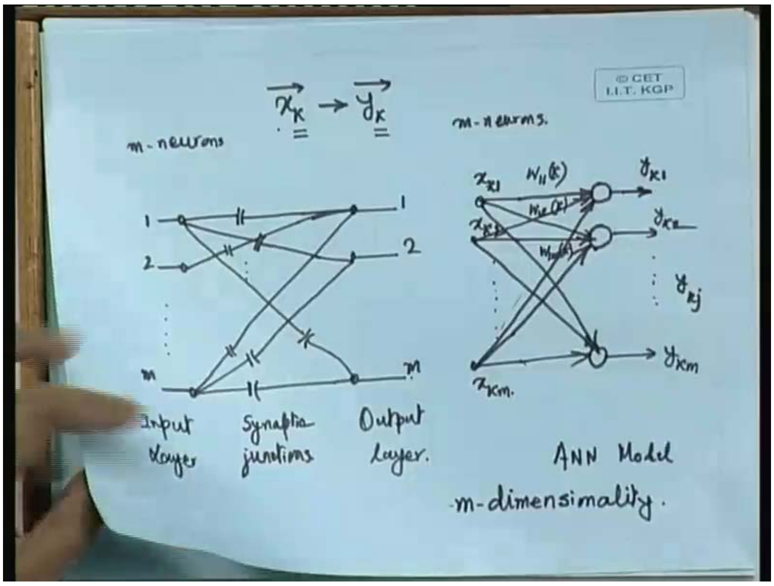
Neural Network and Applications
Prof. S. Sengupta
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 07
Associative Memory Model

In the last class, we were discussing about the Associative Memory and I mean, just the introductory part of it. And today we are going to continue with the detailed discussion on associative memories. And in particular we will be taking the associative memory model, both from the structural aspect as well as from the mathematical point of view.

Now, yesterday we were listing out some of the characteristics of the associative memory. And one of that I think you must have noted, that is to say that the associative memory is highly distributed in nature. So, it is a distributed memory system it is not confined to one place, it is distributed and that is of course, motivated from the biological neural consideration. So, let us look at some modeling that one can make about the associative memories realized out of biological neurons.

(Refer Slide Time: 02:03)



Let us take certain number of neurons, let us say that we take m neurons, which are inter connected in an associative memory weight. So, what we will be having in that cases, that let us first consider the input layer, so we will be having from the input several neurons, which will be connected to the neurons at the output layers. So, here we will be having the input layer and here we will be having the output layer and in between these two we will be having the synaptic connections.

So, all this synaptic junctions will be existing like this, that supposing we indicate the synaptic junction in this manner like this. So, these are I mean all these things nothing but, the synaptic junctions, so how is the association or the memory storage actually taking place. We are feeding the patterns at the input layer and we are obtaining the response from the output layer.

So, whatever is stored is actually stored in the form of the synoptic weights. Now, this is one area of the memory, which is responsible for the storage of information and there what is being done is that we will be having a set of synoptic weights. But, this set of synoptic weights will act in such a way, that if you are feeding any specific pattern, let us say that you are feeding a specific pattern x k in form of a vector.

By vector I mean to say considering all these m elements, let us say that there are a m elements at the input and there are m elements at the output. It is not mandatory that the input number of neurons and the output neurons, they have to be same in number they could be different. But, without any loss of generality one can consider it this way that we have I mean the same numbers of the input and output I mean perhaps just to ease out our analysis part.

But, in fact the analysis can be carried out for even different number of inputs at the input and the I mean a different number of neurons at the input and the output layers. So, what happens is that, this memory or this synoptic junctions will be responsible for the storage of patterns. In the sense that, if we are feeding any input pattern let us say that a particular pattern x k in the form of a vector that we feed.

In that case we should get the corresponding y k vector which is supposed to be the output. And if we get that, then one can say that there is an association between the input x k and the output y k both in the form of vectors. So; that means, to say that we are essentially looking for the association between the x k vector and the y k vector, where x k is the input and y k is the output.

So, in affect what it is doing is that, it is as if to say memorizing the x k y k pair, but not explicitly what it does is that, in the form of this synoptic weights. So, that when x k is straight over here you get y k, similarly when x 1 is pair here you get y 1, so whatever pattern you feed, you can recall that pattern by feeding that pattern as a stimulus, you can recall that pattern I mean recall it is response from the output.

So, that is what it is, so this is from the biological model point of view. And whenever we consider the ANN model, it is almost the same, but only that the representation is slightly different. So, there too also we can consider as an example, again in the case of m neurons and let us say that we consider any pattern, any particular pattern k. So, here by this k I mean to say the index of the pattern.

So, x 1, x 2, x 3 and all these things will mean the first pattern, second pattern, third pattern like that and these are all in the form vectors. So, let us take the kth pattern and we will be having x k 1 has the first input let us say. Then, we will be having x k 2 as the second input and likewise we will be having x k m as the mth input, so these are available at the input layers. So, likewise will be having an input layer for the ANN model also.

So, this is the ANN model of associative memory and these will be connected to the neurons or you can say the output neurons, which will do the processing part. Processing means, what kind of a processing in this case it is fairly simple, that take any neuron. Let us say that we take the neuron, that generates the output y 1 or in this case we will write y k 1. Because, for the pattern k the response that we are getting from the output neuron index task one.

So, y k 1 will mean the output from neuron index task one for the pattern k. So, we will be having y k 1, y k 2, etcetera up to y k m. Just, because in this particular case we had considered the number of input neurons and the number of output neurons to be the same and as I tell told you that they could be different also. Now, what happens is that every neuron will be having the connections from the other inputs.

So, this will connected to x k 1 x k 2 and up to x k m, likewise y k 2 will be connected to x k 1, x k 2 up to x k m and likewise y k m continuing this way, y k m will be connected to x k 1, x k 2 up to x k m, this is the ANN model. So, it is essentially the same only and in this case what happens is that we will be designating each of this connections by their synoptic weights.

So, in this case what we have to consider is that, if we are taking a particular pattern k and we are considering this connection let us say, this connection means what, that it is a connection that is there between 1 and 1, output 1 and the input 1. So, we will be writing it as w 1 1, but just to indicate the pattern number also, what we do is that within parenthesis we give a pattern number. So, we will be calling this synoptic connection by w 1 1 k, what we will be calling this connection that is x k 2 to y k 1.

Student: ((Refer Time: 10:18))

W 1 2 k, so like this, this connection from x k m to x k 1 we will be calling as w 1 m k. And now going over to the second output neuron, that is output neuron number 2, we will be having the synoptic connection as I mean from between this x 1 and the y 2 the synoptic connection will be w 2 1 again within bracket we have to write k. So, the very purpose that we are writing k is that, these weights, at least we know that these weights have been designed.

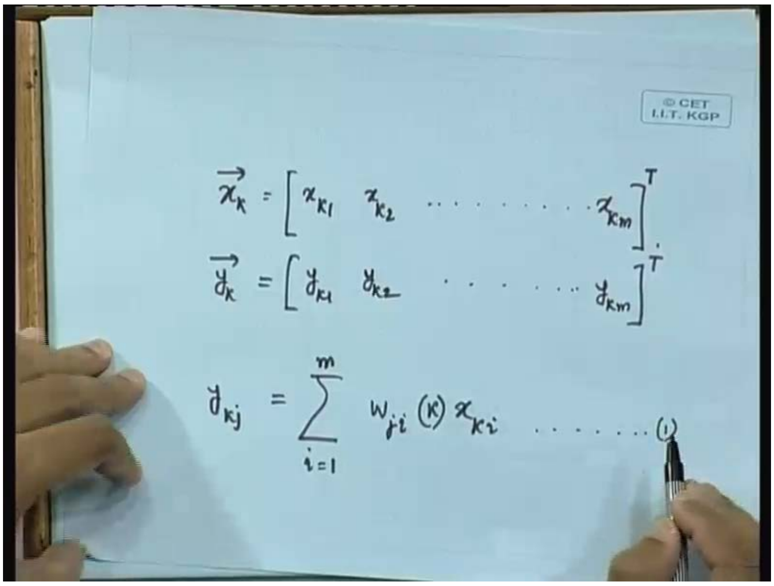
So, that for the pattern k, this weights are suited to this pattern k, so that when the input x k is applied you get the output as y k. But, there lies a question does this network have only I mean memory for only one pattern, certainly not we are going to store a large number of patterns within this memory. But, then why is it that we are writing it as w 1 k, because the moment we write anything with a bracket or suffix or superscript whatever with k.

That means, to say that it is only for the kth pattern that I am considering, for the time being we consider one pattern or rather the kth pattern only. But, in effect when we design the total combination of these weights, when we design this set of weights that time we have to consider the effects of all the pattern. So; that means, to say that the final weights that we will be having will be from the combined contributions of all the patterns.

This I am writing as k just to indicate that for the association of x k with y k, this will be the synoptic weights w 1 1 k, w 1 2 k like that up to w 1 m k again w 2 1 k, w 2 2 k up to w 2 m k finally, up to w m 1 k, w m 2 k up to w m m k all right. Now, let us I mean in these lines we can easily formulate the mathematical part of it, I think the representation now is going to be pretty simple. Let us see, that what is going to be our definition of the vectors as such.

Now, we started with two vectors, the input stimulus which we called as the x k vector and the output corresponding output, which is associated with this input is y k vector.

(Refer Slide Time: 13:02)



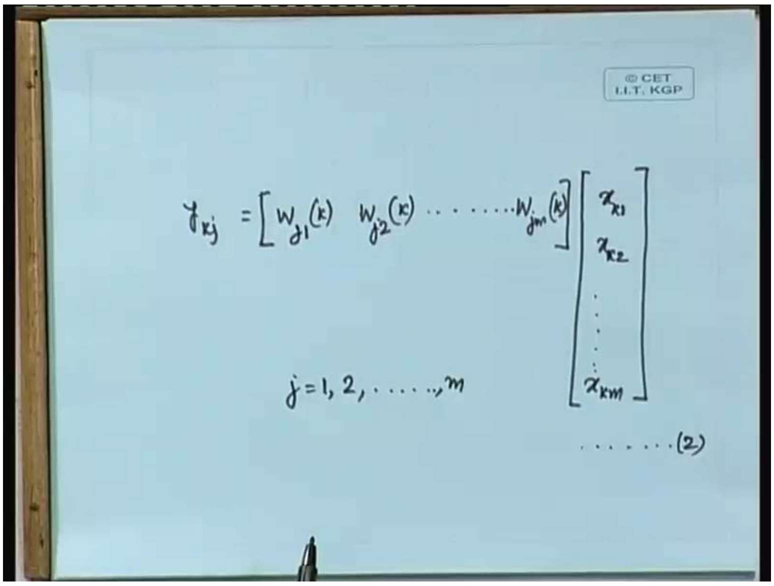
So, what is this x k and y k. So, x k vectors definition is essentially it is x k 1, x k 2, etcetera up to x k m. And we will be having the, in fact we can write this as I mean it is custom attitude write this as transpose meaning, that by vector we will be meaning the column vectors. So, likewise y k vector will be defined as y k 1, y k 2, etcetera up to y k m again the transpose of this.

And so now, we consider the response for any one of the neurons, ((Refer Time: 13:57)) let us say that we consider the response of I mean these are the outputs y k 1, y k 2 up to y k m. So, we consider the response of the jth neuron in the output layer for the pattern k again, so we are considering y k j all right. So, what can be the representation of y k j here y k j we could represent as summation of w, what j and then again some other index we to say.

So, we can say w j i for the pattern k and here we have to write x k, what x k i and we have to summit up over i is equal to 1 to m. So, what is the index i, the index i in this case is for the inputs, so i is equal to 1, i is equal to 2 up to i is equal to m, these are the input numbers. So, these are all the x k i’s and then these are the w j i’s are the corresponding weights between the output neuron j and the input neuron k and the neuron i for the pattern k and we are adding it up this from over i is equal to 1 to m.

So, this is what we are getting for y k i and this in the matrix form we can translate like this.

(Refer Slide Time: 15:50)

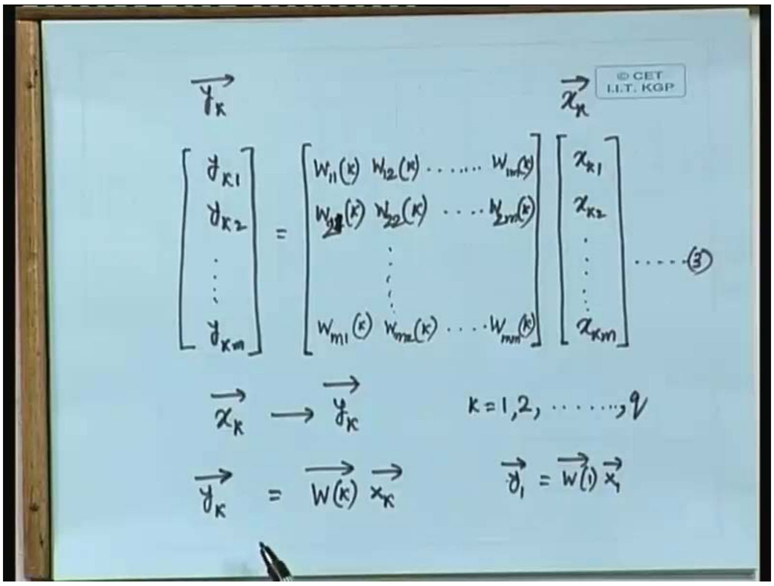


So, the same y k j we can now write as y k j equal to w what are we going to write here, w j 1 k, w j 2 k, etcetera, etcetera up to w j m k these being the weight vector. So, weights are also written in the form of a vector and this will be multiplied by what this will multiplied by x k 1, x k 2, etcetera, etcetera up to x k m. So, this is the matrix representation of the same equation ((Refer Time: 16:39)) that we had described earlier.

So, if call these as equation number 1 in that case equation number 2 is nothing but, a matrix representation of equation number 1 only. And mind you, that this equation will be valid for j is equal to 1, 2 etcetera, etcetera up to m all right, so if this is valid for j is equal to 1, 2 up to m. In that case, it is possible for us to write down m such equations we can write y k 1 is equal to in this case we will be having w 1 1 w 1 2 up to w 1 m y y k 2 will be equal to w 2 1 w 2 2 etcetera up to this.

So, we can write m such equations and if we write m such equations and take all the left hand sides of it. So, all the left hand sides will now indicate y k 1, y k 2, y k 3, etcetera up to y k m, so if all this outputs we take together and represent the outputs in the form of a vector, then what do I get, then I will be getting the vector of the output y as follows.

(Refer Slide Time: 18:07)



So, we will be having the y vector represented as y k 1, y k 2 up to y k m all right and that will be equal to something, we have to see that what is that. But, essentially all these individual inputs will be equal to I mean will be a function of all these x k 1s ((Refer Time: 16:37)) is not it I mean in every equation this part will be common, may I right. This vector I mean it has to be I mean the individual weight vector, this weight vectors will be different.

Because, for the case of j is equal to 1 it will be w 1 1 w 1 2 up to w 1 m, for j is equal to 2 it will w 2 1 2 2 up to 2 m like that. So, this will change with every j it changes, but this remains the same, so essentially since this is going to change, then for every such j's we are going to have a different weight vector. But, in effect; that means, to say that it will give rise to a weight matrix.

So, we are going to write it down as w 1 1 k w 1 2 k up to w 1 m as the first row of this and in the x vector we will be having x k 1 x k 2 up to x k m. So, as you can see that y k 1 is going to be equal to w 1 1 x k 1 w 1 2 x k 2 up to w 1 m x k m, so all this will be added. So, this is what we want is not it, so this really verifies the equation.

So, whatever we had if we put j is equal to 1, you can get w y k 1 from the multiplication of this with this all right. And likewise y k 2 you will be getting as w 1 2 k w 2 2 k up to w 2 m k.

Student: ((Refer Time: 20:25))

That is very correct yes, you have pointed out a mistake it should be w 2 1 k, because all the time our notation is that whenever we are writing the weight it should be first the output index, in this case 2 and then the input index in this case 1. So, it is 2 1 2 2 up to 2 m and is this will be w m 1 k w m 2 k the last line, the last row will be w m 2 k up to the last element will be w m m k.

So, this is the full matrix representation which gives us effectively the relationship between y k vector this is nothing but, the y k vector and this is nothing, but the x k vector. So, this relationship gives us the representation or rather the associations between the pattern x k and y k, so this is giving this association. Now, this is not the only pattern we are going to have many such patterns, let us say that we have got patterns which can be indicated by k 1, 2 etcetera, etcetera up to q all right.

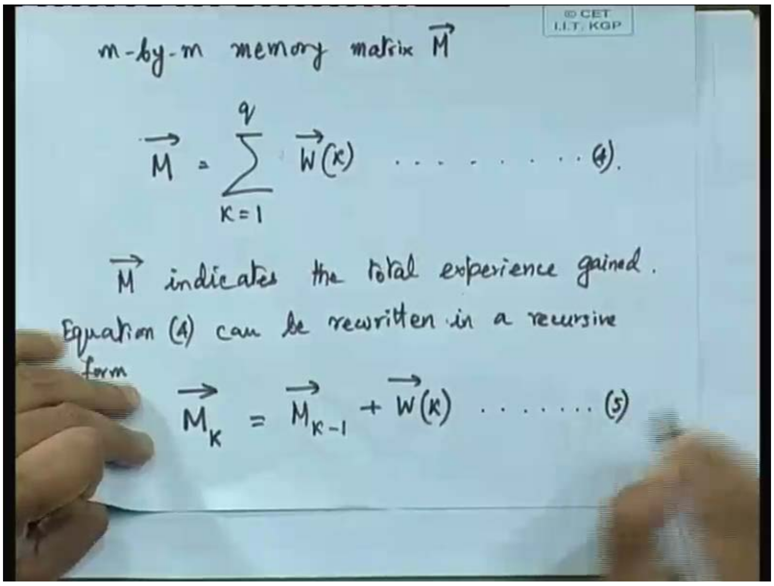
So, we will be having lot of such patterns, but now let us look at this matrix equation that we have got. So, how we can represent it, we can simply represent it as y k vector will be equal to what w k, in this case w k will be a matrix this particular matrix that we have got. So, this matrix w k times what the x k vector, so it is w k and x k vectors product which will be y k.

So, what we are going to have is that y 1 vector will be equal to w 1 vector x 1 vector like that as if to say that with every pattern, we are going to have a different set of weights. As if to say that for every pattern it has to memorize a different set of weights, but that is not a case. In fact, what happens is that I mean as we keep on feeding the patterns. Now, for every pattern it learns it associates this y k with x k; that means, to say that it forms w k, but then it keeps on adding to it is existing weight.

So, initially we can say that we begin with the weight as w 0 could be that all the synoptic connections, they are initialized to 0. And we then feed the first pattern get w 1, feed the second pattern we get w 2 and then, whenever we are getting the second pattern we are adding w 1 with w 2. When, we feed the third pattern we are adding w 1 w 2 and w 3.

So, what we are putting into the memory is memory also is going to be in the form of a matrix now is not it, all this w matrix elements will go into the memory. But, in the memory what we will be actually storing is the summation of weight matrixes.

(Refer Slide Time: 23:56)



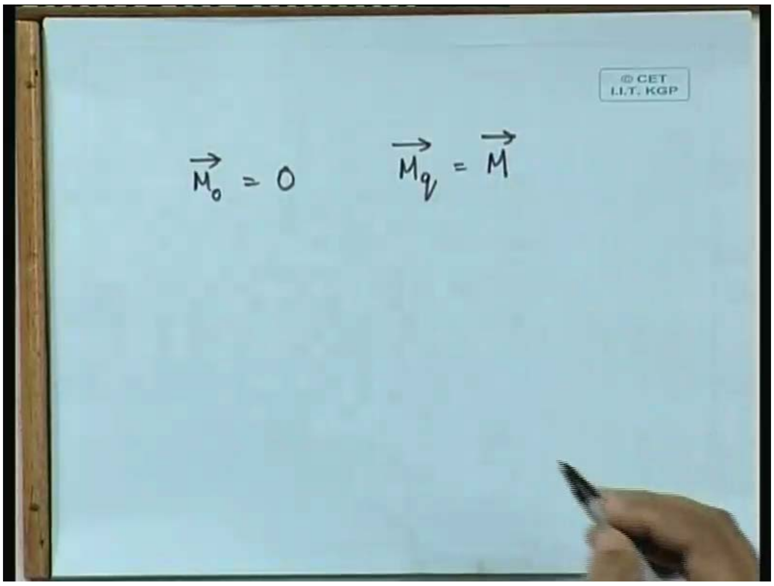
So, in the memory we will be storing like the M matrix will be equal to summation of all the w matrix w k's, but for k is equal to 1 to q assuming that there are q such patterns. So; that means, to say that what is this, so this what this M matrix is an m by m memory matrix. So, m by m memory matrices definition is this, that it indicates say summation of weight matrices.

Now, in effect; that means, to say that this m matrix basically indicates to us the total experience that is gain out of q different patterns. So, M indicates the total experience gained, now how do we gain experience, we feed patterns one by one. And we formulate their weight matrix and we add that weight matrix to the m matrix, with to the existing M matrix. In fact, this equation which we can now write it as equation number 4, equation number 4 can be written in the form of a recursive relation.

So, equation 4 can be rewritten in a recursive form and what is it that we can write, we can write that the M vector I mean the M matrix after feeding the kth pattern. Now, kth pattern means we are assuming that we are feeding the pattern from 1, 2, etcetera, etcetera in sequence. So, 1, 2 up to q we are feeding, so we are feeding the kth pattern, now after feeding the kth pattern the memory matrix composition will be the memory matrix composition which we had at the end of k minus 1 patterns plus the weight w k that we have gained from the experience again.

I mean kth patterns experience gives us the weight w k, which we add to this M k minus 1. That is, the previous experience that we had gained plus the present experience that we are getting is our new experience which is M of k. So, this is in the form of a recursive relation, so; that means, to say that if we write it in this recursive form in that case 1 I mean the two boundary conditions that we have to note is that.

(Refer Slide Time: 27:15)



M 0; that means, to say that before feeding any pattern, we are assuming that the memory was 0. That means, to say we did not have any experience to begin with and finally, I mean since we are assuming q number of patterns. I mean the experience that we have gain after feeding the qth pattern will be nothing but the M matrix I mean ((Refer Time: 27:41)) given that the m matrix definition is this summation from k is equal to 1 to q of this W k.

So, these are the boundary condition, now what you were doing is that ((Refer Time: 27:55)) you are adding this W k to M k 1. Now, the question is that can you really I mean retain the identity of this W k in this M k matrix, you are only adding to it. And if there are large number of such patterns and you if you are adding one by one, it is as if to say like adding droplets to the ocean I mean you have already got a large collection of patterns and you are only adding the experience of the new pattern to it.

So, it is an incremental addition, but again this incremental addition is necessary, why because ultimately this M matrix indicates our total experience gained, this M matrix will now indicate the set of synoptic weight. So, again if we are taking the associative memory model as we had done in the beginning. So, when it goes through all the q patterns, then we will be having a matrix M indicating the total synoptic inter connection over here, which will be such that you feed a pattern x k next time, you should get the pattern y k retrieved from that I mean it should have an exact retrieval of patterns.

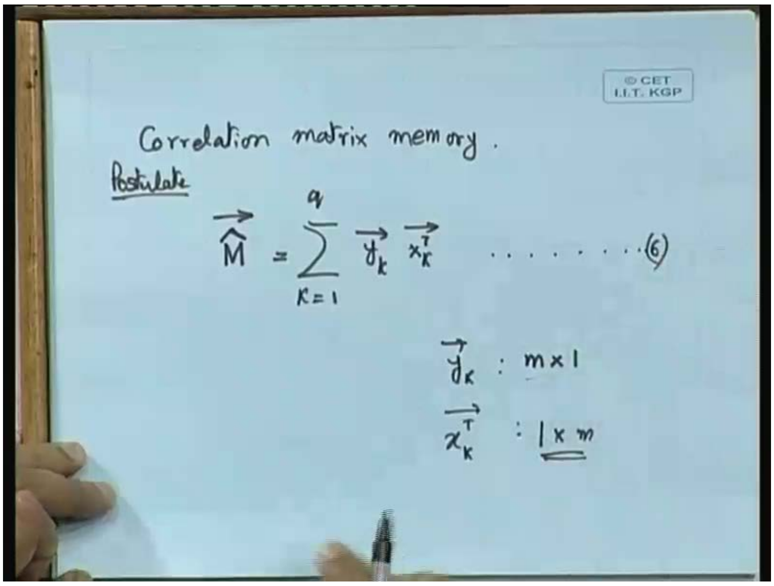
Now, whether we can do the exact retrieval or the our retrieval is prone to some errors, that is something that we have to see. Again, one can also ask another question at this point, ((Refer Time: 29:40)) that well and fine we are feeding such kind of individual patterns we are feeding with k is equal to 1, 2 etcetera, etcetera. Now, can we guarantee the exact retrieval aspect I mean that is what I think I was trying to talk to and we will be coming to that.

So, let us see I mean this particular equation, that we have written in a recursive form let us remember that I mean I just now remember what I was trying to say over here. Well and fine I mean we know I mean we have already said that, this M matrix is going to be the total experience gained. So; that means, to say that we are as if to say computing the weights, but how do we compute the weights on what bases are we computing the weights in order to reaching this figure.

We have not formulated any learning rules or not that it is as per the I mean arbitrary weights that I mean we did not define any updating equation. So; that means, to say that as if to say that we have to somehow estimate that weights in some manner. And then, once again when we estimate the memory matrix; that means, to say in that case instead of M matrix we are going to have M cap matrix as the estimated matrix.

So, again it depends that how good our estimates are our estimates may be good our estimates may not be all that good. So, how do we estimate that, so this has given rise to a concept of what is called by the correlation matrix memory.

(Refer Slide Time: 31:40)



So, that is what we are going to discuss now all right, now.

Student: ((Refer Time: 31:55))

Yes, please any question.

Student: ((Refer Time: 31:57))

Yes.

Student: ((Refer Time: 31:59))

Yes.

Student: ((Refer Time: 32:06))

I mean let me repeat the question, because I mean the audience might may not be all the time audible. So, let me the repeat the question, the question which was asked to me just now is that, we are formulating the memory matrix as a summation of the individual weights. Now, the question was that what is the guarantee, that when we feed the kth pattern as a stimulus, we will be retrieving the kth component of the weight.

Because, the kth component of the weight is no longer there, it has all got mixed up into the memory matrix M only. So, this was the question, in fact just to clarify your doubts I mean similar doubts, which others also may be having, we are not retrieving the weights, please do not make this mistake. What we have to do is that, given that memory matrix M now the W k component is lost we have got the updated memory matrix M.

Now, the question is that given the memory matrix M is it possible for us, that if we feed the input pattern x of k to it, can we retrieve the output pattern y k exactly. If the given matrix M permits us to do that our job is done, because we are only noting the association between y k and x k, we are not bothered about. Whether, we could really identify the W k component or not, you are adding a spoon of sugar to water.

Now, you are ultimately interested in testing a sweet water, you do not want that what is the exact sugar that you have put it to it something of that I mean I do not know whether this ideology is too simplified or not. So, is that understood anybody having any similar doubts to it.

Student: ((Refer Time: 34:41))

Yes.

Student: ((Refer Time: 34:43))

Yes.

Student: ((Refer Time: 34:48))

Yes, so what happens is that yes the y k is equal to yes, y k we are saying to be equal to w k x k. And then, we are adding this w k to the memory matrix and then, we are also saying yes that y k is equal to M x k, we are saying that I mean; that means, to say that this M is containing all the w's. So, this M is the total experience of learning, so now, yes indeed that is what is going to happen, that we are now going to retrieve the pattern out of this combine memory matrix M forgetting about what we had as w's very right.

So, we can discuss about the correlation matrix memory and in this case, we have to see that how do we estimate this memory matrix M. So, we come up with an estimate of the memory matrix and estimate we are writing as M cap, so M cap is the estimated memory matrix, again indicating it with the vector notation. Because, it is the matrix and we have just make a postulate.

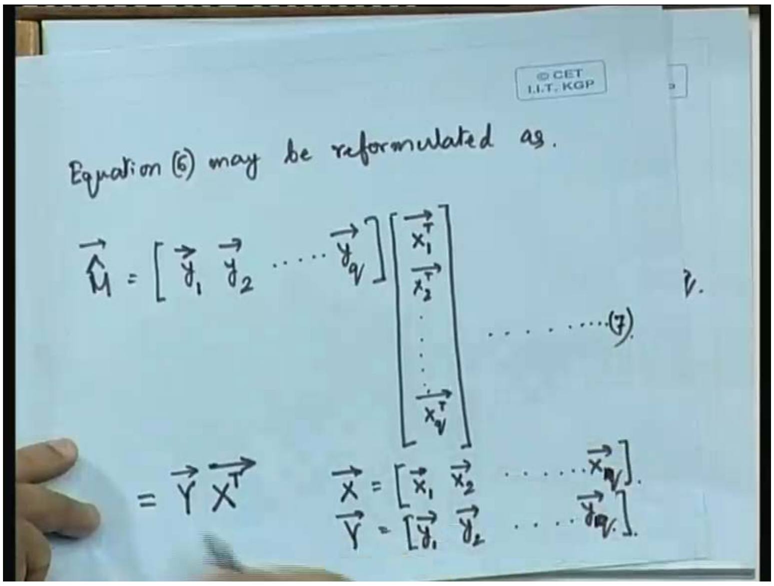
So, this is a kind of postulate that we are presenting over here that this M matrix could be considered as a summation of k is equal to 1 to q y k vector times x k transpose vector and summation of this, let us come to the interpretation of this. Now, what was our problem what I said was that, we are not knowing any explicit weight, so we are going to estimate these weights w k's. But, now we are saying that we are going to estimate this total matrix, we are going to have we are going to come forward with an estimate of this matrix M, which we are calling as M cap.

Now, what is this expression, this is defiantly a vector multiplication and what form of product is this, is it an inner product or is it an outer product. You see, this 1 y k vector is what this y k vector is an m by 1 vector, so this is y k which is m by 1 vector. And then, we are considering x k transpose vector and what is that, that is 1 by m vector and we are taking the outer product of this two and summing it up for all the patterns k is equal to 1 to q.

So, when you do the product of an m by 1 vector with 1 by m vector, what do you get you get m by m matrix. So, this is the matrix that you will be getting out of this and you are summing up all the matrices well and fine. Because, what you actually have is the input stimulus and the output response, you are having only these. So, essentially you have stored the response in response to a stimulus.

So, you take the outer product of that and you store it into the memory. So, this memory M will ultimately contain the summation of all the outer products of this pattern association. Now, this equation can be now I mean this is a summation of all this outer product the vectors and we can reformulate this equation as follows, now what was our last numbering of the equation. Remember, that was 4, in fact this recursive relation we can make as number 5. So, that now we can number it to be equation number 6, now I can reformulate the equation 6 in this way.

(Refer Slide Time: 39:20)



So, equation 6 may be reformulated as M cap again we have to take the M cap as the matrix ultimately. And writing this way y k x k transpose inevitably means, that we could indicate it by the combination of vectors. So, we can I mean since it is going to be a summation for K is equal to 1 to q what we can write is that, we can write the individual vectors that is y 1 vector, y 2 vector up to y q vector again.

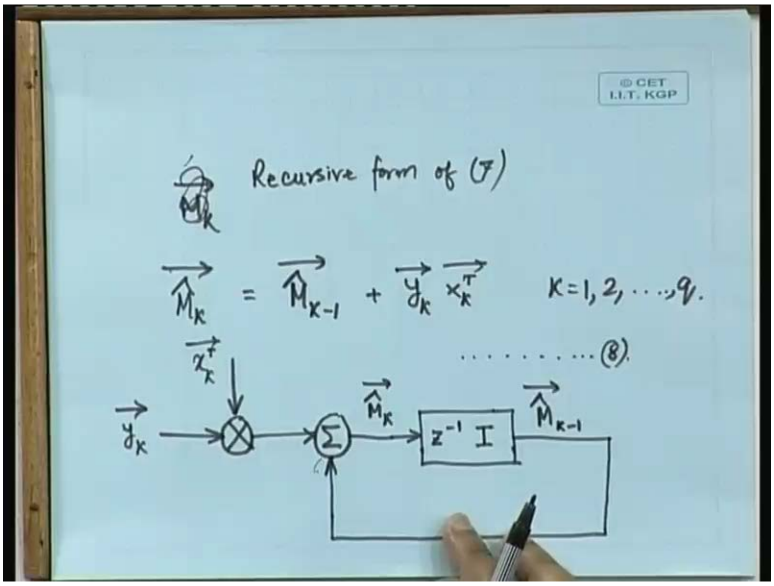
Because, we have got q patterns weight to the system and the input stimuli's corresponding to that are x 1 transpose vector, x 2 transpose vector up to x q transpose vector. This we are going to represent in the form of a column vector I mean arranging all this vectors into again another column vector. So, this in affect can be written as y vector, where y vector is actually whose each elements are all this individual pattern vectors is it understood.

So, the elements of this capital Y vector is all this individual pattern vectors times, we are going to write again going to define another vector, which will be the capital X vector. And capital X vector will be containing the elements as x 1 vector x 2 vector like that, so effectively we are going have the X transpose vector. So, Y vector times X transpose vector, in this case the inner product of these two vectors is going to give us the memory.

In fact, since essentially the product of this y 1 and x 1 transpose, they are going to give us the M by M memory matrix, they are going to give us M by M matrix. Essentially it will mean to say that the dimensionality of this M cap is going to be of the order M by M. So, here I think I have already told, but just note it down that the definition of x vector is that it is x 1 vector x 2 vector up to x k vector and y vector will be nothing but, y 1 vector y 2 vector etcetera up to y k vector y q thank you very much, this is y q.

Because, we are going to have q patterns, k is only an index which will vary from k is equal to 1 to q thank you for this correction. Now, again this expression that we have got that M cap matrix is equal to Y vector matrix times X transpose vector.

(Refer Slide Time: 43:02)



This could be restructured again in the recursive form, how we can write it in a recursive way as M corresponding to the iteration k. That means, to say that after feeding the kth pattern we I mean we should write it as M k cap matrix will be equal to M k minus 1 cap matrix plus y k pattern. So, you have fade the kth pattern, so this was the earlier experience gained.

And you have now fade the kth pattern and you are taking the outer products of the response and the stimulus. And this you are going to have for k is equal to 1, 2 up to q ((Refer Time: 43:51)). So, this is nothing but, a recursive form of equation 7, so if I take it to be equation 7 this or this whatever this is nothing but, the recursive form of 7 is going to be this.

So, we can call it as the recursive relation we can call it as equation number 8. And just we can depict it pictorially as follows, that we can feed as an input to this system the y k vector coming from here. Again, from the other input we are getting the x k, in fact I mean I should have told it like this, see here x k is the stimulus for the kth pattern and the corresponding association or the corresponding output is y k.

So, we are trying to memorize this y k, so how that we have got an association of x k and y k already. So, we are taking the product of these two and then what we are doing, we are taking the outer product and we are accumulating that into the M array. In what manner, we are going to have let us say that this is the M cap matrix that we have got, this we can put through a unit delay.

So, which we can write as z to the power minus 1 indicating an unit delay and multiplying it by the identity matrix I. And this we will be getting from the output as M k minus 1 cap vector, because this is one delay letter. So, if this is M k then this is going to be M k minus 1 the earlier, so what we are doing is that this M k minus 1 we are adding to the present pattern.

So, this is what we have already gained and we are adding it to the present association of y k and x k and we are updating the memory matrix as M k all right. So, this is just a pictorial representation of this anybody having any doubts yes please.

Student: ((Refer Time: 46:25))

Yes.

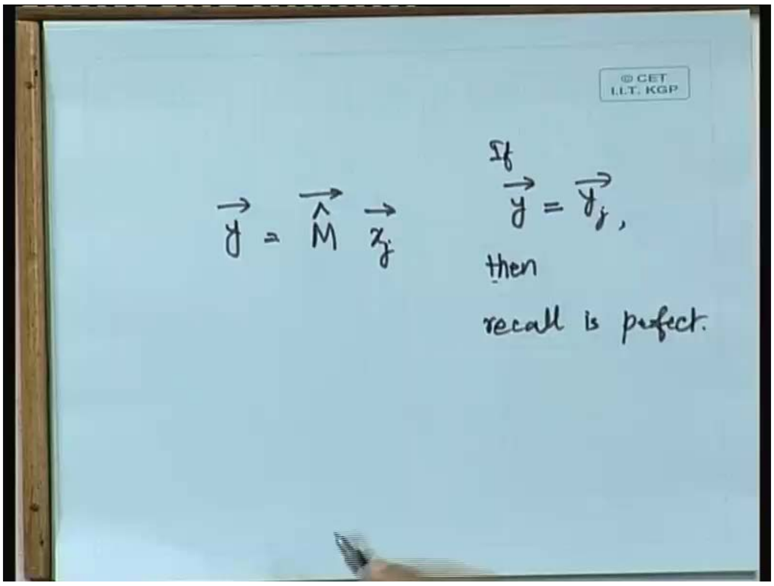
Student: ((Refer Time: 46:29))

We are getting no, we are getting see effectively what it means is that, you have got an association of y k with x k. And you are taking the outer product of that association and you are storing it into your memory, I mean you are incrementally adding that your memory. Now, once that memory is there the updated memory that you have got which is in fact, indicating the sum total of all the experience that is gained.

Now, what you are having is that you are I mean you will later arrive, once you have a memory matrix M already available with you. Then, you can feed a stimulus, let us say that you again repeat a stimulus x j from this set of patterns which you have already gone through. Now, if your learning is proper, then it should give in response to this x j if you can retrieve the pattern y k I mean y j exactly in that case, that is we are wanting is not it, that later on any pattern association that we are making we can get it back.

So, the estimate of M k is coming from x k and y k, you are very correct. And then, ultimately what we are doing is that, when we want to retrieve any pattern we will be feeding an x k and we will be seeing that if out of that we get the y k or not.

(Refer Slide Time: 48:23)



In effect; that means, to say that if we get I mean if we feed a pattern, let us say that we have already formulated the memory matrix M. So, this is M cap our estimated memory matrix and we have fade a pattern x j I mean after learning I mean after we have already stored. So, this is a total I mean this is the last iteration that we gone through, so we have gone through all the q patterns. And now in order to test this system I take I just happen to pick up the jth pattern out of it and I feed the jth pattern as an input.

So, my output corresponding output will be again a vector, which will be y vector which will be equal to M cap matrix times x j vector. Now, what I should ideally have is, that if y vector is equal to what, y j vector if y vector is equal to y j vector, then I can say that the recall is perfect, is that understood all of you agreed. So, that is what we would like to investigate now, that whether our recall is going to be perfect or not, so we are going to study the recall aspect.

Now, let us see that if it is, so or not and if not then what is the condition, that we have to fulfill for perfect association, this is what we are going to investigate. So, we do not know yet that if this is going to be true, so that is why I have just put a general vector y I did not intentionally designated by y 1 or y 2 or y j I mean I expect y j, but I did not intentionally put y j, because I do not know yet that whether I will get the exact y j or not.

Now, let us expand this basically what we can write in place of the M vector I mean M matrix is this expression. I mean this was our definition is not it, I mean it is either you follow this or I mean a better expression is this ((Refer Time: 50:50)). So, this is the expression of the M cap matrix, which is the summation of all the outer products for k is equal to 1 to q. And now, what we are going to do is that in this expression we are going to replace this M cap vector by this expression M cap matrix by this outer product expression.

So, if we put in that case we can write down the y vector as summation over k is equal to 1 to let us say that we have got M different patterns. So, this we can have as y is that y k vector times x k transpose vector and x j, this x j remains the same, so this x j I am keeping as undisturbed and only in place of this M cap I have just put forward this.

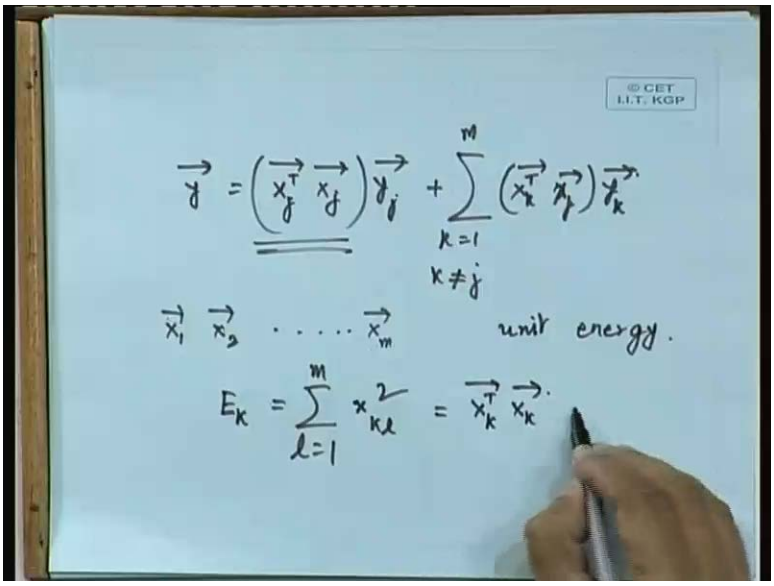
Now, you can see over here that I did not put the limit as k is equal to 1 to q instead I have put the limit as k is equal to 1 M. And what is M, you remember that as per our original definition this was our associative memory model ((Refer Time: 52:29)). So, we had M inputs and we had got M outputs, so that we can say that this network is having an M dimensionality.

And we have somehow restricted, the total learning within it is dimensionality. That means, to say that if there are 100 neurons in this associative memory model, we are saying that we are not going to teach this more than 100 patterns. We can then see that whether our assumption is correct, if it is capable of learning more, we will teach him more, but if it is not then we have to restrict. So, that is why just to make a restriction I say that, if there are M inputs I am not going teach you, more than M at least should be less equal.

So, I mean we can just teach up to M patterns only and now what we can do is that, this summation that we have got it is very much possible for us to rewrite this product expression. And I can say k is equal to 1 to m and I can now associate x k transpose x j transpose, now x j x k transpose x j's. So, I associate this and then I multiply it by y k I can do that, I can do that and this now I expand, this is for k is equal to 1 to m.

Now, very interestingly when in summing it up for k is equal to 1 to m, one of the indices that it has to pass through in the summation is when k is equal to j. And when k is equal to j, then very interestingly this term becomes x j transpose x j times y j and then other than j there will be all other M minus 1 term. So, what I do is that this summation that we have got, we are simply separating it out between the jth term and the non j terms summation.

(Refer Slide Time: 54:51)



So, we can write it as y vector this could be represented as x j transpose x j y j. So, taking the jth term out and then keeping the summation as it is from k is equal to 1 to m only restriction is that not for k equal to j. So, it is for k not equal to j that way sum up the rest of the terms keeping as it is x k transpose x j y k this terms remain as it is. Now, what is it, what is this term can you interpret this, this is the yes this the magnitude of the jth pattern, the squared magnitude of the jth pattern and what is that in effect it is the energy.

Now, we can say that if each of the patterns x 1 x 2, etcetera, etcetera up to x m if all these patterns are normalized to have unit energy. So, if this is having unit energy, then what is the energy expression, the kth vector let us say the kth x vectors energy will be summation of x k l square n and l summed up from 1 to m. In this case this is going to be x k transpose x k.

So, if this is equal to 1; that means, to say what that in this expression we substitute x j transpose x j to be equal to 1. And in that case what do we get, you get y is equal to y j plus this term, now you expected y to be equal to y j and there is one additional term, which is coming in between, which is creating the trouble for us, this additional term that we are getting is essentially a noisy term. So, in next class next class we will see that how to deal with this noisy term.

Thank you very much.