Deep Learning for Visual Computing

Prof. Debdoot Sheet
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 21
Cost Functions

So, welcome. We had done till the previous few lectures on using down different kinds of methods, and we learned on how to train autoencoders. And the simple way of doing that was we while we did one part of it which was just for representation learning, there was another part of it which we were extending to multilayer perceptrons as for classification purpose. And over there you did learn about using something called as a cost function and that cost function which we are using was just a two norm or the Euclidean distance of the predicted to the actual one.

Now, standing on top of that I did speak out that we will be covering down more details about other types of cost functions as well. And that just trying to find out the mean square error or the Euclidean norm is not the only way of deducing out what are the different kind of cost functions which when used. And you have it varying as well. So, you will be using some of them for classification, some of them for trying to solve a regression problem.

So, today’s class stands on top of the stuff which we have done. And we will be getting into what is called as what are called as cost functions. And then through that the subsequent next lecture in which we would be covering down using the different kind of cost functions for solving the same kind of a problem. And typically for the case of an auto encoder what we will be doing in the lab is that for representation learning where it just tries to encode the patch itself we will be using a mean square error or an l 2 norm. And then eventually for the part of classification, we will be using a different kind of a cost function may be a binary cross-entropy or any of one of those. So, without much of detail let us get into what we are going to do in terms of this topic called as cost functions.

(Refer Slide Time: 02:04)

NPTEL ONLINE CERTIFICATION COURSES
Indian Institute of Technology Kharagpur | Department of Electrical Engineering

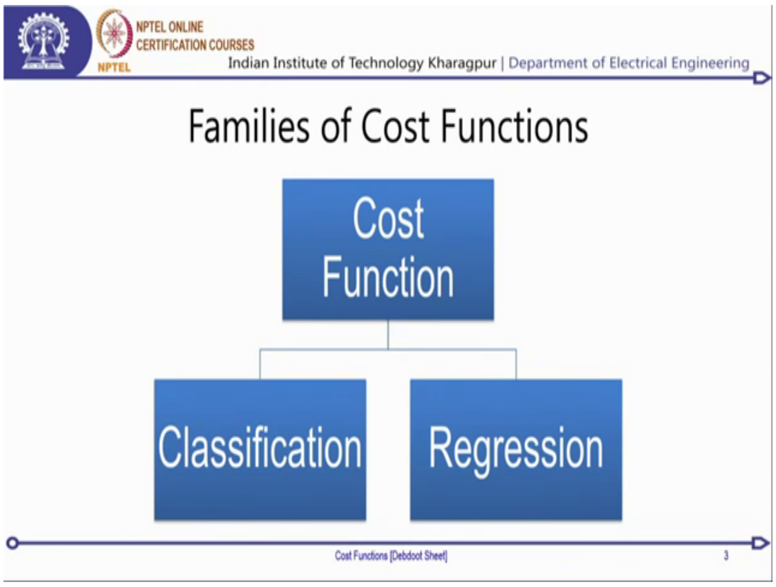
Organization

- Families of Cost Function
- Classification
- Regression

Cost Functions [Debdoot Sheet] 2

So, typically cost functions are grouped down into two broad families and that is called as either they are the ones which are used for classification or the other ones are used for regression kind of problem.

(Refer Slide Time: 02:18)


NPTEL ONLINE CERTIFICATION COURSES
Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Families of Cost Functions

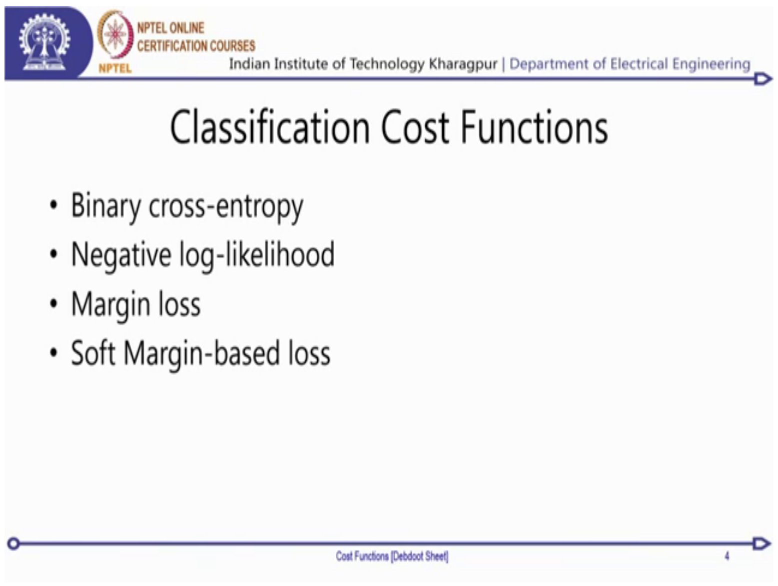
$$\begin{tikzcd} & \text{Cost Function} \arrow[dl, dash] \arrow[dr, dash] & \\ \text{Classification} & & \text{Regression} \end{tikzcd}$$

Cost Functions [Debdoot Sheet] 3

So, typically this is what it comes down to. Now, when you go to the fact that you have a cost function which is a classification cost function; now, what typically happens with a classification cost function is that you would be solving a classification problem or where you would need to associate a class label to a given kind of an image. So, if I have an say for the mnist kind of a problem where you had handwritten digits of zero to nine and you just had to identify which digit it was. So, this is a typical case of a ten class classification problem and that is something where different kind of cost functions which are actually of the nature of classification are the ones which will be used.

Now, the other one is what is called as a regression problem as well. And regression problem is typically the kind of auto encoder problem for representation learning which we are doing in which you would try to predict out what is the state of a variable on the other side of it. And then those kind of predictions can be done typically you with regressions or even a restoration problem such as . So, and all of them have different properties which you would like to see and that is what guides down the cost functions in its own way.

(Refer Slide Time: 03:24)



NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Classification Cost Functions

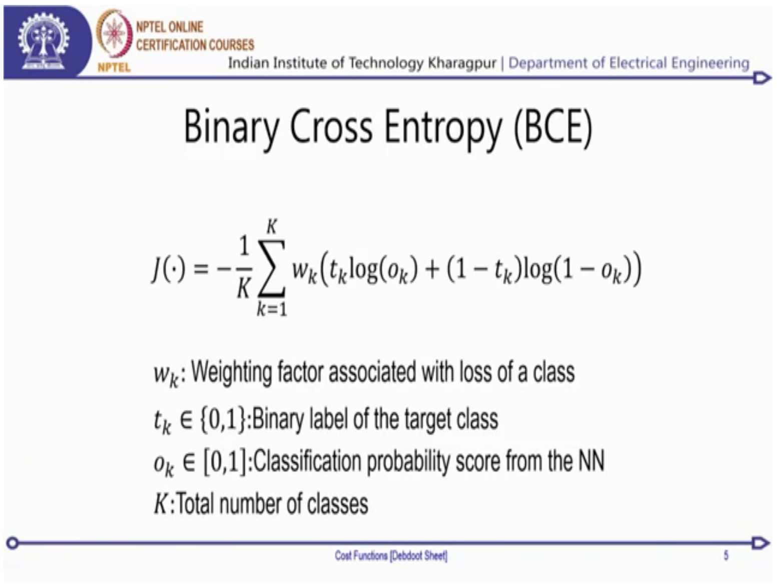
- Binary cross-entropy
- Negative log-likelihood
- Margin loss
- Soft Margin-based loss

Cost Functions [Debdoot Sheet]

4

So, for classification the typical ones which are used are called as so the most common one is binary cross-entropy the just successively following is the negative log-likelihood that is a margin loss and the soft margin-based loss. Now, we will be getting into what is the mathematical form for each of them as well as what are the different attributes which come out when you are going to use each of these cost functions. And that will more or less guide like what is the kind of application scenario in which you will be using a particular kind of a cost function. So, it may not always necessarily be so that using any cost function you will be coming down to your best performance over there. So, not just guaranteeing that you will be getting none the best accuracy over there; now the cost function does have a significant role to play in terms of identifying what you are trying to solve.

(Refer Slide Time: 04:14)


NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Binary Cross Entropy (BCE)

$$J(\cdot) = -\frac{1}{K}\sum_{k=1}^{K} w_k\left(t_k\log(o_k) + (1-t_k)\log(1-o_k)\right)$$

w_k: Weighting factor associated with loss of a class

$t_k \in \{0,1\}$:Binary label of the target class

$o_k \in [0,1]$:Classification probability score from the NN

K:Total number of classes

Cost Functions [Debdoot Sheet] 5

So, let us get into the first part of it. So, the first one is what is called as a binary cross-entropy. And now how it is defined is typically of this one, where you have a weighing factor associated with the class of a law with loss of for a particular class. So, say that I am looking at so this is typically where you take a k class classification problem. So, for my mnist this K is equal to capital K is equal to 10, because I just have ten classes of digits 0 to 9 available to me. Now, w k will be a weight associated with loss of a class. And what that typically means is that if I want to make classification for one of the classes more stronger which is say I want to very accurately classify my zeroes whereas, I can have some sort of error if I am trying to classify between the other classes. So, maybe if something is written down as 7 and that gets wrongly classified as 2, it is not so erroneous, but if I am wrongly classifying that as a 0 then it is highly erroneous.

If that is the case then what I can do is and these are very dependent on the kind of problem you are dealing with. Typically for digit recognition it is where it is an ISO equivalence problem which means that all classes are equally important and then you would try to put down equal weight to each of these classes. But then it may be in a case say I want to detect and identify if a particular image given down is cancerous or it is not cancerous. Now, in those cases its really risky to say a cancerous patient as non-cancerous. Whereas, if you have a non-cancerous patient you say that as cancerous, there is a mental dilemma associated with the patient obviously. But then telling a non-cancerous patient to be cancerous in some way does not as such hamper the life of the patient, because anyways they would be going down through successive more tests where it would be identified that this patient is not cancerous. But in case that the patient is cancerous and you classified them as non-cancerous they are not going to go through any subsequent tests and just move out. So, I mean that is a life taking point over there. So, each class is not necessarily always equivocal or equally valued, but then certain classes might be having more importance than the other classes. So, in that case of a thing is where you would be putting making use of this factor w over here.

Now, t k is basically the binary label associated with the target class. Now, you remember that in the classification problems which we were solving till now and the way we were defining the target output vectors over there, so it was always called as a number of classes cross one. And it was a one hot vector which means that that particular class which belongs which is the class level of that sample which is being used for classification. So, which is being used for training or say whenever you are classifying, so that particular element of the tensor is what is going to remain as one everything else is going to go as 0.

So, in my minist classification problem if I have 0 to 9 classes, and the digit given down over there is say the number 9. So, it means that all the previous 9 elements over there will be 0, and the 10th element is going to have a class level of 1. If the digit given down is 0, then the first element of that tensor of that 10 cross 1 tensor is going to be one rest everything is going to be 0, so that is what is t k over here and the subscript k is for each of these classes.

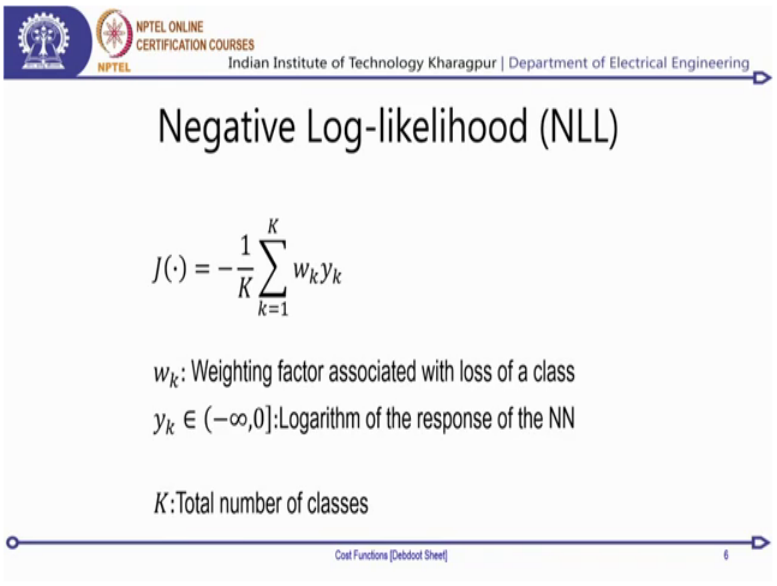
Now, on the other side is a classification probability score from the neural network. Now, if you look over here for t k this is just a value of 0 or 1, you do not have a floating point value going down over there. So, there is no continuous value in the range of 0 to 1, but it is either 0 or 1. Whereas, when you are classifying or whatever is coming as the predicted output of the neural network that can have any value in the range of 0 to 1. Now, keep this in mind that binary cross-entropy necessitates that the output of the neuron coming down over here lies in a continuous range of 0 to 1 and that would also mandate that you use an appropriate transfer function over there. So, the non-linearity which I impose over that needs to be in a range of 0 to 1.

Now, from your previous classes or non-linearities when we were dealing down with simple perceptron model, you can definitely recall that sigmoid is one of those kind of a functions which will impose a value of 0 to 1. So, if input to a sigmoid tends towards minus infinity, it puts a value to 0; if it tends towards plus infinity it scales up and puts it to a value of 1, but it will never exceed the value of 1 or go below the value of 0. Whereas, if you have a tan hyperbolic then it ranges in the value of minus 1 to plus 1 and that is not something which you can use for binary cross-entropy. Neither can you use a rectified linear unit or Raleigh kind of a function which will come down eventually when we start studying about convolutional neural networks, they are more famous. So, this kind of transfer functions are necessarily very important over here and you need to choose a particular one which does impose your output to also stay within this range.

Now, what typically you do over here is if you look into one pair of this cost over there. So, t k and is basically what is the probability that the class k exists. So, and then one minus t k is the probability that the class k does not exist and log of 1 minus is the probability that it was not predicted. So, you take a marginal product of these two and then you weight it down by that particular class over there. So, in any case if your prediction is wrong, then you would be seeing that it comes down to 0, 1 and vice versa is what happens over here and then because on by virtue of this log, and this o k value over here. So, o k is in the range of 0 to 1. So, 1 minus o k will also be in the range of 0 to 1 and that makes that this output of this log is basically a negative number on both the cases. And in order to get this cost function as a positive one you have this balancing minus over here which balances out the total contribution coming down from this whole thing.

So, this is a typical function which is very commonly used for classification problems and in fact, in the next lecture, where we will be doing more on coding aspects of it. So, we will start down with the auto encoder like architecture using a mean squared error cost function over there. But eventually when we do a transfer or just transfer it out, transfer the weights which have been learned for representation part over there to create a multi layer perceptron, we will be changing over to use this kind of a cost function which is called as binary cross-entropy. And see what is the change happens. And in fact, there would be changes in terms of the dynamic range of the errors which come down, because here you can see that the dynamic range of the error is anyways limited in the range of 0 to 1 and then it cannot cross that, so that is like really an interesting factor which will come to play.

(Refer Slide Time: 11:05)



NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Negative Log-likelihood (NLL)

$$J(\cdot) = -\frac{1}{K}\sum_{k=1}^{K} w_k y_k$$

w_k: Weighting factor associated with loss of a class

$y_k \in (-\infty,0]$:Logarithm of the response of the NN

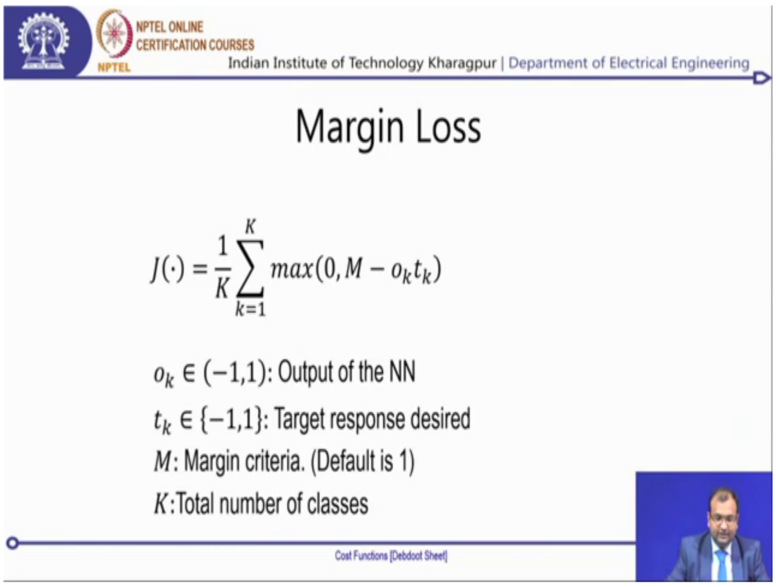
K:Total number of classes

Cost Functions [Debdoot Sheet] 6

So, the next part of it is what is called as a negative log-likelihood error. So, how this goes is that you have also a weight over here, because it goes down by the same argument that this weight is pretty much dependent on which particular class. So, if you need to have one particular class very perfectly classified then you will be putting down your a very high weight to that one or if all of the classes whichever you want to classify are of the same kind of a contribution then you put down the equivocal weight or once basically for all of them. So, y k over here is basically the log of the response of the neural network. So, what it means is that in the earlier case say you took the sigmoid which was in the range of 0 to 1.

So, what we do over here is you put down a logarithm after that and that scales down this output value. Now, if your output from the neural network before the log was in the range of 0 to 1, you know that definitely taking a logarithm of some kind either to a base e that is a natural logarithm or base 2 or any of your base 10 even would work. So, that would scale it down the maximum value which is 1 a log of 1 is always 0; whereas, the any value which is close to 0 that is something which tends towards minus infinity with your log scale. So, this is where it puts it down and this function is typically what is called as a negative log-likelihood. So, the negative term is what comes down over here, and this part is just the log-likelihood for each of them. And this is a very straight forward function which can also be used.

(Refer Slide Time: 12:39)



NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Margin Loss

$$J(\cdot) = \frac{1}{K}\sum_{k=1}^{K} max(0, M - o_k t_k)$$

$o_k \in (-1,1)$: Output of the NN

$t_k \in \{-1,1\}$: Target response desired

M: Margin criteria. (Default is 1)

K: Total number of classes

Cost Functions [Debdoot Sheet]

The next part of it is what is called as a margin loss. So, in margin loss typically what you would do is that you try to find out the margin between your so it is like you have your output of the neural network which can be in the range of minus 1 to plus 1. So, here it is pretty different because here you see that its no more than 0 1 0 1 kind of a vector, but it is basically minus 1 and plus 1. So, whichever class exists that is what is associated with plus 1; whichever class does not exist is what is associated with minus 1. So, your target which is the actual class label for a particular sample that will be either minus 1 or plus 1, whereas the output of the neural network over here can be in the range of minus 1 to plus 1.

Now, typically for using this kind of a cost function of margin loss what you would be making use is that the non-linearity has to be definitely tan hyperbolic non-linearity which can generate in the range of minus 1 and plus 1. If you use say a sigmoid kind of a non-linearity that is not going to solve your problem in any way. So, in order to keep it between the range of minus 1 and plus 1 you will have to use down a tan hyperbolic non-linearity over here.

Now, what we do over here is that you take a product of these two o k and t k. Now, say that both of them are minus 1 over here. So, this total product over there is what leads down to plus 1. Now, you have another factor, which is called as a margin criteria. Now, the default for it is always one this margin is basically the amount of tolerance you would allow down. So, if both of them are minus 1 and this is also one, this part becomes basically a 0. Now, maximum of that is what is called on as a 0 and that means, that so this interior part within the summation is what goes down as 0, so your classification is also 0.

Let us look at the other part of it which is for one of these classes where your output was 1 and the target was also 1 in that case also you will be getting down the same thing coming down. Whereas, if you have the opposing criteria, so your target was actually plus 1; whereas, your output came down as minus 1. So, in that what you get down is that this factor over here becomes down to negative quantity say this becomes minus 1. So, minus of minus 1 is what makes it plus 1. So, you have a m which is 1 plus 1 and that is a value which is higher than 0, and greater value which goes down over here.

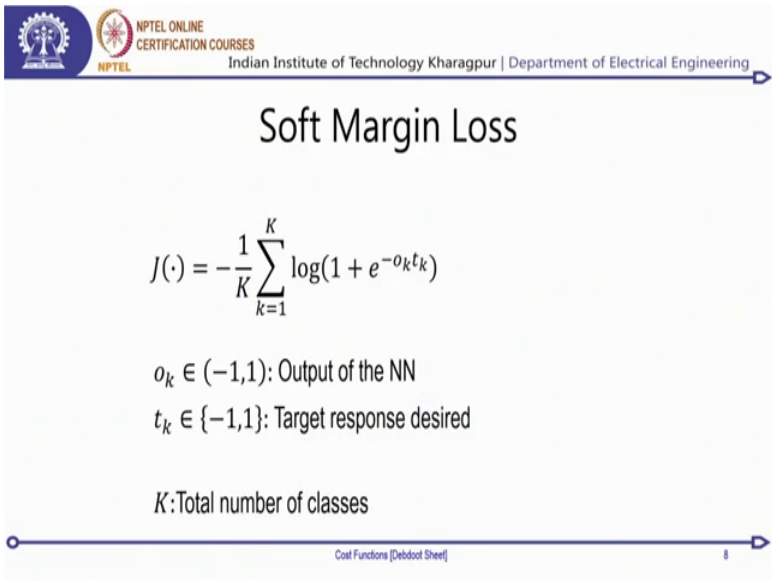
Now, this max part over here is what limits that if in case this difference over here m minus o k t k this is the value which is lesser than 0 under certain condition then the error will be made down to 0 itself. So, that is sort of the minimum value of error which we will be getting down at any given point of time. And for all other cases it is a different one.

Now, typically this is a function which you would be using when your non-linearities which you are going to deal with are in the order of minus 1 to plus 1. And whereas, again choosing the non-linearities is also always not in your hand, because you would choose a non-linearity in order to match down the kind of a response you are looking at. So, if your say images which we are using they somehow come down in the range of some minus value to a plus value and this is not so complicated say you take down synthetic aperture radar images or you take down your ultrasound images or take any of these MR signals. So, these are the ones where they do have a negative to positive range over there, and this is a sign number system in which your data exists over there.

Now, whenever you have a sign number system in which the data exists and say you normalize it gets normalized in the range of minus 1 to plus 1 where 0 corresponds to basically that some sort of a normalization factor and a standardization factor. So, zero typically say for our c t, 0 is basically the hounds of the Hounsfield unit which is associated with water now for sorry not with water so with air. Now, if you are an m r that will be the Hounsfield unit which is associated with water and this is what is used for calibrating your system over there.

So, in those kind of image families where you do get a negative to positive, you would try to put down at non-linearity as well which is ranging in the range of negative to positive. And in that case your margin function also has to be different, because you cannot use a binary cross-entropy which would not allow you to have minus 1 to plus 1. It needs you to have between 0 to 1. You cannot use a negative log-likelihood which also has a similar problem. So, this kind of a cost function does come to a huge respite in those kind of scenarios.

(Refer Slide Time: 17:10)



NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Soft Margin Loss

$$J(\cdot) = -\frac{1}{K}\sum_{k=1}^{K} \log(1 + e^{-o_k t_k})$$

$o_k \in (-1,1)$: Output of the NN

$t_k \in \{-1,1\}$: Target response desired

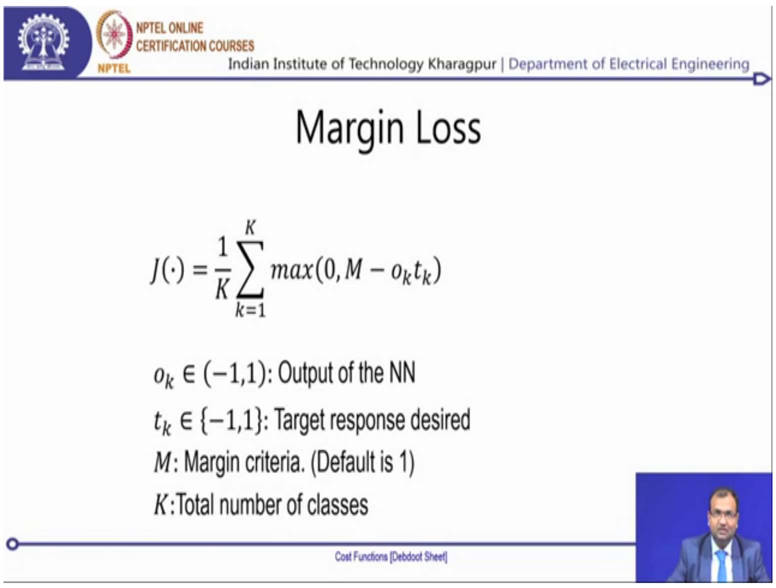
K:Total number of classes

Cost Functions [Debdoot Sheet] 8

So, now standing on top of that the next one is what is called as a soft margin loss. And in a soft margin loss, what you typically do is that while you have your o k and t k over here and they are in the range of minus 1 to plus 1. So, what you would do is you take another log of 1 plus of this number. Now, typically remember in case of your earlier margin loss, so it was a max of zero comma m minus which was your margin minus that net factor over there and then you made it that it is going in some sort of steps and then the minimum value which comes down as 0. Whereas, if you look at this soft margin loss over here, what it typically does is it raises this total product over here the marginal product to a n e power of something and then adds a 1.

Now, what that would necessitate is that if I do not have this one over here, so this becomes e power of minus o k t k and then this value over there. So, you have a log of that factor. So, this basically becomes of o k t k summation of o k t k and that will be a value of zero either 1 or minus 1 or plus 1 that is what its typically going to be nothing else. Now, here by doing this when you are going to scale it down in that particular range. So, if I have my this minus o k t k if both of them are negatives then this becomes a positive quantity e power of 1. So, this becomes log of 1 plus e power of 1, so that is a number which is greater than one that is clearly a number which is greater than 0 as well as a number which is greater than 1 as well. Now, these kind of functions, so this log soft. So, this is sorry the soft margin loss over here which you have. So, the soft margin loss basically allows you to give a very continuous kind of a function.

(Refer Slide Time: 18:50)


NPTEL ONLINE CERTIFICATION COURSES
Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Margin Loss

$$J(\cdot) = \frac{1}{K}\sum_{k=1}^{K} max(0, M - o_k t_k)$$

$o_k \in (-1,1)$: Output of the NN

$t_k \in \{-1,1\}$: Target response desired

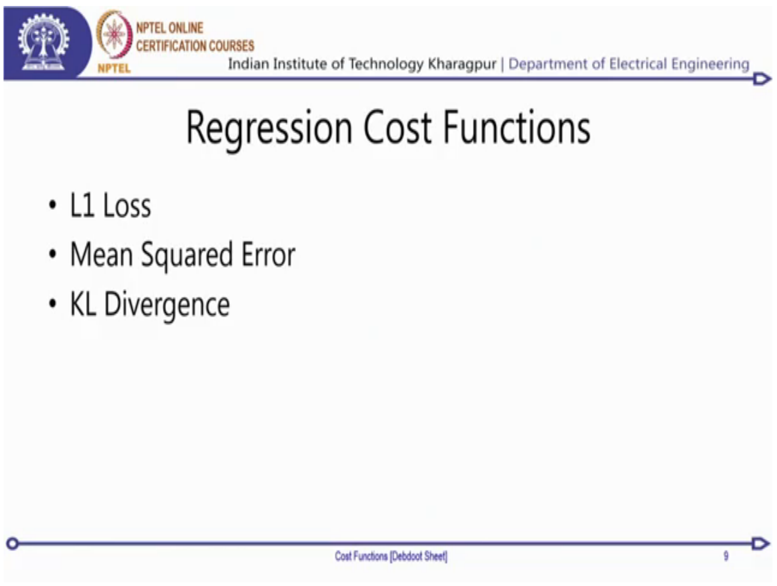
M: Margin criteria. (Default is 1)

K: Total number of classes

Cost Functions [Debdoot Sheet]

Now if you look into the margin loss in the earlier case which we had done, you also know that for any kind of a loss function you need to have its derivative. Now, this derivative does have a discontinuity because of the max function coming down over here. Whereas, if you look into the soft margin loss function, this is a very continuous function does not typically have a discontinuity and then you have a perfectly derivative computed out. Now, so computing our derivatives is what I would rather leave to you and we will be having exercises the weekly assessment exercises where you would be getting down questions want to find out these derivatives as well. But this is a this is sort of an experience and a learning which you will just be gaining down by taking down these functions and we are not solving out derivatives because that is clearly much on the prerequisite side of it that you know how to find out derivatives of certain functions falling down its chain rule.

(Refer Slide Time: 19:44)


NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Regression Cost Functions

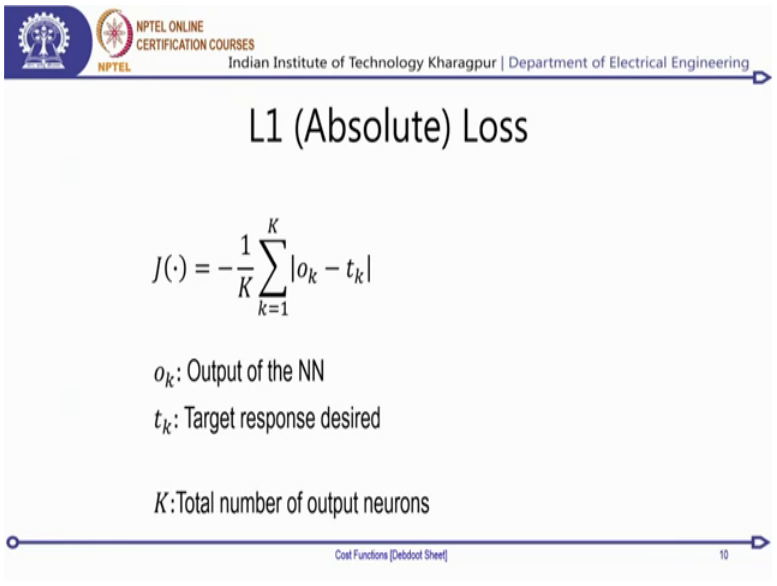
- L1 Loss
- Mean Squared Error
- KL Divergence

Cost Functions [Debdoot Sheet] 9

Now, that was all we had to speak with classification related cost functions. The next one is what is called as a cost functions, which are related to regressions. And here there are basically three very distinct category of cost function which we will be using so two of them are basically which are geometric in nature L 1 loss, and the mean squared error or the L 2 loss, and KL divergences which is probabilistic in nature.

So, you remember that clearly the earlier ones which we were doing on classification. So, your binary cross-entropy was an information theoretic because you were using cross-entropy. Negative log-likelihood was a probabilistic function, which you are using. The margin loss and the soft margin loss these two were against structural cost functions because they were just trying to look into your geometry properties like there is a fixed point and how far are you from that point in terms of your cost space. So, there are these different kinds of distance metric families based on which these cost functions get derived. So, we follow the same thing for a regression problem as well.

(Refer Slide Time: 20:45)


NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

L1 (Absolute) Loss

$$J(\cdot) = -\frac{1}{K}\sum_{k=1}^{K}|o_k - t_k|$$

o_k: Output of the NN

t_k: Target response desired

K: Total number of output neurons

Cost Functions [Debdoot Sheet] 10

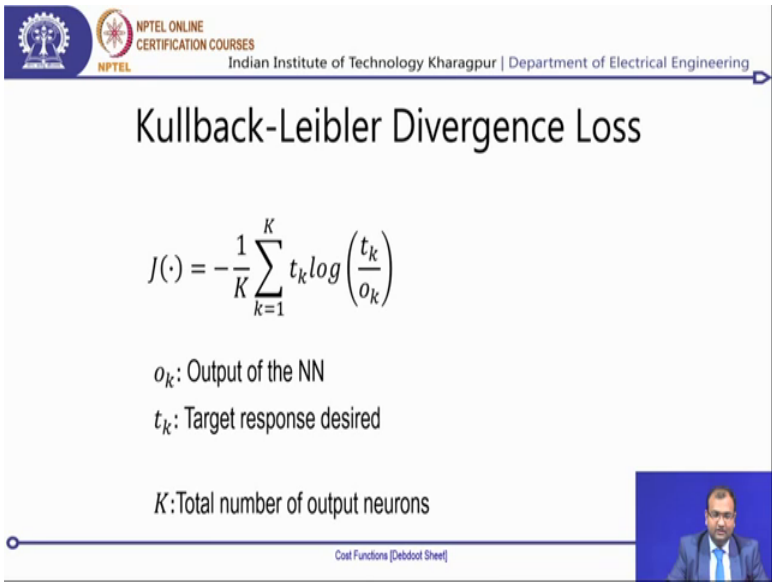
So, an L 1 or an absolute loss function is what is something defined of this particular form. Now, what you can see is that whatever is the target output and whatever is the output of predicted output of the neural network, you are just going to take an absolute of the difference between them. So, this absolute of the difference will mandate that this value stays in the range of in a in a positive number over there. And then no issues occurred on. So, there is a slight error, so this minus should not have been there. This is just one by k times of this average of this one.

Now, one thing clearly you can note down is that if this difference is negative or this difference is positive in both the cases you will be getting a positive number and this will come down to the minimum value which is 0 and that is how this cost function is bounded in itself. That it is the lesser the least value which you can get down is zero and the higher value can be anything. Now, the good part about this one is that in the earlier cases on classification you did see that it was mandating that whatever you predict out your o ks they need to be in the range of 0 to 1 or minus 1 to plus 1 strictly within a bounded range. You do not have an open range over there.

So, if you are taking down say images and I want to predict down the full scale of dynamic range of the image then I do not get that option. So, if I put on some input of 200, and 0 to 255 and 8-bit integer, the output side cannot be 0 to 255, but the output will be mapped down to 0 to 1 range or say minus 1 to plus 1. Whereas, in case of regressions that is not something which we are generally interested at and this kind of a cost function does allow me to have flexibility on choosing what can be my range of outputs on which I would like to play.

The other one which is called as a mean square error loss or the l two loss. So, what it does is it takes a square of this amplitude over that. So, this is also going to be in the range of 0 to 1, 0 a greater than 0 basically not necessarily in the range of 0 to 1 and this minus will not be that this is just a error in putting it down. So, what this would mandate is that you will always be having a positive loss coming down out of these kind of functions totally. And this is also an unbounded one and that is what makes it very interesting for your regression problems in it is in own way.

(Refer Slide Time: 22:54)


NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Indian Institute of Technology Kharagpur | Department of Electrical Engineering
Cost Functions [Debdoot Sheet]

Kullback-Leibler Divergence Loss

$$J(\cdot) = -\frac{1}{K}\sum_{k=1}^{K} t_k log\left(\frac{t_k}{o_k}\right)$$

o_k: Output of the NN

t_k: Target response desired

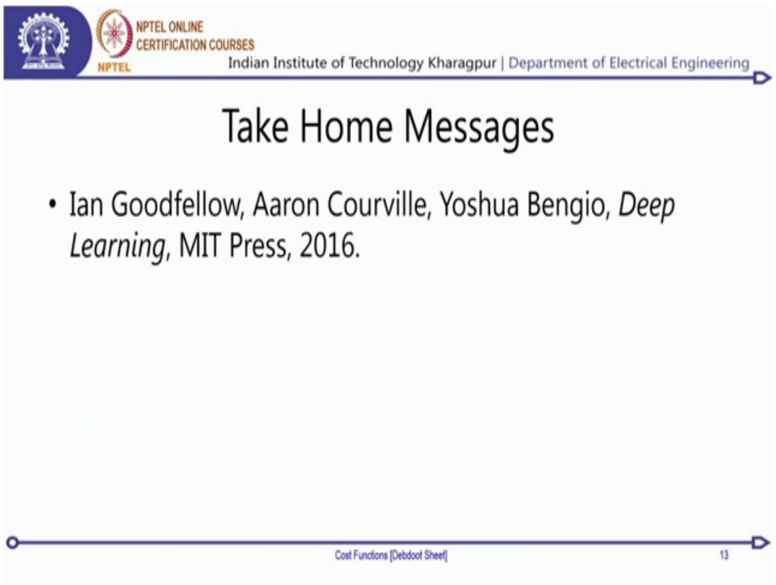
K:Total number of output neurons

Now, the other one is what is called as a Kullback-Leibler divergence, and this is a information theoretic loss function which comes down. So, you have your t k which is the target response which comes out, and o k which is the output from the neural network and this is again a multi hot vector or all your outputs over there will be mapped down. So, your k can be a particular location which you are taking down on your predicted class over there, or it can be even a class label if you would like to take it in that way. And here one thing which generally mandates is that o k and t k also need to be in the range of 0 to 1 though it is not very strictly imposed in terms of this equation. But coming down from the fact of Kullback-Leibler diverges actually being an information theoretic one we would prefer that we still treat o k and t k in terms of probabilities or in the range of 0 to 1. So, this typically goes down into that form and does it.

There are other ones as well. So, you can take down Jensen-Shannon divergence loss as well and what that would mean is that here while the probability which were weighing down with the mutual information. So, t k by o k is going to give you the basically the mutual information content over there, and then you weigh it down by t k which is your target response. Now, if you would like to weight by o k, you can still do it. So, you put down o k over here put down o k here and a t k over here, but that does not give you a totally balanced one.

So, if you want to get down a balanced one, you can take a sum average of that and that is what is called as a junction channel. So, we will have down exercises where we deal more about different kinds of these statistical divergence based and information theoretic measures for costs when you are trying to solve a regression problem and that sort of brings us to the end of it and we have the take-home message.

(Refer Slide Time: 24:36)


NPTEL ONLINE CERTIFICATION COURSES

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Take Home Messages

- Ian Goodfellow, Aaron Courville, Yoshua Bengio, *Deep Learning*, MIT Press, 2016.

Cost Functions [Debdoot Sheet] 13

So, if you want to get more into cost functions and learn more about the details and the suggestion is to refer back to the textbook on deep learning by Goodfellow, Courville and Bengio that has a consolidated description of different cost functions, their ranges and the cases where you would be applying them. So, with that we come to an end for today’s class.

Thank you.