Pattern Recognition and Applications

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

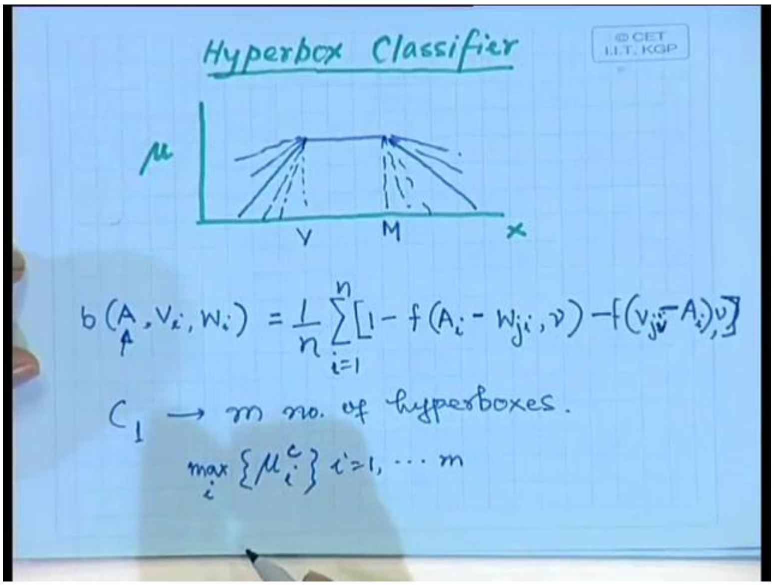
Lecture - 31
Hyper box Classifier (Contd.)

Good morning. So, today we will continue with our discussion on hyper box classification. So, in the last class, what we have said is if the feature samples the feature vectors belonging to different classes, they are mixed in an arbitrary manner. And then we cannot really design a linear classifier or a quantity classifier to classify different samples to different classes.

So, in such kind of situation, the kind of classifier that we have talked about is hyper box classifier. We divided the feature space with the number of hyper boxes. The different hyper boxes will be given different task levels. Then when an unknown sample comes if the unknown sample falls one of hyper box is well within hyper box, which belongs to particular class, then this unknown sample will also be classified as belonging to the same class. The task level is given to that particular hyper box.

If the unknown sample falls outside the hyper boxes outside any of the box, then find out what is the distance of this unknown sample from the hyper boxes. Based on the distance, we can give off fuzzy membership grade. To whichever class the fuzzy membership function is maximum, we can classify the unknown sample to that particular task.

(Refer Slide Time: 01:51)



For that, we have said yesterday that offers membership function or trapezoidal membership function that can be employed, which the direction of the feature. On this side, I put the membership value, which is made under the fuzzy membership function. If I define something like this, a trapezoidal fuzzy membership function for these are the min points and max points because the hyper box can be represented by a min point and a max point and accordingly it is called min max hyper box.

So, if I have a fuzzy membership function, a trapezoidal fuzzy membership function something like this, we have given an analytical expression for the sort of fuzzy membership function. So, we can give an analytical expression of the fuzzy membership function like this for A is input sample. The vector vi presents the min point in the ith dimension and wi represents the min point the max point in the ith dimension. Comma is a parameter, which represents the degree of fuzziness. So, because the slope of these lines that depend upon the value of comma, so if the value of comma is more, then the fuzziness will be less. It will be more towards hard classification.

This is because if I increase the value of comma, I will have these slanting straight lines. If I reduce the value of comma, then I will have the straight lines something like this, which influence the fuzziness. So, this comma actually controls the degree of fuzziness. So, you can compute given a mini max hyper box where I know the min point vi and the min point of w. This is what I have given on the membership calculation. The membership calculation only the ith dimensional shows. Such membership competition has become among all the dimensions added together and then normalize with respect to n.

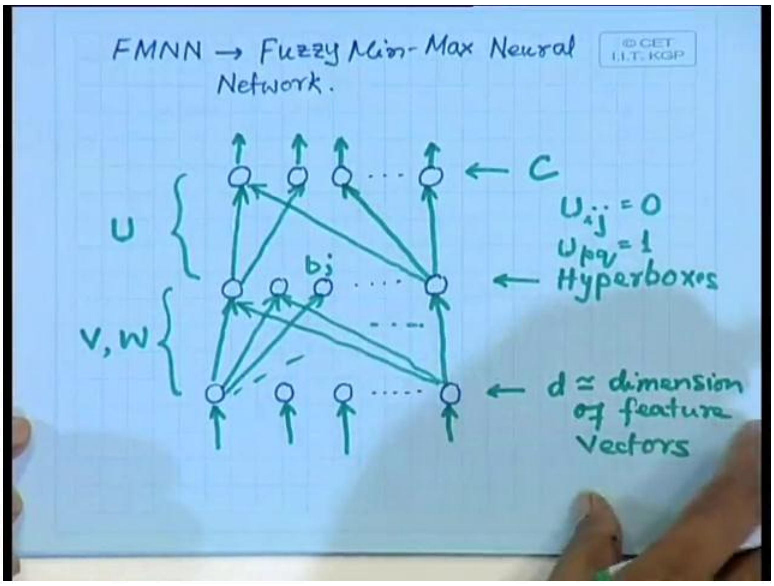
So, once I have this sort of analytical expression for competition of the membership, I can compute the membership of an unknown sample to different hyper boxes or because in this case I will have for every class and number of hyper boxes. So, considering a particular costs c1 has say m number of hyper boxes, so class c has m number of hyper boxes. So, accordingly for each of this hyper box, I will have a membership function of this 1. So, there will be mu i for class c. If I put it like this, it is mu i c for i varying from 1 to m. As I have m number of hyper boxes, so for each of these hyper box, I will get a membership function.

So, I will have m number of membership function like this. The maximum of this maximum over i gives me what is the membership function of the hyper box belonging to class c 1, which is nearest to this unknown sample. This is because obviously a hyper box, which is nearest to unknown sample that hyper box will get the maximum of membership value and hyper box, which is what this unknown sample that hyper box with the minimum membership value.

So, I take the maximum of these m number of membership values given by m number of hyper boxes belonging to class c. This maximum value is the membership value of this unknown sample to class c 1. So, what I have to do is I have to compute membership function with all the hyper boxes. Then particular costs are to take a maximum of the membership functions and that gives the membership of value to that particular class.

So, that way when I get the membership values to all of the classes, whichever class gives the maximum value, I classify this unknown sample x to that particular boxes. So, this is how this hyper box classifier has to work. Now, in the year 1992, Patrick Simpson said that this concept can be converted in the form of a neural network. So, accordingly neural network architecture is fuzzy min max neural network.

(Refer Slide Time: 07:56)



So, FMNN or fuzzy min max liberal network for pattern classification was proposed by Patrick Simpson in the year 1992. So, what is this concept of fuzzy min max neural network? The fuzzy min max neural network has got 1 input layer and 1 output layer and 1 so called hidden layer. Now, this output layer is same as the number of classes that we have. So, this is equal to the number of notes in output layer is same as the number of classes.

So, if I have c number of classes, there are c number of notes output layer. Similarly, number of notes of the input layer is same as before that is equal to the dimension of the feature vectors. So, this one number of notes in the input layer will be d, which is nothing but dimensionality of features, feature vectors. The hidden line notes actually represent hyper boxes. It is the responsibility of the hidden line notes to compute this membership function.

So, if the hidden line notes is to compute this membership function, then the hidden line notes or every hyper boxes saying that these hidden line notes represent the hyper boxes. They have to know what corresponding min point is and max point to the min point. The max point is represented in the form of connections. So, the connection between the input layer notes and the hidden layer notes will be represented by 2 matrices corresponding to the min point v and max point w.

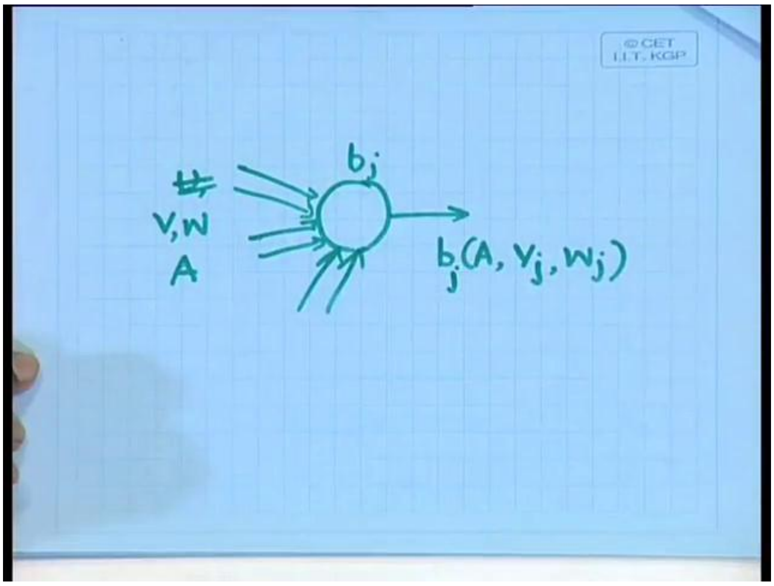
So, I will have all these connections from input layers to the middle hidden layers for something like this for these connections actually represents the min points and the max points. This connection weights are represented in the form of 2 matrices; 1 matrix corresponded to the min point and 1 matrix corresponding to max point. The connection from the hidden layer notes to the output layer notes with different classes.

So, if hyper box say bj is given the class level of c line or class c, then only I will have a connection from this hyper box bj to the responding output notes. For a hyper box, which does not belong to a class of particular clause, there will be no connection from that hyper box to the corresponding class. So, I can represent this connection with forms the hidden layer notes to the output layer notes by another matrices say u. The elements of the matrices u can be either 0 or 1.

If it is 0, say uij if an element uij is equal to 0; that represents that I do not have a connection from the ith note in the hidden layer to the nth note in outer layer. That means the corresponding hyper box, ith hyper box does not belong to that class cj, where as if say up q that is equal to 1 that indicates the hyper box. The pth hyper box in the hidden layer belongs to the qth class. The corresponding element in the matrices will be set to 1.

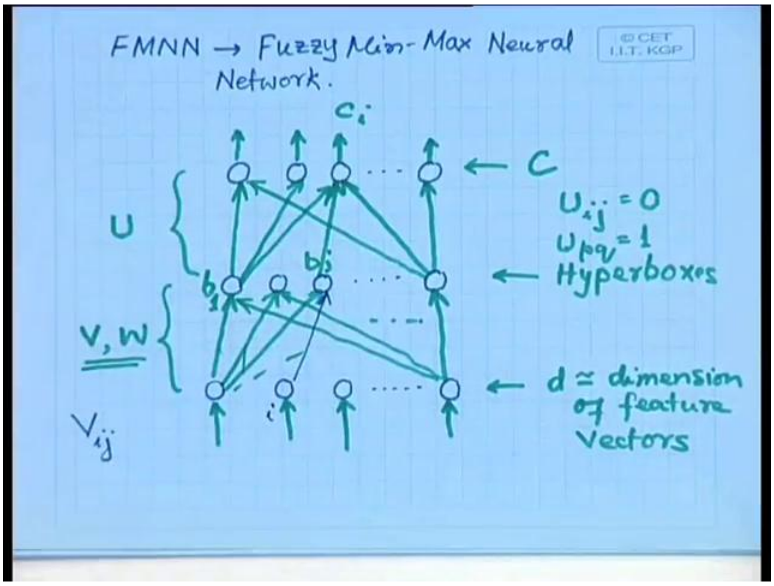
So, accordingly I will have connections from the hidden layer notes to the notes in the output layer. So, this being binary either I will have a connection or I will have no connection. So, this connection weights are either 0s or 1s. So, as before, you feed the input vector to the input layer note. So, this input vectors are passed on to the hidden layer notes. The hidden layer notes know what are the min point and max point of the responding hyper box to these matrices elements v and w. They compute the corresponding membership function. So, the architecture of a node in the hidden layer will be something like this.

(Refer Slide Time: 14:09)



It will have the connections corresponding to u and v for the imports u, v and w, which are the min points and the max points. It will also have the input vector unknown vector A. It will give me an output, which is nothing but this function b j A vj wj. So, if this note this is the j th hyper box notes, let us represent it by bj. So, in this architecture, each of these hidden layer notes will compute the corresponding membership. So, bj is a jth note because all these hyper boxes are repeated during the training operation or learning operation.

(Refer Slide Time: 15:29)



So, if this bj is connected to say an output class ci, so I shall have this corresponding connection with equal to 1. So, this output membership value is collected by this output node output and notes ci. Similarly, if I have another note says b 1 that might also be labeled as class ci. So, this hyper box is 1. So, it will also calculate the corresponding membership function. That membership value will be passed onto this note ci. So, this notes ci actually gets the membership values from different notes, which are computing the membership values in the hidden layers.

Finally, the responsibility of this notes ci will be to make a fuzzy union of all this membership that it gets. This fuzzy union is nothing but max operations. So, it finds out what is the maximum of the membership values. That maximum is taken as the membership value of this unknown vector to this class ci. So, that way, each of these output layer notes will compute the corresponding membership value. The outputs of this output layer note can also have membership values computed output layer notes that it can be taken as fuzzy class equation.

If I want to form a hard classification, I will take the maximum of this values and whichever output note gives the maximum membership value, I classify my note to that corresponding class. So, how the training process of the learning process will continue? Suppose that I have got m number of samples training samples. The training samples are labeled as belonging to different classes. I take the samples one by one. So, whenever a sample is taken for the first time, I had to create 1 hyper box note. That means I have to create 1 note in the middle layer.

So, initially what I have is I have this output layer notes. I have this input layer notes. The number of output layer notes is fixed that is same as the number of classes that I consider. The number of input layer notes that is also fixed. It is same as the dimensionality of the feature vectors. The notes in the middle layer of the hyper box notes are created while training that is in the form of connection weights. That is given in the form of connection weights over here unlike in other neural networks like perceptron single layer perceptron of multilayer perceptron where we make a weighted summation of the input feature components. The linear equation here is not a linear equation.

Each of these notes has to compute this membership function, this membership value. So, whether you call it as weights connection weights or some information available to the input layer note or to the hidden layer notes, the information is saying is represented in the form of 2 matrices v and w. For matrices, v will give you the min point information and w will give the max point information. So, a particular element vij represents what is the min point. I consider that this is the information going from the ith note in the input layer to the jth note in the middle layer.

If I consider these concepts of 2 weights are associated with every note 1 corresponding to the min on corresponding to the max point, so this hidden layer notes, they only need to know this information min point max along with the feature vector classified. So, while training as we are giving a number of labeled samples for which the jumbling is already known. Initially, what I have is I have this output layer notes. I have this input layer notes. I have 3 empty matrices; 1 corresponding to u, 1 corresponding to v, 1 corresponding to w. These matrices are empty initially and this hidden layer that is also empty. There is no note in the hidden layer.

So, while training, I take the first training sample and because this is the first training sample that I am counting, so in case of hyper box creation, I have to create 1 note in the hidden layer. Corresponding to that, I have to fill up these v matrices and w matrices. What will be the v and w matrices? Initially, v and w will be identical. As I have only 1 hyper box corresponding to single point, so min point and max point will be same.

I also have to establish the connection from this created hyper box to this 1 of the hyper boxes in output layer or 1 of the notes in output layer. So, what I will do is if the first point has been taken that belong to class c 1, I will make the corresponding entry in the u matrices equal to be 1. All other elements in the u matrices are zero.

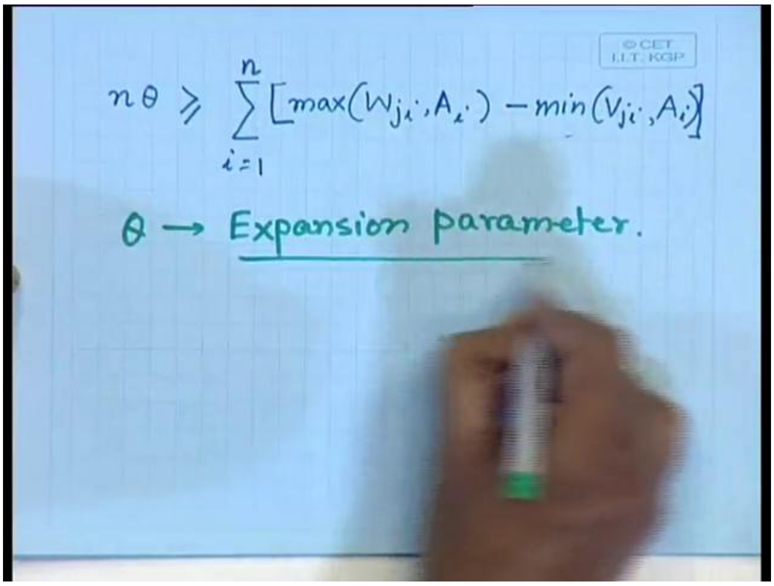
That means I am making a connection from the hyper box, which is created in the middle layer to the class notes come in the output layer, whereas other connections from the hidden layer to the output layer are missing. All of them are represented by 0 in the u matrices. Next time, I get the second point, second point either may belong to class c 1 or it may belong to another class. If it belongs to another class, then what I have to do is I have to create another note in the middle layer corresponding to another hyper box. That will be a point hyper box.

This is because I have not encountered any point belonging to that class earlier. So, that would be a point hyper box. Accordingly, I have to enter information in v matrices w matrices and u matrices. I get the third point assuming that this class of the third point I already created the hyper box in the middle layer. I assume that one or more hyper boxes are created in the middle layer.

Then, I have to see that whether this third point has at least 1 hyper box that is already been created whether third point is will be accommodated in those hyper boxes or not. If it falls within the hyper box, which is already created, I do not have any problem. I do not have to create anymore note any new note for that. If it is outside the hyper box and for that class 1 or more hyper boxes has already been created, then first what I have to check is whether this new point can be included in one of the hyper boxes or I have to create a new note for this new point.

So, the way this new point can be included in one of the hyper boxes is that you select a hyper box, which is nearest to that point and expand the hyper box whether if the expansion is permitted or not. So, if I have to have criteria to decide whether the expansion is permitted or the expansion is not permitted, if we need to stage a hyper box, then there is a risk that samples belonging to other classes will also be included in the same hyper box. So, accordingly, I have to put some constant or how much a hyper box can be expanded.

(Refer Slide Time: 24:49)



So, this expansion criteria for hyper boxes is given by this in theta, which is greater than or equal to summation maximum of w j i Ai minus minimum of v j i Ai, i varying from 1 to n when I have n number of dimensions, whereas before w j i represents the ith dimension of the max point of j th hyper box. Similarly, this is the ith dimension of the min point of the j th hyper box. Ai is the ith component of the feature vector, which is to be included in the j th hyper box. So, I expand a hyper box. Then naturally its min point and max point, they are going to be disturbed.

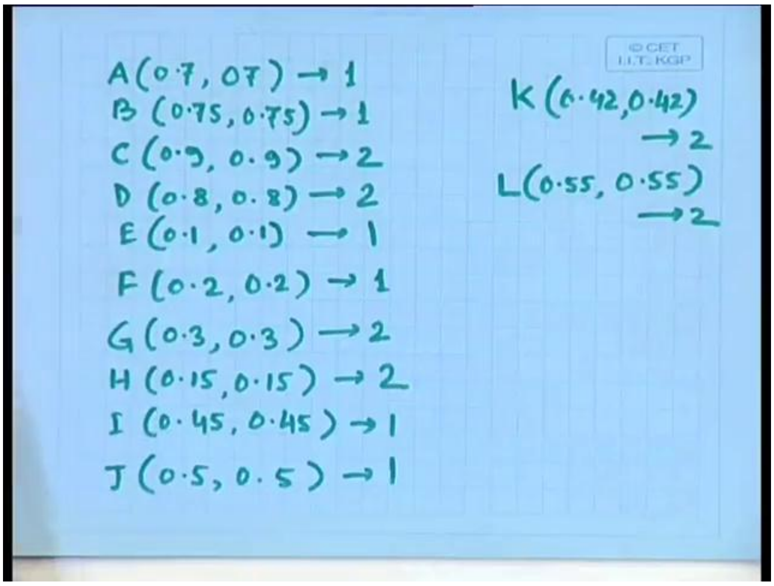
Max point is nothing but the maximum coordinate value of all the feature vectors of hyper box. So, the new max point in the ith dimension will be maximum of these 2. Assuming that this max point will be going to shift to Ai, so that maximum these 2 will represent the new max point and minimum of these 2 will represent the new min point.

The difference between the min point and max point that is going to tell what is the spread or width along the ith dimension. So, sum of all these width in different dimensions in all the n dimensions that must be less than n into theta. As long as this condition is satisfied, we are allowed to stage or expand the hyper boxes. The moment I find that this value exceeds n into theta, then expansion of hyper box is no more possible and no more permitted for this theta. It is called the expansion parameter. So, while trying to corporate if feature vector in a class for which some hyper boxes are being created are first to see whether the hyper boxes are expanded to include this feature, if the expansion is permitted, then I simply expand the hyper box. Over here, expansion of hyper box means I have to modify the min point and the max point.

So, I do not have to create any more any new hyper box. What I have to do is I have to change the min point and max point of an existing hyper box. That means I have to change the values in the v matrices and w matrices. That is the only thing I have to do. I do not have to do anything else. If I have to create a new hyper box, then I have to make a corresponding entry in the v matrices w matrices. I also have made an entry in the u matrices.

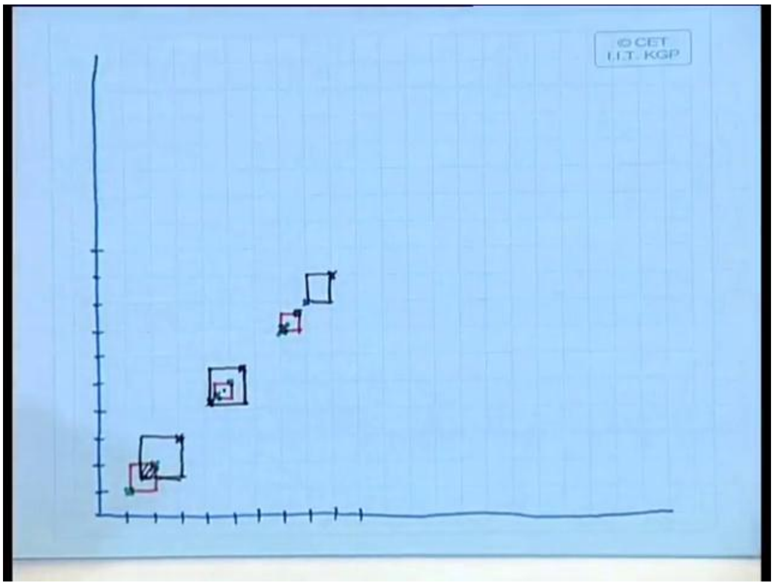
In this case, u matrices entry already exists without 1 of the hyper box, which has to expand. Only the min points and max points are to be modified. So, that modification, I will do in the v matrices and w matrices. So, that is how the learning will continue. When doing learning as there is a profusion of expansion of hyper boxes, other problems may come in that is it is possible that hyper box and expanding that overlaps with another hyper box. So, let us just take an example to see that how this overlapping of expansion may affect our performance of the 1000 minutes. So, let us take few points belonging to say 2 different classes class 1 and class 2 points.

(Refer Slide Time: 29:37)



So, I take a point A whose coordinates are say 0.7, 0.7. Suppose that this point A belongs to class 1. I take a point B whose coordinates are 0.75, 0.75. Suppose that this also belongs to class 1. These are the points of feature vectors 2 dimensional feature vectors, for which these of the class belonging that is specified. So, let us assume that these are the training vectors using, which I had to generate the hyper boxes. So, let us take these points one by one. So, I take this 2dimensional feature.

(Refer Slide Time: 31:51)



So, let me consider this 2dimensional features. So, find that the first point that I have that is whose coordinates is 0.7, 0.7, this sample belongs to class 1. So, I take 0.7 over here. This is 0.7, 0.7. This sample belongs to class 1. The next point is 0.75, 0.75 that also become to class 1. This 0.75, 0.75 is over here at this point. So, this point belongs to class 1. So, initially, I will have 1 hyper box. The min point and max point will be same as 0.75, 0.75 next time another sample belonging to same class having points as 0.75, 0.75 because it belongs to the same class. So, I have to say that whether this hyper box and expanded to include this particular point.

Now, assuming that depending up on the value of theta, this hyper box expansion is permitted. I shall have a new hyper box, which is given by this. So, this will be a hyper box responding to class 1. Next, I have a point given by 0.9, 0.9, which belongs to class 2. I have this point respond 0.9, 0.9, which belongs to class 2 over here. The next point is 0.8, 0.8; that also belongs to class 2.

So, 0.8, 0.8 belongs to class 2. So, initially I will have 1 hyper box for this. Next, I have to see that whether for this 1 hyper box and expanded to include this point. Let us assume that this expansion is also permitted. So, I will have this box corresponding to class 2. Then I will have this point for class 1 that is 0.1, 0.1 that is somewhere here will prove to cost 1. Then I have this 0.2, 0.2. I have already an existing hyper box, this 1, which also become to say fighter to include there is by expanding this. You find that the hyper box size is quite large.

So, here this expansion may not be permitted by our expansion criteria. This expansion is not permitted. Then I have to create a new hyper box office. So, assuming that this is new hyper box is created, so the next point that I take is 0.2, 0.2, which also belongs to the same class 1. So, I have 0.2, 0.2somewhere here. Now, I have 1 hyper box responding to class 1 and another hyper box corresponding to class 1. So, I have to try to see which one can be expanded to accommodate this.

So, here I find that this is the nearest hyper box. Assuming that this expansion is permitted, I get hyper box of this point. Then I have to take other training samples. So, I have this 0.3, 0.3, which belongs to class 2. So, 0.3, 0.3 is somewhere over here, which belongs to class 2. I have 0.15, 0.15, which also belongs to class 2. So, 0.15, 0.15 is somewhere over here.

So, when I created this hyper box, this hyper box out of class 2 already existed. Again supposing that this expansion is not permitted, this was not expanded, so I have to pay no hyper box yet another hyper box, if this expansion is not permitted. Now, let us assume that this expansion is permitted.

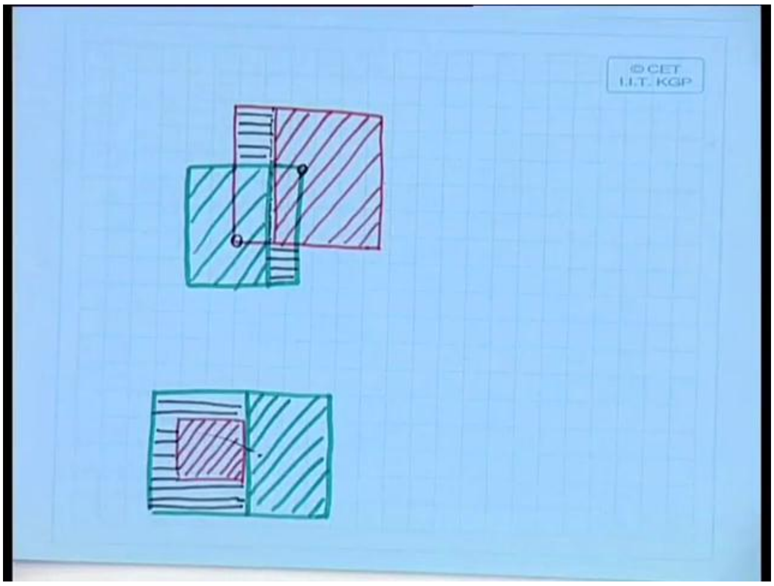
So, I have the new hyper box, which is something like this. So, here you find that what I have is an overlap because this hyper box belongs to class 1, where as this is hyper box belongs to class 2. How to tackle such overlap situation? So, let us take other training samples. So, again I have this 0.45, 0.45somewhere over here, which belongs to class 1. Then I have 0.5, 0.5 that is somewhere here that also belongs to class 1. Suppose that these 2 formed 1 hyper box. Next, I have this 0.42, 0.42. This was 0.45. So, 0.42, 0.42 will be somewhere over here. Then I have 0.5, 0.5, which is somewhere over here. Assuming that these 2 form a single hyper box, I have a situation like this.

So, here to find that 1 hyper box belonging to class 1 that is completely enclosed within another hyper box belonging to class 2. So, as the expansion of the hyper box is permitted, such types of overlap are likely to occur. So, the question is how to tackle such overlaps. So, this is what is called containment because 1 hyper box belonging to 1 class is fully contained in a hyper box class belonging to another.

If I expand this definition that it may not be necessary that the hyper box is completely contained within another hyper box, it is also called containment. At least 1dimensional upon hyper box is completely contained within similar to the same dimension of another hyper box. So, here you find that if I have this overlap containment with in hyper boxes belonging to the same class that is permitted because hyper boxes belong to the same class. If the hyper boxes belonging to different classes either there is contentment or there is an overlap, then I have a problem in computing them membership function in this overlap region at this region. I cannot compute the membership function because over this region if the sample falls, get to the membership function of 1 for class 1. I will also get the membership function of 1 for class 2.

So, over this entire region, the classification cannot be performed. Similarly, over here, if a sample falls within this to which class should be there whether the sample falls in belongs to class c 1 or sample belongs to c 2. So, I have similar kind of problem. If such types of overlap of containment occurs while training this puzzle, so I have ways of dealing with this sort of problem. So, what you have seen so far?

(Refer Slide Time: 41:09)



I can have either an overlap of this form that is I have a hyper box of 1 class. I have a hyper box of another class. These 2 hyper boxes overlap. I can have a situation something like this. I have a hyper box of 1 class, which is fully contained within the hyper box of another class. So, this sort of situation when this sort of situation occurs;

then seems ahead that you go for contraction. If the hyper boxes belong to different classes overlap or one contains another, such a kind of containment of overlapping cannot be permitted.

So, you go for contraction of the hyper box. When you go for contraction, you have to go for the dimension of min overlap, so that the disturbance incorporated in the hyper boxes, which is already created that is min. So, when you contract, then what have to do is I have to break it somewhere in the middle over here assuming that this is the dimension of minimum overlap.

So, I get new hyper boxes. 1 hyper box will be this and the other hyper box will be this. So, this is 1 hyper box after contraction and this is another hyper box after contraction. For the contraction, what I have to do is I have to simply modify min points and max points nothing else. So, by this contraction, you find that overlap between the 2 hyper boxes belonging to 2 different classes that has been removed.

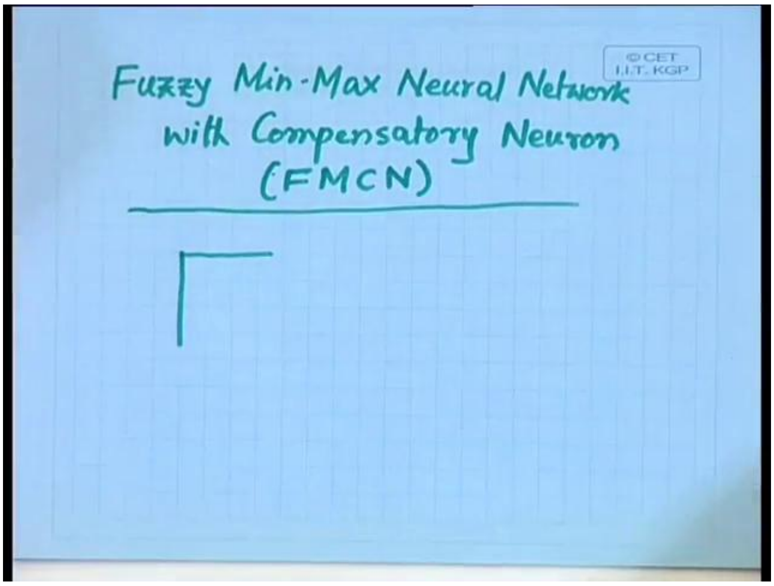
At the same time, I am incorporating some amount of error at the time of training itself. This is because this region for which information was already gathered 1 is this region and another is this region. For these 2 regions, information was already given. So, by this contraction process that information is and not only that when I have created this min much hyper box, this was the min point of this hyper box and this was the max point of the this hyper box. By this contraction, you find that min box point of class 1 has been put into class 2.

I am not saying first feature along the feature along, which the overlap is minimum, it may be first feature or it may be second feature, may be third feature. It may be any feature. It is along the feature dimension along which the overlap is minimum. So, you find that this min point of class 1 has been put into the hyper box of the second class. The max point of class 2 has been put into the hyper box of the first class.

So, while training it, I am introducing some amount of error. So, if I introduce amount of error while training, it is quite natural that point testing or maybe, I get unknown samples. The performance will be poor because I cannot maintain the accuracy during training process. Similarly, over here also, I have to go for contractions.

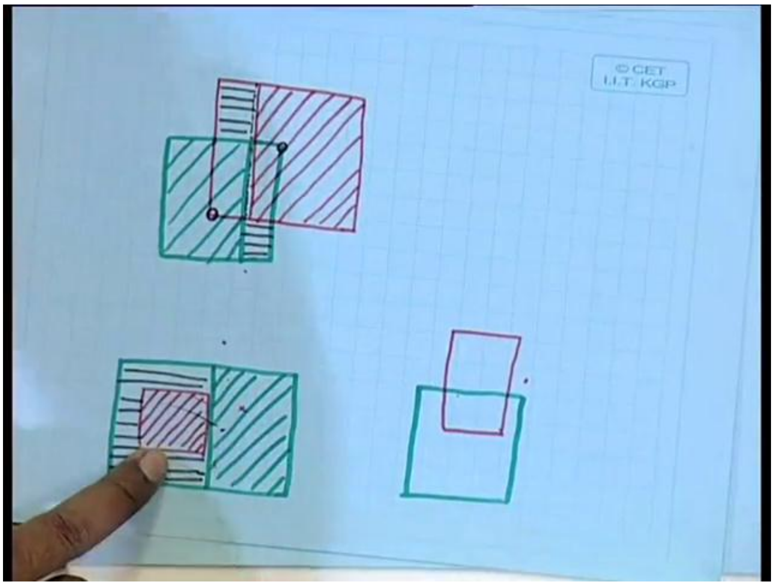
So, after contraction, the hyper box may be something like this. So, the new hyper box will be this 1 for 1 class. It will be this 1 for another class. So, we find that the information, which is already gathered for this region, this information is lost. So, as this information is lost, so obviously I am incorporating error while training operation. As I am incorporating error training operation, obviously during the testing or classification of unknown samples, the performance will not be that good to solve this problem. We had suggested an architecture, which is called fuzzy min max neural network with compensatory neural.

(Refer Slide Time: 47:05)



So, that is fuzzy min max neural network with compensatory neural. In short, it is FMCN. So, in case of FMCN, what you have said is if I have similar kind of overlap and when I said this sort of containment, it is not necessary that 1 hyper box has to be fully contained in a hyper box. It is also possible that a long one dimension contained with another hyper box. So, I can have a situation like this also possible.

(Refer Slide Time: 48:12)



So, containment is only on 1dimension, not it is fully contained. So, in case of FMCN, what is suggested as whenever there is an overlap of this form or whenever there is a containment of this form, do not go for contraction of the hyper boxes to afford this overlap at this containment. What you do is you try to re compute the values within overlap region, whether it is here or it is here, the logic is as the hyper boxes created using the information of min point and max point.

So, when you try to re compute this membership value, you try to retain maximum membership value along the min point and max point. As you move away from the min point and max point, you try to reduce the membership. So, in the graded region, the membership value will actually be upgraded membership at this point, which is the max point of hyper box belonging to class 1. The membership value of a sample near to this point will be more to hyper box 1 more to class 1 than towards class 2. Similarly, an unknown point, which is near to this max point, this is max point of the hyper box belonging to class 2.

So, if an unknown point is more towards the max point of this hyper box, its membership value to class 2 will be more than its membership value of class 1. So, putting this logic, what we have suggested is we have upgraded membership in this overlap region where the membership value near this min point will be maximum to class 1 and minimum to class 2. The membership value of an unknown point, which is near to this the max point of class 2 of this hyper box in class 2, it will have maximum membership to class 2 and minimum membership for class 1. In between, the membership value will vary.

Similarly, over here, if an unknown sample falls within this region, what you do is you do not disturb the structure of the hyper box. The structure remains the same. If an unknown point falls over here, it is within this hyper box belonging to class 2 though it is also within hyper box belonging to class 1. The nearest hyper box is this 1, the red 1. So, it will have a membership value 1 to this red hyper box. The membership value will be 0 to this green hyper box.

If an unknown point falls over here, it is not within this hyper box. It will have membership value 1 within this green hyper box. I am not saying it will have a membership value 0 within the red hyper box. It will have a membership value to this red hyper box depending upon its distance from the red hyper box, but obviously less than that. So, by max operator because this 1 will have membership value to 1 to green hyper box, which will obviously, be greater than the membership value to red hyper box.

So, max operator obviously, this will be filtered out. The class membership will be given as the class belonging. This will be given to the green class. When it falls outside this somewhere over here again, its membership will be more value than its membership value of this. Naturally, this will be classified to this. Similarly, over here, if the hyper box falls in this region, in the overlap region of the content region, it will have a membership value of 1 to this red class, membership value of 0 to green class.

If the point falls somewhere over here depending upon its distance from these 2 different classes, its membership value will be computed. It will be put into that class for its membership. So, what we are suggesting is that do not disturb the structure of hyper boxes. This is because disturbance of structure bar of boxes introduces more amount of error. By retaining the structure of the hyper box, you try to modify the membership value competition. So, we will have upgraded membership within the overlap region as well as with the content region. But, in the content region, it will not be greater membership value of either 0 or 1. In the overlap region, it will have upgraded membership. So, accordingly, when we suggest this min max neural network with compensatory neuron, this is actually inspired by the reflex action of the human brain.

What is reflex action? The moment you suddenly touch a hot object or suppose you put your foot on a nail usually, if you have a stimulus that stimulus information sends to your brain. The brain instructs what to do next, but in such sudden event, the instruction does not come from brain. So, sense the signal instead of going to the brain, there is a bypass. So, immediately, there is a short circuit thing. You get an instruction to your muscle. What to do next? So, it is an instant action or a reflex action. This reflex goes on decreasing with your age. As you become older, the reflex action becomes weaker.

So, it is a similar type of concept that whenever you meet such a kind of situation, you take an immediate action. Such a kind of concept has been introduced in this FMCN architecture that if an unknown sample falls within this overlap region or within the content region, we consider these to be such a kind of sudden situations. So, immediately we have to take action without going for much of computation. So, to do this for every such overlap region or every such content region, you introduce a new hyper box or a new node in the hidden level, which are called compensatory amounts.

So, whenever an unknown sample falls within this type of regions in the overlap region or the content region, then only that compensatory neuron will be active. It will try to act as a modifier of the membership computation of the membership computation of the output layout over here. So, I will have a set of compensatory neurons. These neurons or these notes or neutrons are actually computing the membership value to a particular class whenever a sample falls within the overlap region or within the content region. Then these compensatory neurons will modify these values. The modified value will be taken as membership function. So, we will discuss about this FMCN architecture in next class.