Pattern Recognition and Application
Prof. P. K. Biswas
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 32
Fuzzy Min Max Neutral Network for Pattern Recognition

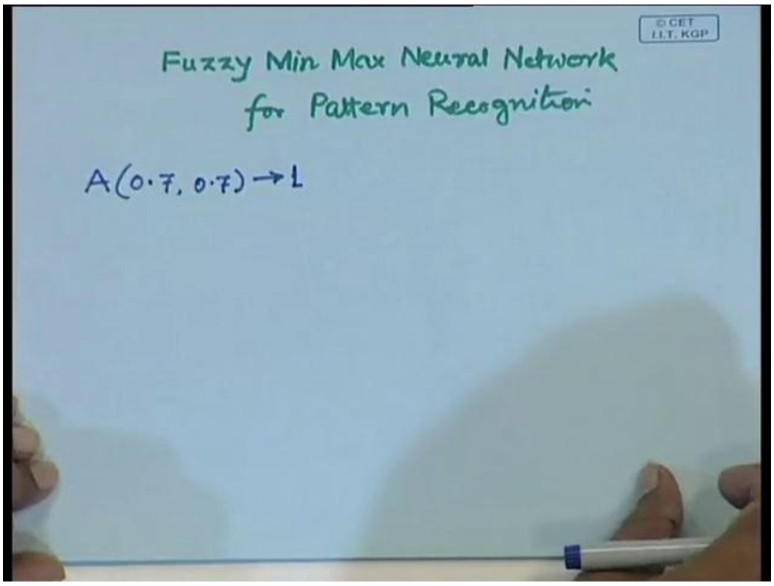
So, we were discussing about the fuzzy neural network, fuzzy min, max neutral network for pattern recognition. And in the last class we have said that how the fuzzy, how the hyper box classified can be extended to fuzzy min max neural network, for every middle end note in the fuzzy min max neural network will compute the fuzzy membership function of an unknown point to a particular hyper box.

Then outputs of all the hyper boxes belonging to the same class they are gathered together in one of the output ((Refer Time: 00:52) nodes or class nodes. But the class node decides about what is the maximum membership given by the different hyper boxes belonging to different classes. And the membership function to that class is this maximum membership of different hyper boxes belonging to a same class.

So, if I have c number of output neurons pertaining to c different classes, then every c neuron will give me a membership function and this membership function is the membership of the unknown feature vector to the corresponding class. And we said that this itself can be taken as fuzzy classification because this is nothing but a membership vector or if we want this classification or hot classification then whichever neutron gives the maximum membership value we can put the unknown feature vector into the corresponding class.

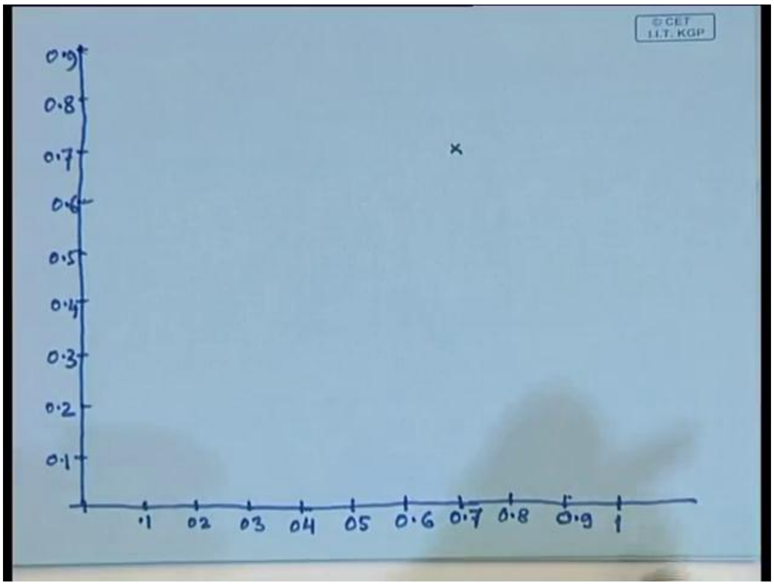
So, we can have a max operator which will give us a binary vector or only one of the components of a binary vector will be equal to 1. And that obviously corresponds to a maximum membership and all other components of the binary vector will be equal to 0. And that is equivalent to our hard classification. So, today what we will do is we will take the same example that we have taken for designing the hyper boxes. And we will take the same example and we will re draw the hyper boxes and simultaneously we will try to design a fuzzy min max neutral network. So, this is nothing but learning or training of the fuzzy min max neutral network. So, the set of points that we had taken in the last class is something like this.

(Refer Slide Time: 02:39)



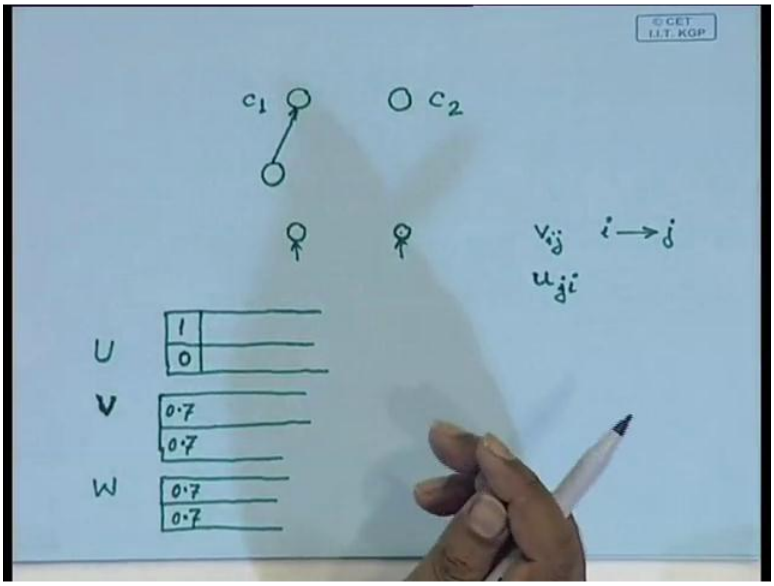
We had taken a point having the components 0.7 0.7 and we said that this belongs to class 1. And let us simultaneously draw the hyper boxes that we will get out of these points that we have taken, okay?

(Refer Slide Time: 03:01)



So, we will do it like this. The lines are very light so I will put it like this is 0.1, 0.2, 0.3, 0.4. So, when we said that this point, this feature vector 0.7 0.7 this belongs to class 1. So, immediately we said that there will be a hyper box for which the min point and max points are same. So, I will have a hyper box somewhere over here 0.7 0.7. So, at this location I will have a point hyper box for which min point and max point is same, that is 0.7 0.7. And for this hyper box let us draw try to generate this fuzzy min max neutral network. So, as we said that I assume that there are two classes, class 1 and class 2.

(Refer Slide Time: 05:24)



So, obviously in the output layer I will have two nodes, one corresponding to class 1and other one corresponding to class 2. The first node corresponds to class 1 and the second node corresponds to class 2. Similarly, at the input layer also I we will have two nodes because I am considering feature vectors of dimension 2. So, at the input layer also we have two nodes. So, I have node 1 and input node and this will have the input feature vectors.

And while designing this fuzzy min max neutral network what we have to do is we have to generate three matrices because from the middle layer to the output layer, the connections are actually binary connections, either 0 or 1. 1 means there is a connection from a middle layer node to that corresponding output layer node. 0 means there is no connection from the corresponding middle layer node to the corresponding output layer node. And I will have two matrices, which represents the connections from the input layer nodes to the middle layer nodes. And those two matrices are actually represented the min point and max point.

So, for min point I will have a matrix U. For max point I will have a matrix W and for the middle layer node to the output layer node I will have, sorry I will have matrix V and

W, not even W. V corresponding to the min points and W responding to the max points. And I will have a matrix U which represents the connections from the middle layer nodes to the output layer nodes.

So, if I use this convention that for every note in the input layer I will have one row in matrix U and one row in matrix W, so this matrix, sorry matrix V and matrix W. So, matrix V will have two rows corresponding to the min points and the number of columns will actually be variable because till now I do not know what the number of nodes in the middle layer is. So, the number of nodes in the middle layer that will actually represent that how many columns I need in this matrices V and W.

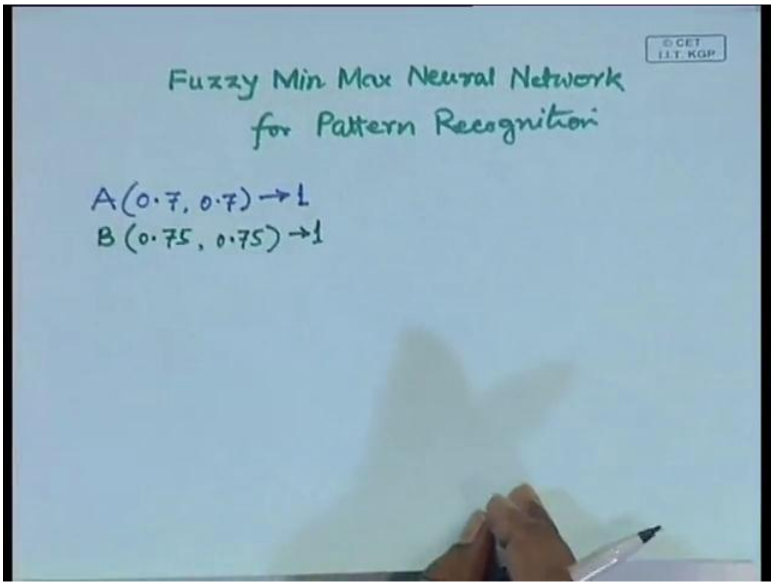
So, matrix V will have two rows. Similarly, matrix W will also have two rows. Similarly, for matrix U also let us assume that we will have two nodes. It depends upon which way I represent the matrix. So, here in matrix U our row will correspond to one of the nodes in the output layer node. The other row we correspond to the other node in the output layer node.

So, from input layer to middle layer a connection V i j will represent a connection from ith node in the input layer to the jth node in the middle layer. Whereas, in matrix U the node the entry u j i will represent the connection from the jth node in the middle layer to the ith node sorry, a connection from u j i will represent a connection from the ith node in the middle layer to the jth node in the output layer. I just said the reverse, jth node in the middle layer to the ith node in the output layer. Accordingly for every node in output layer I will have a row.

So, in this matrices what I have to go on adding is the number of columns. So, for the first point that you considered, this point is 0.7 0.7. So, min point and max point that is there. So, accordingly in matrix V, I will have to introduce 0.7 0.7 because that is the min point. Similarly, matrix W, I have to introduce 0.7 0.7 because that is also the max point.

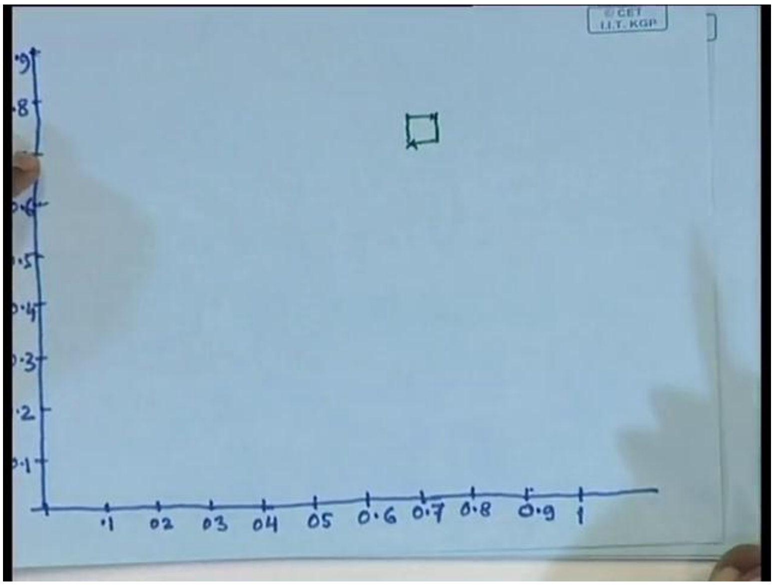
And I have to introduce one node in the middle layer corresponding to this particular feature vector 0.7 0.7. So, the corresponding entries in matrix V and W, I have already established. It is known that this feature vector belongs to class 1. So, as this node belongs to class 1 so the corresponding column in matrix U will be 1 0 indicating that this node is connected to c 1 only. This note is not connected to c 2. Similarly, when I consider the second point.

(Refer Slide Time: 11:10)



The second point that I have is point B which is 0.75 0.75 and this belongs to again class 1, right?

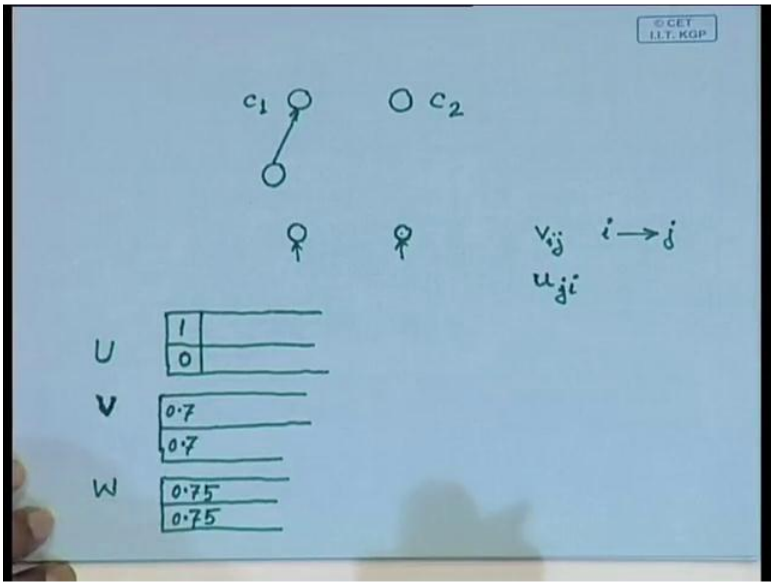
(Refer Slide Time: 11:31)



So, coming over here 0.75 0.75 is this point. This also belongs to class 1. So, I generate a hyper box. Assuming that this expansion is permitted, I generate a single hyper box containing these two points. Now, find that the earlier min point and max point that we heve generated, this min point and max point gets changed and I am not introducing a new node in the middle layer. Only the min and max points I have to change and because I am not introducing any node in the middle layer so the matrix U will remain unchanged, okay?

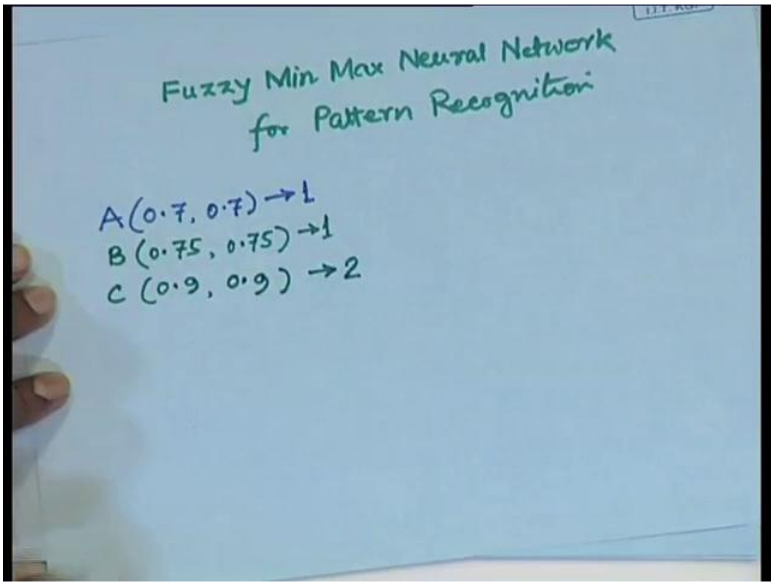
So, what I have to do is over here, the min point is now minimum of these two. In the first dimension the minimum of 0.7 and minimum of 0.75, it remains 0.7. And similarly, minimum of 0.7 and minimum of 0.75, it remains 0.7. Whereas, the max matrix that will be maximum of these two. So, it will be 0.75 0.75, right? So, the matrix W though initially we had set it 0.7 0.7.

(Refer Slide Time: 12:41)



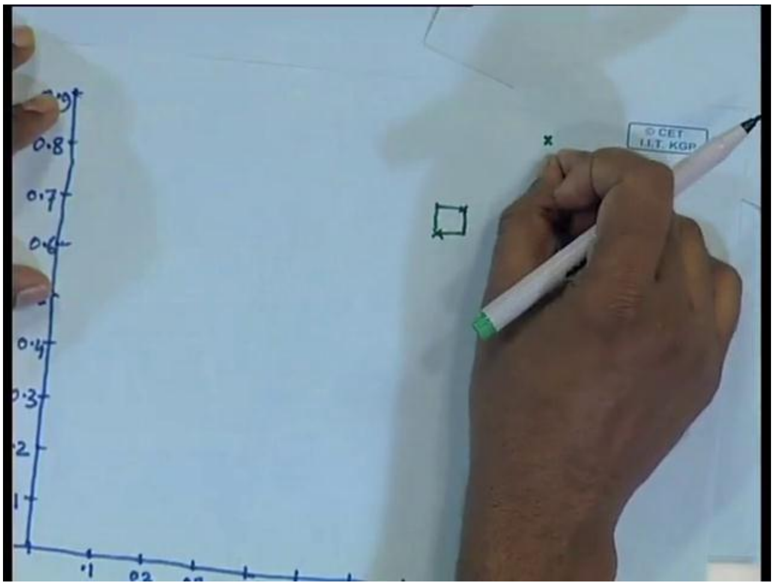
Now, it will be 0.75 and 0.75. So, any time the min point and max point gets changed. I have to change the corresponding entries in matrix V and matrix W. Now, comes the third point.

(Refer Slide Time: 13:02)



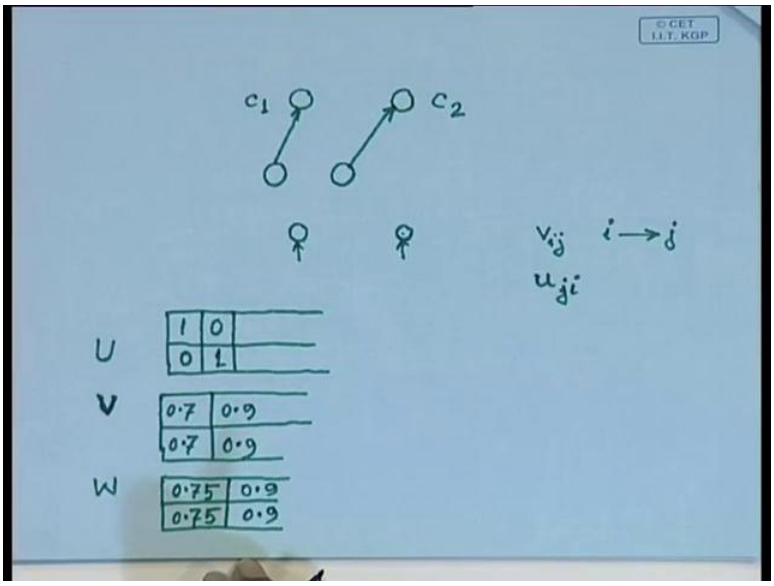
The third point that we considered was point C which is 0.9 0.9 and we said that this point belongs to class 2. So, I will take 0.9, 0.9 over here. So, 0.9 and 0.9 is this one.

(Refer Slide Time: 13:27)



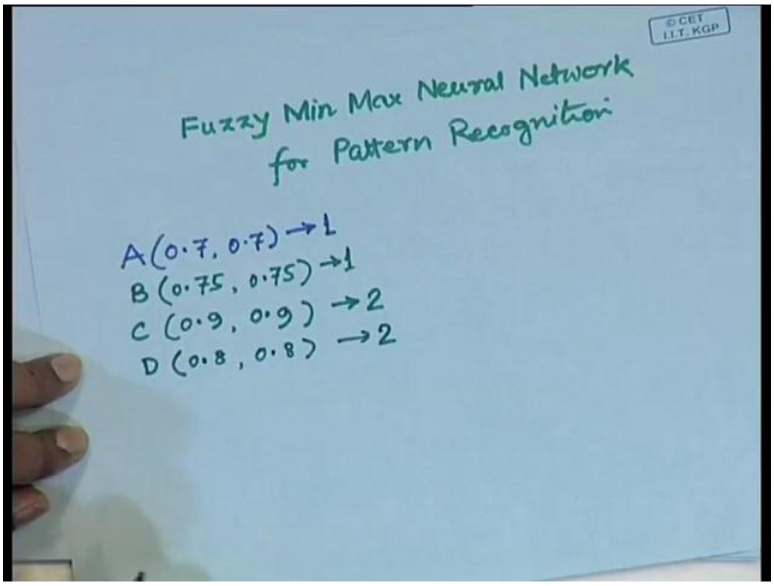
And this belongs to class c 2. So, obviously I cannot expand any of them to include this particular point. And this is the first point, the first feature vector that is considered class 2. So, naturally I have to introduce a new node in the middle layer, right?

(Refer Slide Time: 13:49)



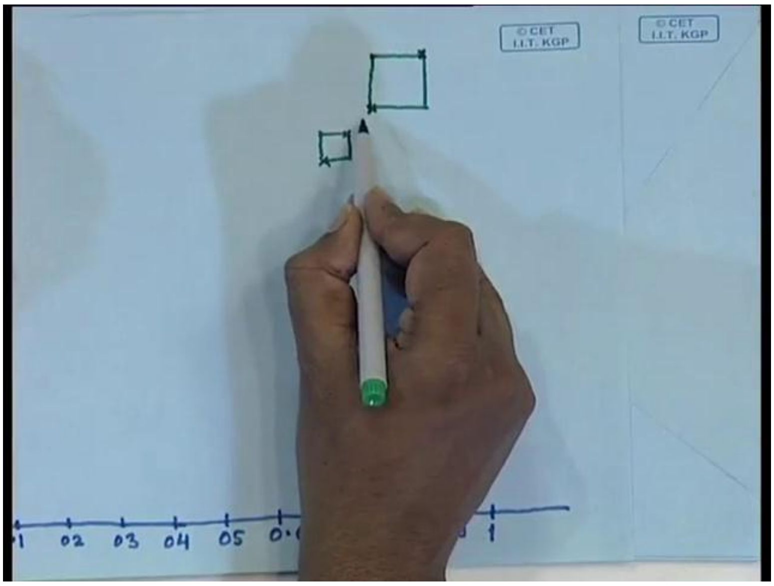
So, I introduce a new node in the middle layer. And as I know that this hyper box will belong to class c 2. So, this will only have a connection from here to c 2. It will not have any connection upto c 1. So, accordingly the corresponding entity in matrix U will be 0 1. Now, I have to set the entries in the other two matrices, that is V and W. Any you find that this is the only feature vector considered so far belonging to class c 2. So, naturally its min point and max point will be the same. And min point max point both of them will be 0.9 0.9. Max point will also be 0.9 0.9.

(Refer Slide Time: 14:54)



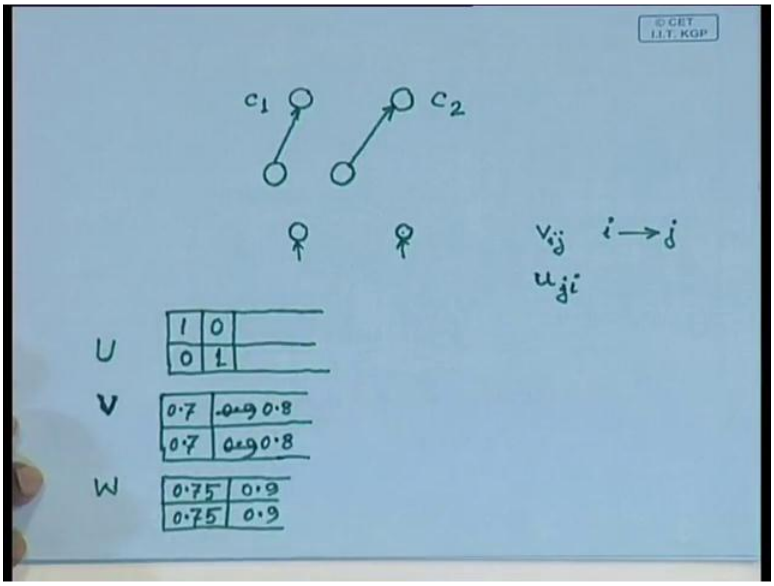
Now, I consider the third point where the third point is D and it is 0.8 0.8 and this also belongs to class 2, right?

(Refer Slide Time: 15:11)



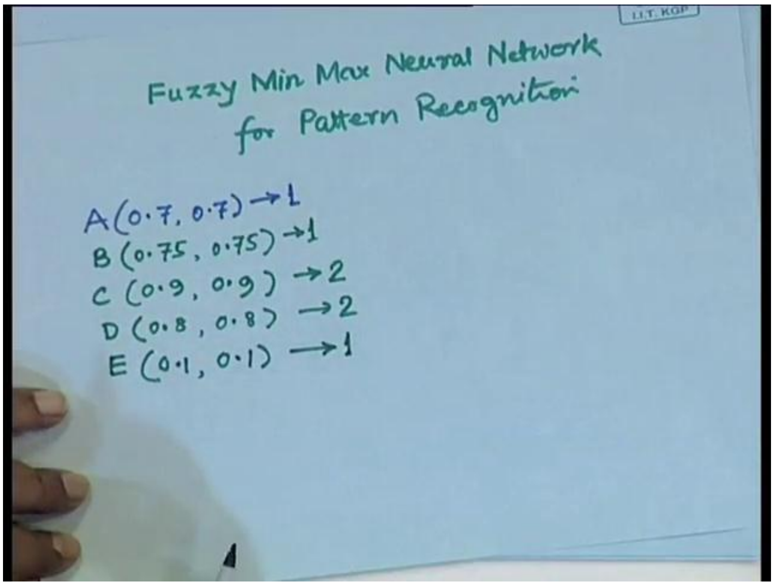
So, corresponding to this I have 0.8 0.8 somewhere over here. And suppose I can expand this hyper box to include this also. The new hyper box that I will get is this one. Now, in this hyper box we find that the max point will remain the same as the previous hyper box whereas, the min point has to change. So, earlier the min point was 0.9 0.9 Now, the min point has to be minimum of these two. So, it will now be 0.8 0.8.

(Refer Slide Time: 15:54)



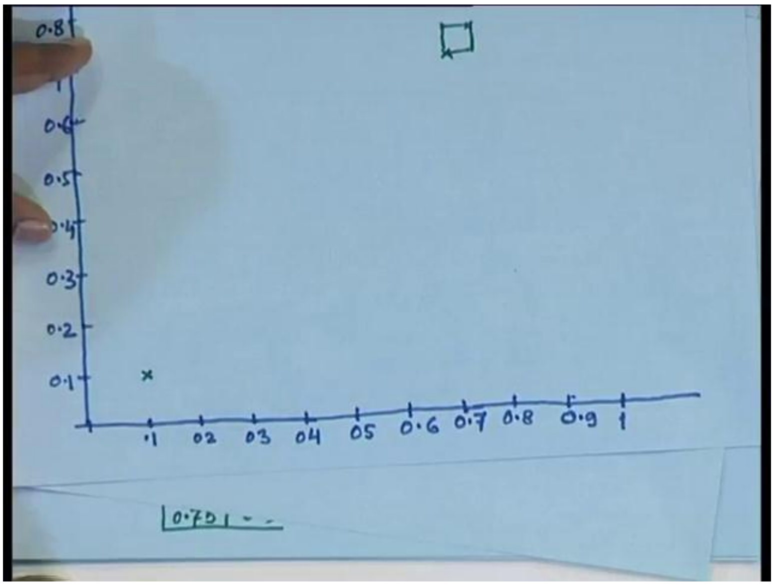
So, earlier this 0.9 0.9, which we had put, this min point is to be 0.8 0.8, okay?

(Refer Slide Time: 16:08)



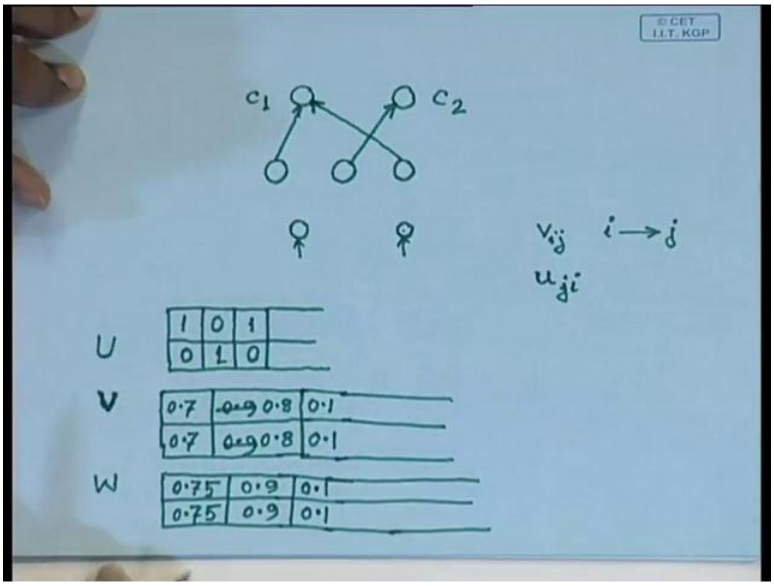
So, now let us take the fifth point E and the vector is 0.1 0.1, and we said that this feature vector belongs to class 1, okay?

(Refer Slide Time: 16:30)



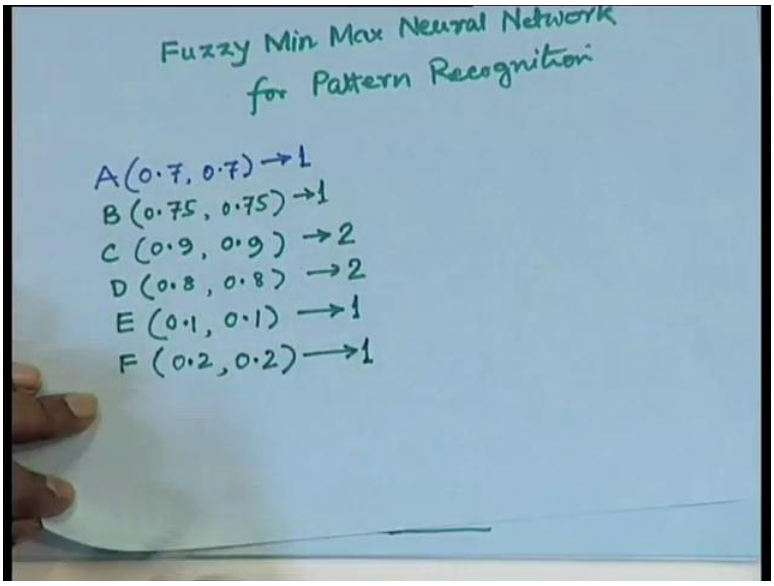
So, for this I will have the hyper box at 0.1 0.1, right? And suppose that this hyper box belonging to class 1, I cannot expand to include this. So, I have to generate a new node for this hyper box and that new node have to generate over here.

(Refer Slide Time: 16:50)



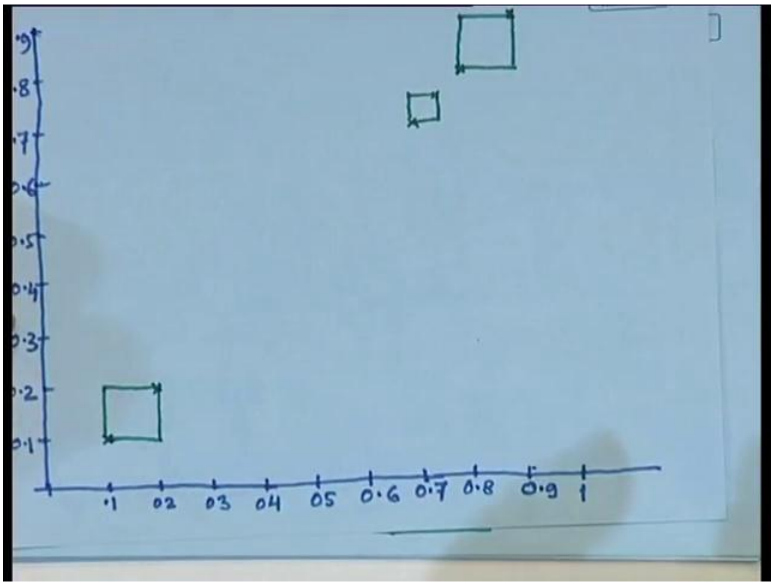
And as I know that this belongs to class c 1. So, the corresponding entry in my matrix U has to be 1 0 indicating that this node belongs to class c 1, right? And then I have to make an entry into the matrix V. I have also have to make the entry into the matrix W. And here again because this is a point hyper box so the min point and max point they will remain the same, that is 0.1 0.1. Here also it is 0.1 0.1. So, then let us consider the other point that is point F.

(Refer Slide Time: 17:56)



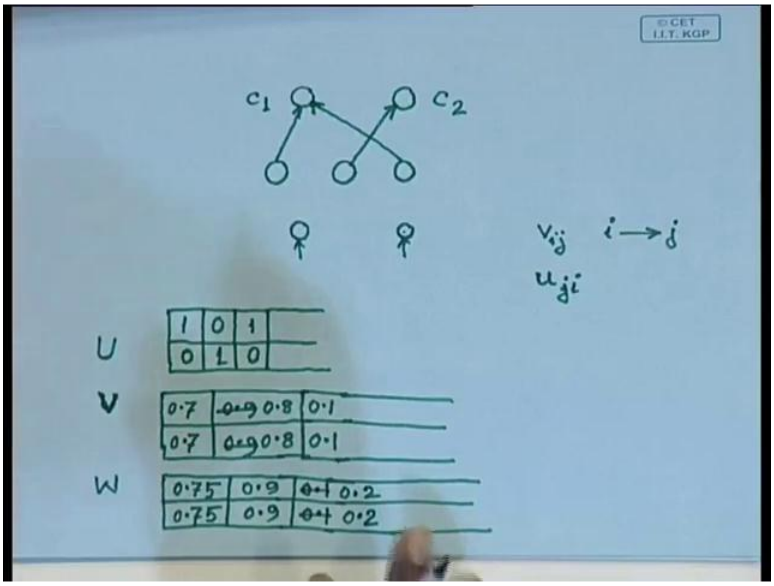
Whose coordinates are 0.2 0.2 and this point also belongs to class 1, right?

(Refer Slide Time: 18:17)



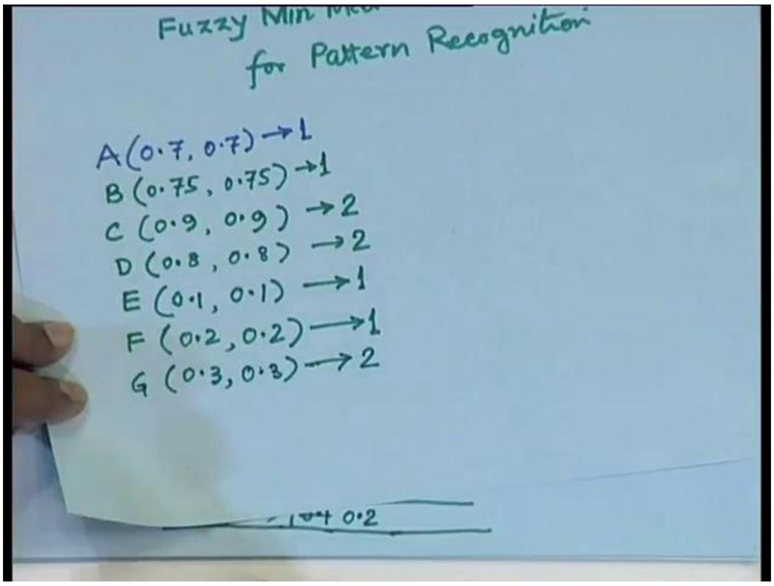
So, over here this corresponds to 0.2 0.2 a point somewhere over here. And assuming that this 0.1 0.1 can be expanded to include this point. I get a single hyper box, and this single hyper box is going to class c 2 and because I am not adding any new hyper box so I do not have to add any node. The previous node that I have only in that node I have to change the min point max point. So, accordingly I do not have to make any entry into matrix U. In matrix V and W change the min point and max point. So, the new min point that remains minimum of these two. So, the min point does not change but max point has to be maximum of these two.

(Refer Slide Time: 19:18)



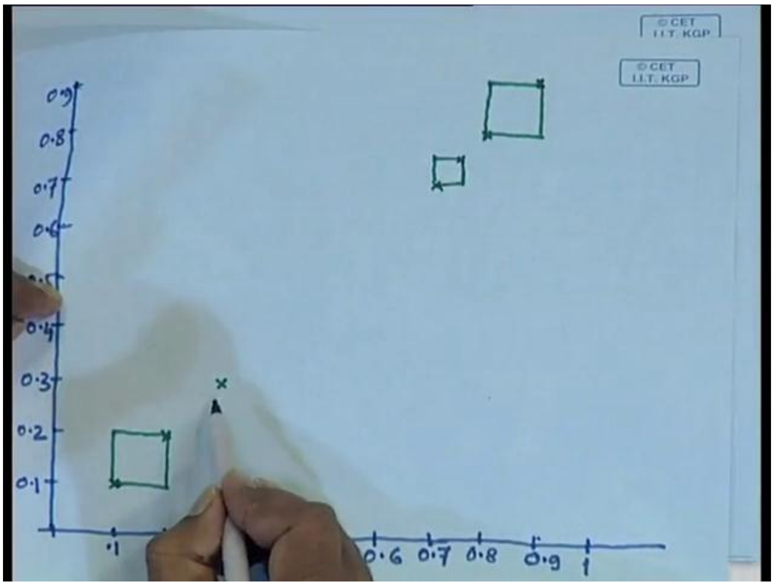
So, the max point has to be changed to 0.2 0.2 from 0.1 0.1. So, the new max point that I get, earlier it was 0.1 0.1 now I have to make it 0.2 0.2. The other matrices remain the same.

(Refer Slide Time: 19:35)



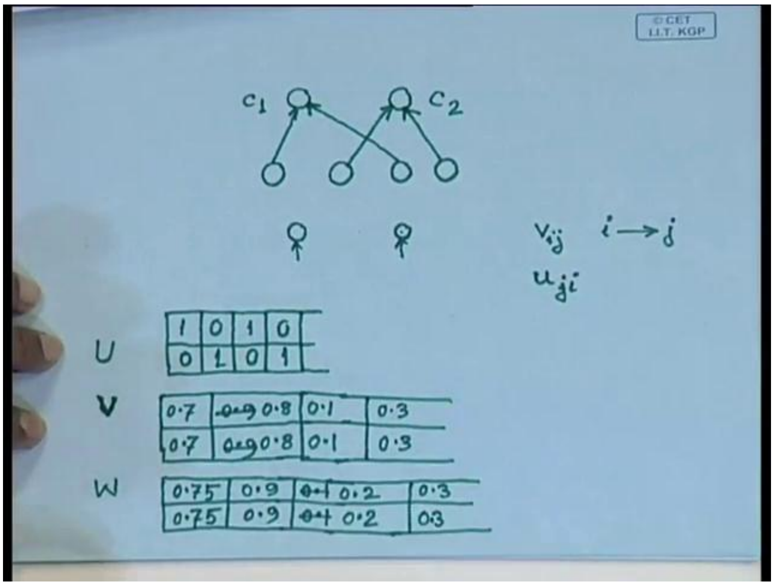
Then let us take the next point, that is G for the vector is 0.3 0.3 and this belongs to plus 2.

(Refer Slide Time: 19:52)



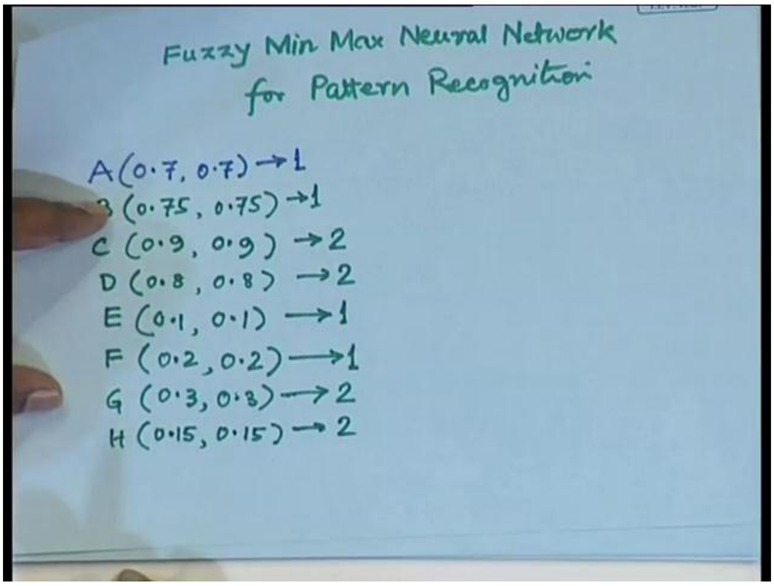
So, 0.3 0.3 if I put over here 0.3 0.3 comes at this location. So, you find that this belongs to class c 2. The earlier hyper box for c 2 was this. Assuming that this cannot be expanded so I have to include this to add a new node in the middle layer, okay?

(Refer Slide Time: 20:13)



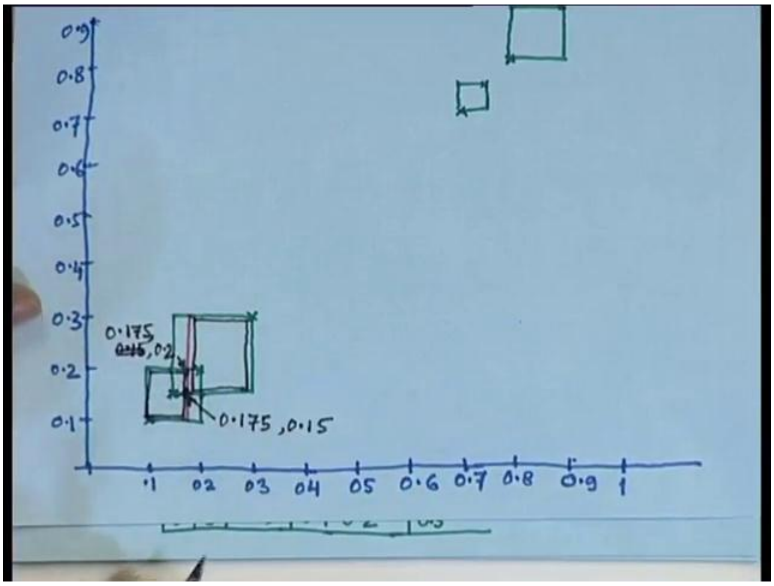
So, the new node in the middle layer comes somewhere over here. And I know that this belongs to class c 2. So, it will only have a connection to c 2. The classifying neurons c 2 and accordingly the corresponding entity in the U matrix will be 0 1. So, this is U matrix. Now, this is a new node and a point hyper box. So, I will have the min point and max point which is same as 0.3 0.3. Max point we will also be 0.3 0.3, the same min point max point.

(Refer Slide Time: 21:07)



Then you come to the next point which is 0.15, 0.15 and this also belongs to class 2.

(Refer Slide Time: 21:23)



coming over here 0.15 0.15 is this, right? And assuming that I can expand this to include this one. So, the new hyper box that I generate is this one. So, this becomes my new hyper box. And now we find that this hyper, this one is hyper box belonging to class c 2. This is hyper box belonging to class c 1.

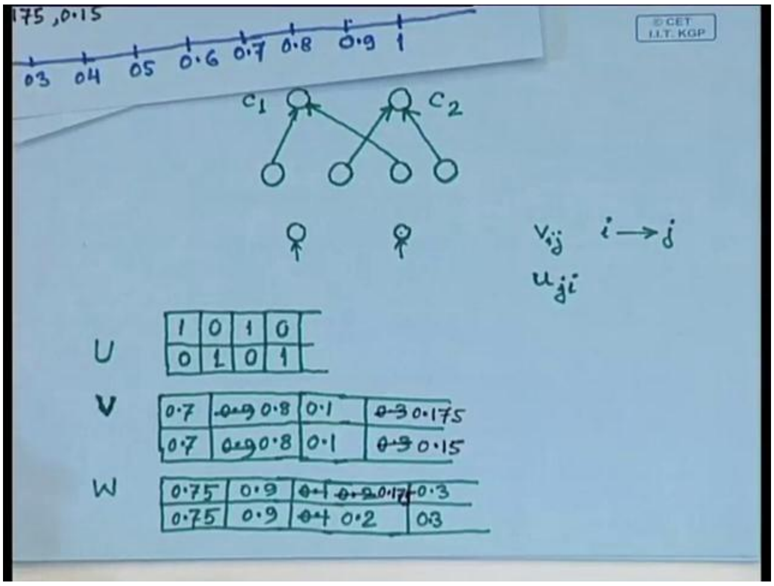
So, whenever I have such a situation, because there is an overlap I have to break this two hyper boxes. And to break these two hyper boxes I have to find out that in which dimension the overlapping is minimum. So, here you find that along the horizontal dimension the overlap is, this 0.15 and this is 0.2. So, the overlapping IS 0.05. In this dimension it is 0.15. This is 0.2. So, the overlapping is 0.05. So, in both the dimensions the amount of overlap is same. So, I can break this hyper box in any of the two dimensions. Let us assume that we break the hyper box along the middle of the first dimension over here. So, when I break this I get two hyper boxes. One hyper box is this one which belongs to class c 1 and the other hyper boxes is this one which belongs to class c 2, right?

Now, over here you find that when I am breaking these two hyper boxes, I am not generating any new hyper box. There was a hyper box corresponding to this and there was a point hyper box corresponding to this. So, it is the point hyper box which has been expanded to include the second point. And while it was expanded there was an overlap because of this overlap I had to contact this hyper box. So, what I have to do is I have to change the min point max point accordingly for this hyper box which was a point hyper box earlier. I have to change the min point and max point for this which hyper box was already existing, right?

So, what will be the min point max point? For this hyper box the min point remains the same, right? What is the max point? Max point is this coordinate, right? And what is this coordinate? This coordinate we be 0.15 and 0.2, sorry max point, max point for this hyper box will be 0.15 0.2 ,right? So, this is 0.15 0.2 and that is the max hyper box. Max point of this hyper box, the modified max point, similarly the min point of this hyper box which is nothing but this location.

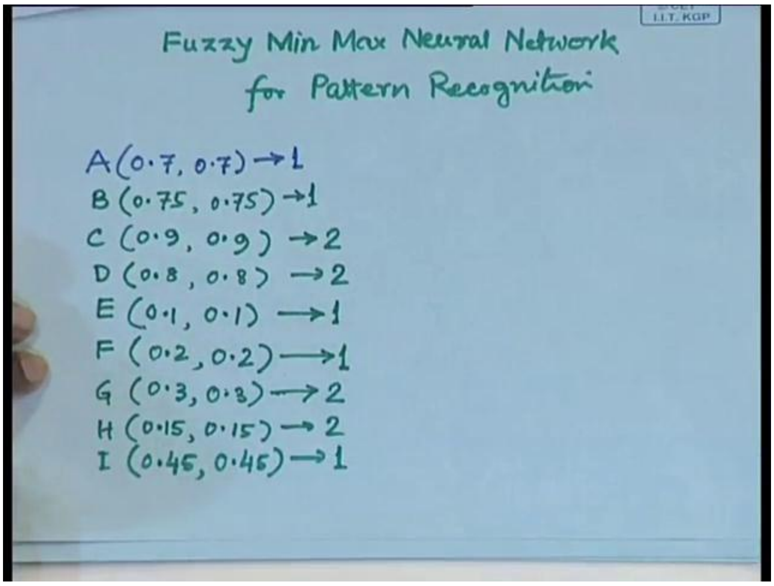
This can be 0.15 and 0.15. Earlier it was 0.15 now it will be 0.175 0.175 and this is why we will remain same as 0.15, right? And this is 0.175 and 0.2. So, the max point of this hyper box is going to change and for this hyper box earlier I had a point hyper box for which the min point and max point was same. Now, the new min point will be this. Max point will remain this. Is it okay? So, coming to our V matrix and W matrix you find that for the earlier one where the min point was 0.1 0.1 and max point was 0.2 0.2. Now, the new max point is 0.175 0.2.

(Refer Slide Time: 26:07)



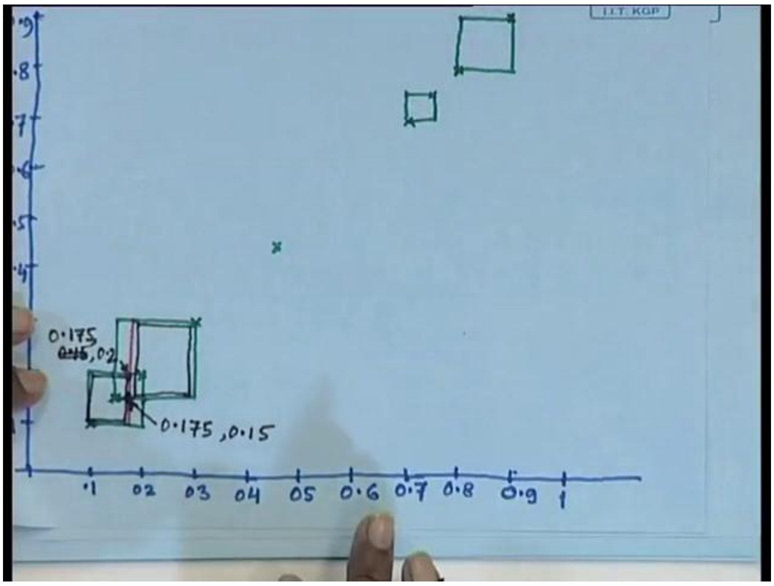
So, this 0.2 now it gets replaced by 0.175. The other point, the other dimensions remains as 0.3, right? For this one 0.3, 0.3 which was a point hyper box. For this point hyper box the max point remains the same. But the min point becomes 0.175 and 0.15 is that. So, the min point will be for this will be changed to 0.175 and 0.15, is it okay?

(Refer Slide Time: 27:05)



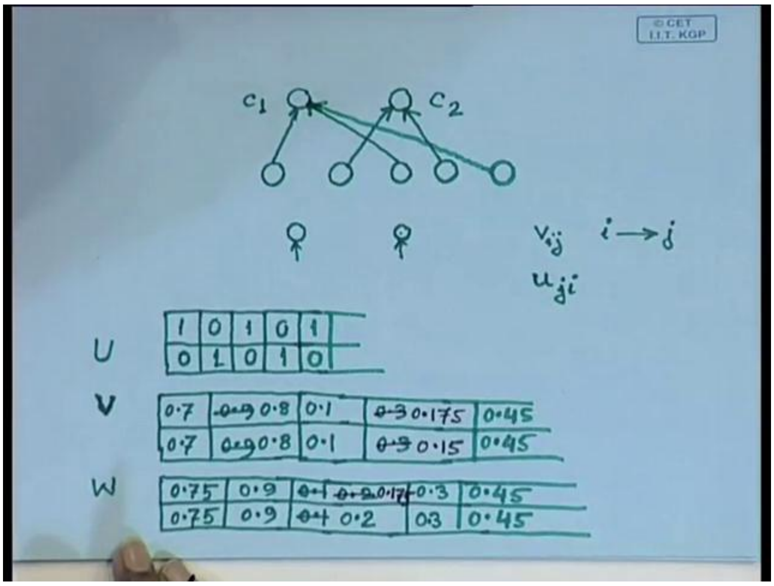
Now, let us take the next point. The next point is I for which the vector is 0.45 0.45 and this belongs to class 1, okay?

(Refer Slide Time: 27:27)



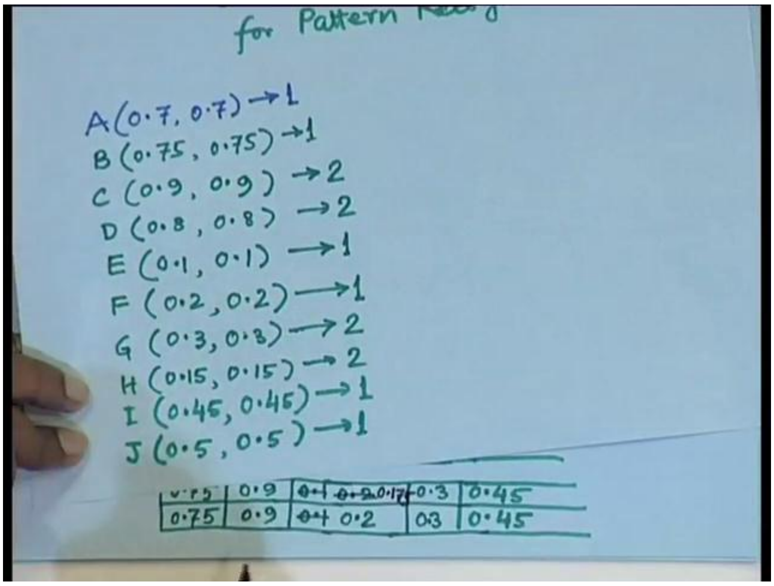
So, for this I have 0.45 and 0.45. This belongs to class 1, and suppose this cannot be included in any of the hyper boxes after expansion. So, I have to make a new node for this particular point whose corresponding to which vector is 0.45 0.45.

(Refer Slide Time: 27:48)



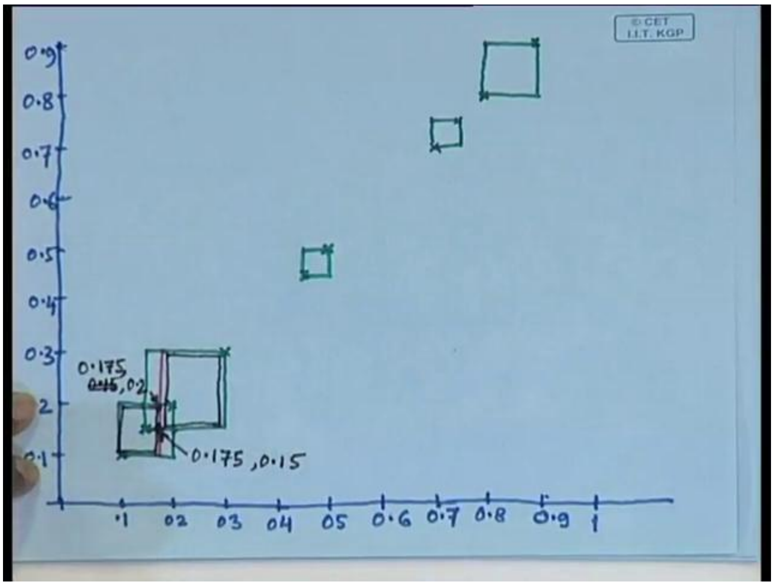
So, I generate a new node and I know that this node belongs to class c 1. So, accordingly had to make a connection from this node to class c 1 and correspondingly matrix U will be modified as 1 0. And I have to enter the min point and max points in this two matrices. And because it is a point hyper box so the min point and max point will remain the same. So, both of them are 0.45 0.45. Here also it is 0.45, here also it is 0.45, okay?

(Refer Slide Time: 28:50)



So, when I consider the next point that is point J corresponding to this the feature vector is 0.5 0.5 and this also belongs to class 1. If I come over here 0.5 0.5, that is this, okay?

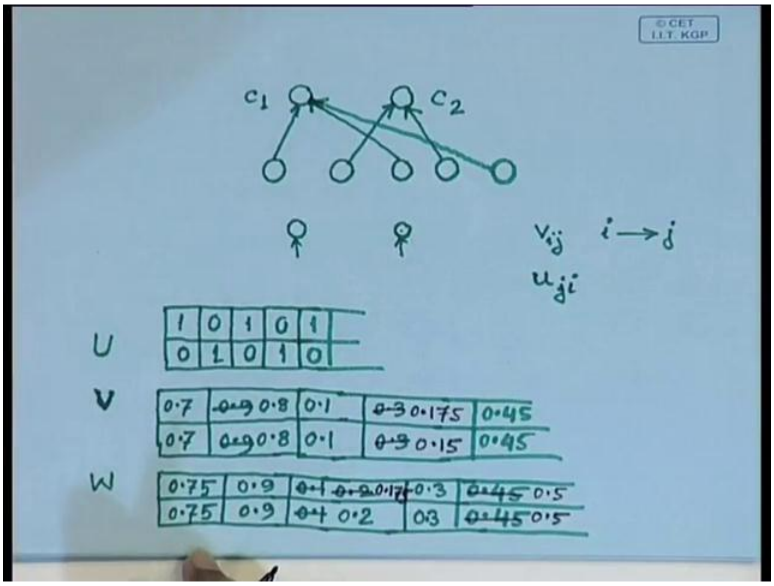
(Refer Slide Time: 29:09)



And assuming that this point hyper box can be expanded to include this. So, I generate a new hyper box and because I am not adding any new node because the previous point hyper box has been expanded to include this. So, U matrix will remain the same.

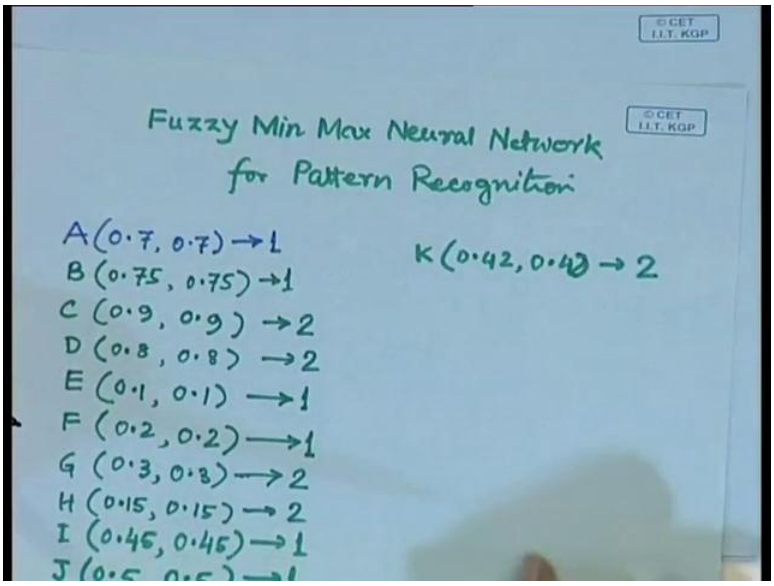
However, I have to look at the V matrix and W matrix. And here you find that the max point that is being changed. So, earlier max point was 0.45 0.45. Now, the new max point is 0.5 0.5, okay?

(Refer Slide Time: 29:48)



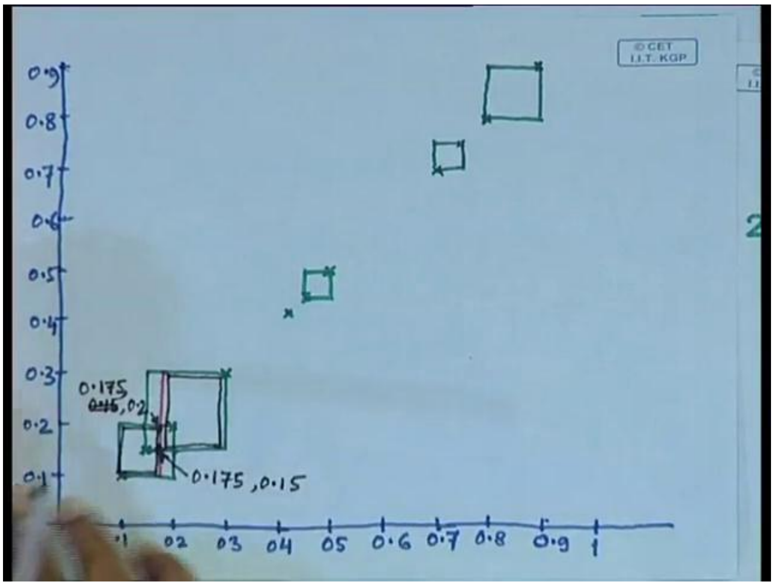
So, I have to change this corresponding entry in matrix W. So, it will be 0.5 0.5, okay? Then let us consider the next point, the next point is K.

(Refer Slide Time: 30:07)



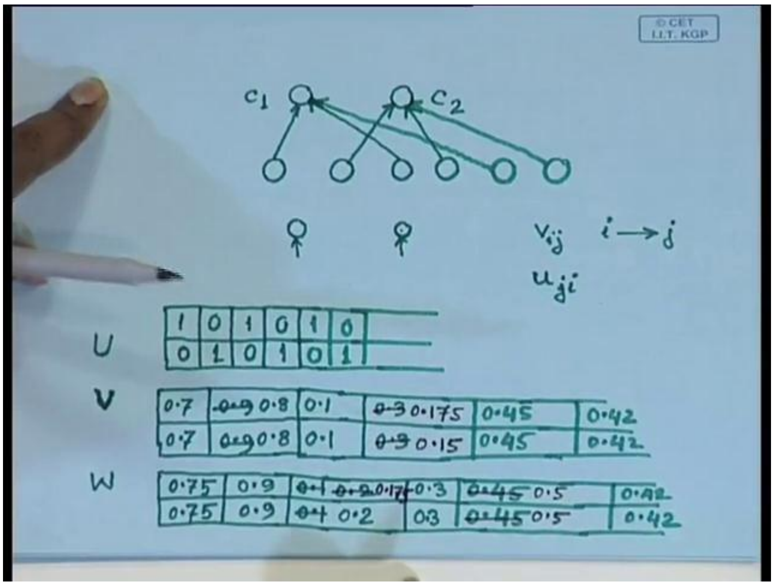
The feature vector is 0.42, 0.42 and this belongs to class 2. So, I come over here. This is 0.45, 0.45. 0.42, 0.42 will be somewhere over here.

(Refer Slide Time: 30:35)



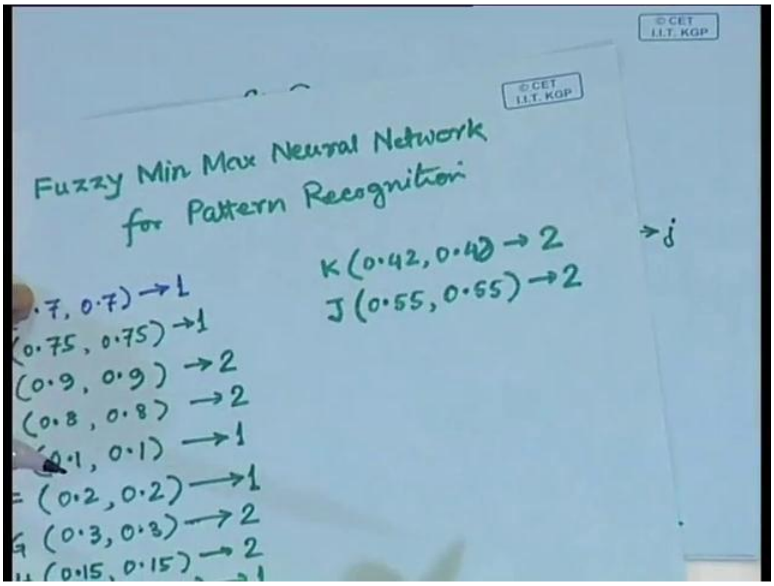
Somewhere over here 0.42, 0.42 and assuming that this has to be point hyper box I have to have a corresponding node in the middle layer, okay?

(Refer Slide Time: 30:51)



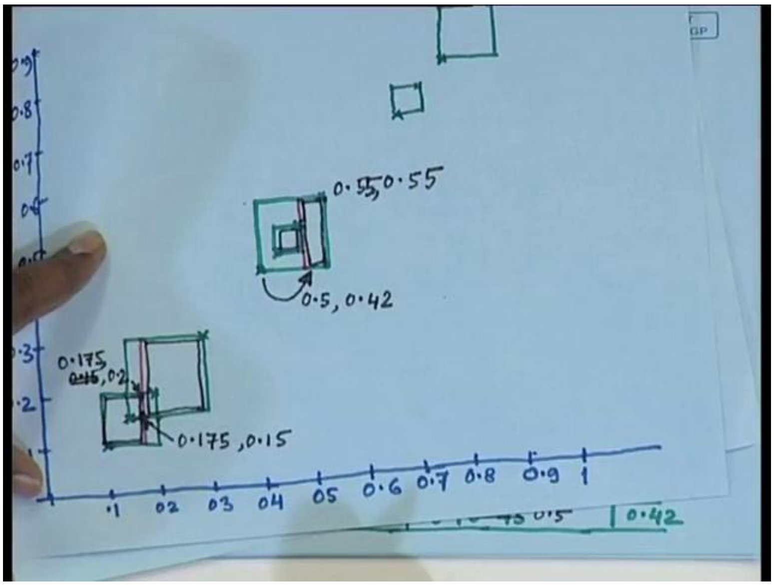
So, have to have a node in the middle layer and as I know that this node belongs to class c 2. So, I have to change the corresponding U matrix. So, I have to make a column, add a column in the U matrix which will be 0 1 because this belongs to class c 2. And in the U matrix, V matrix I have to generate, I have to add a column corresponding to this particular point, so this 0.42, 0.42. Here also it is 0.42, 0.42. Then I take the next point.

(Refer Slide Time: 31:43)



The next point is O.55 0.55 and this belongs to class 2. So, according to this I have over here 0.55, 0.55 somewhere over here.

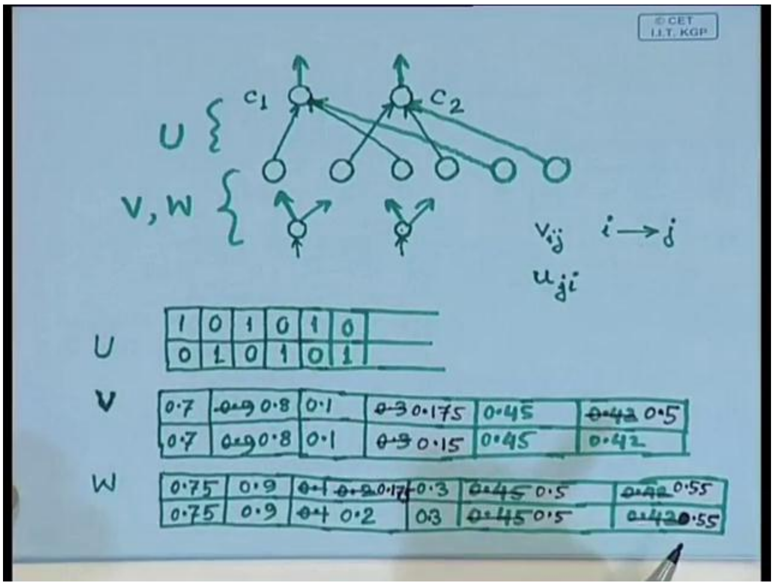
(Refer Slide Time: 32:02)



And assuming that this can be expanded to include this I have this new hyper box, the modified hyper box which comes like this and now we find that there is a containment. So, because there is a containment I had to contract this hyper box. So, this new hyper box we be contracted. Let us assume that we contract it from this location. So, in this side I have one hyper box belonging to class c 1 and on this side I have another hyper box belonging to class c 2. So, over here you find that this hyper box was already existing and for this hyper box and min max point has not been changed.

So, accordingly the earlier entry in our matrix V and W matrix that remains the same, that is 0.45 0.45, 0.5 0.5. This remains unchanged, right? Whereas, for this new hyper box I had earlier the min point, which is 0.42 0.42 that corresponds to this particular point. And when I have expanded this to include the second point and because of this expansion there was a containment and due to this containment I had to contract this. So, as I contracted this hyper box the min point has now changed from here to here, right? And this is my max point. So, what is this new min point? The new point is 0.5 0.42 that is the new min point. And the max point is 0.55 and 0.55. So, accordingly I had to make the modification over here.

(Refer Slide Time: 34:40)



Now, the min point becomes 0.5 0.42. So, this will be 0.5. 0.42 as it is and the max point which earlier was 0.42 0.42 because it was a point hyper box. Now the new max point will be 0.55 and 0.55, that is this one. So, this max point, new max point becomes 0.55 and 0.55.

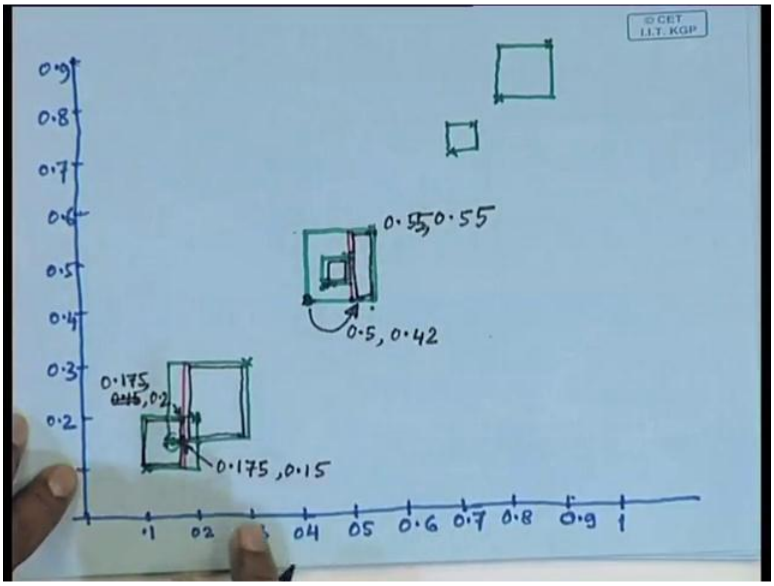
So, with this I have included all the training points that were given. So, at the end of the training, the three matrices that I have generated are like this. So, I have the new matrix which represents connections from the middle end nodes to the output line nodes. And I have two matrices V and W corresponding to min points and max points of the nodes which are generated in the middle layer. So, I over here these are represented by the matrices V and W. And over here the connection is presented by matrix U and I have the final memberships computed to the two different classes c 1 and c 2 which comes from the output layer units, is that okay?

So, we find that your network has been obtained in a single pass unlike in case of multi layer perceptron or single layer perceptron. What we have done is, we have changed the weight vector or the weight matrices iteratively every time a given feature vector becomes misclassified then following the gradient descent approach or following the back propagation learning we had to change the weight matrices for different layers. And once it is changed I have to re-check with all the samples all the feature vectors which was used to train the neural network before this. That means I have to come to conclusion that the multilayer perceptron or single layer perceptron has been trained properly only.

When in a pos all the samples are correctly classified, all the training samples are correctively classified or the error that you get is less than certain specified value or less than the tolerable value so that any algorithm of the neural network in such cases was an iterative algorithm. Whereas, in this case the neural network is restrained in a single pass because every time.

And what I am doing is if there is an overlap immediately contacting the existing hyper box, which takes care of the fact that there is no overlap or no ambiguity during classification or during testing, however as we said before that the problem with this fuzzy min max neutral network is that as we are going for contraction.

(Refer Slide Time: 38:07)



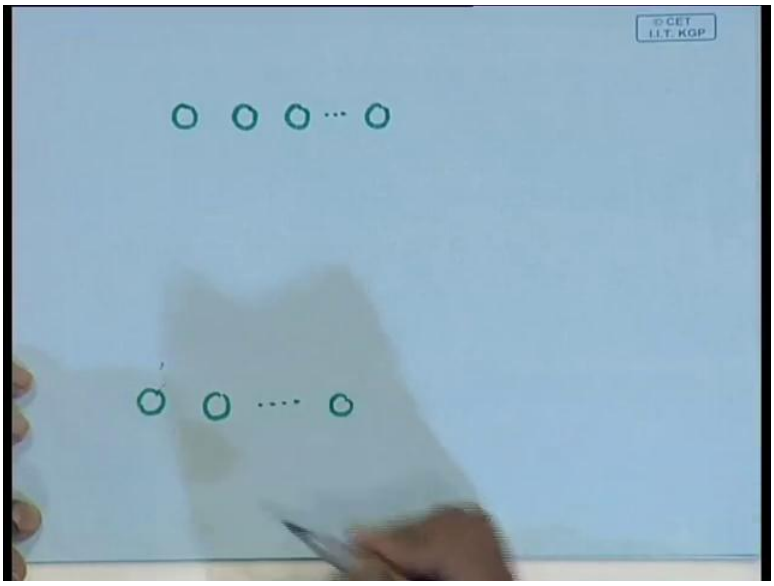
So, here to find that say this point which we know that these belongs to class c 2. But because of overlap and contraction finally this point has been classified into class c 1. Because now if I after this contraction operation if I present the same point to this neural network, the neural network will say that this point belongs to class 1, though I know that this point belongs to class c 2. Similarly, for this I know that this point belongs to class c 1. But after creation of this hyper boxes and contraction of these hyper boxes, if I present the same point to the neural network it will say that this black point belongs to c 2. Though I know that this point belongs to class c 1, okay?

So, that is the disadvantages of this kind of neural network. And such a kind of problem you get both over here and over here. But earlier we know that this point belongs to class c 1 but after creation of this hyper boxes and contraction. Now, if I present this point to the neural network, what it will do? Is this obvious point? It will try to calculate the fuzzy membership function to this hyper box and also the fuzzy membership function to this hyper box. And now you find that this point is nearer to this hyper box, then this hyper box. So, the fuzzy membership function for this hyper box will be more than the fuzzy membership function for this hyper box.

So, naturally if I go for hard classification. This point will be classified to class c 1, though you know that it actually belongs to class c 2. So, after this expansion and contraction process you say that the training samples which are used to train the classified, train the neural network those samples themselves will be misplaced or some of those samples themselves will be misplaced.

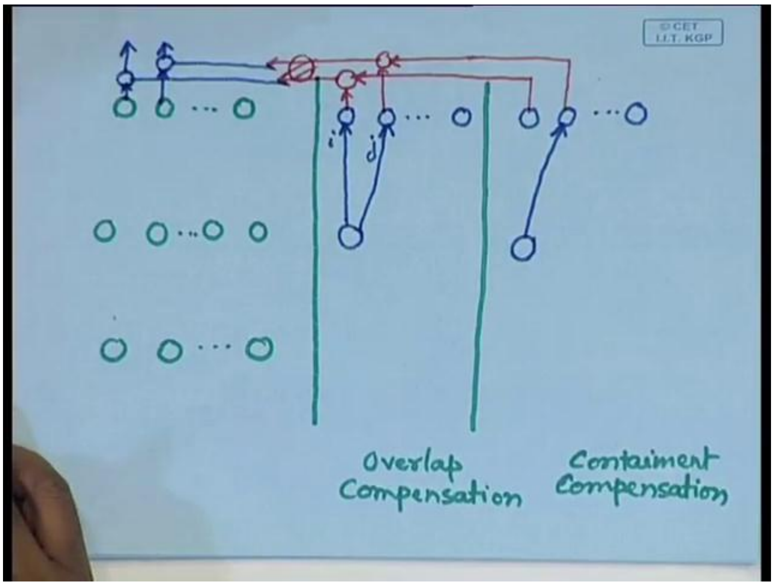
So, naturally due to this contraction we are introducing some amount of error in our learning process. We have suggested some modification to this kind of algorithm, that is by introducing the concept of compensatory. So, what we have done is the basic structure or the classified section of this neural network called reflex neural network remains the same as what has been suggested by practical assumption, that is fuzzy min max neural network, okay?

(Refer Slide Time: 40:44)



So, in the new architecture I have as before a number of classifying neurons in the output layer. A number of input layer neurons. Let me draw in another paper.

(Refer Slide Time: 41:12)



So, I have a number of classification neurons. I have a number of input layer neurons. Class is same as the number of classes we are considering. The number of input layer neurons is same as the dimensionality of the feature vector that we are considering. So, the major part of this reflex fuzzy neural network is same as the fuzzy min max neural network which has been suggested.

And we have this middle layer nodes which are basically hyper boxes. And as we have just seen that these number of nodes in the output layer and the number of nodes in the input layer they are fixed by the problem. Whereas, number of nodes in the middle layer that we create during the training process and accordingly we generate matrix U, matrix V and matrix W.

Now, what we have suggested is, we have suggested two more sections in this neural network which is called the compensation section, and in compensation section again as we have just seen that we can have two types of situations. One is overlap situation and the other one is containment situation and the kind of compensation that you have to impart is different in case of overlap compensation. Overlap compensation we have to give some sort of compensation.

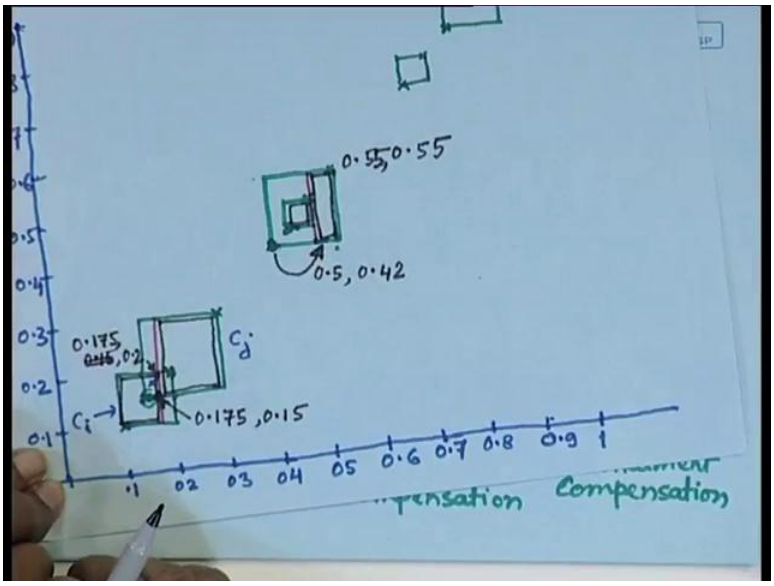
In case of contentment we have to give some other type of compensation. So, accordingly this compensation section that we have generated, this compensation section is again divided into two parts. One of them we are calling as overlap compensation, the other one is called contentment compensation. So, I will have some additional nodes. One in the overlap compensation section. This we call as overlap compensation and the other one is containment compensation.

So, what is done in this overlap compensation and containment compensation? Whenever there is an overlap like here while training if we find that there is overlap in the previous case, what we have done is we have contacted the hyper boxes to know that problem. Now, we do not go for contraction. But we go for the gradation of the membership function within that overlap. So, whenever there is overlap one new node is introduced in the overlap sections.

So, over here in the overlap section we will introduce a new node and this new node will give one compensation to the classifying neurons. Corresponding to this hyper box and another compensation to the classifying neurons. Corresponding to this hyper box because I cannot get the membership of only one hyper box. I have to get the membership of both the hyper boxes. Because if a point falls within this region, the overlap region normally or classifying neurons. In this section, classification section both of them will give membership function to this class and membership function of one to this class, okay?

I have to modify both of them or compensate both of them. So, this new neuron that we are introducing in the overlap compensation section that will actually generate two outputs. One output is meant for compensation of one class and the other output is for compensation of the other class.

(Refer Slide Time: 45:36)



So, if I have class c i and class c j, the hyper boxes belonging to c i and the hyper boxes belonging to class c j boxes there they overlap. Then output of the compensation neuron that will give compensation to both c i and c j and amount of compensation to c i and the amount of compensation to c j will be different. The reason is if this point is nearer to this point then the membership function to c i to c j should be more. And if this point is towards this, than the membership function to plus c i should be more, okay?

So, the amount of compensation that you give to plus c i and the amount of compensation that you give to plus c 2 or c j, they cannot be same. They have to be different and while calculating this compensation values you have to take care that, to reach of the points your unknown point is nearer. So, what you are doing is, in this section also I will have a number of classifying neurons or compensation neurons which is same as the number of classes that I have. This will get output from the input. This will get input from the input layer neurons and output of this will actually go to two of the output layer neurons because I have to compensate for ith class as well as the jth, okay?

Similarly, in the containment section also I will also have a number of output layer neurons which is same as the number of classes I have. Here again whenever there is a containment like this I will put containment compensation neuron. Now, why I said that the amount that the type of compensation that have to give in this case is different from the type of compensation that you have to given. In this case is that, here I said that I have to compensate the outputs, the membership functions of both class c i and class c j, right?

Whereas, in this case if any feature point lies over here we are saying that it should get a membership value of 1 to this class. Whereas, the membership value to this class has to be 0. And my hyper box in the classifying section that itself will compute membership value of 1 to this class. It will also get a membership value 1 to this class.

I have to make the membership value to this class equal to 0. And the membership value to this class should remain unchanged. That means I have to only compensate the membership value of one class which is containing the hyper box of another class. So, the hyper box that is contained, the membership value of that should not be changed. However the hyper box which contains the other hyper box, the membership value of the class of the hyper box that contains that should be made equal to 0. So, accordingly output of a compensation neuron in the containment section should go to only one class, that is the class of this particular hyper box.

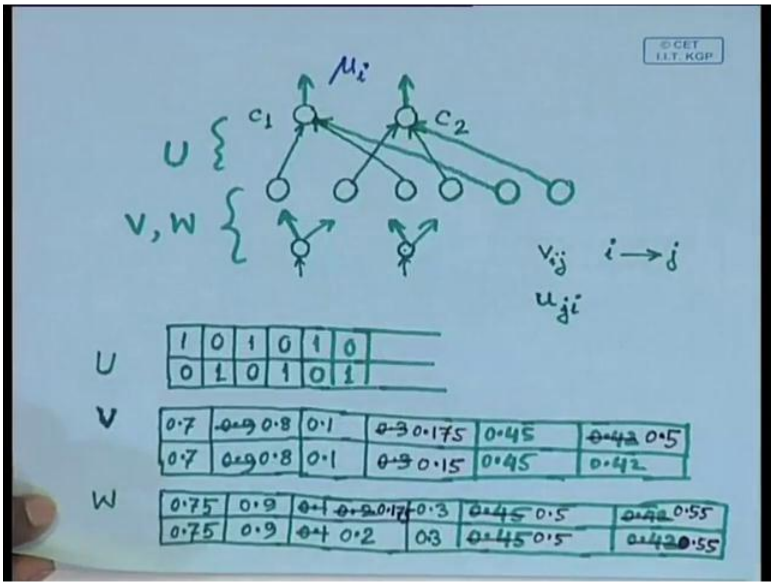
It should not go to the membership function. It should not compensate the membership function of this class. So, here in the overlap compensation section the output of this compensating neuron goes to two classes. Whereas, in this case the output of the containment compensation neuron will go to only one class, right? So, I get two types of compensation; one from the overlap compensation neurons and the other one for the containment compensation neurons. Now, I have to combine these two compensation.

So, I have to have a combiner, right? So, over here, that belongs to the class of smaller hyper box, because that defines its class membership more vigorously than the cost membership of the other one. So, I will have to combine the outputs shuttle have to combine the outputs of this compensation neurons. So, I have to have a number of combiner, okay?

So, I will combine the output of this and the output of this. So, these are outputs of the output layer. Neurons of the compensation class belonging to the same class. So, this is class c 1. This is class c 1 you combine these two. Similarly, class c 2 class c 2 you combine these two. Finally, after combination combining these outputs I know that what is the overall compensation that had to give to the compared to the membership values generated by these neurons. These neurons are generating the membership value as if there is no containment or there is no problem. We are trying to modify these outputs, these membership values as computed by this compensation values, okay?

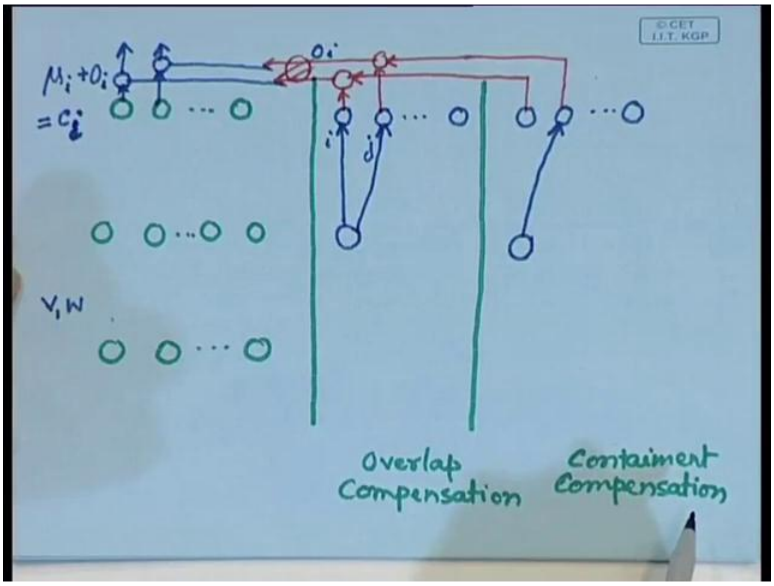
So, what I have to do is, I have to add these values to these membership values. So, this will come over here, this will come over here and then finally I get the final membership value. So, we find that in case of this neural network without any compensation section if the membership value to a particular class the ith class was mu i.

(Refer Slide Time: 52:29)



In this case the membership value after this compensation to ith class will be mu i plus o i, where o i is the combination of the outputs of the overlap compensation neuron and outputs of the compensation neuron.

(Refer Slide Time: 52:43)



So, over here I will have this output as o i, okay? So, the final membership value will be mu i. mu i is what is computed by these neurons. I have to modify this with the outputs of the compensation neurons of the final membership value becomes mu i plus o i. And that I can take as the final membership value say C i.

Two class to the ith class for any of the unknown feature vectors which have been put up in the neuron. So, when this compensation section neurons whether it is overlap compensation or containment compensation, the computer compensation values they will make use of the same V and W matrices, the min points and max points that we have generated for these hyper boxes.

So, they will make use of the same min point and max point to compute what should be the compensation given. And this composite compensation computed only when an unknown feature vector falls within either the overlap region or the contract region. This will not be computed if the unknown feature vector falls outside the overlap region or falls outside the containment region. And when it get this the final membership value your classification will depend upon this final membership value. So, the classification instead of doing based on mu I it will be based on c i, okay?

So, in the next class we will see the details of the membership computation or the compensation computation as done by this overlap compensation neurons and the complete containment compensation neurons. So, we will take the same example to design our reflex for the min max neural networks with. I will take the same example because I have the case of overlap I also have a case of containment. So, with that we will illustrate that how this neural network with compensation neurons that can also be generated. So, we will stop here today.

Thank you.