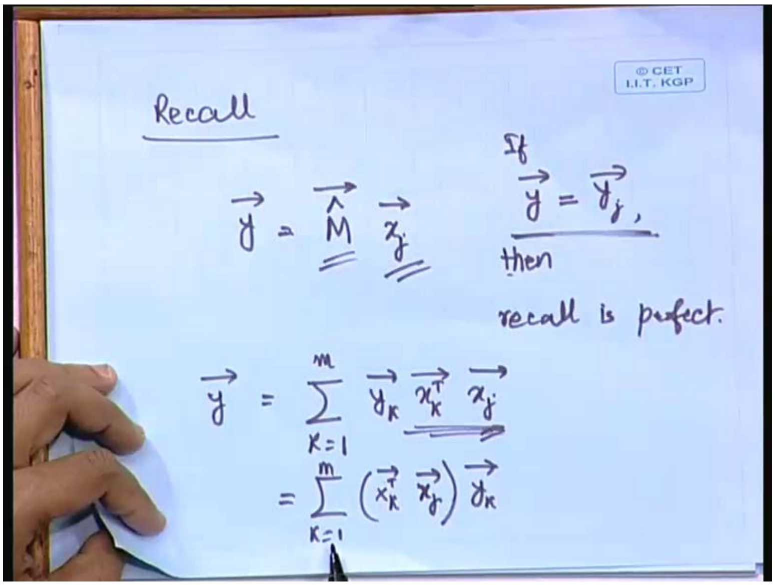
Neural Network and Applications

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

Lecture - 08
Conditions for Perfect Recall in Associative Memory

Conditions for Perfect Recall in Associative Memory. Actually after going through the condition for perfect recall, I am going to tell you about the next topic that we are going to cover that is about the statistical aspect of learning. So, that will in fact, complete the chapter on the learning processes. So, let us try recollect what we were doing in the last class.

(Refer Slide Time: 01:15)



Recall

$$\vec{y} = \hat{\vec{M}}\,\vec{x_j}$$

If $\vec{y} = \vec{y_j}$, then recall is perfect.

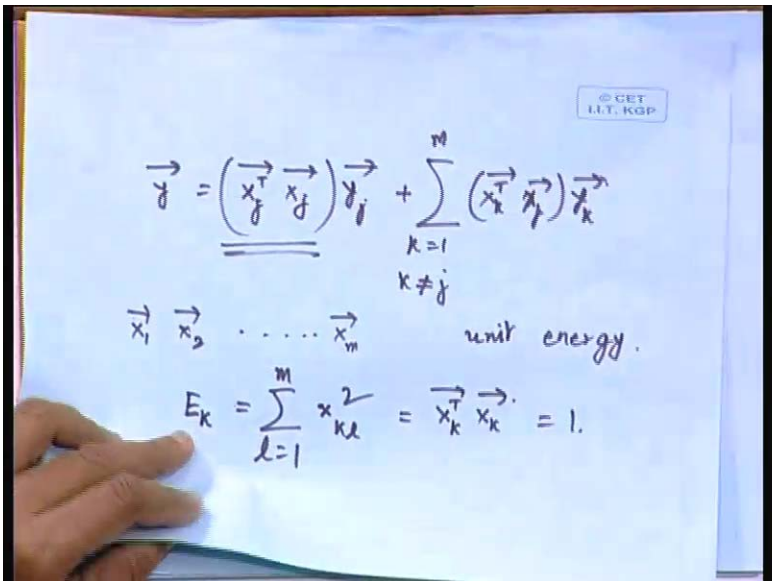
$$\vec{y} = \sum_{k=1}^{m} \vec{y_k}\,\vec{x_k^T}\,\vec{x_j}$$

$$= \sum_{k=1}^{m} \left(\vec{x_k^T}\,\vec{x_j}\right)\vec{y_k}$$

So, as you remember that we had and estimated memory matrix M cap which was in fact, realized out of the correlation in the way that, we had consider that M matrix to be the outer product of y k and x k transpose. And then what we did was that we had fed pattern x j we have just happen to pickup particular pattern x of j. And then, we got the output y as the product of this M matrix and x j vector.

Now, what we are expecting out of this is, that y is equal to y j of vector and let us see that what be obtained in the process. So, what we did was just to apply the definition of this M matrix, this M cap matrix the way be defined it as the out outer products, so by replacing this with the outer product of the vectors y k and x k transpose. And then, we just associated this x k transpose with x j and this is what we had got, as the summation x k transpose x j times y k.

(Refer Slide Time: 02:38)



$$\vec{y} = \underline{\underline{(\vec{x}_j^T \vec{x}_j)}}\vec{y}_j + \sum_{\substack{k=1 \\ k \neq j}}^{M} (\vec{x}_k^T \vec{x}_j)\vec{y}_k$$

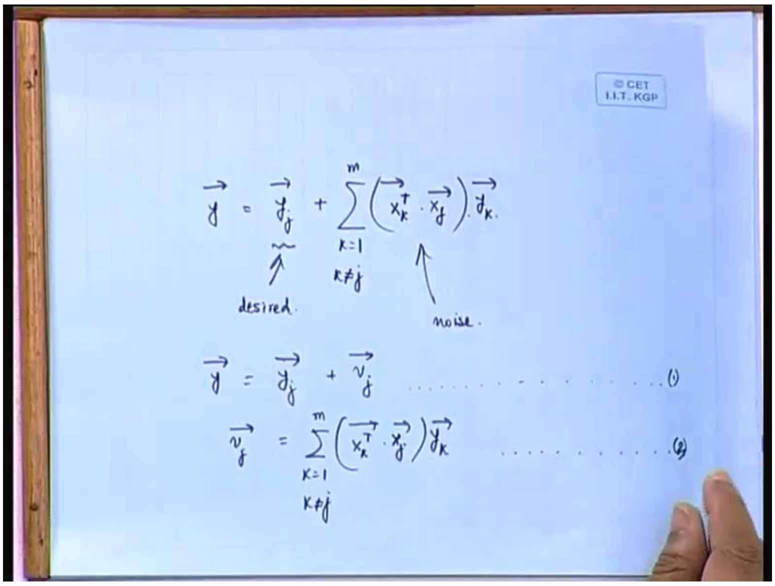
$\vec{x}_1 \ \vec{x}_2 \ \ldots\ldots \ \vec{x}_m$ unit energy.

$$E_k = \sum_{l=1}^{m} x_{kl}^2 = \vec{x}_k^T \vec{x}_k = 1.$$

And this was from k equal to 1 to M which we had split up into the j term and the summation of the non j term k not equal j and k equal to 1 to M we took as one side and on the other out of this summation we have taken out, the case where k is equal to j. So, that from this expression we were obtaining by substituting k is equal j we were obtaining X j transpose x j times y j all vectors mind you plus the summation term of this one remaining as it is except for the fact, that here we are summing up for k not equal j for k is equal to 1 to M.

And we had in order to simplify we had further more assume, that all these vectors they are having unit energy. So, that the norm of all these vectors will be equal to 1, in other words if we calculate the E k of the kth of any vector, let us say the kth vector. Then, E k will be equal to summation of x k l square l equal to 1 to m and we just added it up and it becomes equal to 1 alright. So, now what we can simply do is that, we can right down that y equal to y j plus this term.

(Refer Slide Time: 04:18)



$$\vec{y} = \underbrace{\vec{y}_j}_{\text{desired}} + \underbrace{\sum_{\substack{k=1 \\ k\neq j}}^{m} \left(\vec{x}_k^T \cdot \vec{x}_j\right) \vec{y}_k}_{\text{noise}}$$

$$\vec{y} = \vec{y}_j + \vec{v}_j \quad \ldots\ldots\ldots\ldots (1)$$

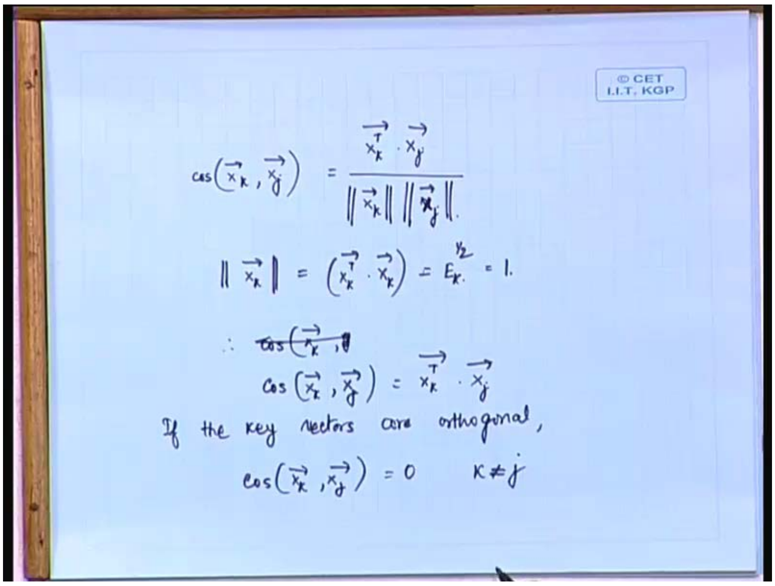
$$\vec{v}_j = \sum_{\substack{k=1 \\ k\neq j}}^{m} \left(\vec{x}_k^T \cdot \vec{x}_j\right) \vec{y}_k \quad \ldots\ldots\ldots\ldots (2)$$

So, let us see that we have already obtained that y equal to y j to plus this term which is for k equal to 1 m and k not equal to j x k transpose vector and x j vector, the product of this two times y k vector. So, this the relation, now this one is very simple this is have a desire term and this is the term that contributes to noise. And if we have talking in terms of the conditions for perfect recall, then what we need to do is that, this term should ultimately be equated to 0.

So, let us see that what conditions are the getting for that let us now see that. So, now let us represent this as y vectors equal to y j the desired vectors plus the noise again in the from a vector we can write v j vector. Because, this after all is this x k transpose x j, this being the dot product, that leads to a scalar quantity times y k. So, in other words this becomes an additional vectors, so it is a vector. So, again m dimensional vector this point will be..

So, here if we right this expression by y j plus v j vector, then v j vector which is nothing but, the noise vector will be summation k equal to 1 to m k not equal j x k transpose vector x j vector times y k that is to say this term all right. Now, we can define the cosine of the angle between two vectors x k and x j as follows, let us you remember this expression. So, we can call this as equation number 1 and call this to be equation number 2 that is with reference to today only I am land talking maybe that I have given some equation numbers already, so let us not come from that.

(Refer Slide Time: 07:05)



© CET I.I.T. KGP

$$\cos(\vec{x}_k, \vec{x}_j) = \frac{\vec{x}_k^T \cdot \vec{x}_j}{\|\vec{x}_k\| \, \|\vec{x}_j\|}$$

$$\|\vec{x}_k\| = (\vec{x}_k^T \cdot \vec{x}_k) = E_k^{1/2} = 1.$$

$\therefore$ ~~$\cos(\vec{x}_k,$~~

$$\cos(\vec{x}_k, \vec{x}_j) = \vec{x}_k^T \cdot \vec{x}_j$$

If the key vectors are orthogonal,

$$\cos(\vec{x}_k, \vec{x}_j) = 0 \qquad k \neq j$$

So, if I take this to equations and then we can also define the cosine of the angle between two vectors x k and x j. And how is that defined, that is defined as x k transpose times x j and in the denominator we will be having the norm of x k times the norm of x j alright. Now, what can we say about this, now certainly since we are taking each of the unit vector, each of the applied vectors to be of unit energy.

So, that is why according to that assumption we are having norm of x k vector, which is equal to nothing but, the x k transpose dot x k which is equal to the energy E k in this case in fact, it will be E k to the power half and this is equal to 1 by our definition by what we are assuming of this one having unit energy. So, therefore, we are getting cosine of x k comma x j that we are getting simply by the numerated term, that is x k transpose x j vector.

Now, can you tell me that what is then the condition of the perfect recall according to this.

Student: ((Refer Time: 09:05))

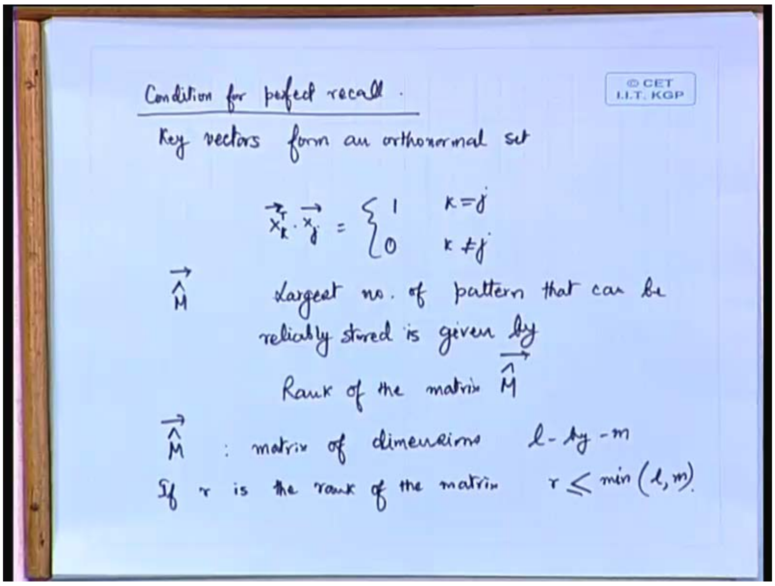
They.

Student: ((Refer Time: 09:07))

Exactly, so thank you very much for the correct answered. In fact, here what happens is that x k and the x j vectors cosine. That is, what we are defining over here and that cosine has to be 0, in other words if the cosine is 0; that means, to say that x k vector and x j vector they are orthogonal to each other. So, if the key vectors are orthogonal, in fact this x vectors, we are calling as the key vectors.

So, if the key vectors are orthogonal in that case we can certainly write cosine of x k comma x j that is equal to 0 for k is not equal to j. And in such a case, we are seen that ((Refer Time: 10:11)) the term v j becomes equal to 0 and y is v j becoming equal to 0, because you simply say that if this term is equal to 0, then x k transpose x j, this term is also equal to 0. And if this term is equal 0, then v j vector is equal to 0 and v j equal to 0 means y is equal to y j that is a perfect recall.

(Refer Slide Time: 10:46)



Condition for perfect recall.

© CET I.I.T. KGP

Key vectors form an orthonormal set

$$\vec{x}_k^T \cdot \vec{x}_j = \begin{cases} 1 & k=j \\ 0 & k \neq j \end{cases}$$

$\vec{\hat{M}}$ Largest no. of pattern that can be reliably stored is given by Rank of the matrix $\vec{\hat{M}}$

$\vec{\hat{M}}$: matrix of dimensions l-by-m

If r is the rank of the matrix $r \leq \min(l, m)$.

So, now that means, to say that the condition that we are getting for the associative memory to work perfectly is that the condition is that the key vectors form and orthonormal set. So, that is what we can say as the condition for perfect recall, so what is the key vector forming and orthonormal set condition, just written in terms of mathematics it is equal to 1 for k is equal to j and this is to equal 0 for k not equal j.

Now, and other aspect that we should have in mind is that, the largest number of pattern that can be reliably stored to the matrix M cap. Now, you remember that in the matrix M cap, we are storing pattern, in what form are we storing in the form of association simply the y k and the x k taking them as the association we were forming this matrix.

Student: ((Refer Time: 12:03))

x k yes absolutely, absolutely correct x k transpose, yes thank very much for the correction. That it should be x k transpose x j, which is equal to unity for k not equal to j and equal to zero for k not equal to for this 1 is equal 1 for k equal j and equal to 0 for k naught equal to j k x. And now for the matrix M cap, the largest number of patterns that can be stored, in fact is given by the largest number of patterns that can be reliably stored.

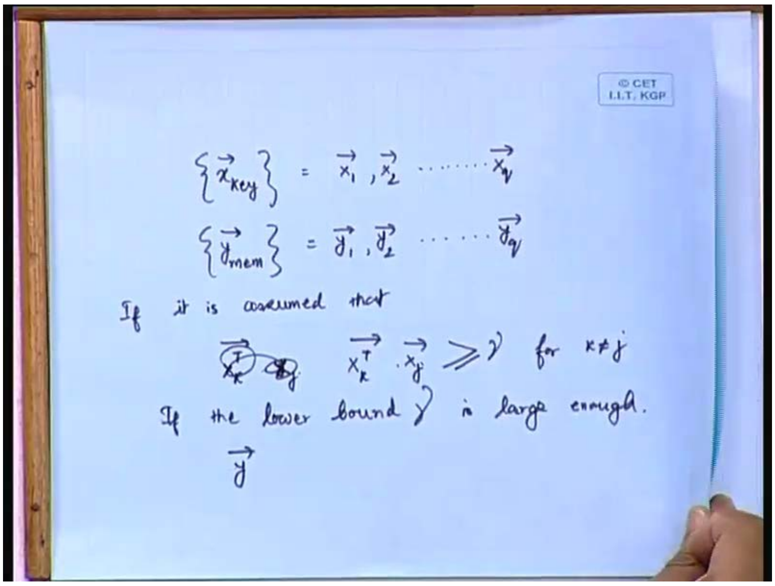
In fact, is given by the rank of the matrix M, so this is the given by the rank of the matrix M. So, let us remember this, it has been shown analytically we have not going in to the derivation of that all right. Now, this matrix M, in fact we can say that this is matrix of the dimension, let us say matrix of dimensions l by m, that is no how assuming this to be a rectangular matrix, if we assume in to be a rectangular matrix.

Then, the rank, rank means what it is defined as the number of independent rows or column of the matrix. So, the number of independent rows and columns must necessarily be less than the minimum of l and m, so that is why if r is the rank of the matrix. Then, we must necessarily have r to be less than equal to minimum of l comma m is that alright. So, that is what we can have as the maximum number of patterns that can be stored.

So, for a reliable recall we must be having the number of patterns to be equal to or less than or equal to this r, which necessarily will be given by the minimum of the l and m should be less than or equal to minimum of this l and m. So, this aspect we must keep in mind and now although we have already established the condition of the perfect recall. The question that should come to our mind is that, use it that to this condition is fulfilled by all the with all the real life cases, in fact they are not.

Many of the real life associations that we will be having, they are the key vectors do not necessarily form and orthogonal set. And I think from the mathematical analyses that we have already carried out, we can emphatically say that if they are not orthogonal, then there is going to be some error in the recall. And let us see that the error in what form that we a getting.

(Refer Slide Time: 16:02)



© CET I.I.T. KGP

$$\{\vec{x}_{key}\} = \vec{x}_1, \vec{x}_2 \ldots\ldots \vec{x}_q$$

$$\{\vec{y}_{mem}\} = \vec{y}_1, \vec{y}_2 \ldots\ldots \vec{y}_q$$

If it is assumed that

$$\vec{x}_k^T \cdot \vec{x}_j \geqslant \gamma \text{ for } k \neq j$$

If the lower bound γ is large enough.

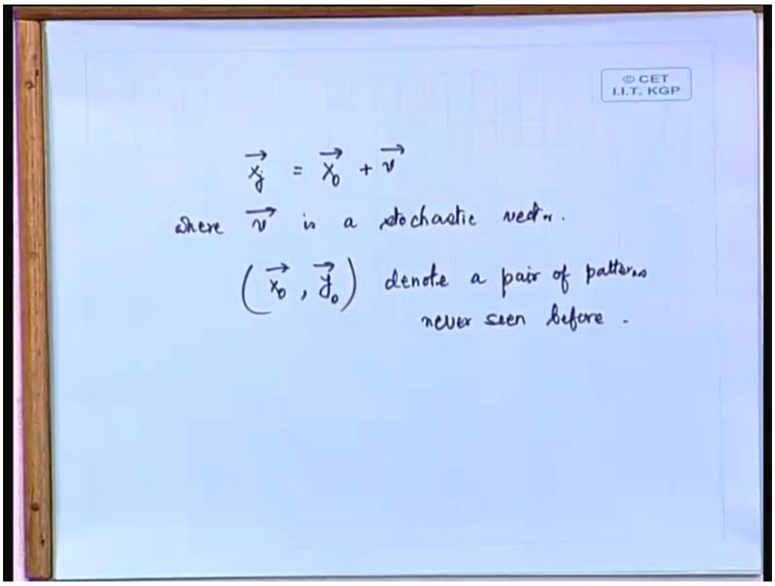
$$\vec{y}$$

Now, let us consider set of key pattern. So, we can define the set of key patterns as a set of x key vectors which is nothing but, consisting of x 1 vector, x 2 vector, etcetera up to x q vector. And we are having the memorized patterns y memorized vector as y 1 vector, y 2 vector up to y q vector the corresponding. Now, if we have this type of a situation that we have x k transpose dot product with x k transpose dot product with x j vector.

So, this one we had assumed to be 0 when there was perfect recall, but if we now say that this is the greater than or equal to some gamma for k not equal j. That means, to say what is gamma, gamma is the some lower bound on this product x k transpose x j. So, if the lower bound gamma is large enough in that case what can be expect out of that, in that case the response that we will get from the response y that we will be getting by feeding any of the key patterns is some response, which will not be there in any of this y 1 to y q's is not it.

If we have the deviation of this to be quite high, if the lower bound of this is quite high already. That means to say, that is in no way it is far deviated from the prefect recall case, then we are not going to find the response in this. So, this leads to the error and another thing that we should keep in mind is that, we are picking of the x vector during the test it to say, we are key picking up the x vector from this key vectors set.

(Refer Slide Time: 19:06)



© CET
I.I.T. KGP

$\vec{x}_j = \vec{x}_0 + \vec{v}$

where $\vec{v}$ is a stochastic vector.

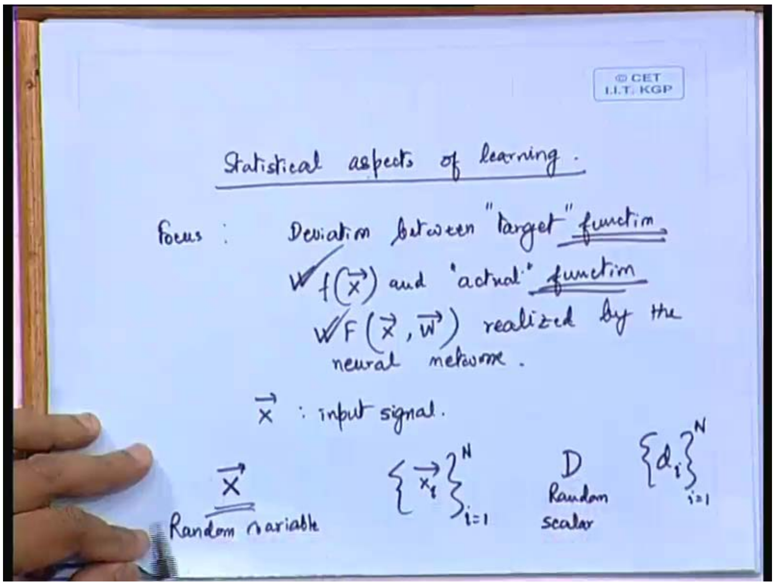
$(\vec{x}_0, \vec{y}_0)$ denote a pair of patterns never seen before.

Let us say that we happened to pick up the x j vector. Now, x j vector may be represented in this form that x j vector could be a stochastic sense to say. That x j vector may be return as x naught vector plus v vector, where v vector is a stochastic vector, so v is a stochastic vector. So, if that is the case that your pattern itself is a stochastic pattern, meaning that it is having one x 0 component and stochastic component which is the v vector.

In that case we will be able to associate the x naught vector with y naught vector. So, x naught, y naught that will form a pair actually what is y naught, y naught is nothing but, the corresponding response to x naught. So, x naught y naught that will denote a pair of patterns never seen denote a pair of patterns. In fact, y where calling as never seen, because it is not there in this set X naught is not there in this set, y naught is not there in this set, but we will be getting an association of x naught with y naught.

So, this is what we have to remember about the pattern association and the recall condition. I think the condition of perfect recall is something that we must understand, because the rest definitely follows logically from here alright. So, before we go to the next aspect that is the statistical aspect of learning is there any doubts that comes to your mind at this stage. Shall we then see the statistical aspects of learning.

(Refer Slide Time: 21:30)



© CET
I.I.T. KGP

Statistical aspects of learning.

Focus : Deviation between "target" function $f(\vec{x})$ and 'actual' function $F(\vec{x}, \vec{w})$ realized by the neural network.

$\vec{x}$: input signal.

$\vec{X}$
Random variable

$\{\vec{x}_i\}_{i=1}^{N}$

D
Random scalar

$\{d_i\}_{i=1}^{N}$

Student: ((Refer Time: 21:37))

Yes please just minute I will get back to your question is what is your question.

Student: ((Refer Time: 21:47))

Correct, x k is the set or no x key is the set of vectors that we are feeding and that is consisting of x 1 vector, x 2 vector up to x q vectors.

Student: ((Refer Time: 22:11))

Right, see this is only one instance ((Refer Time: 22:22)). Now, you see that when I put forward this condition x k transpose x j; that means, to say that take any key and take any j. The condition that you have is with k naught equal to j, so this should be valid for all combinations of k and j. And we only talk to about a lower bound of this one, in the earlier case and in the first case we were talking of this condition.

I mean when we were talking of this condition x k transpose x j; that means, to say that we have to consider that for all the combinations of k and j we have this condition full filled ((Refer Time: 23:12). Because, you see only when all these are full filled, then only you are going to get a perfect recall why, because you just see this expression k equal 1 to m we are already taking. That means, to say that this orthogonality condition must be valid for all is it clear.

Student: ((Refer Time: 23:24))

X j both x j and x k belong to the key set x key both of them belong to we just pickup any two and then show that they are orthogonal for the condition of perfect recall. They are all belonging to this set and we are getting the recall error, that is an association of the x naught and y naught, which are not there in the set. Only when there is an error condition, only when the orthogonal duly conditions and not full filled followed that any other doubts.

Student: ((Refer Time: 24:12))

The question is that why did we take q, why did we take m instead of taking q. Because, we were taking q patterns, but m was number of inputs, simply to see that we do not exceed m. Because, ultimately the maximum storage capacity is restricted to the rank of the matrix m and the rank of the matrix is given by l by m or in our particular keys, where we had assume the matrix to be a square matrix we took l equal to m.

So, m in other words was the dimensionality of the matrix. So, any way you cannot exceed the total number of patterns q to be more than that of m alright, so that leads to a anything that is more than m number of patterns, anything that exceeds the rank of the matrix is something that we cannot recall. I mean that is maximum amount of storage that is possible if you exceed that.

Student: ((Refer Time: 25:33))

If you have q by m matrix, that is not possible. So, let us now goes to the statistical aspect of learning. In fact, in the statistical aspect of learning the focus that we are going to have is on the deviation between a target function f of x, deviation between the target function f of x. In fact, we will be taking f of x vector, again x vector that denotes the input signal.

So, what is f of x vector simply a function, simply the regression function. So, in fact if we have x vector to be the input signal and we have y vector to be the y to be the desire response, in that case we can the map this x to y by a function f of x is not it. So, f of x vector is define to be the target function and how we are achieving that function, we are not exactly realizing that function.

We are only using a neural network and that neural network, in that what are we doing, we are feeding a set of inputs that is again the x vector, we are feeding as the input to that x vector and we are realizing in a actual function. So, we are interested now in finding out the deviation between the target function, which f of x and the actual function which is going to be F of x vector comma w vector, which is realized by the neural network followed.

So, again I repeat what it is x vector the input signal F of x vector, that is the target function what you can describe mathematically. But, you are realizing the target function or a very close approximation to the target function by actually working through a neural network. Where, the neural network does a mapping form this x vector to the output in fact, output could be a vector or scalar anything.

Then, that mapping function the neural network is realizing, that will be a function of x vector as well as the w vector the weight matrix, weight vector or weight matrix whatever weight is. So, this is what we are focusing upon in the statistical aspect of learning, mind you here we are not exactly considering the evaluation of the weights w, which we were considering earlier.

But, here we are only interested about finding the deviation between the target function and the actual function. So, our focus is on the function level mind you, now what is after all a neural network, now we are having our set of observations, we are doing an experiment. And we are not knowing, we are not exactly knowledgeable about the mathematical form of the function, but we have set of some experiment and we are having a set of observations out of that experiment.

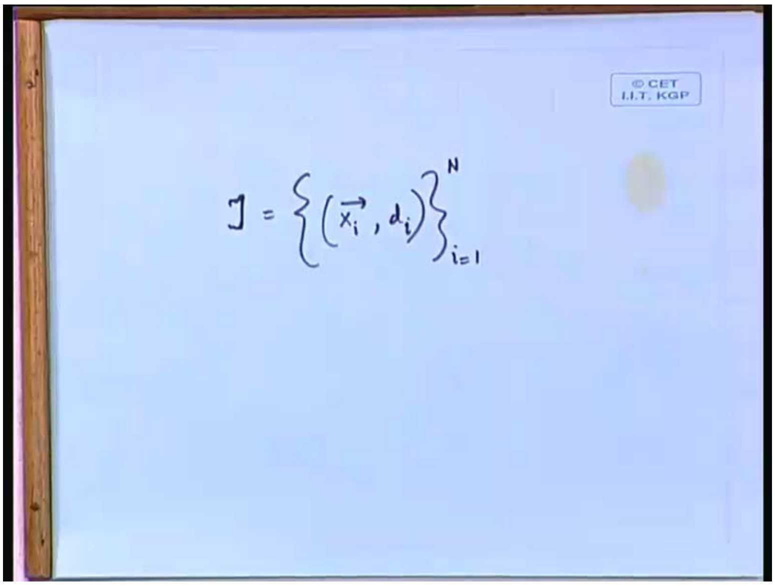
And we are only gathering, so; that means, to say that the knowledge that we are acquiring out of this is only an empirical knowledge. And how are we representing that empirical knowledge, that empirical knowledge we are representing by a set of training samples. Now, in fact to say that we are defining x vector to be a input signal, now we can say that we can denoted by a capital X vector which is nothing but, as a random variable vector where, each of these where the instances of that vector is this x vectors or rather to say x i vectors.

So, I can define a random vector x vector, whose instances are all the x i vectors and I can just happen to pickup in such instances of the random vector. So, I get a set of x i vectors for i equal 1 2 N meaning what, meaning again x vector is a random is a vector and I am taking it is particular instance x i, x i is the ith instance of the vector and how many such instances have gathered i equal 1 to N.

So, the entire input the input is now being modeled as variable. In fact, to say that if x vector is defined as a random variable, this x vector is defined as a random variable, in that case what we are taking is just nothing but, a subset of that random variable. So, we have only fed a limited number of instances i equal to 1 to N and also we are defining a random scalar output, if we assume to be a scalar.

Let us a that the output or rather to say we define a random scalar. So, D is nothing but, a random scalar quantity and this random scalar quantity we are defining as the set of d i's, where d i is the ith instance of that scalar i equal 1 to N. So; that means, to say what that we have got of set of training samples given by x i vector comma d i for i equal 1 to N. And this realization that constitutes a set of training samples.

(Refer Slide Time: 33:13)

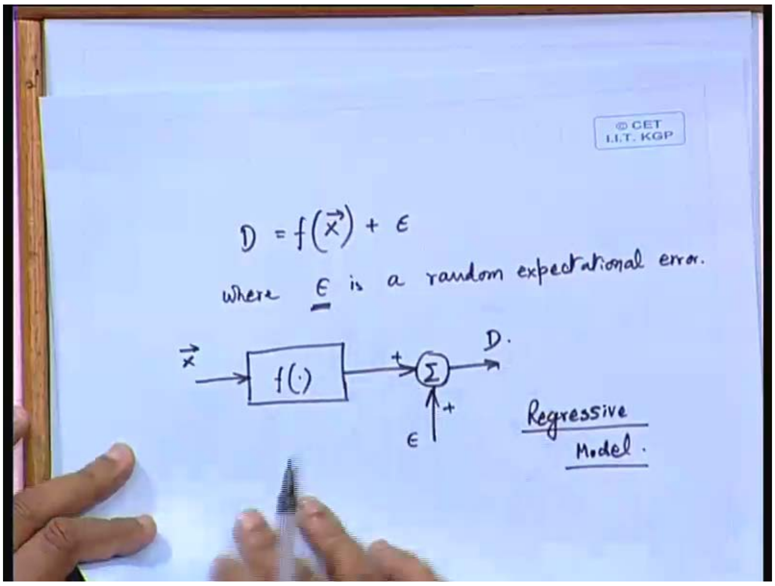


$$\mathcal{T} = \left\{\left(\vec{x}_i, d_i\right)\right\}_{i=1}^{N}$$

So, the set of training samples we denote by T equal to a set of x si vector comma d i and we can have it as i equal to 1 to N, this set is the set T that we can consider. Now, again what we are doing, let me come back to the discussion that is doing a little while back, that after all in a neural network what are we doing ((Refer Time: 33:54)). We are actually gathering the empirical knowledge and that empirical knowledge we are storing in the form of the training samples.

So, now this training samples after all finally, what this training samples how do we represent the training samples or rather how do be encode those training samples. Those studies of this will be ultimately encoded by the set of the weighed vectors, but we will come to that later on. Now, our idea is to find out the deviation between this to. So, we must have a good knowledge about f x and this f of x comma w.

(Refer Slide Time: 34:55)



$$D = f(\vec{x}) + \epsilon$$

where $\underline{\epsilon}$ is a random expectational error.

So, let us first take this f x thing, that what do we see out of this, so for this we can a model like this, where the dependent what did we take again ((Refer Time: 35:02)). We had taken x vector to be a random variable, so this a random variable vector and this is a random variable scalar. And what more we have to take is that, this x vector is an independent random variable.

Whereas, this D what we are getting is a dependent random scalar quantity, D is dependent and x is independent, we must have that in mind, so.

Student: ((Refer Time: 35:36))

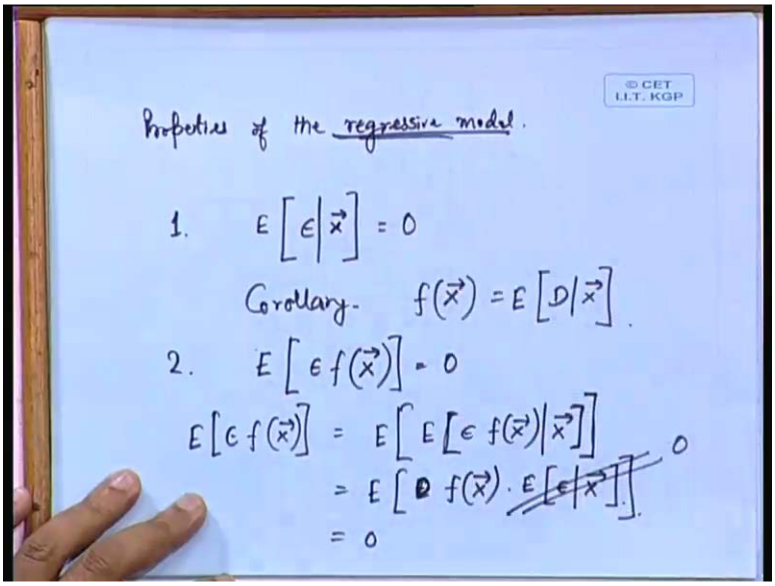
D is dependent on x actually, because in our case what happen says that D is very much dependent on x. So, x is the independent and D is the dependent, so the dependents between D and x that can be model us F of x vector plus epsilon, where epsilon is the random expectational error. And this we can simply depict as follows, that here x vector is the input and then, we are having a function this f x vector function.

And what we are getting this f of x vector, which we are getting as a output of this block that is getting add a two. So, here we will be having a summer and we will be adding the random expectational error epsilon to it and then, we will be getting the D as the output. So, this in fact, is called as the regressive model, it is a mathematical regressive model, that if we have I mean what we have as the input is x and what we have as output is D and in between it is this function f x which models this.

Student: ((Refer Time: 37:27))

Yes, in this case this f is nothing but, the target function, so this is the mathematical model in the form of a regression. So, this is definitely the target and we are having just a random expectational error added to it. Now, this regressive model that we have presented, that has got two useful properties.

(Refer Slide Time: 38:03)



Properties of the regressive model.

1. $E\left[\epsilon \middle| \vec{x}\right] = 0$

Corollary. $f(\vec{x}) = E\left[D/\vec{x}\right]$.

2. $E\left[\epsilon f(\vec{x})\right] = 0$

$$E\left[\epsilon f(\vec{x})\right] = E\left[E\left[\epsilon f(\vec{x})\middle|\vec{x}\right]\right]$$
$$= E\left[f(\vec{x}) \cdot \overset{0}{\cancel{E\left[\epsilon|\vec{x}\right]}}\right].$$
$$= 0$$

So, properties of the regressive model, the first property is that the mean expectational error given any realization of x vector is 0. So, mean of expectational error; that means, to say what, expectation of epsilon, epsilon is the expectational error, so epsilon is the expectational error. So, mean of this expectational error given what any realization of the x vector. So, given any realization of the x vector and what is it going to be this will be equal to 0.

That is the first property of the regressive model, that is mean of the expectational error epsilon given any realization x vector is equal to 0. And as a corollary to that ((Refer Time: 39:20)), we can have by simply applying this definition that D equal to f of x vector plus epsilon. So, if we take T expectation of this on the left hand side and right hand side, then what it leads to is that f x vector that is equal expectation of D given x vector, quite understandable is not it.

Because, what happens is that we are take a expectation of D given x vector. So; that means, to say what expectation of D given x vector will be equal to f x plus expectation of epsilon given x vector. And expectation of the epsilon given x vector is already equal to 0, so that is why this will be equal to expectation of D given x vector. And the second interesting properties that, the expectational error is un correlated with function f x.

So; that means, to say what, that expectation error is the expectational error epsilon, that is un correlated with the function f x. So; that means, to say that expectation of epsilon f x vector is equal to 0, in fact it is quite easy to see that, this you can simply see like this, this expectation of epsilon f x vector. That will be actually we can write it as expectation of epsilon f x vector given a realization x vector, just this epsilon f x vector definition itself will say, that this can be realized as expectation of this.

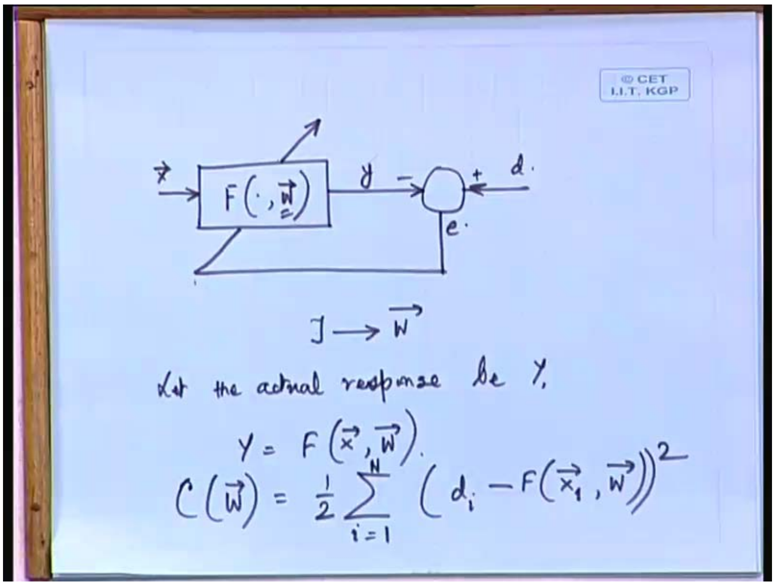
So, this square bracket is for this expectation and outside we have this one. So, this means to say what that this means say, that this is becoming f x vector goes out of the expectation operator. And on the inside we will be having expectation of epsilon given x vector and this quantity we know is already is equal to 0. So; that means, to say that this is equal to 0, so the expectational error is un correlated with respect to the regression function f x.

So, this is the property of the regressive model ((Refer Time: 42:26)), now this is the regressive model. Now, we are not getting this function f x we should have got that or rather the exact mathematical model of this is f x. But, instead of getting that f x, we have got a neural network to do our job. And a neural network in fact, is realizing a function f which is function of x vector, as well as the weighed functions w.

And what we are doing, we are adjusting the w in accordance with what, in accordance with the deviation that we are getting between the target output or the desired output and the actual output. Now, we are already fed with the training pattern x i comma D, x i being the instances of the random variable x vector and d i being the corresponding the desired response that, so we already know that about the d i are.

So, what you simply have to do is just to take the difference between the actual output from the neural network. And the desire response that is d i take the error of that and in turn that error is going to update the weights that is w. So, just giving pictorially the neural network model that we are having as an approximation to this regressive models. So, this being the regressive model, the neural network that approximates that is the regressive model can be shown as follows.

(Refer Slide Time: 44:14)



$\mathbb{J} \longrightarrow \vec{W}$

Let the actual response be Y,

$$Y = F(\vec{x}, \vec{W}).$$

$$C(\vec{W}) = \frac{1}{2}\sum_{i=1}^{N}\left(d_i - F(\vec{x}_1, \vec{W})\right)^2$$

Here too the input will be the x vector, then we are going to have the function, the neural network function represented as f of x as well as w. And here y is going to be the actual output and d is going to be the desired output, so plus this as minus and this is the error and we are going to pictorially show it as follows. The way of showing this invariably indicates, that we are going to adjust this weights w vector in accordance with this error e.

Now, what it means that here empirical knowledge is represented ultimately is encoded by this weight vector w. So, what we it can say is that as if to say that the training samples or the training set T that your feeding I define it earlier you remember ((Refer Time: 45:38)). So, this T is nothing but, the training set for i equal 1 to N x i comma d i.

So, this training set T in fact, in this case mapped into the weight vector. That is, what to we are doing by the process of learning in the neural network and let the actual response be y. So, that we have y equal to f of x vector comma w vector, now we are having already d to be the desired response. So, we can calculate the deviation between the desired and the actual and in fact, I think we did in our first 1 or 2 lectures only, that we had in fact, found out the some of the squared errors.

So, what we have do is that we are defining a cost function see a cost function C which is to be express as a function of the weight matrix only. So, C as a function of w will be return has half, half also you remember that it is only for from mathematical convenience that we have been using this half. And summation of what d i minus f of x i vector comma w and square of this; that means, to say what, that the sum of all the squared errors. And some computed about what, what is the limit of the summation i equal to now not m.

Student: ((Refer Time: 47:45))

Not q i is equal to 1 to N, where N is the training set I now understand that y you set q, because earlier I was taking the training set to be i equal to 1 to q. So. In fact, in that since you are right, but now you should have been attentive in finding out that in this case we have define that training set as i equal to 1 to N. So, it is this capital N for which we have to do this summation.

Now, realizing the cost function like this can be interpreted as follows. That means, to say what that you have got totally capital N number of trainings samples and you are adding of the squared error of all this training samples. And then combinedly you are getting a single index, single cost function C which is being represented like this. And what you are going to do now, in order to reduce the error, you must now take a derivative of this C w with respect to w and then you will adjust the weights.

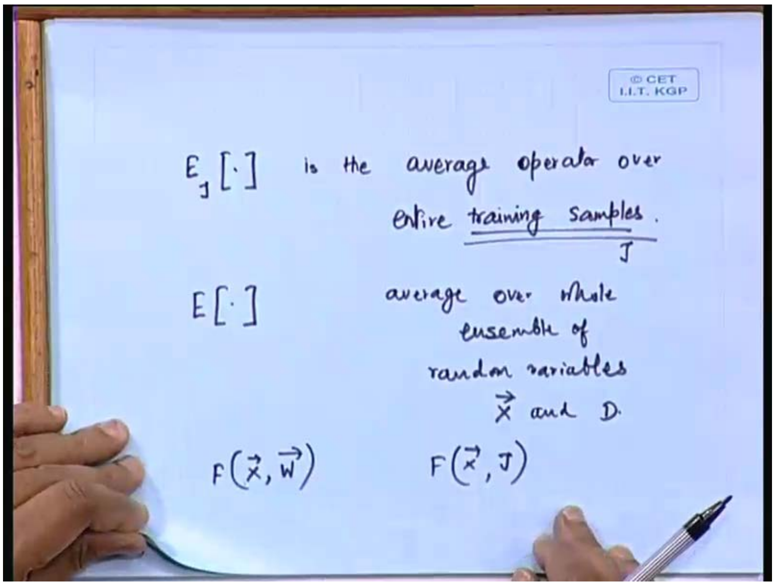
You did it earlier also, but in earlier case viewer feeding each and every example we were giving I mean here we have training set. And a training set means, there are N number of examples already, in earlier case what we were doing feed one example. And then, adjust the weight, feed the next example adjust the weight and so on. That means, to say that by feeding in different examples, we would have adjusted the weight N number of times.

Whereas in this case, in this sort of a representation what we are doing is that, we are accumulating the sum of squared errors. For all the N examples which are they are in the set and once this N examples are all presented and we have accumulated the some of the square error, then we are going to adjust the weights. So, this C w is a combine cost function for all these patterns put together.

So, this means to say that this one what sort of learning can be tell this as a batch learning, this is a learning by the batch. You feed all the N examples and then you adjust at the end. Whereas, the earlier learning what we discussed presenting one sample and then, adjusting the weight, again next sample again adjusting weight, that could be called as an online learning, whereas this one can be called as batch learning.

So, let this be the batch learning expression, now what we are going to do here we are getting the only the error between d i and f. But, our objective is to get an expression, where we have a relation between f x or error between f x and this f. So, we have to reach there, so it is not very difficult to obtain that.

(Refer Slide Time: 51:40)



© CET
I.I.T. KGP

$E_J[\cdot]$ is the average operator over entire training samples. J

$E[\cdot]$ average over whole ensemble of random variables $\vec{X}$ and D.

$F(\vec{x}, \vec{w})$ $F(\vec{x}, J)$

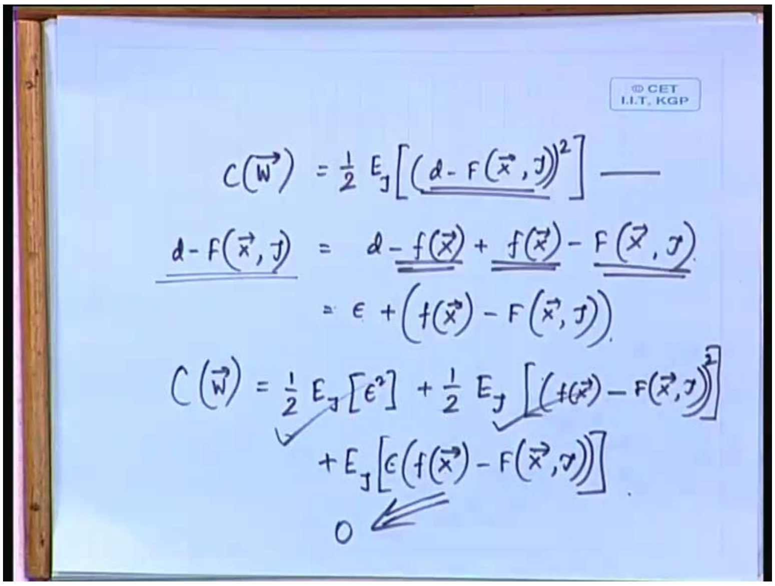
So, before we do that again some definitions that we have to some things which we have to define. Let us say that E T is the when be right it is like this, this we are defining as the average operator over the entire training sample. So, this is, so when I write a E T within third bracket something, this is nothing but, the average expectation or this is nothing but, the average operator over entire samples.

The training samples is T and that is why we have given E suffix T and writing this as E suffix T is different from that of writing as the E operator directly. What is E operator, E operator is nothing but, it is the average over the whole a ensemble of random variables, what are the random variables, which a random consideration X and D. So, there is a difference between calling it has E and calling as E T.

Because, E T is define over the training samples and in fact, training sample is nothing but, a subset of the ensemble. And another thing ((Refer Time: 53:32)) that here we have been calling it as f x comma w and after all what is w indicating, in this kind of an empirical knowledge representation, w is actually an encoded form of the training that you are giving.

So, we can instead of writing it as x i comma w, we could also write or rather F x comma w could be interchangeably return as F x comma T, where T is nothing but, again the set of training samples. So, we that we can write down the cost function as follows.

(Refer Slide Time: 54:26)



© CET I.I.T. KGP

$$C(\vec{w}) = \frac{1}{2} E_{\mathcal{T}}\left[\left(\underline{d - F(\vec{x}, \mathcal{T})}\right)^2\right]$$

$$\underline{d - F(\vec{x}, \mathcal{T})} = d - \underline{f(\vec{x})} + \underline{f(\vec{x})} - \underline{F(\vec{x}, \mathcal{T})}$$

$$= \epsilon + \left(f(\vec{x}) - F(\vec{x}, \mathcal{T})\right)$$

$$C(\vec{w}) = \frac{1}{2} E_{\mathcal{T}}[\epsilon^2] + \frac{1}{2} E_{\mathcal{T}}\left[\left(f(\vec{x}) - F(\vec{x}, \mathcal{T})\right)^2\right]$$

$$+ E_{\mathcal{T}}\left[\epsilon\left(f(\vec{x}) - F(\vec{x}, \mathcal{T})\right)\right]$$

$$0$$

That we can say, that C w is equal to half of again the half is coming, because of this ((Refer Time: 54:36)). Now, mind you what we are doing that we are adding of all these things, but leaving aside we can add out and then we can divided by N. So, some normalizing vector N if we consider over here, in that case this will represent nothing but, the average square error over the training samples is not it.

If we simply say, that if we simply define a quantity as 1 by N normalize by N 1 by N C w. Then it becomes nothing but, half into 1 by N into this and 1 by N into this is nothing but, the average cost function. So, that an associated a that a average of this square error and this average of square errors. So, we can represent C w in terms of the average of the square error as follows.

So, here this average is over the training samples that is the set of training samples. So, it is average E over T. So, here this thing we can represent as E T d minus F x vector comma T, because I have already set that instead of x vector comma w vector I could interchangeably write it as x vector comma T. So, this thing square I can write that only thing is that I did not considered that 1 by N multiplication that is implied.

So, this is the way where by the cost function could be represented. Now, we have to calculate this expecting average of this quantity, average of this quantity can be calculated in a easier way as follows. That, let us see that what is argument of this, it is d minus f of this function, so d minus f of x vector T, this could be return as d minus f x vector plus f x vector minus f x vector T, just I added one f x and I subtracted one f x, nothing more than that.

Now, let us interpret this what is d minus f x, d minus f x is epsilon. So, it is epsilon plus what is this f x minus f x comma tau. In fact, this is what we looking for, the error between the actual function or rather the regression function and the approximation that we are doing by applying neural network. So, this is epsilon plus f of x vector minus f capital F of x vector T alright.

So, now if I substitute this, so instead of d minus F x comma T if I substitute this expression into this alright into the expression for C w. In that case, what do we get first C w, we are get a C w is equal to you can just see instead of d minus f, we are going to write as epsilon plus f x minus f x tau. And that leads to many quantities I mean that leads to in fact, it is this plus this whole square for.

So, they are will be a term which will be return as E T half of E T.

Student: epsilon square.

Epsilon square, yes very much epsilon square term will be there plus there will be another term, which will be half of E T and square of this quantity, that is to say f x vector minus f x vector T this whole square, this quantity. And then, there will be another term which will be epsilon times this and I can easily here, that some of the students have already said that this is going to be equal to 0.

So, why this is equal to 0 I thing for those who have not understood, we can have brief discussion in the next class, because time is running out for today. So, let us just complete this expression and in the next class we will continue from this point, so this is x vector comma T. So, this is the expansion of this one that we are getting, so this one is a term to be considered, this one is a term to be considered and this one some students have felt that this should be equal to 0 and why that one we will see in the next class. So, till then good bye.

Thank you.