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

Lecture - 33
Dimensionality Reduction Using PCA

Our topic for today is Dimensionality Reduction Using the method of Principal Components Analysis, short form we write that has PCA. So, this is what we will be discussing in this lecture. Now, we already introduced the basic theory. That is, behind principal components analysis. In fact, we say that the basic objective is to have a reduction of data, if the input data that we are having is m dimensional.

We should remap this, in this input data into a lower dimensional space. In such a manner, that we can eliminate in that mapped space. We can eliminate the those components, which are contributing to very low variance. So, that was the idea, so what we said is that we have to design the best kind of a transformation t, which can do this kind of a job that. So, that if f x vector is mapped into t x vector over t is an m by n matrix.

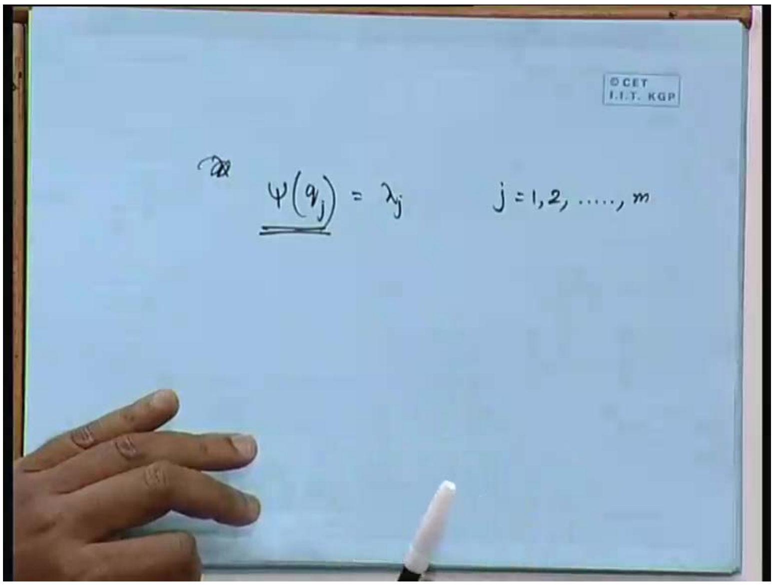
The basic idea is to find the matrix t. And what we did in that process is that we decided to project this vector x. On to a set of vectors, on to a set of m dimensional input vectors, which we call as q y's. So, we were projecting that into q y's. And then, the solution that we had got under the extremal case mind you, we said extremal. Because, there we did not specifically in fact as very rightly pointed out by one of the students. That, the extremal does not necessarily, mean that it is a minimum. We, derive the condition of extremality of the variance.

And that condition laid to that the solution that we are having, basically boils down to the solution of the Eigen values of the co-occurrence matrix R. So, by a having the correlation matrix R and finding the Eigen values of that, where Lambdas or the Lambda Is of the Eigen values. And their corresponding Eigen vectors or the vectors on to which we would like to project the input vector into those vectors.

So, the thing is that the solution naturally is obtained for m number of Eigen vectors. Because, if our correlation matrix is of size m by m. In that case, we are going to have m different Eigen values solutions for that you know. So, the thing is that unless we definitely know, that how to reduce the dimensionality. We should first except that till know we have not saved anything. We had started with m dimensional x vector and after projecting into the Eigen space.

We have got m such different unit vectors, which are possible Q 1, Q 2 up to Q m those are the m different vectors, that we have to a unit vectors we have to which basically this x can be projected.

(Refer Slide Time: 04:55)



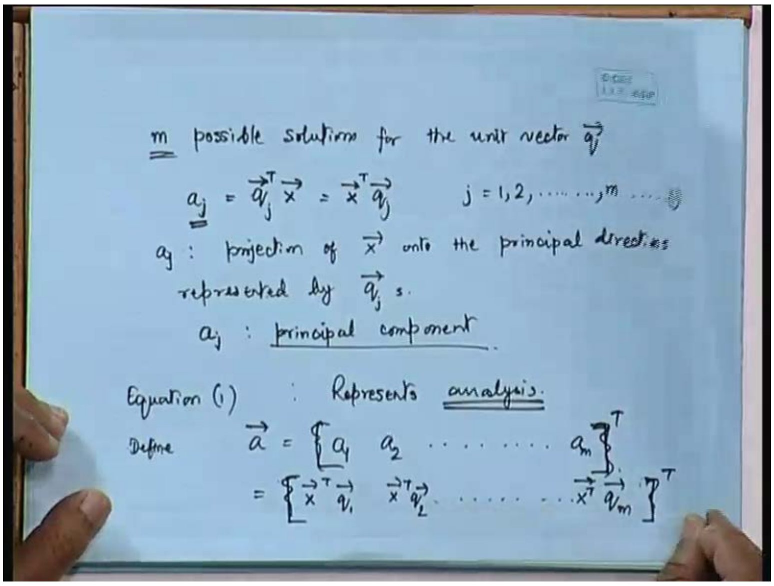
And another very important result, that we had obtained there was that the variance we had used the q j as a variance probe. So, we used this thing as psi of q j, where psi was used as the variance probe. And we had shown a very important relation, that psi of q j is equal to Lambda j, where Lambda j is the jth Eigen values. So, we definitely have such things done for j is equal to 1, 2 up to m, since there are m such Eigen value.

So, this is very interesting to note that what is psi of q j, it is the variance probe. So, the variance is equal to the Eigen value. So; that means, to say that if we can arrange the Eigen values in a descending order, that is to say Lambda 1 being the highest Eigen value, Lambda 2 the next one up to Lambda m. Lambda m being the lowest of all the Eigen values that we have got.

So, if we can arrange the Eigen values in a decreasing order. In that case, we should be able to decide that, if we decide to eliminate the last m minus l, where we decide to keep l Eigen values and we decide to eliminate the last m minus l Eigen values. In that case what we are essentially eliminating a are the components with low variance. Because, now in the projected space, in the projected q j space eliminating low values of Lambda j, means that we are eliminating the components with low, we are eliminating the components with low variance in the projected space.

So, this we have to keep in mind. And now, we have to talk about the way whereby we can do the dimensionality reduction. So, for that matter what we do is that, we have how many we have got m possible solutions for the unit vector.

(Refer Slide Time: 07:08)



So, there are m possible solutions for the unit vector q. So, there are m possible projections of the data vector. So, what we get is that, if a j is the jth projection or projection of the x vector into the jth unit vector. So, what we are looking for is by projecting this x vector into jth unit vector. So, what we have to do q j transpose, q j transpose x, that becomes the jth projected value of x into this q j.

And this can be also represented as this is one and the same way of writing x transpose q j. And we have got since there are m possible solutions, this we have to do for j is equal to 1 to m. Where, a j's are the projections, so a j's we call as the projection of the x vector on to the principal directions as represented by q j. And as a j's are projections into the principal directions, as a j's are projection into the principal directions. This a j is called as principal component.

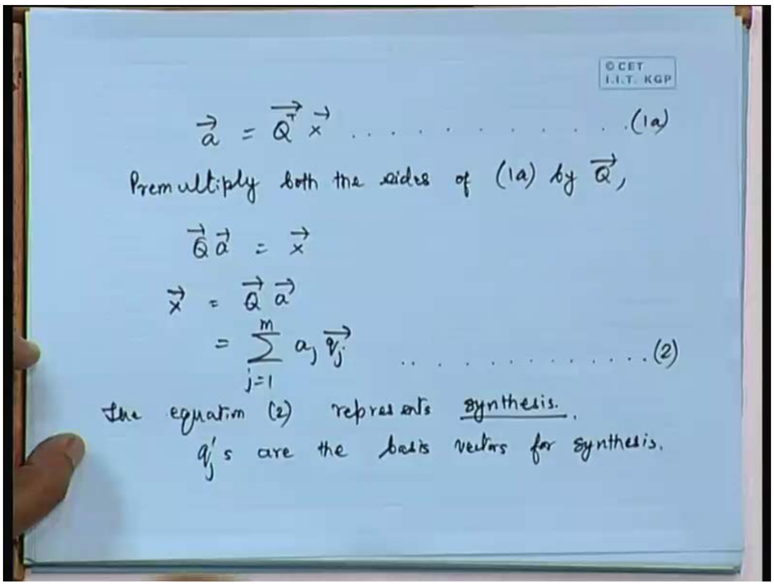
So, that is why the name principal component originates. So, basically we are going to have m such principal component. And out of that, which ones are very important to us, those are the things which we will be discussing very shortly. So, now that we have got this equation call it equation number 1. Then, this equation number 1 we are going to call as an analysis equation.

So, equation 1 basically represents the analysis. So, what is meant by analysis, where we are projecting x into a new space. We are projecting x into the q j space and then the projections that we are getting are the principal components. So, this is called as the analysis. And our objective should be that, if we can project this vectors into this q j space and do the analysis, we should be able to do the reverse of that also. What is the reverse of analysis? Synthesis.

So; that means, to say that in the synthesis, what we are going to do is that from these a j's we should be able to recover the x. So, in order to recover that, so what we do is that, since there are m such number of a j’s. Since, a j varies from j is equal to 1 to m, we can form a vector consisting of the projections. So, we can define a projection vector. So, we define a projection vector call it as a vector, whose elements are going to be a 1, a 2 up to a m, there are m such projections.

In fact, these in the expended form one can write as x transpose q 1 vector. The second element will be x transpose q 2 vector. Like that the m'th element will be x transpose q m vector. So, we write it like this customarily we should represent it with as square bracket. So, this we should represent as transpose this also is transpose.

(Refer Slide Time: 12:02)



And this is one in the same as writing it in the matrix notation as we can write it as a vector is equal to Q matrix, Q matrix is something that we have already define in the last class. How is the Q matrix found? The Q matrix is found using the Q Q i vectors, using all the Q i vectors, Q 1, Q 2 up to Q m, we had found the Q matrix. And Q matrix had got a unique property what is that? It is the...

Student: ((Refer Time: 12:31))

Orthogonal matrix. So, Q matrix was an orthogonal matrix, so we have got basically that a is represented, this is one an the same thing as the analysis only, what we have done ((Refer Time: 12:48)). So, basically there is no difference between our equation number 1. And if we call this thing as let us say, equation number 1 a, basically there is no difference, it is only combining all this j terms, 1 to m all this m different a j's are been combined in the form of a vector.

And all that Q i's are combine in the form of that Q matrix, nothing else it is the same thing. Now, we do one thing, now we pre multiply both the sides of this equation using Q, what will you get? If you pre multiply both the sides of equation 1 a by Q. What are you getting on the left hand side?

Student: Q.

You are getting Q a, and what are you getting on the right hand side?

Student: X.

X, you are getting x. Why, because it is Q Q transpose and Q Q transpose is equal to the identity matrix. Because, Q matrix has been already prove into be the orthogonal matrix. So, that is why this is Q Q transpose Q Q a is equal to x, so in fact, x can be represented now as in the expanded form, we are going to write it as summation of a j q j vector for j is equal to 1, 2 m. This is the expanded form of the matrix equation, that we have written in this step.

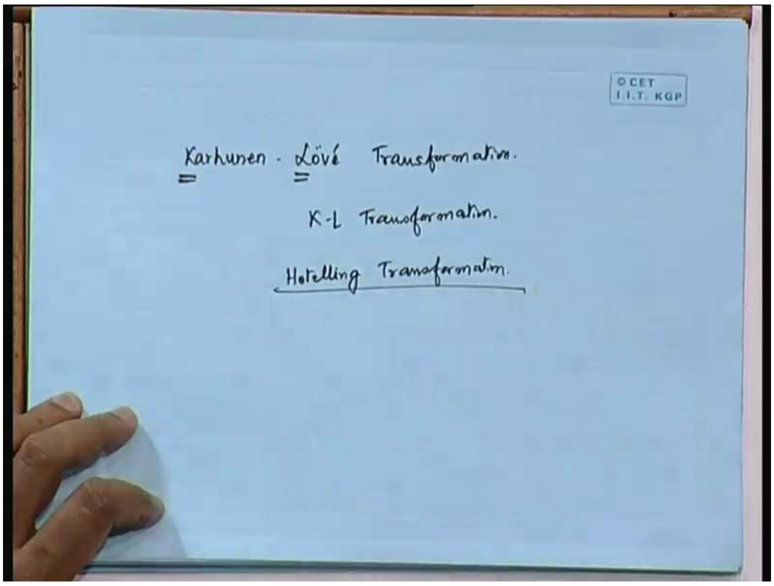
It is just the expanded form of this. And let us call this thing as equation number 2. So, what is equation number two representing? Synthesis, so the equation 2 represents synthesis. And in order to have this synthesis, what are our basis vectors for synthesis? The q j's act as the basis vectors for synthesis, q j's are the basis vector for synthesis.

This is a very interesting observation, that you see unlike the typical orthogonal transform kernels, that you use Fourier discrete cosign sign transform etcetera, etcetera.

You do not use any specific function as the basis function. Instead you are deriving this by analyzing the correlation matrix. You are finding the Eigen values of the correlation matrix. And the corresponding Eigen vectors that you are getting are essentially forming the basis vectors. So, you are observing you are forming the basis vectors from the random process itself, there are lies the beauty, you do not have to depend upon any external kernel to do that.

You are basically forming the basis vectors, out of the correlation matrix of the same random variable. So, in that sense this transformation technique has become very popular. In fact, it is used in many domains of signal processing. In fact, in the signal processing community this approach of principal component analysis is very often refer to as Karhunen Love Transformation.

(Refer Slide Time: 16:50)



In fact, people do not use the long names of Karhunen and Love a. And there, they use K and L, so people simply call it in the short form as KL transformation. In fact, in the image processing community people follow the two dimensional version of this. And there the transformation that they use is also known as the hoteling transformation, this also is based on the same principal component analysis.

So, in different variants it is available and people have been using this approach. And essentially it is main use is dimensionality reduction. And because of this property of dimensionality reduction it is used very effectively for the purpose of data compression. When we need to compress large amount of data, especially speech data, video data, which has to be transmitted over a very limited band width, there we essentially required this type of transformation techniques.

Now, stills some doubts must be in our mind that, this a dimensionality reduction or data reduction, data compression type of techniques. Then, why are we at all covering it under the topic of neural network. Till now that question must be cropping into our mind, which I will be explaining pretty shortly. Just to show you, that essentially the neural network, if we can construct a neural network using Hebbian learning mechanism. Hebbian learning is something that we have discussed much earlier in the beginning part of our lectures.

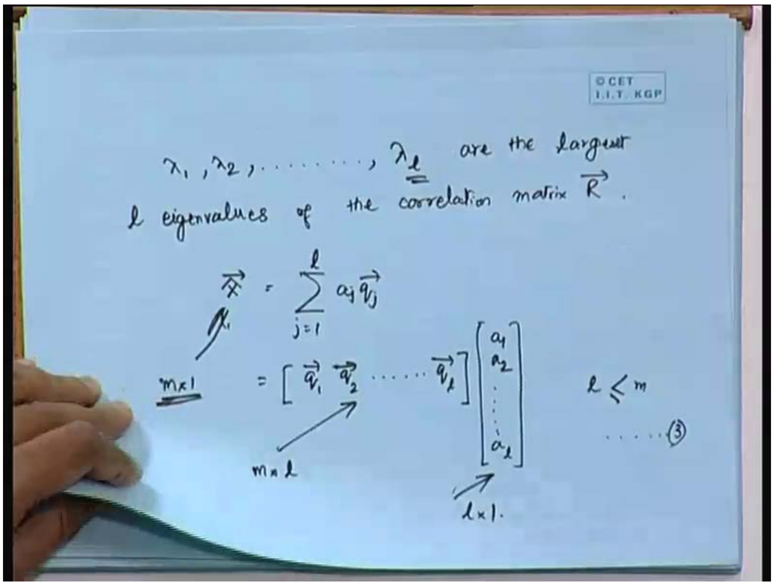
We have talked about, while explaining about the different learning mechanisms, we are talked about Hebbian learning. In fact, if we have a Hebbian learning network, we consider a single neuron and we consider m inputs connected to that neuron, it can be shown by analyzing mathematically. This was a very break through result one can tell, that it can be prove that using the Hebbian learning mechanism. The a single linear neuron, essentially performs the task of finding the first principal component.

What is first principal component? That is corresponding to Lambda 1, because if we arrange the Eigen values in the decreasing order and Lambda 1 is the highest Eigen value. Then, with Lambda 1 we will be having an associated vector Q 1. So, it is we are referring to the maximum Eigen value. So, it leads to the maximum Eigen value solution, neuron which is learnt with Hebbian learning mechanism.

So, that is why principal component analysis problems can be very easily modeled using neural networks, which follow Hebbian learning. That, we will be discussing little later on off course, without going into the details of mathematics. Because, the mathematics part of it will be fairly involved it itself will cover quit a few number of lectures. And that may disturb the trust of this course.

So, that is why I will be only giving you some references, from where those who are interested in learning, the inter cases of the mathematics can go through that. Any way before we do all those things, we have to first of all talk about the dimensionality reduction, which was in our I can refer today. So, what we do is that we have got in the we can project the vectors into the Q i spaces and we got all the different projections.

(Refer Slide Time: 21:28)



And let us say, we have the Eigen values which are arranged in the descending orders. So, we have got Lambda 1, Lambda 2 up to Lambda l, which denote the largest l Eigen values of the correlation matrix R. So, Lambda 1 to Lambda l are the largest l Eigen values of the correlation matrix R. In that case we can approximate the data vector, see the solution of the data vector x is we know that already ((Refer Time: 22:24)) the synthesis equation says that, you have to have summation of this a j q j. And this summation mind you please note here is j is equal to 1 to m.

Only if we sum up for j is equal to 1 to m. Then, we can get back the exact solution of x. But, now that we have arranged all our Eigen values in a descending order and the corresponding Eigen vectors, which are associated with it are q 1, q 2 up to q l we choose the first l. Now, let us decide to truncate this summation series considering the first l Eigen vectors only.

So, if we do; that means, instead of m terms we take the first l terms, naturally we are not reaching at the exact solution. So, whatever solution we find for x cannot be called as x any more, we have to call that as x hat. So, this is x hat as vector would now be the summation of same thing a j q j only, but a j q j summed up for what?

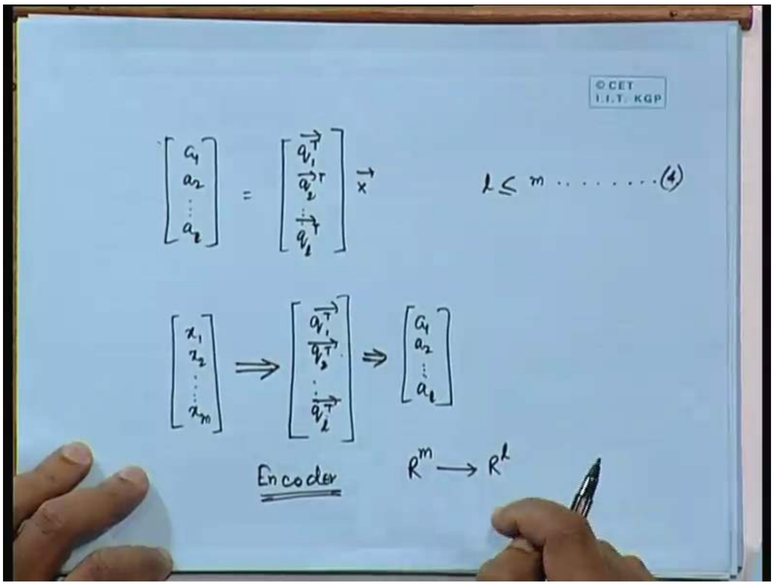
Student: ((Refer Time: 23:47))

One to l yes j is equal to 1 to l. Because, we are taking the l terms, it is up to Lambda l please note. And this is the summation representation and the same thing can be represented like this also in the form of just showing in the vector expression itself. That we can write it as q 1, q 2 up to q l. And then, here we are having the a vector arranged as a column vector. So, q arranged as row vector and a is arranged as the column vector. And what is the relationship between l and m? L is...

Student: Less than m.

Less than m. In fact, in the limit l could be equal to m also, if l is equal to m in that case we are not doing any truncation effectively. So, if l is equal to m, then x hat becomes equal to x only, but in general when which happen to choose l to be less than m. Then, x hat is different from that of x. So, this we are calling as the equation number 3. Now, how we can compute this a l's, that we already know. That in order to compute this a l's first off all what we have to do is to project this x vectors on to we did not have to project to all the m q i's, we can project then in to the first l qi's.

(Refer Slide Time: 25:38)



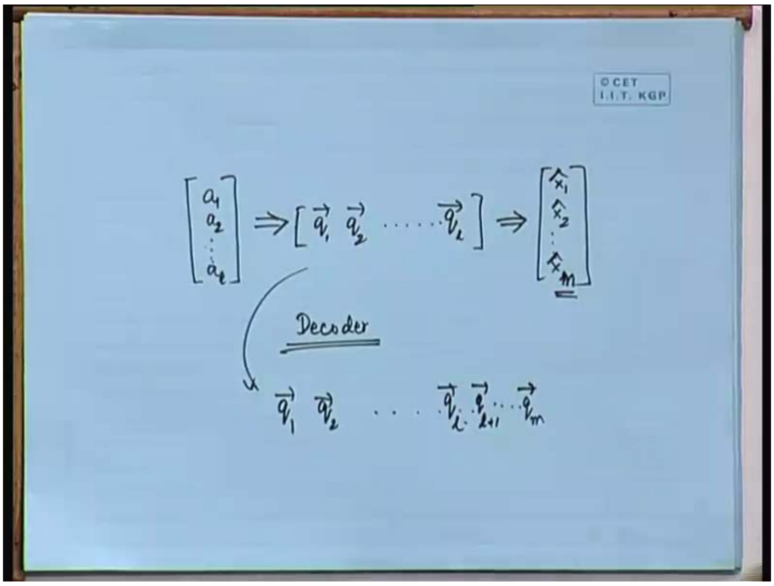
So, we can obtain the a vector as a 1, a 2 up to a l this vector can be formed as q 1 transpose q 2 transpose up to here q l transpose, this multiplied by the x vector. And this is for l less than or equal to m. So, this we can call as the equation number 4. So, basically what does it mean, if we so; that means, to say that using the q i's, we can transform the x vector, x vector is of original x vectors dimension is m.

So, we will keep all the m elements of this x vector. But, we will project all this m elements or we will project the x vector together along with all this m elements into an l dimensional space and those l dimensions are found out of the q i's. So, we will take like this, so we will be taking x 1, x 2 up to x m this is our input. And then, this input will be transformed by this q 1 transpose q 2 transpose up to q l transpose.

And mind you a l is less than m, so; that means, to say the this dimension is less than this. So, as a result of that what we are getting is that, this will get mapped into a 1 a 2 up to a l. So, from an m dimensional input space, we are getting an l dimensional a component projections. So, this much will be done by the encoder, this is our analysis part, so the analysis part will be done by the encoder, which basically maps the m dimensional input space.

So, we call it as R m from R m it gets mapped into R l space. So, this is m to l dimensional transformation being done by the encoder. And in the decoder part just the opposite of that is going to happen.

(Refer Slide Time: 28:30)



So, there what we are going to do is that our input will be the a's a 1 a 2 up to a l that is going to be our input. And we have to transform it using what? In order to get the x q vectors, last time it was the q transpose and this time we are going to take the q vectors. So, if we take all the q vectors q 1, q 2 up to q l this ultimately gets transform into x 1 hat x 2 hat up to x l hat.

And why hat again, because this is not an exact solution, this is an approximated solution, approximated, because of the truncation of the l terms. So, this is getting mapped into this, so this is our decoder part. In fact, the this way one can practically use this kind of principal component analysis, for doing the encoder and decoder and many practical image and video coders or speech coders can be realized using this principal component analysis technique.

Now, the only difficulty that one can find about this is the million dollar question is that, how to determine all these q's practically, it is easy to talk that we are talking in terms of the Eigen values of the correlation matrix. But, somebody has to really form the correlation matrix by making several observations of the input variable. And then, finding the Eigen values of the correlation matrix is also a computationally involve problems. So, let us not...

Student: ((Refer Time: 30:43))

Yeah sorry should be yes, somebody has very rightly pointed out that we are getting here up to x l. And in fact, just see that it is x l only that no it is not x l, ultimately we are getting the x m. So, this is no yes thank you for the correction. Because, this definitely it will be clear if you look at the original expression only, we have all this a j's q j's. Only thing is that we are summing up to j is equal to 1 to l here ((Refer Time: 31:37)), j is equal to 1 to l we will sum up.

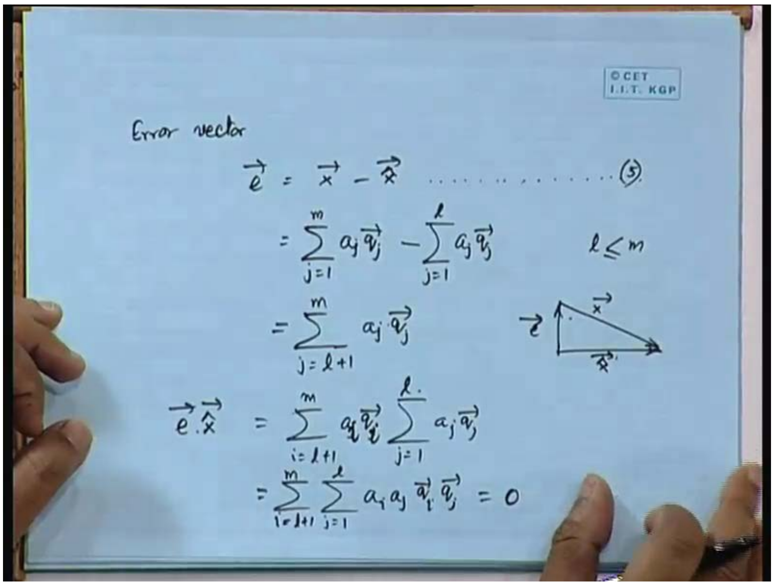
But, this leads to an x vector not an exact x vector. But, an approximated x vector, but the dimension of x vector will be preserved. So, what we are going to get is ultimately an m dimensional x vector only, but because it is different from the original x vector elements. So, that is why I am writing it as x 1 cap x 2 cap up to x m cap, thank you pointing this out, so this has to be x m.

Let us come back to what we were discussing that the encoder and the decoder structure looks pretty simple ((Refer Time: 32:24)). That one has got this or one has got this in the decoder and then one can easily do the transformation. Because, the transformation is nothing but, a projection and that is very easy to do. But, the million dollar question is that how to determine all these q i's? Because, finding the Eigen values problem is a computationally involved problem.

So, that is the reason why one has to study, that whether this could be carried out using neural network or not. Now, before we go into the neural network implementation of the principal component analysis, little bit of time we should spend at the error. Because, we have just now seen that what we are getting at the reconstructed thing is not the exact x vector, but it is the x cap vector. So, if it take the vector difference between the x and the x cap. That is, going to be our error vector.

So, let us try to find out the expression for the error vector. And in fact, as expected naturally I will not tell you that, you will automatically come to know when you find out the expression for that.

(Refer Slide Time: 33:50)



You see the error vector will be represented as e vector equal to x vector minus x cap vector. And let us call this thing as since we have gone up to equation number 4, let us call this as equation number 5. Now, if we substitute the equations 2 and 3 into this 5, you say why we are looking for equation 2 and 3, equation 2 says this you say ((Refer Time: 34:29)) equation 2 tells us that x is equal to summation j is equal to 1 to m a j q j.

So, if we now write it down this is equal to j is equal to 1 to m a j q j, this is the substitution for this x. And the substitution for the x cap term is going to be you see ((Refer Time: 34:59)) x cap is summation of a j q j only, but j is equal to 1 to l. So, this is equal to j is equal to 1 to l a j q j. And in this case what we are having is that l is less than or equal to m.

So, what we can see from this is that essentially this can be represented as yes somebody can tell me compact way of represent, in this summation j is equal to 1 to m.

Student: ((Refer Time: 35:34))

And summation j is equal to 1 to l and l is less, so.

Student: ((Refer Time: 35:39))

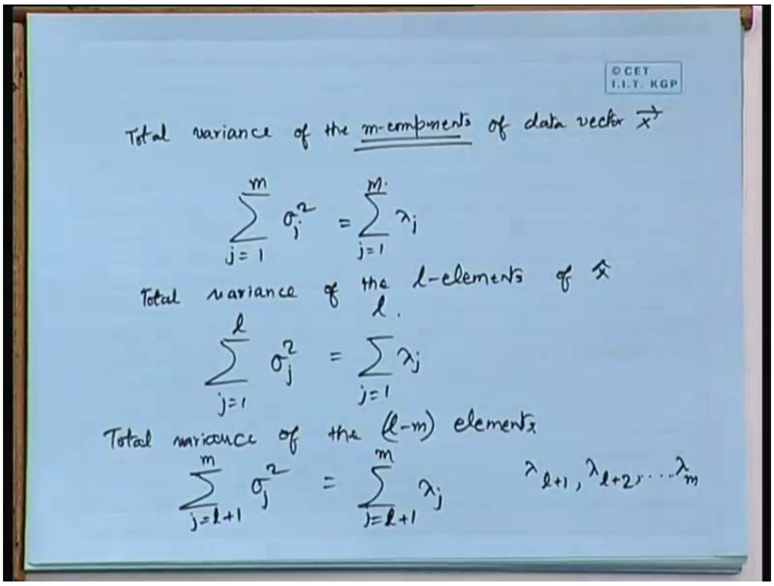
l plus 1 to m. So, I can represent the same thing as j is equal to l plus 1 to m a j q j vector. So, this is our error term and now let us calculate the errors dot product with the x cap vector. So, what are you expecting just see, but you are getting, this is i is equal to l plus 1 to m, we just simply substitute the expression that we have got for e. So, e is a j q j vector. And what is the next term that is x vector, x vectors term is a j q j. In fact, here this is a j q j, j is equal to 1 to l.

Now, yes somebody has already guessed that it is becoming 0. That means, to say that error and the x cap vector, they are orthogonal to each other expectedly. In fact, you see that this is the x vector, if you see and if this is the x cap vector, then this is going to be our error vector. And just see that this one essentially becomes i is equal to l plus 1 to m. In fact, this a i a j's they are all scalar quantities, so no problem. And here we have j is equal to 1 to l i am taking this summation.

And then it is a i a j they remain as it is and from here, we are getting what we have done is this one I have made it wrong. You did not point it out this is a i q i, because here I have chosen the index to be i and here I have chosen the index to be j. So, although I substituted I should have written down here as the index as i. So, it is a i a j we have taken from here, but then we are having q i and q j. And basically because q i's are all orthogonal to each other.

So, q i q j product is definitely going to be 0 in this case. Because, here you see that i is not equal to j very clearly. And we are getting this term to be equal to 0, so that the error is orthogonal to x. And another thing that one needs to observe is about the variance.

(Refer Slide Time: 38:46)



You see the total variance if we now look at it. That is to say, variance of all the j terms, if we total variance of the data vector of the m components of data vector, original data vector. That, variance we can write it as sigma square j, j is equal to 1 to m. So, this is the total variance. And what is that what is the sigma square expression for us, remember Lambda j, for us the sigma square j is as same as the Eigen values.

So, this is equal to summation of Lambda j, j is equal to 1 to m and so if this is the variance of the m components, then the total variance of the l elements of the approximating vector x. So, total variance of the l elements of x vector, if we take the l elements of that, this is summation j is equal to 1 to l sigma square j which is equal to summation Lambda j, j is equal to 1 to l.

So, from here we can find out that the total variance of the l minus m elements. This is going to be summation j is equal to l plus 1 to m of sigma square j, which is equal to summation j is equal to l plus 1 to m of lambda j. So, this is the variance that we are chopping off. Because, what we are doing is that the Eigen values which are represented as lambda l plus 1 lambda l plus 2 up to lambda m, these Eigen values are the smallest Eigen values.

So, whatever we chop off out of the l minus m terms. In fact, it will not be very correct to say, that this is the total variance of the l elements of x vector is one on the same as l elements of a. In fact, we are considering l elements of a. And because, we have chopped off the l elements of a vector, the variance of that is also equal to summation of lambda j.

So, in terms of the total variance this is the difference that we are getting by eliminating this lambda l plus 1 lambda l plus 2 to lambda m. And these are the smallest lambda's, which we are eliminating. So, we are eliminating the minimum variance terms mind you, we could have done the reverse also if we had chopped off the, if we had accepted the extremal solution. Because, Eigen value says that it is an extremal solution.

Now, some body can take the first l Eigen values and chop off the rest m minus l, somebody can say that no I will take the first l. And I will eliminate the rest, because last class I remember somebody was talking in terms of maximization of variance that, was interesting. And maximization of variance yes somebody can do if we keeps the minimum variance terms and chop off the terms contributing to the things.

So, that way even maximum variance solution is possible. But, we are going in for a minimum variance solution. And the very fact that is the minimum variance solution is obtained like this, that order all the lambda's in the decreasing order and then pick up the first l components of that, take the corresponding Eigen vectors. Project the x vector onto those Eigen vectors, onto those dominant Eigen vectors now we can say or what we are getting are the dominant principal components.

So, the first one to which we have got the projected result is a 1, this we will be calling as the first principal component, a 2 we will be calling as second principal component and so on a 3 as the third principal component and so on. Provided lambda 1 is the highest Eigen value, ((Refer Time: 44:11)) in lambda 1 is the highest Eigen value. So, this is the way so, by chopping off the l minus m.

Student: ((Refer Time: 44:21))

You are very correct, because m is larger than l. So, we have to chop off the m we have to chop off the m minus l elements. So, that is very correct, so this is what we get. So, now this way they dimensionality reduction will be the most efficient. Because, we are only eliminating the low variance terms by cleverly projecting our data into the Q space anyone having any doubt about this approach yes please.

Student: ((Refer Time: 45:09))

In the decoder yes, the decoder output is getting the x cap components.

Student: ((Refer Time: 45:17))

You are no, you are chopping the a components followed just look at the encoder and the decoder block again I do not want any confusion in your mind. You see ((Refer Time: 45:42)) this is the encoder m elements getting mapped into l elements, it was originally suppose to be m elements here. But, I decided to chop off, because I am chopping of this in accordance with minimum variance criteria.

Now, I have got this a l's and using this a l’s I transform this things with q i’s ((Refer Time: 46:10)). And my summation the expression that I have got for x vector, says that it is a q that is what we are getting as the solution of x.

Student: ((Refer Time: 46:24))

Q’s are yes q’s are of m dimension this is a very interesting point, that we are not keeping the a m dimensions of l, we are keeping m dimension of q we are keeping only the l dimensions of q. So, what is happening to the rest the rest’s...

Student: ((Refer Time: 46:53))

Exactly, the rests are all the 0 vectors, you see from q 1 to q l you are remapping them into q 1, q 2 up to q m. How up to q l you are keeping the original matrix. And q l plus onwards you are chopping off. So, you are as if to say, assuming that those vectors are equal to 0 vectors. So, that is why you are getting as a q as a solution of a q, that is what we get just a minute we go back to the...

Student: ((Refer Time: 47:41))

We go back to the synthesis equation, that is what I was looking for. Now, we will go back to equation 2 only ((Refer Time: 47:51)) equation 2 is our synthesis equation, you see there some of these vectors forming this Q matrix, they are becoming 0 vectors. So, but they are still of dimension m. In fact, in effect what is happening is that, you are chopping off the m minus l elements from here, you are chopping off m minus l elements from here to. But, you are keeping them as zero's and then you are in effect getting the x vector solution.

Student: ((Refer Time: 48:33))

One minute yeah that is right.

Student: ((Refer Time: 48:55))

Yes, yes, yes.

Student: ((Refer Time: 49:02))

Yes.

Student: ((Refer Time: 49:07))

Yes.

Student: ((Refer Time: 49:09))

X cap you have to get as m dimension.

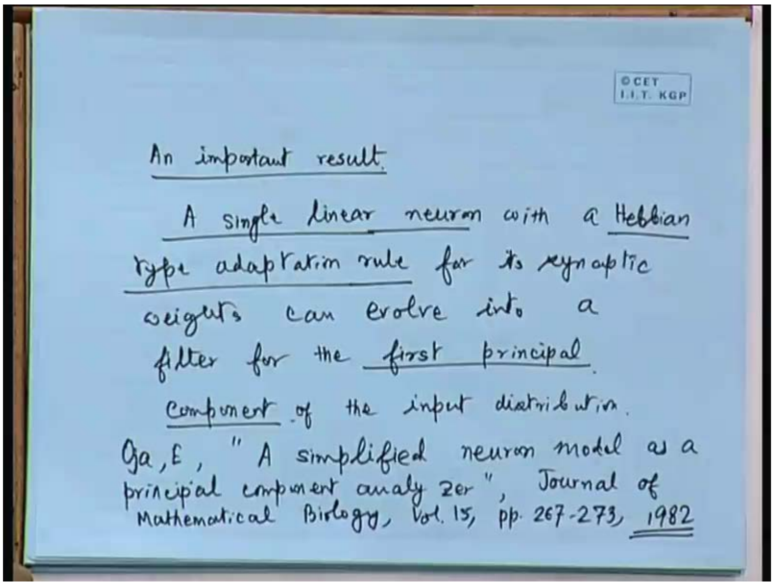
Student: ((Refer Time: 49:13))

Yeah this point is no effectively it is that only, you see that we are keeping the l components of a's and we are similarly keeping the l components of this Q vectors. So, essentially yes you are right that these things are all of m. So, these are m by l matrix.

Student: ((Refer Time: 49:49))

So, this is m by l and this is l by 1. So, dimensionally x vector is going to be an m by 1 vector. So, even after the chopping we really get m dimensional vectors. So, that there is no confusion. So, you do not have to really going to the complication of making those things 0, because those things zero's we do not need it, we can directly take this transforming to the q l's and we get all the x 1 to x m's is that clear in peoples mind now.

(Refer Slide Time: 50:41)



So, now yes one important result what I was telling you is that. So, this is an important result, which is of relevance to us in the neural network. That a single linear neuron with Hebbian type of learning, Hebbian type adaptation rule for it is synaptic weights, can evolve into a filter for the first principal component. First principal components definition I have already given.

So, a Hebbian type adaptation rule single linear neuron having Hebbian type adaptation rule, will give you the first principal component. This is a very important result and this was pointed out by Oja. That is the name of the scientist who did that and the paper were he described this is a simplified neuron model as a principal component analyzer. And this was published in journal of mathematical biology, volume 15, page numbers 267 to 273 and it was a 1982 paper.

So, this is a very interesting result. And we will be accepting this result as it is without going in to the detailed mathematic analysis for that. So, in the coming class what we will be doing is, that we will take a linear neuron model and go through the Hebbian type adaptation rule. And we will show that, how this theory can be expanded into obtaining not just first principal component. But, if we decide to keep l number of output units in that case, we can show that we can extract the first l principal components.

So, whatever we intend to do. So, computation can we very easily modeled using the neural networks with Hebbian adaptation rule.