we have solved some numerical examples.

Now, let us see how to develop the neuro-fuzzy system for this Mamdani approach and Takagi and Sugeno's approach?

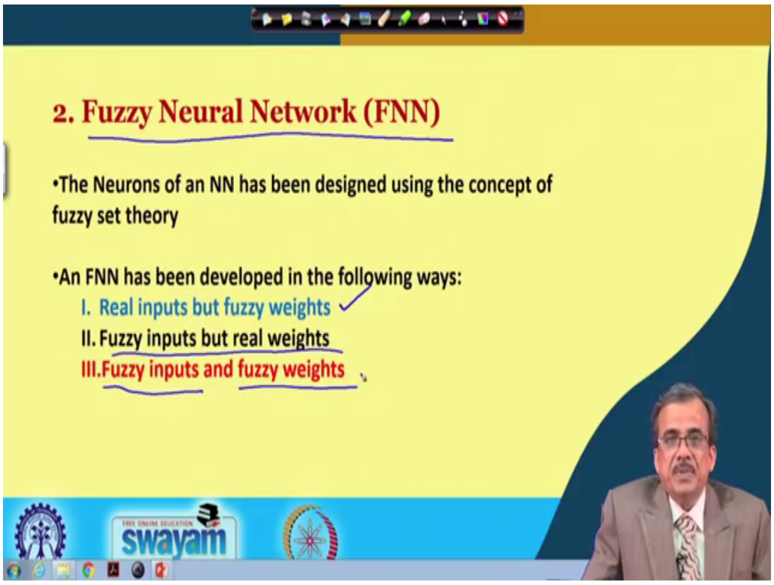
(Refer Slide Time: 01:11)



Now, let us concentrate on this combined neural network and fuzzy logic. Now as I told that this Neuro-Fuzzy System, so, actually in short this is known as NFS, here the purpose is to improve the performance of these fuzzy reasoning tool and what we do is, the fuzzy reasoning tool or fuzzy logic controller we represent using the structure of a neural network and this network is trained using either a BP algorithm or a genetic algorithm or any other nature inspired optimization tool. But basically the main purpose of developing your the neuro-fuzzy system is to design and develop your the fuzzy reasoning tool or fuzzy logic controller.

Now, if you see this literature, now neural network and fuzzy logic have been combined in fact, in 2 different ways; one is called the neuro-fuzzy system and another is called the fuzzy neural network.

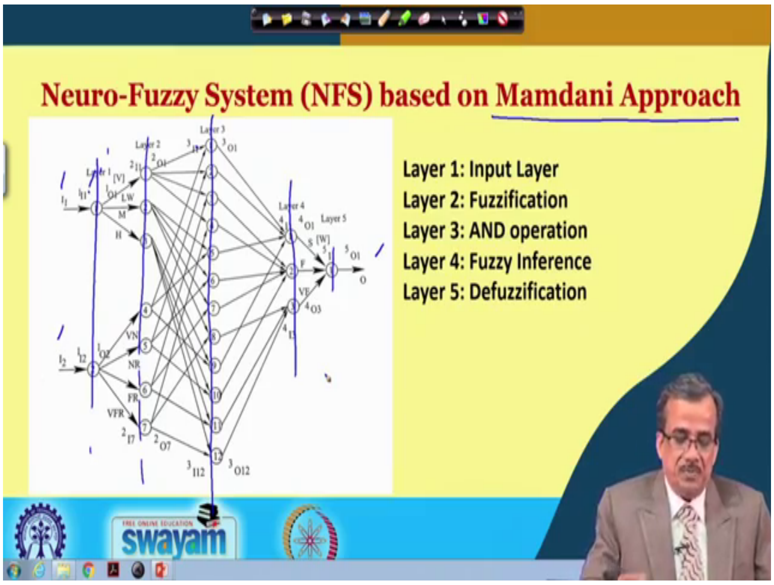
(Refer Slide Time: 02:27)



Now, if you see the fuzzy neural network, now here actually what we do is? We try to represent the neurons of a neural network using the concept of your the fuzzy set theory. And in your fuzzy neural network actually there are 3 different ways through which we can develop, we can consider the real inputs and fuzzy weights this is one way of developing the fuzzy neural network then comes we can consider the fuzzy inputs, but real weights and we can also consider fuzzy inputs and fuzzy weights. Now let me repeat.

So, this fuzzy neural network can be implemented in three different ways as I told. But unfortunately so, this fuzzy neural network could not reach much popularity on the other hand your the neuro-fuzzy system that is your NFS could reach a huge popularity just to solve your the different types of the real world problems and that is why actually in this course so, I am just going to consider or the neuro-fuzzy system.

(Refer Slide Time: 03:57)



Now this Neuro-Fuzzy system as I told that this is known as NFS and your here I am just going to discuss the neuro-fuzzy system based on the Mamdani approach. Now let us see the Mamdani approach a fuzzy reasoning tool or fuzzy logic controller, how to represent using the structure of a neural network and so, that we can develop the neuro-fuzzy system. But let me repeat once again. The main purpose of developing this neuro-fuzzy system is to design or evolve rather one the fuzzy reasoning tool very efficient fuzzy reasoning tool, which can perform the input output relationship.

Now, if you remember in a Mamdani approach we have got a few steps. For example, say we try to identify the inputs and outputs. Now once those inputs and outputs are identified then we actually went for some sort of fuzzification. And a once we have got you are the fuzzified values for the input parameters then we in fact, implemented your the inference the fuzzy inference. And through this particular fuzzy inference we got some output and those outputs are nothing, but the fuzzified outputs. And once you got that fuzzified output we went for the defuzzificztion. So, that we can find out the criss value of these particular the output.

Now here so, this shows actually the schematic view of one neuro-fuzzy system, very simple neuro-fuzzy system having your the 2 inputs and 1 output and this particular network it consists of 5 layers. So, we have got this input layer. So, this is the input layer the first layer is the input layer, the second layer is actually the fuzzification layer, third layer is known as the AND operation layer, then comes the fourth layer is nothing, but fuzzy inference and the fifth layer is nothing, but is your defuzzification. Now look wise this is similar to a particular the network, but truly speaking the is nothing, but the Mamdani approach of fuzzy reasoning tool.

Now what is the reason that why do you consider the structure of the neural network to represent a particular the fuzzy reasoning tool? Now the answer is very simple we use the structure of this particular network to implement or to actually represent the fuzzy reasoning tool or fuzzy logic controller, based on Mamdani approach so, that we can modify this particular network in order to train the Mamdani approach of fuzzy reasoning tool using the principle of either the back propagation algorithm or we can use some setup nature inspired optimization tool like genetic algorithms and others. So, that is the main purpose why did we take the help of so, this type of the structure of the network.

Now here if you see we have got 2 inputs like your I 1 and I 2. So, on the first layer, so, here you can see I have written I I 1 that is the input of the first neuron lying on the input layer and this is I I 2 that is the input of the second neuron lying on the your the first layer and here I have got actually 1 O1 that is the output of the first neuron lying on the first layer and then 1 O2 that is the output of the second neuron lying on the first layer. So, the first layer is nothing, but the input layer. Now here so, this is actually the second layer is the fuzzification layer. Now the purpose of fuzzification is to find out the your the normalized value corresponding to these particular the inputs.

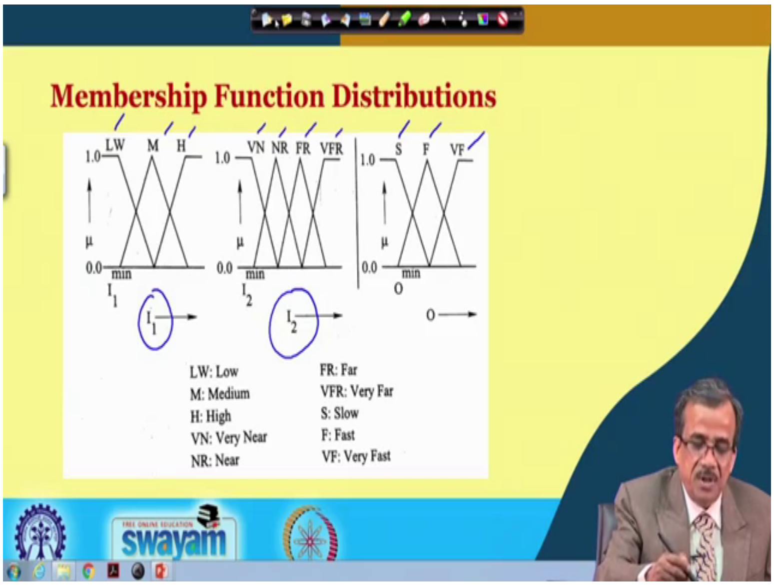
Now how to implement? So, that I am going to discuss in details. Now once you have got the fuzzified output and so, here you will be getting the outputs of layer 2 is nothing, but the values of the memberships. Now these membership values for the 2 inputs I 1 and I 2 I will have to use here as input of the layer 3. And on layer 3 actually will have to carry out the AND operation. And if you remember in AND operation so, what we do is?

So, we try to compare the 2 mu values and we try to find out the minimum. Now this minimum of these 2 mu values will be considered as output of layer 3 or the third layer and layer 4 is nothing, but the fuzzy inference layer. And as we have discussed the purpose of fuzzy inference is to select the set of fired rules corresponding to one set of input parameters.

Now in the rule base we have got a large number of rules. So, out of these large number of rules only a few will be fired depending on the set of inputs. Now fuzzy inference is going to decide which once the rules are going to be fired out of the maximum number of rules present in that particular the rule base. Now once we pass through this particular the fourth layer or the layer 4. So, I will be getting actually the output here and these outputs are denoted by actually 4 O1 that is nothing, but the output of the first neuron lying on the fourth layer and so, on and here those things will enter as input to the fifth layer or the defuzzification layer and as output of this defuzzification layer. So, I will be getting some crisp output.

Now this is the way actually it works in short, but I will be discussing in much more details and after that we will be solving one numerical examples also. Now let us see now let us see this the principle in much more details.

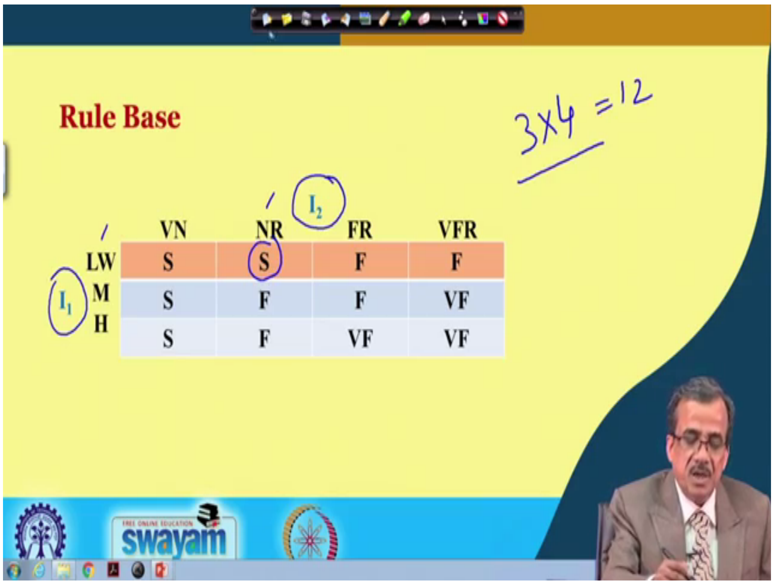
(Refer Slide Time: 11:13)



Now, here for simplicity we have assumed that both the inputs and the output are having actually the triangular membership function distribution for example, say. So, these are nothing, but the membership function distribution for your I 1. So, the first input are expressed using 3 linguistic terms like your low, medium and high. Similarly 3 other 4 other linguistic terms are used to represent I 2 for example, we have got very near far and very far and the output is actually represented using 3 other linguistic terms. So, S is nothing, but slow F is nothing, but fast and VF is nothing, but is your very fast.

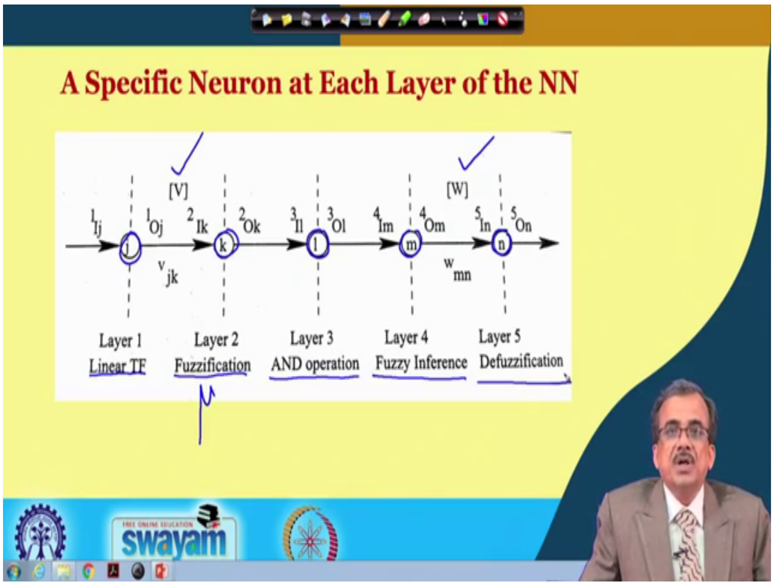
So, for these 2 inputs and 1 output so, we used actually the triangular membership function distribution. Now let us see how to proceed with this your the designing of the this neuro-fuzzy system.

(Refer Slide Time: 12:31)



Now, as there are 3 inputs for this particular other 3 linguistic terms for this I 1 and there are 4 linguistic terms for this your I 2. So, we have got 3 multiplied by 4 rules. So, we have got actually 12 rules. The rules are as follows if I 1 is low and I 2 is very near then the output is your S. So, S is nothing, but the slow. So, this is the way actually we designed this particular the 12 rules. So, these 12 rules are designed something like this.

(Refer Slide Time: 13:17)



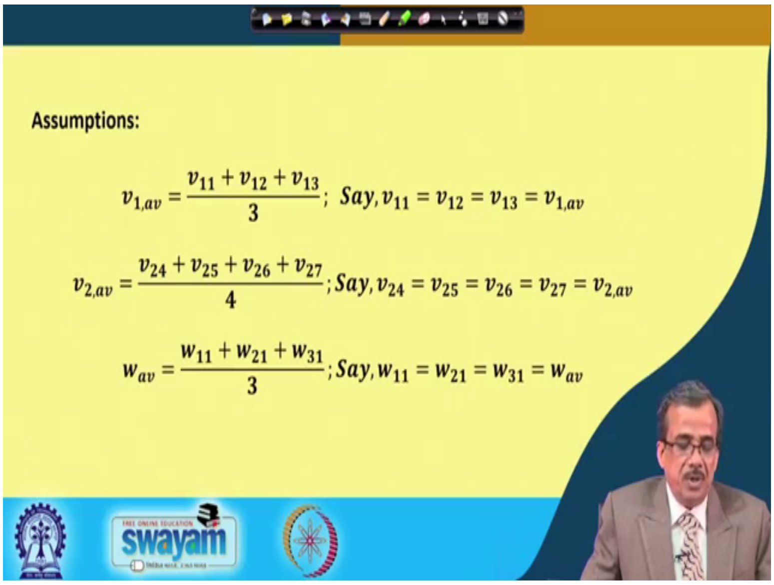
Now I am just going to represent actually each particular neuron lying on each of the layers. Now for the purpose of analysis. So, here we are going to consider the jth neuron lying on the first layer, then comes here the jth neuron lying on the first layer then the kth neuron lying on the second layer, lth neuron lying on the third layer and mth neuron lying on the forth layer and nth neuron lying on the fifth layer.

Now the connecting weight between the layer 1 and the layer 2 is denoted by V, the connecting weight between your layer 4 and the layer 5 is denoted by your the W matrix; now the nomenclature let me explain once again. So, 1 I j is nothing, but the input of the jth neuron lying on the first layer then 1 Oj that is the output of the jth neuron lying on the first layer, then your 2 Ik that is the input of the kth neuron lying on the second layer.

Then 2 Ok that is the output of the kth neuron lying on the second layer, then 3 Ik the input of the first neuron the lth neuron lying on the third layer, the 3 O l that is the output of the lth neuron lying on the third layer, then 4 Im that is the input of the mth neuron lying on the fourth layer, then o 4 Om that is the output of the mth neuron lying on the fourth layer, then 5 I n that is the input of the nth neuron on lying on the fifth layer, then 5 On that is the output of the nth neuron lying on the fifth layer. And as I mentioned the first layer we use some set of linear transfer function that is output is nothing, but the input in layer 2 we carry out fuzzification; that means. So, we try to find out the membership function value denoted by mu which I have already discussed.

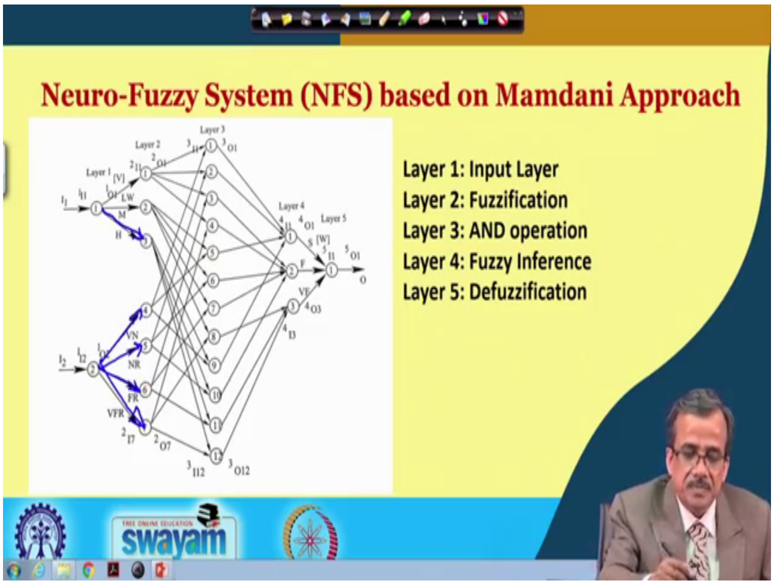
Layer 3 actually here it indicates all 12 rules and we carry out. So, this particular the AND operation; that means, we compare 2 mu values corresponding to the 2 inputs and we try to find out the minimum. The layer 4 is nothing, but the fuzzy inference and layer 5 is the defuzzification so, that we can find out like what should be the crisp output corresponding to that fuzzified output.

(Refer Slide Time: 16:21)



Now, let us see how does it work? Now if you once again look into your so, this network.

(Refer Slide Time: 16:27)



So, here you can see that we have got some connecting weights like your v 11 then comes v 12 then comes v 13 we have got the connecting weights like your v 24 then v 25 v 26and v 27. Now these are all connecting weights between layer 1 and layer 2 and these particular connecting weights lie in the range of 0 to 1.

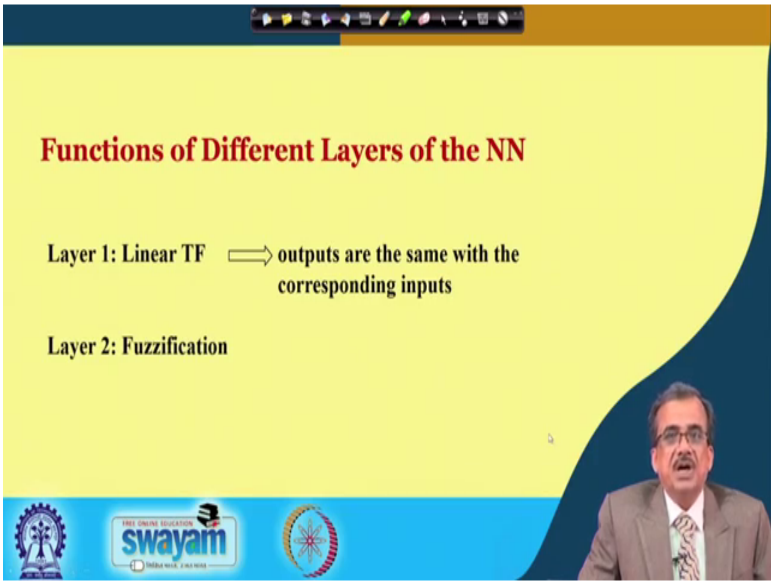
Now, here actually what we do is. So, this connecting weight expressed in the normalized scale may have some other value in the real scale and that particular value in the real scale is going to represent, what should be the spread of that particular the triangular membership function distribution; that means, your if it is a right angle triangle. So, its base width is base width will be represented by the connecting weight. If it is isosceles triangle so, its half base width will be represented by so, these particular the connecting weights. Now truly speaking v 11 could be different from v 12 could be different from v 13. Now what we do is we try to find out the average of these particular three and then we assign that v 11 is nothing, but v average, v 12 is nothing, but v average and v 13 is nothing, but is your v average.

Now, this is done actually just to ensure that we have got the symmetrical triangular membership function distribution. Now if we consider a symmetrical membership function distribution. So, you need not consider the average and you need not assign the average value to each of these particular the connecting weights. The same thing actually is done for so, these connecting weights also which are going to represent the base width or the half base weight of the triangular membership function distribution used to represent your the I 2. Now let us see how does it work? Now if you see the way it has been implemented is as follows.

So, here as I told that we try to find out first. So, this v11 v12 v13 find out the average. Now this particular average you assign to v11 v12 and v 13; that means, v11 equals to v12 equals to v13 is nothing, but v 1 average the same principle we follow here also. Like v 2 average is nothing, but v 24 plus v 25 plus v 26 plus v 27 divided by 4 and after that we assign v 24 equals to v 25 equals to v 26 equals to v 27 and that is nothing, but v to average.

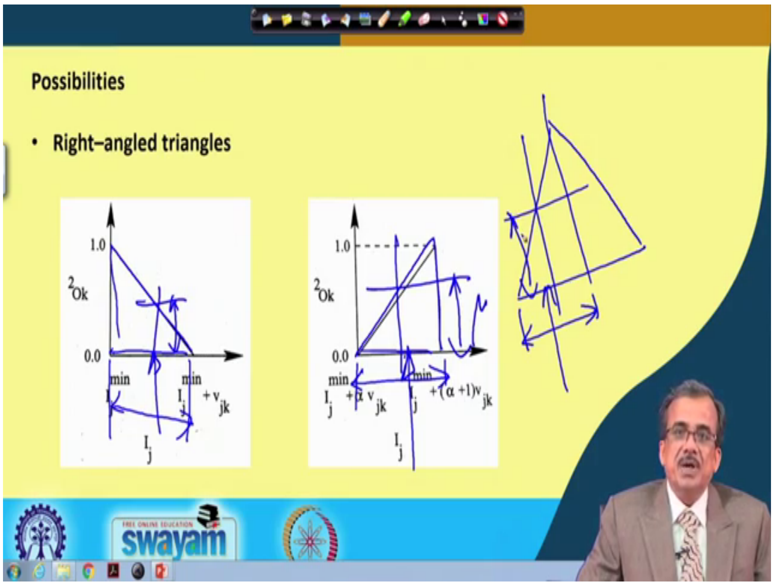
And this w average we calculate. In fact, for the connecting weights between the fourth layer and your the fifth layer and these particular connecting weights or the w values are going to represent what should be the size of this particular the triangular membership function distribution representing the output. The same principle if follow here so, we find out the average of w 11 w 21 w 31 and then we assign w 11 equals to w 21 equals to w 31 is equal to w average. So, all such things we do now actually you are the membership function distribution for the inputs and the outputs are ready and once they are ready, now actually we are in a position to pass the set of input parameters.

(Refer Slide Time: 20:51)



Now here actually the all the steps all the 5 steps. So, once again we have written in this particular fashion. So, layer 1 we consider actually the linear transfer function and here the outputs are kept same with a the corresponding inputs. Then layer 2 is nothing, but the fuzzification module and here in this fuzzification module actually what we do is? We try to find out what should be the membership function value.

(Refer Slide Time: 21:39)

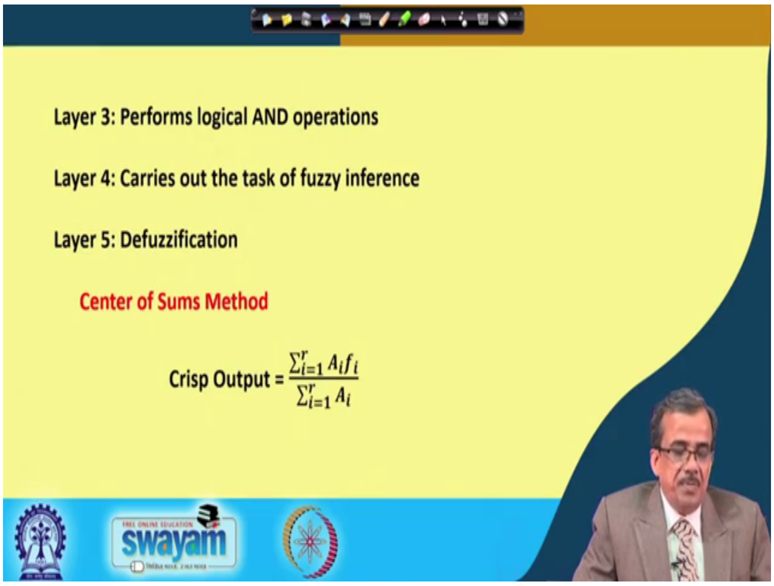


Now, if you concentrate on the membership function distribution of the input parameters that is I 1 and I 2. So, you will be getting actually three types of triangle. So, one is this type of right angle triangle you will be getting and you will be getting this type of right angle triangle also and you will be getting actually these type of isosceles triangle also.

Now if you are getting this type of right angle triangle. So, during the optimization or the during the training. So, we are going to optimize so, this particular your the base width. And if you are getting this type of right angle triangle. So, we are going to optimize so, this base width and if you have the isosceles triangle. So, we are going to optimize the half base width of this particular the triangle. And the moment we pass actually a particular the input for example, say I am here.

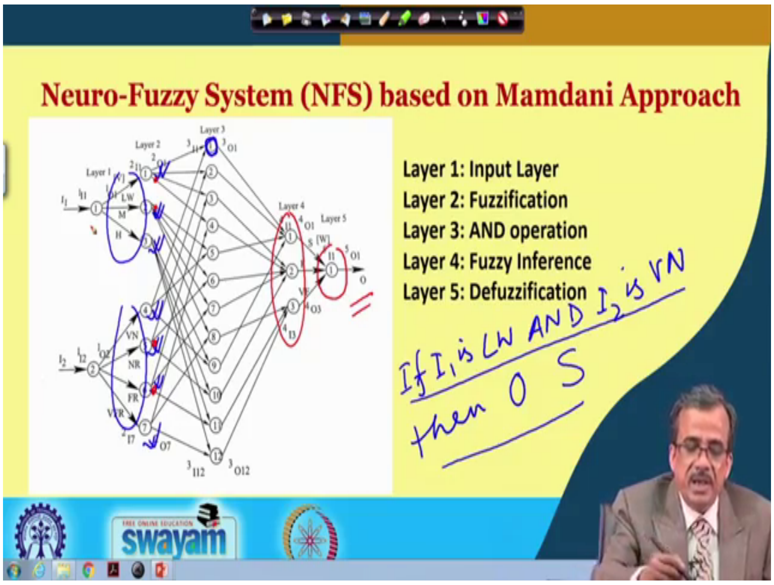
So, very easily I can find out what should be the mu value; and because the range for mu is your 0 to 1. So, using the principle of similar triangle so, very easily you can find out the mu value. This we have already discussed in much more details similarly corresponding to this if I pass one input so, I will be able to find out so, this particular the mu value. Then corresponding to this actually if I pass then I will be getting so, this as the view value or the membership function value. Now the same thing actually we follow just to carry out that fuzzification the step.

(Refer Slide Time: 23:21)



That means corresponding to the set of inputs you will be getting some mu value and once you got that particular mu value, then you go for the logical AND operation.

(Refer Slide Time: 23:45)



Now let me once again go back to the your the schematic view of the neuro-fuzzy system and you will understand here actually what we do is. So, you are passing this I 1 and I 2 So, this output is nothing, but the input and these particular connecting weights are going to represent the membership function distribution, the shape and size of the membership function distribution and the output of the second layer is nothing, but is actually your the membership function value or the mu value.

And as I told so, this particular AND operator or the layer 3 is going to represent the AND operator. So, we have got 12 rules. So, each of the 12 rules actually we are going to represent here. For example, if I concentrate on the first neuron of third layer. So, this is nothing, but if the input I 1 is low and the input I 2 is your very near then the output is nothing, but so, this is your slope. So, corresponding to this particular the first neuron lying on the third layer. So, this indicates if your I 1 is your low and I 2 is your very near, then the output O is nothing, but is your S. So, output is nothing, but slow. So, this is the rule which is actually represented by the first neuron lying on the third layer.

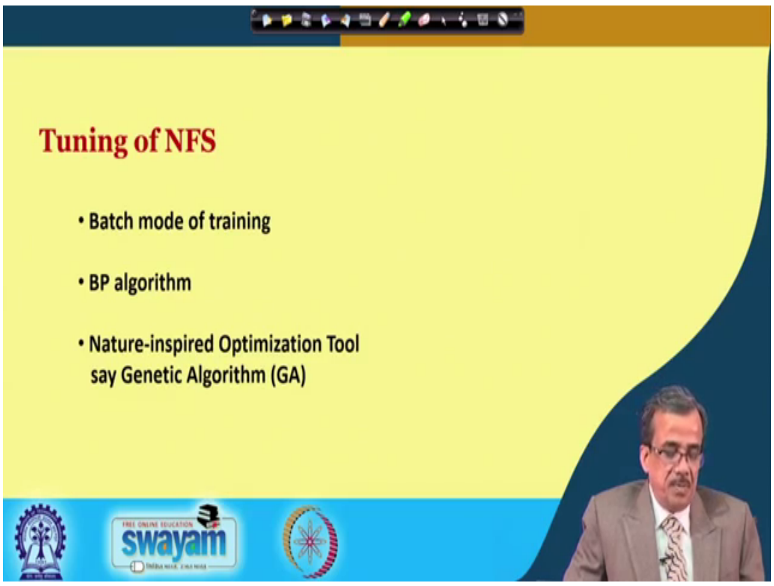
Similarly, we have got 12 such neuron just to represent the 12 rules. The moment I pass one set of inputs here I 1 and I 2. So, there is a possibility I will be getting actually your two such nonzero mu here. So, for example, I have I may get these has the non zero mu this has the mu similarly corresponding to I 2 actually I may get this has the nonzero mu this has the non zero mu. Then if I get the non zero mu here and the non zero mu here similarly nonzero mu and non zero mu here. So, there is a possibility that out of these 12 rules only four rules are going to be fired. And corresponding to each of the fired rules so, we are going to find out actually what should be the fuzzified output and that is actually the purpose of using your the fuzzy inference.

So, using the fuzzy inference. So, we can find out the output the fuzzified output of each of the fired rules four fired rules and after that we carry out defuzzification here just to find out what should be the crisp output corresponding to these set of inputs. Now this is the way actually your; so, this particularly neuro-fuzzy system works that is the Mamdani approach works.

Now here if you see so, till now we have discussed up to this. Now layer 3 we perform the logical and operation therefore, we carry out the task of fuzzy inference and there will be defuzzification on layer 5. There are several methods of defuzzification, now here I am discussing how to use the principle of the center of sums method. The center of sums method in fact, I have discussed in much more details while discussing the fuzzy reasoning tool particularly the Mamdani approach.

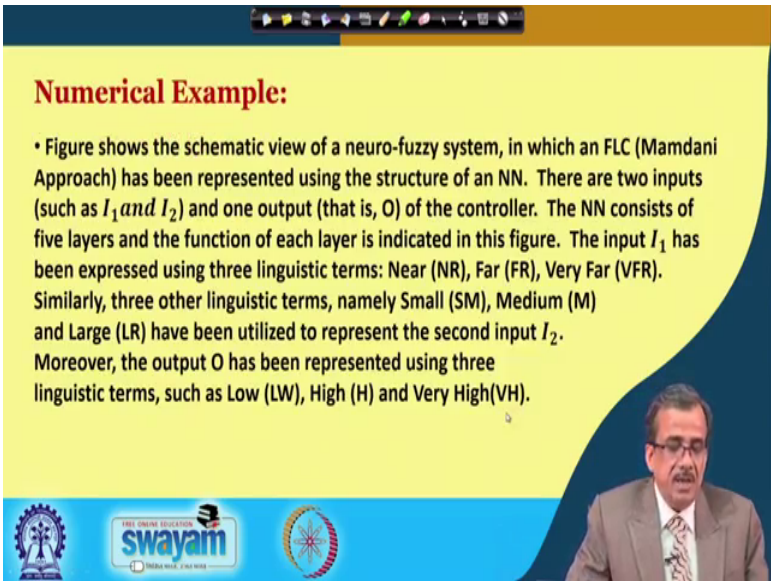
Now, the crisp output is nothing, but summation your i equals to 1 to r A i f i divided by summation i equals to 1 to r A i. So, we can find out actually what should be the crisp output and once you have got this particular the crisp output. So, this particular crisp output can be used for your the controlling purpose.

(Refer Slide Time: 28:11)



Now, let us discuss little bit how to tune or how to train this particular the neuron fuzzy system. Generally we consider the batch mode of training to train this type of neuro-fuzzy system, we can use either back propagation algorithm or we can use some nature inspired optimization tool like your genetic algorithm or particle swarm optimization and so on. And using this particular the tools actually we can optimize.

(Refer Slide Time: 28:53)



And once you optimize so, now you are in a position actually to train this particular network; that means, we are basically giving training to the fuzzy reasoning tool which works based on this particular your Mamdani approach.

Thank you.