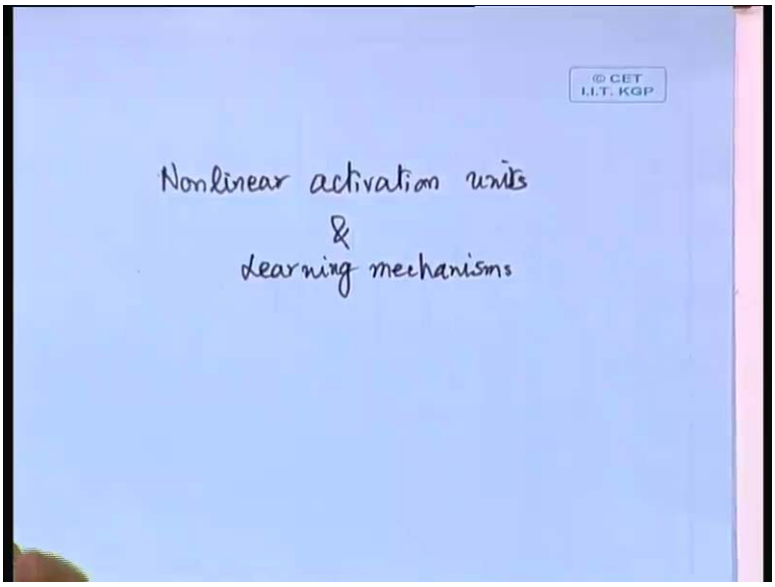
Neural Network and Applications
Prof. S.Sengupta
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 04
Nonlinear Activation Units and Learning Mechanisms

Discussions on the Non-linear Activation Units.

(Refer Slide Time: 01:00)



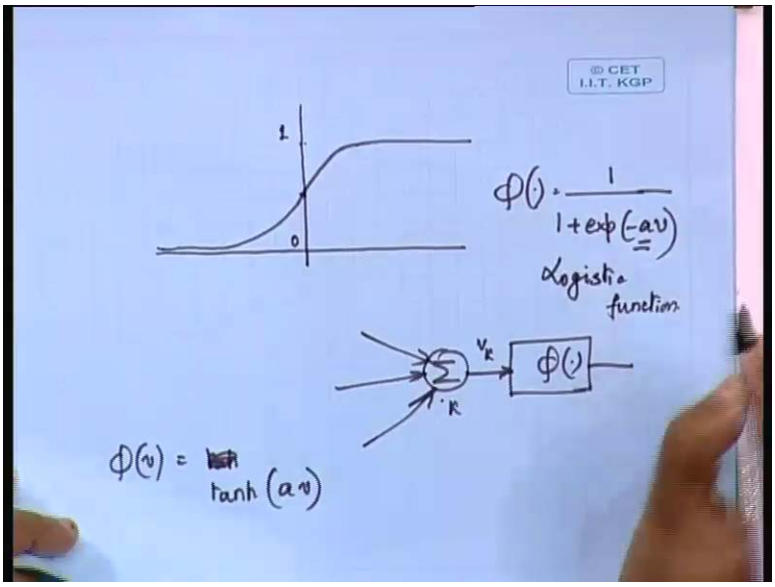
The topic which we had said, in fact introduced in the last class and also in this lecture, I am going to introduce the learning mechanisms and I think this aspect that is learning mechanism cannot be completed in this lecture. Alone, I think we are going to spill over again to the next lecture, in order to complete the topic of learning mechanism. So, we begin with the non-linear activation units.

And in fact, what we are going to discuss in particular about the non-linear activation units is also the aspect of stochastic neural networks, because whatever we have considered, so far are all deterministic in nature. Now, as in the last class, we were discussing about the non-linear activation units in fact, that is what we are needing very much for the realization or approximation of non-linear functions and typically, that is what is going to be.

So, typically we are going to have for an output, we are going to have a non-linear approximation in the N dimensional space, given a set of data points, given a set for

observations, that is what we are going to do. And for that, we have to take some non-linear activation unit and we decided to take the sigmoidal function, which is having the s kind of a shape as the activation function. So, the thing which we had considered is a function.

(Refer Slide Time: 02:43)



Somewhat, like this and this function will be given by 1 by 1 plus exponential to the power minus a v, where what is the v, v is the input, the input means that it is after the linear sum. That means, to say that v is what we are obtaining here, we call it as v k, if this is the kth neuron in the system. And then, what we are doing is, to pass it through an activation function phi and in fact, this is one of the realizations of this phi, that we can have it as 1 plus exponential to the power minus a v.

This means to say, that when v tends to minus infinity as you can very clearly see, that this becomes this phi function becomes equal to 0. And when, v tends to plus infinity, in that case we are going to have this terms that is exponential to the power minus a v to be exponential to the power minus infinity. And that means, to say that it is the phi function value is going to 1, so it is indeed bounded between 0 and 1.

And what is this a; a in this case, we considered to be a positive quantity and a basically determines the slope of this curve in fact, if we are making a higher and higher, what is the limit, that we are going to reach. That is right, if we are making a very high, then in a

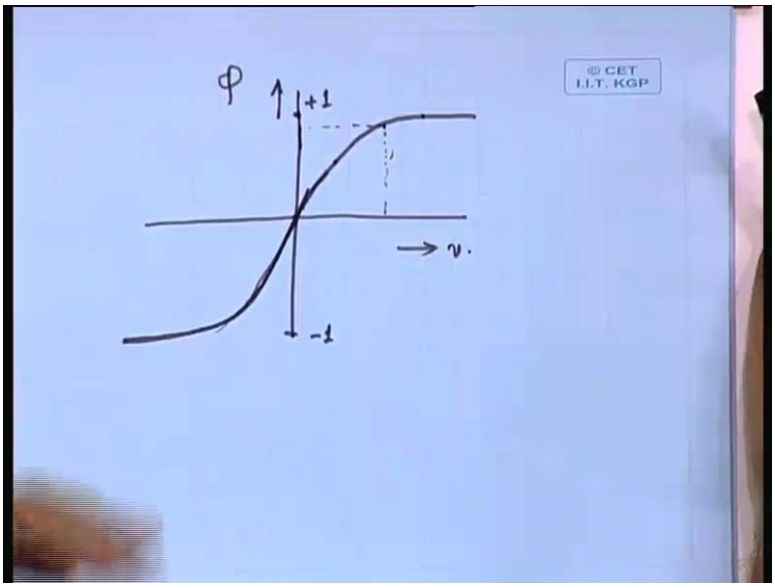
limit, it reaches the threshold function. So, it is the McCulloch-Pitts model that we are reaching, if we are making a tending to infinity.

And if we are making a tending to 0, then that would mean that, it is becoming a linear activation unit. So, this is in between the two and this is, as I also discussed that is it is going to be differentiable, it is going to be continuous, monotonically increasing. These are some of the characteristics that we can point out about this sigmoidal function in fact, this is not that, it is the only realization of the sigmoidal function. The sigmoidal function, in fact can be realized in using many functions.

Something, like this s shape characteristics, it has to give not necessary that the equation for that has to be 1 by 1 plus exponential e to the power minus a v, it could be any realization. But, another advantage is the tuning of the slope of the curve using this a and now we are going to see that, just like the way. Sometimes, we required for the binary activation units we had seen that, we not only defined the threshold function or McCulloch-Pitts model.

But, we also considered the signum function, where the activation values are finally restricted between plus 1 and minus 1. So, likewise, we can think of a non-linear function, that ranges between minus 1 to plus 1in fact, the best function for that is going to be as a phi function, if we take the tan hyperbolic. If we consider the tan hyperbolic of a v, then that is going to have sigmoidal shape only, but it will range between minus 1 to plus 1.

(Refer Slide Time: 06:31)



So, it is characteristic, the tan hyperbolic a v, again in that case to a is going to decide the slope of the curve. So, it is going to be like if this is minus 1, so on this axis, we are plotting the phi function and on this, we are plotting the v function, v could be positive, v could be negative. And in fact, when v is equal to 0, it is going to be 0 and in all other places, it is going to asymptotically reach minus 1 in this end and if this is plus 1, then it is going to have a shape like this. So, this is the realization of the tan hyperbolic activation function.

So, here the range is plus 1 to minus 1, otherwise shape wise it is very similar to what we have considered as a sigmoidal function realized using the exponential function like this 1 by 1, thus exponential to the power minus a v. So in fact, this function has got a name, this is often known as the logistic function, this realization 1 by 1 plus exponential to the power minus a v. This is also known as the logistic function, whereas this is the typical tan hyperbolic function.

Now, the thing is that we now have a choice, that we can use the activation units to be either McCulloch-Pitts model, that is to say a threshold function, we can use. We can use linear activation units; we can use non-linear activation units. The thing is that, which of it we are going to use, that is again dependent upon several factors in fact, what we sometimes do is that, in a typical neural network, which is realizable out of multiple layers.

In fact, we are going to talk more about that, later on in this course. That where you can have one input layer from which the neurons are feeding their inputs and then we can have some output layer at the end from which we are going to realize the final output functions. And in between the input layer and the output layer, we are going to consider several neurons, which are only taking part in some intermediate computation which are not exactly interface to either the input or the output.

In fact, those are the neurons or those layers, we are going to define as the hidden layer and typically in a neural network, there could be one or more number of hidden layers. More about it, we are going to see later on, when we discuss the typical neural network structures, especially the multi layer feed forward networks. When, we see that time, we will be discussing about all this things.

But, what I try to say is that, one could have a kind of system, where one can have the intermediate layer neurons, which operate on the non-linear activation units. Because, it is the intermediate layer neurons, which ultimately realize the function that you are going to approximate. And then finally at the end, you can combine, you can linearly combine the outputs, which the intermediate neurons or the hidden layer neurons are generating and then, we can realize the end function.

So, it could be a combination of the linear units, it could be a combination of linear threshold and the non-linear activation unit realizable in the sigmoidal function, logistic functions forms. So, all these things are possible and we are going to see that later on in our implementation. Now, another aspect, since we are discussing about the neuron model so far.

Another aspect that we should discuss is that, so far the neural network models that we have considered, they are all going to be deterministic in nature. Deterministic in the sense that, if you are feeding a fixed input, like say for example, if this is the net activation v, that we are going to feed, from the linear in input, output from the linearizer output over here. If this is the v, that we are going to feed, we know that as the output is definitely going to be this.

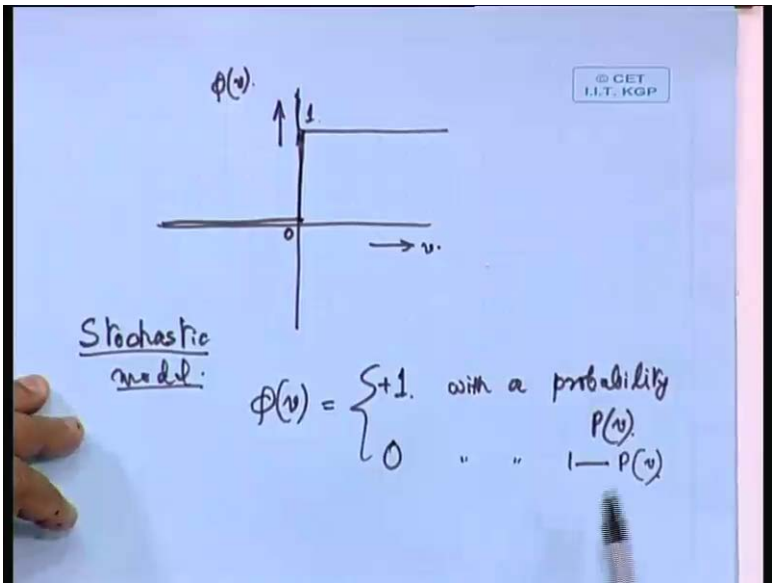
In fact, if we are going to operate the neural network under different conditions. However, whatever the conditions may be operating it under different condition, still whatever v, it will give the output as far the function that you have realized out of it. It is

never going to act unpredictably or it is never going to act that. Sometimes, it is going to give you an output equal to this or sometimes it is going to give an output equal to this, for the same input.

But, the neural network model could also be thought of as a stochastic model in fact, that is how our biological neurons do behave, there is a psycho physical consideration people have conducted experiments. And they have concluded, that yes our biological neurons, very often act as a stochastic model. I think in a very simplified form, we can understand it like this, that given a set of surroundings or if we are placed in some environment is it that, all the time we are going to act in the same manner, we are not.

It could be that given the similar situations, we sometimes act on in one particular way and given the very similar situations or may be exactly identical situation within to behave in a little bit of different manner. May not be totally deviated from the manner in which we had acted earlier, but may be similar to it, but quite different from, what we did last time, so it is a stochastic neural network model.

(Refer Slide Time: 13:35)



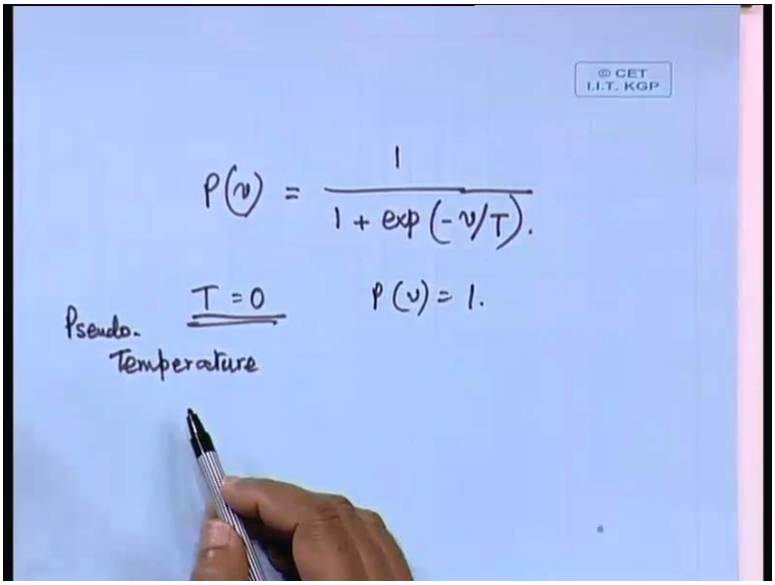
And in fact, the way that you can think of as a stochastic neural model is like say for example, you take the McCulloch-Pitts model only where, we are going to have the function. As the you can have, this is the v and this going to be the phi of v and in McCulloch-Pitts model, we are going to have this to be 0 and when v is greater than 0, we are going to have the value of phi v to be equal to 1. Now, it is so far what we have

learnt is that as long as v is greater than or equal to 0. We are going to have phi v equal to unity and when v is less than 0, we are going to have phi v equal to 0, that is what we have learnt and it is always going to be true.

Now, with the stochastic model, we are going to define in this way, that phi v is going to be 1. We can say that, it is plus 1 with a probability p v and it is going to be 0 with a probability of 1 minus p v. What that means, that we are going to define a probability, given that v is greater than equal to 0, we are going to consider that phi v will be equal to plus 1 with a probability p v.

So, it is not always true that given that v is greater than equal to 0, it is going to have a value equal to unity. It is probabilistic in nature, so that is what the stochastic model tells us.

(Refer Slide Time: 15:27)



And this probability is often defined this way, that we can consider the p v, the probability of activation to be 1 by 1 plus exponential to the power minus v by T, v. We all know, it is the same v, that is the activation value and it is 1 by 1 plus exponential to the power minus v by T. The term T in this case is a parameter that very much controls the probability.

You can just very clearly see it, that if we are having, let us say if we take T is equal to 0, what is going to be the probability, in that case the probability is going to be, this

expression is going to be exponential to the power minus infinity. So, it is 0, so in that case p v is going to be equal to 1. That means to say, that it was down to the deterministic McCulloch-Pitts model.

Because, if we have that phi v is equal to plus 1 with a probability 1; that means to say that it is same as that of the McCulloch-Pitts model. And so, this way we are going to have a more general form that McCulloch-Pitts model is going to be one special case, where we put T is equal to 0 and if we keep on increasing the T. Let us say that, we start increasing T from 0 onwards and we are going to reach higher and higher values of p.

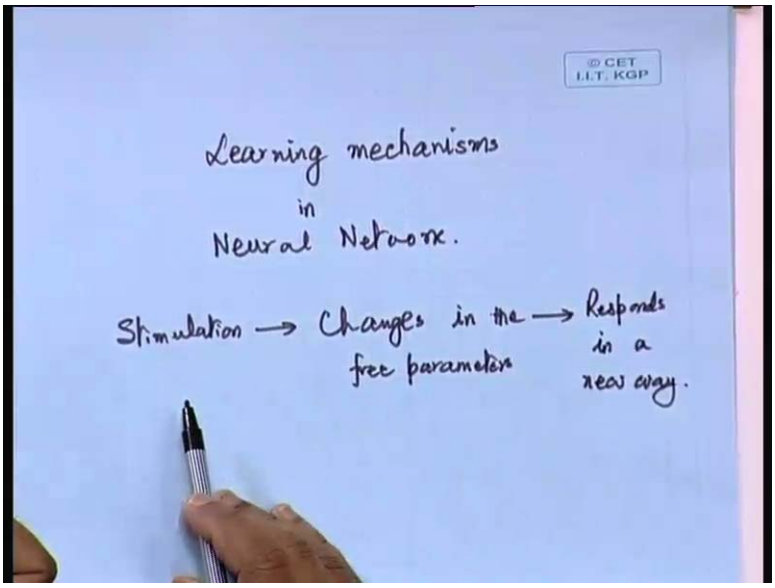
And what does it mean; that means to say that the network now deviates from it is deterministic behavior and goes most more towards the stochastic behavior. So, as we keep on increasing T, the probability in fact decreases, with the increase of T. Now, what this T is, so far I have deliberately told this as a parameter, but people defined this T to be a temperature, do not feel surprised that we are relating the temperature to it.

In fact, it is not the temperature, the actual temperature of neurons or the temperature of our body, anything like that. This temperature is actually a pseudo temperature; in fact, it is very similar to what we can define as a thermal noise model. In the sense, that if we increase the pseudo temperature, then the system behavior becomes noisy and that is why it is response is going to be probabilistic in nature.

So, this T is defined as a pseudo temperature and at T is equal to 0, it is noise free. So, it is absolutely deterministic in nature and with increase of T, it is going to be more noisy in it is behavior and there the stochastic model is coming in. So, these are the things that I wanted to tell you in connection with the neuron activation unit. So, we learnt about the threshold units, we learnt about the linear units, we learnt about the non-linear activation units and now the stochastic neural network models.

In fact, all these things will be needed for us, as we progress more through the lectures. And is there any question, regarding this topic of the neuron models in general, it linear, non-linear, whatever we have discussed, is there any question. So, in that case we can proceed to the next topic for us, which is the learning model. So, we are going to consider the learning mechanisms in neural network.

(Refer Slide Time: 19:29)



Now, something about the learning, we can intuitively know. Because, we all go through the process of learning, when we are transformed from a child to an adult, we go through the process of learning every day, every moment in fact and we know what it is. But, it is very similar kind of learning mechanism that we have to think of for the artificial neural network models. Also, because ultimate objective is that the reason, why we picked up artificial neural networks topic is that, we can mimic, what goes on.

Mimic in a very random version, certainly not the exact realization of the human brain, but we can just mimic, it in a smaller scale in order to incorporate the learning mechanism in to it. So, what are the essential components of learning, these things we can list out. First is that, for any learning to take place, we give a stimulation, there is a set of stimulations that we are receiving from the input from the environment.

And what we are doing, after obtaining the stimulations is that, we are making changes in the free parameter, so changes in the free parameters and free parameters of the neural network. In a case of biological neuron, it is the synopsis that is going to change the synaptic behavior or the synaptic connection strengths, which adapts according to the stimulation that we receive.

And in the case of the artificial neurons, which we had already seen, when we were discussing about the linear neurons, we had seen that dependent upon the error that we are going to make, between the actual and the desired response that combined error

function. And then, we are going to adjust the W i j's, that is to say the synaptic weight connections. In accordance with the error, that we are making or we are proceeding in the direction of gradient descent and then, we are adjusting it is parameter.

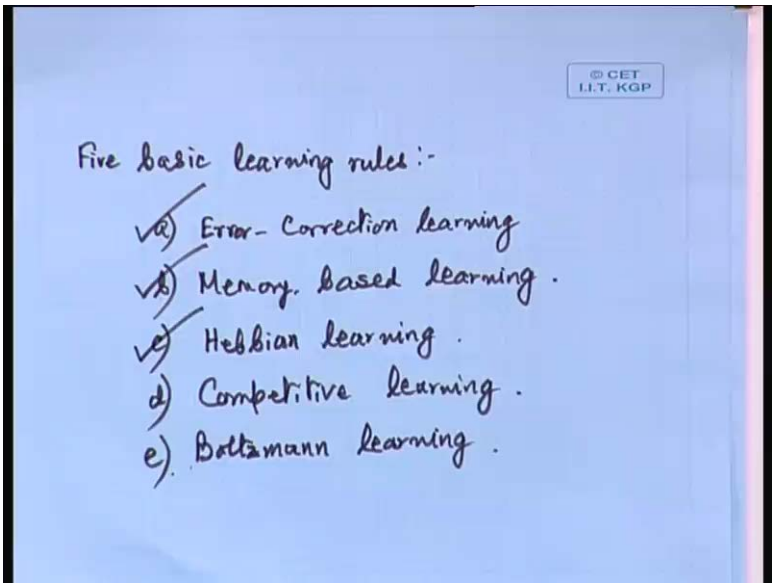
In fact, the parameters that we adjust by free parameters, we define the synaptic weights and also the bias or the weight that is associated with the bias and the model that we choose. There all the free parameters are this synaptic weights and bias that we are going to adjust. And then, when this change has been done in fact, we do it in an interactive way, meaning that, we make changes in the free parameter.

Then, again we see that, how much we have deviated from the actual, then again we change, it is an iterative process, that goes on and it takes in fact, several iterations. But, we go through the process of learning and once it is learned, then what it does is that, it responds to the environment in a changed way. So, it responds in a new way, that is what we can tell about it.

So, these three are the essential components about the learning, that is to say, you have to get the stimulation and what is most important is that you have to make changes in the free parameters of the system. And then, once those are changed and then it responds in a new way, meaning that, now it is in a position to accept the kind of inputs or patterns, which it has not encountered before. That means to say that, it is in a position to accept the test patterns.

And then, dependent upon the test patterns, it is going to give you the outputs a correct classification or a correct response, it is going to give. Because, already it has adjusted, it is free parameter in accordance with the stimulation that it has received from the environment. Now, there are some basic learning rules, which we are going to consider.

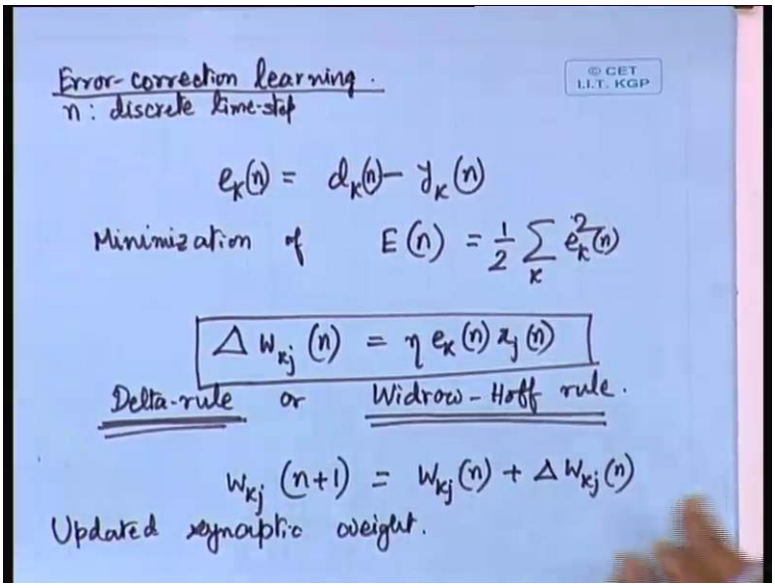
(Refer Slide Time: 24:35)



In fact, we can list out, five basic learning rules and we can list them out as follows that at first we can list out, which is very important that is the Error Correction Learning. We will come back to each of these in a more detailed way, but first let me list out the five basic learning rules. The second that we are going to consider is the Memory based learning, the third category is the Hebbian learning, the fourth category is Competitive learning and the fifth and the final category is the Boltzmann learning.

And we are now going to take each of these categories, each of these basic learning rules in a 1 by 1, we are going to consider that. So, what is after all the error correcting or this first one that is error correction learning? I think, we have all ready seen about it in the last lecture, we had seen the error correction learning, because essentially, what we did was that we took the error.

(Refer Slide Time: 26:25)



Let us say, that the error e k, that is the error and we are going to write it as d k or I think we might put the notation as t k that is target or d, you can call it as desired minus y k. And in fact, a more customary way of writing this type of expression is that, we put in parenthesis as this n and what does this n mean, this n indicates the discrete time step, so n is the discrete time step in fact, you can imagine that this is the number of iterations that is taking place.

Because, what we are doing is that, we are finding out the error, we are adjusting the synaptic weights. And then, after the adjustment is done, we are again going through the error computation, again adjusting synaptic weights and these things we are doing iteratively. So, this n is the iteration number or we are calling it as a discrete time, step discrete time step in the sense that in one time step.

Like, if say we take n is equal to 1; that means to say that in that time step, whatever things are being done will be written. And then, the next time step will be recognized as a time step 2, when we will be doing the second iteration. So, this is the error expression that we had considered and then ultimately what we had thought of is the minimization of this E n, this is the error that we wanted to minimize, the sum total of error and that was nothing but summation of e k n square.

Because, we are going to have different units and this is the kth unit that we consider and we have to sum it up, so this is e square k n. And then, we had in fact, derived this in the

last class that the change of weight that is going to take place, which we can write as W k j. So, W k j for the discrete time step n is going to what, remember eta, eta being the learning weight. Then, e k that is the error term, e k n we are going to write multiplied by input, which is going to be x j of n.

So, this is the rule, the weight updating rule that we had discussed. This is in fact, based on the gradient descent; we have derived that yesterday from the gradient descent consideration. And in fact, this rule is also often called in the literature as the delta rule or it is also known as Widrow-Hoff rule. So, different books or journals could be just calling these functions, calling this equation as either gradient descent or delta rule or Widrow-Hoff rule, whatever name they may say.

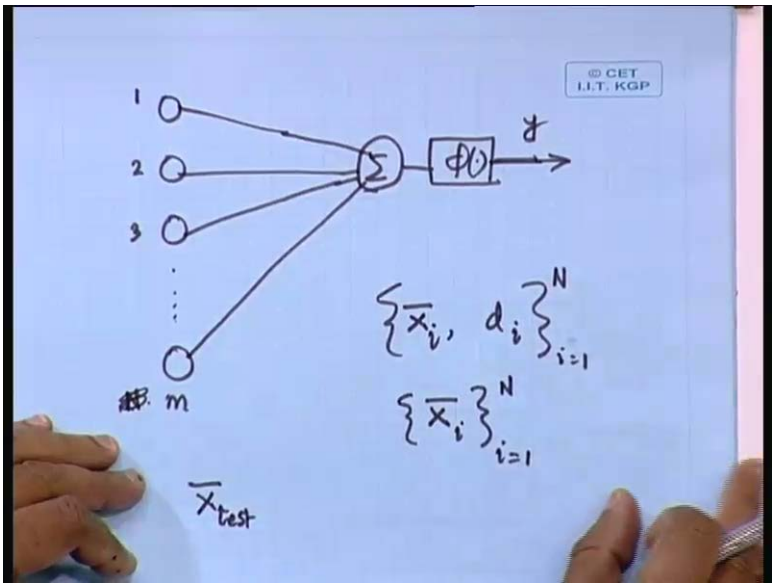
But, ultimately it means that, that this is the adjustment or the increment in weight that you are going to do. And then, the updated synaptic weight, that we are going to get is, that based on this delta W k j, the W k j is going to be updated. So, we are going to say that W k j, at the step n minus 1, should be equal to the W k j which we had in the time step n plus this delta W k j, at the time step n. So, this is going to be the updated synaptic weight. So, this is the simplest form of what we had discussed as the first category that is the error correction learning.

And the next 1, we are going to consider is the memory based learning. This is what we are going to consider now, as a matter of fact, if you are asking me that, which is the one that the human brain really follows, which kind of learning mechanism in fact, it is very difficult to say, it is such a complex process. And I do not think that so far, people have understood it in the exactly correct way.

Because, it is seen, that from different experiments people concluded that the learning mechanisms, they are different under different conditions. And it is thought that, sometimes we do the learning in this manner, sometimes we do memory based, sometimes we do Hebbain learning and all these things are essentially biologically motivated. So, the human brain in fact has been doing, have been adopting all this kind of learning mechanisms in some way or the other.

Now, for the case of the memory based learning, what we are doing is that, we store or rather memorize a set of patterns.

(Refer Slide Time: 32:51)



Let us say that, when we are considering a network, let us say that we have a network with N different inputs, supposing N and here we have the inputs 1, 2, 3in fact, the input will be normally called as the X 1, X 2, X 3 like that. So, these are the input numbers, I am saying 1, 2, 3 up to N and this are going to the linearizer activation unit and all that. But, what to say is that, this N or does not let us call this as m, different inputs we will say.

So, X 1 to x m are going to be a set of inputs and for that, we are going to get some output and that output is going to be the output y, ultimately we are going to get a y, for these set of inputs. Again, we take a different pattern, we take a different X 1 to x m and we get a different output in fact, in the learning mechanism, what we do is that, for the memory based learning, that we try to memorize the input vector.

The association between the input vector and the output, the desired output, that means, to say that you are feeding a set of values as input that defines a pattern. So, when I say a pattern, that pattern is consisting of X 1, X 2, X 3 up to x m. So, this pattern in fact, mathematically or expression wise, we can call it as a vector. So, we are in fact feeding a x vector.

So, when I say that I am feeding x vector; that means, to say that I am feeding this input and for an x vector, we are obtaining a desired output, that is d. Now, if I have multiple such an output unit, then even d is also going to be a vector. But, without loss of

generally T, I can consider the case of a single neuron as a output and it is response, I am calling as d.

In fact, d could be a vector, d could be a scalar, but just without loss of generality, I am considering d to be a scalar quantity. So, d assumes some value, associated with this pattern x vector, we are going to have d as the desired response. Now, for a system behavior, we might have decided to store, let us say N such patterns. So, that means to say that I am having a set of this x vectors and I call this as X i and i is equal to 1 to N. So, this N indicates, what the total number of patterns that we are going to feed to the system.

And for every pattern or every vector, every input vector X i, we are going to get a desired response, which is going to be d i. So, X i will be associated with 1 d i, X i is a vector giving the inputs and d i is the desired output of it. So, we are going to have an association between this X i and d i and again, this will be for i is equal to 1 to N. So, in a memory, we are in a large memory, we are going to store all these patterns, which we memorize.

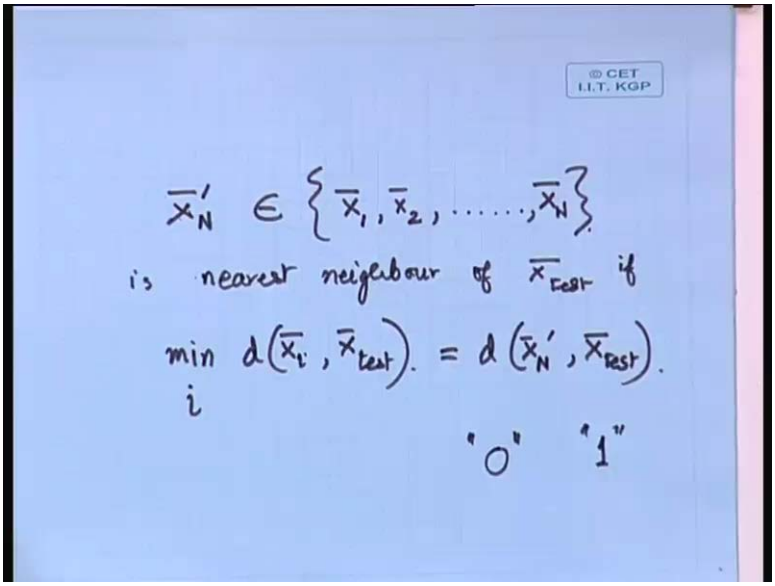
Now, the thing is that, when we are presented with a pattern that we have not encountered before. That supposing, we take this system to an unknown environment, but unknown environment in the sense that unknown vector that we are obtaining, but yes environment is going to be the same. Because, it has been trained with some environment only and it has received the inputs from a very similar environment.

But, we have got a, let us say new test pattern X test, X test vector that we are feeding as a as an input to this system and this X test is not anyone out of this set of exercise. And what we are going to find out is that, when we give this X test as a vector input. Then, what is going to be the corresponding output to it; that is the question that we are going to ask.

So, if the learning is proper, so here the learning is based on memory. So, what it does is that, it now finds out from the memory that which of these vectors is going to be closest to this X test. Closest in what manner, naturally we have to adopt some distance measure and the simplest of distance measure that we can think of is the Euclidean distance. This is in fact; all this x vectors are essentially m dimensional vectors, because we are considering m different inputs to the system.

So, it is m dimensional vectors and we are going to find out the Euclidean distances between the X test and each one of these X i's, which are already stored in our system. So, we are going to examine, that if X test is closer to X 1 or X 2 or X 3 or to X N and the 1 that is closest to let us say that x k happens to be or x j happens to be the closest. So, we can say that out of these vectors.

(Refer Slide Time: 39:49)



If we are able to find out a vector X N prime let us say, the X N prime essentially belongs to this set, the set of X 1, X 2 etcetera, etcetera up to X N, for N different vectors or N different patterns. We can say that this X N prime is the nearest neighbor is nearest neighbor of X test. If the minimum of the Euclidian distance between X i and X test and the minimum is searched over i space, that is varying i from 1 to N. And this is the minimum of all these d's happen to be, the d between X N prime and X test.

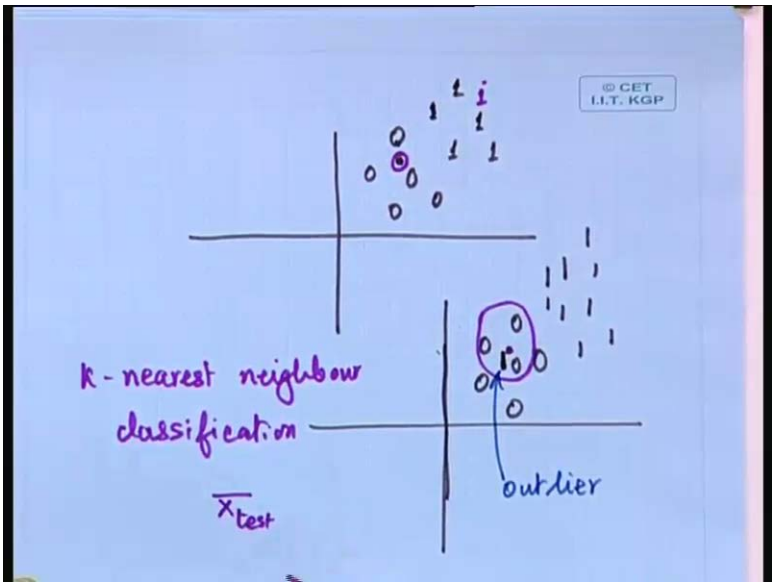
If it fulfills the condition, then we are going to say that X N prime is going to be the nearest member of this X test is this clear. It is one and the same as telling in a very simple language that the minimum of Euclidian distance that we are getting out of this set of X 1 to X N. In fact, it is just a simple mathematical representation of that, that you take the minimum of this distance Euclidian distance between X i and X test and this minimum, you find out over the i space and then if that happens to be X N prime X test.

Then, X N prime, which belongs to this, is going to be the nearest neighbor to this X test. So, then what we are going to do, we have found out the X N prime. So, then the

corresponding d N prime, which is the response then that is going to be the response, that we are going to consider for X test is that understood. Let me give you a very simple example; that supposing we have got different patterns and ultimately we are going to classify the input patterns into 1 of 2 classes.

Let us say, that the classes are such that some of the patterns, we are going to call as the pattern 0, means binary pattern classification problem. So, some of the patterns, we are going to classify as the class 0 and some of the patterns, we are going the call as the class one pattern. And let us say that we have got all these set of observations, which we have stored in the memory.

(Refer Slide Time: 43:04)



And let us say that, we just plot it in some space, a space containing all these inputs and let us say that for one particular pattern, we have got the classification 1 here, classification 1 here and classification 1 here, so these are all the positions of the vectors. So, for this position of the vector, the output is 1, this position the output is 1, this position is it is 1, may be for this position the output is 0, may be for this position the output is 0, may be for this position the output is 0. So, like this I have got many observations.

So, for some we have put it into the one class and for some we have put it into the 0 class, this is what we memorize. And now a test vector is presented to the system and that test vector supposing is this position. Now, what the nearest neighbor criteria really

tells us or the memory based learning tells us is that. At this position, it is going to find out that, which is the nearest neighbor.

Supposing, if the nearest neighbor is this 1, then we find out, if after performing this test, that is determining the nearest neighbor, we find that this 1 happens to be the nearest neighbor, then we are going to adopt this class to X test is that understood. So, that means, to say that, in that case we are going to classify this, I am just writing it with a different pen, just to indicate that it is a test.

So, all these black 1's are the training patterns or the patterns, which have been memorized, which are all ready there in the memory. And this is a test pattern that we have presented and the test pattern will now be classified as 0. If in this region, we present a test pattern that will be classified as 1, because it is nearest neighbor is going to be having a classification equal to 1.

Now, it is quite logical, I do not think that there is any reason to mistrust this approach, but if we analyze it very closely, we can follow, we can just have some flaws. You see after all, all these classifications of 0's and 1's, how have we done, we have done it based on some observations and our observations may be 99 percent correct. But, it may be 1 percent of time the observation may be going wrong also.

What to say is that, it could be that in the pattern association process or when we are putting it in to the memory. We have put let us say all these regions are having responses of 0's and one of these has got 1 and then these things are all 1's. Some response like this, where this one that has come up is indeed one of our wrong observations or we can call this one to be an outlier, so this is an outlier in our pattern presentation process.

So, when we are presenting this X 1 to X N there itself we have got one outlier, which is lying over here. But, whenever we are training the system, whenever we are putting in to the memory, we are not really finding out very closely. That whether, we have got a correct pattern or whether we have got an outlier, we accept everything as it is. Now, just imagine that, we are presenting a test vector and supposing the test vector happens to be placed over here.

If we adopt the nearest neighbor criteria, in that case what is going to be the nearest neighbor of that, that outlier can act now as a nearest neighbor to X test vector. And in

that case, the classification that this X test vector is going to get is 1, which is not a very correct classification. Because, we are making the classification, based on an observation based on an earlier observation, which is already wrong, which is already an outlier and it will then be characterized into the outlier, which is not correct.

So, naturally this approach of finding the nearest neighbor has got some problem. So, what are we going to do, we should find out a variant of this method. Now, any suggestion that can come from you people.

Student: ((Refer Time: 48:25))

Correct, it is very correct, as one of your friends pointed out that, it is going to consider the neighbors not neighbor. So, that plural itself means that the answer that your friend has given is quite correct. In fact, if this is the position, where we are going to have the X test vector, we are going to define a neighborhood around which we are going to consider.

In that neighborhood, what are the different X N's that are available, we are going to observe the X N's in this neighborhood. And in this neighborhood, if we find out that majority of those are going to be 0's, like this is the case. That supposing the nearest neighborhood is like this and then we find out that out of all these nearest neighbor classes, which is the 1; that is more predominant, in this case it is 0. So, in that case, we are going to classify this, X test vector as the class 0.

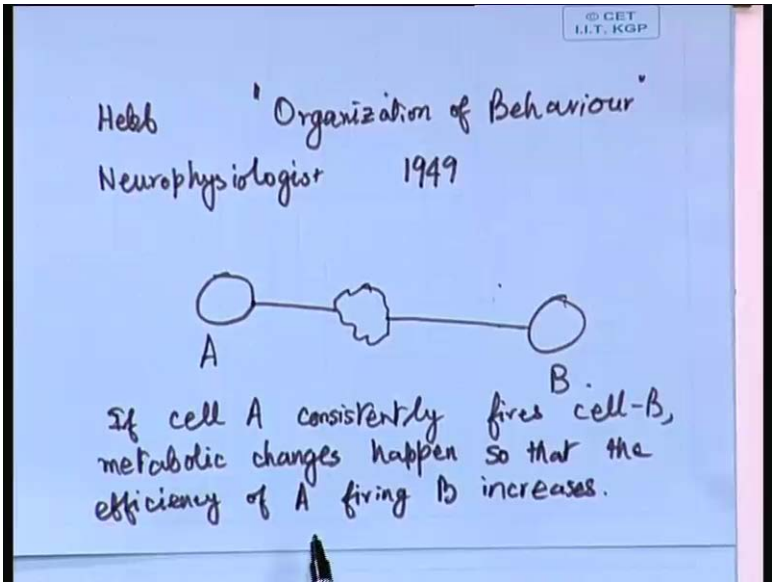
So, in that case, the outlier is not going to play any significant role, because the outlier will be outnumbered by the correct classes. So, in fact this kind of an extension to this idea is called as the k-nearest neighbor classification. And what is meant by k- nearest neighbor is that, we are not going to consider one neighbor, we are going to consider k different neighbors.

And it is the best out of the k nearest neighbor, it is out of the k-nearest neighbors, we are going to consult it is d's. So, we are going to consider the values of the outputs, which are there for the nearest neighbors and then depending upon the majority choice, majority output. We are going to decide that, what is going to be the response to this X test.

So, for X test, we are going to find out the best value of d by considering the nearest neighborhood, you can take three nearest neighborhood. Four nearest neighborhoods like that, in which it means to, say that, you consider not the nearest neighbor alone. But, three or four nearest neighbors or defining a nearest neighborhood zone and then ultimately classifying it.

So, this is the basic philosophy, behind the second learning rule, out of the 5, this is the second one that we have considered so far. And now we can come to the next learning mechanism, which is called as the Hebbian learning. In fact, Hebbian learning, it is thought that Hebbian learning is the 1; that is closest to our biological neuron learning mechanism.

(Refer Slide Time: 52:24)



In fact, all these theories basically originated from Hebb, in fact Hebb is a neurophysiologist, who in the year 1949, he came up with a book, which is called as the Organization of Behavior. In this book of Organization of Behavior, Hebb has in fact, proposed the kind of model that possibly our human brain neurons adopt. All that, it says is that, supposing we have got a neuron cell, that is A and we have got another neuron cell that is B. And this cell A and cell B, they are connected to each other using a synaptic weight.

So, what Hebb says is that, if the cell A happens to fire the cell B. Fire means, if cell A is cell A's activation at a given time can activate the cell B. In that case, the metabolism,

the metabolic changes that happen should be such that A's efficiency in firing B increases. So, all that it says is that, if the cell A, consistently fires, the cell B, then metabolic changes happen.

So, that the efficiency of A firing B increases, means whatever happens, so if cell A fires cell B, in that case the synaptic weight will be. So, increased that next time the cell A has got a better probability of firing the cell B. And I am not writing the reverse of that, the reverse of that is also true. If cell A does not fire cell B, in that case that synaptic weight or that synaptic connection should be weakened.

The metabolic changes will be such that, that synaptic connection will be weakened and ultimately the synaptic connection could be lost also. So, this is quite an interesting thing that Hebb had proposed. In fact, all that it means is that, whatever behavior happens with the behavior being cell A firing cell B, that is supported. So, that next time, you have got a better possibility of doing it.

Now, this postulate of Hebb could be very easily translated into our neural network mechanism also, in what manner. In that case, we can say that, we can define a presynaptic neuron and we can also define a postsynaptic neuron. So, the presynaptic and the postsynaptic neurons, they are connected to each other using the synaptic weight. So; that means, to say that, if we observe that the presynaptic neuron and the postsynaptic neuron, they are both activated at the same time step.

At the time step N, if we find that the presynaptic neuron and the postsynaptic neuron both have got activated. In that case, we are going to increase the strength of this W k j. And if the reverse of that happen, that if cell A and cell B, they are activated asynchronously, that is not in the same time step. If this happens in N and the activation of B happens in a different time step or rather to say in time step N, if I say that cell A is activated,, but not cell B or vice versa.

In that case, we are going to reduce the strength of this W k j, so that is the way, it is being adopted into the artificial neural network behaviors also. So, we are going to consider some details about the Hebbian learning mechanism and the other aspects of learning in the coming lecture.

Thank you very much.