Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-18

Yeah, early stopping I think I had told you data set augmentation I said L2 norm on the weights is where we started, right. So this regularization as we move forward, so one of the other ways to actually regularize is to introduce a regularizing term. Regularize means, means you know various things for example, restoration problems and all. The classical way of calling what regularization is, is a little sort of you know this one different. Here by regularization as I said what we really mean is that we would like, I mean what you say right, we would like to kind of reduce the weights right which we are using, the total number of weights. You want to kind of make sure that only those weights play a role right that really have to play a role.

And one way to do that is to actually introduce an L2 norm on the weights and which is what which then leads to a modified sort of a cost function. So then you know like I said I think last time I had written some L tilde theta for that right. So if you go to that expression, so what we had was yeah L tilde, okay if you just talk about weights I will write it as omega okay. So to be more kind of specific if you are talking about the weights, L omega plus let us say some beta by 2 norm of omega square, norm of w square.

So this L omega is still the same old cost which could be a mean square error or which could be a cross entropy loss right. So this can be an MSE or a cross entropy depending upon what kind of a problem you are solving. By the way right, so this cross entropy we are talking about this cross entropy there is something that you should be aware of which I will just you know there will be a brief sort of a digression but then we will kind of quickly come back. There is something called a softmax which is used when you talk about a cross entropy okay this is like a module that we actually put I mean so right now what you have is let us say if you are doing a classification problem right you will come till the end till the output layer and in the output layer right I mean you can actually think about because you have let us say right I mean you have a network right which takes some input and then let us say right you have got let us say the simplest case is what I mean you can think about think about right 2. So you can think about just you see 2 outputs okay in the simplest of cases.

Let us say I am just trying to do a 2 class problem I want to say whether it is a foreground

or a background right. So then what you can then in such a case right you will have maybe a success for let us say a foreground is P then a success for the background is 1 minus P right and this layer right which you have at the output that is actually sending you that value I mean you can think of because you need a probability as the output because you are trying to solve a classification problem you might want to say that I would rather use a sigmoid there because I know that a sigmoid will go from 0 to 1 therefore right I would just want to use a sigmoid as my activation function at the output so that the values right are actually you know they would be a probability they will sum to 1 between 0 and 1 all that right. But what can happen is if you kind of so right this will be a binomial case but if you had you know a multinomial case where let us say you have got you have got you know several classes right I mean typically one does not limit oneself to just a foreground or a background. For example, that image net kind of a classification problem that we had 1000 classes right. So then I mean it no longer makes sense to talk about P and 1 minus P and so on it although the no is they should all sum to 1.

So in so it is it is in such a situation that the softmax comes in and the softmax actually right enables you to not just limit yourself to a sigmoid I mean you can have any activation in fact with the output okay even if it is a classification problem. Because for the earlier case it looks like you should have a sigmoid because that is what will output a value between 0 and 1 but you know that is not actually necessary especially when you go to multinomial case right when you want to see a multinomial case then what is done is. So for example, you have your outputs coming in from the activation of the at the output then let us say that that those are okay I mean what we will be using some Y hat right. So let us say let us say what you are using is some Y1 hat last time I think that toy example that we took we had Y1 hat and Y2 hat but let us say we are we are we have more of them you know Y2 hat and let us say we have all the way up to some whatever you know let us say Yn hat. So we have got n classes right it is a multi class problem now and now what we can do is now because of the fact that you see these Y1 hats could be limited to values between 0 and 1 or they could even come from some linear activation function right we do not know it could be anything for that matter then what is done is for so for example right you would convert this Y1 hat into let us say let us call this sigmoid let me just call it as some sigma then what I will do is I will take sigma of Y1 let us say okay in fact for any Yi right Yi hat I will write this as e raised to Yi hat by summation e raised to Yj hat j going from 1 to n okay for the n number of classes.

So the thing is that what you do is you actually take the outputs that that you get for in fact all the classes Y1 hat Y2 hat and then you sum them up right at the bottom denominator and then numerator you raise e to the I mean you raise you know you keep it as e raised to Yi hat. Now the idea is that if you now one thing is right now Yi hat whether it is negative or positive it does not really matter it could carry any value finally sigma Yi hat will now

be a number between 0 and 1 because if you sum up you know if I do summation sigma Yi hat over all the i's equal to 1 to n right that will be equal to 1 because this will this if you keep adding it will be the same as whatever is so that you can see. So then it will be like summation e Yi hat and then at the bottom you have summation e is e what is that Yj hat and they both are equal and therefore these values will sum to 1 and it is a kind of you know clean way to clean way to sort of go from a binomial case where you would have had a situation like P and 1 minus P and most likely you will think of having a sigmoid there but a multinomial case you can directly go using the software softmax not software softmax. So I think of the softmax as an additional module that is coming in so you are still free to do whatever you want prior to that whatever activation you want to use do not limit yourself to 0 to 1 or anything that can also this Yi hats can be so they can come from a real number. So they can be whatever they are there is no constraint that they have to be positive and all.

And then eventually you then kind of you know convert them to a probability and then what happens is right so when you do this cross entropy loss rate I mean you know we saw it we had like Yi log Yi hat right that is what we had summation i going from the number of classes that was here that would be your cross entropy loss. So this Yi which is actually the true label right see you have a network and let us say right I pass the example of let us say a horse image right inside. Now my output vector right the sort of a target vector that would be what suppose my let us say suppose this entry is for horse which I will have to keep it as fixed right through all my experiments. So what I will do I will make this as 1 and then kind of right reduce all of all the others to 0 because I will say that you know this is actually a horse and therefore that particular element so right this is going to be your Y okay and Y at some let us say kth entry will have a 1 okay and therefore right when you when you do a cross entropy loss it will do like this right. So what we will do is you will have like Yi log right I mean Yi hat right summation over i.

So now what will happen is so such a vector so this is a target is what you want ideally to be that is when you pass a horse you want your Y hat to look like this you should have 0s everywhere except that in that one place whatever kth position it should be 1 and then rest it may not happen right it will try to get there. So when you compute the sigma Yi hat you may end up with the Y hat right so this is a target so this is a target and what is being estimated is some Y hat right which is this Y1 hat Y2 hat and so on and this need not exactly look like Y but then it will try to get there because that is what that is what right that is what we talked about earlier right for cross entropy you know so you want this Yi hat to come as close as possible to Yi right and such a vector is called actually a one hot vector okay. So a vector of this type is called a one hot vector what it means is that only one element is really active right rest of the rest of all or all 0s right so it is called a one hot vector so if you encounter one hot okay you know do not so you should know that

what is being meant right and therefore so I just wanted to mention that talking about these losses for L of W right when you are talking about the loss functions of the weight which is typically the original cost right that we have been talking about till now it could be MSE it could be cross entropy whatever it is but if it is cross entropy then very likely that you are dealing with a classification problem or whatever a segmentation problem where you have got multiple classes and right you could be you could be labeling every pixel or you could be labeling an entire images whatever you know to know right to which class that particular object in the way that image belongs so in that case remember that there will be a softmax right at the after the output layer and then you convert everything to a probability okay. Now see this I am just going to indicate an outline okay as to how one goes around so the idea is that right by introducing this additional norm on the weight right so what you are actually effectively doing is so suppose I look at the gradient of L tilde omega because right that is what I will have to now use right till now in my iterations I was using dou L by dou dou dou W now I will have to use dou you see L tilde by dou W because that is my new cost so if you look at this right this is going to be gradient of L W with respect to W plus you know last time it will be sorry it is going to be beta W everything is a vector here okay. So now the argument right if we if I start deriving right it is going to take time but it is a little tricky but not so tricky so what you have to do is you know you have to do suppose let us say see what you want to what you want to understand is with respect to this new cost function okay the new cost function if there is an optimum that let us say what is the what is the symbol that I am using for that tilde W tilde okay if that is what is the actual optimum that you get by actually minimizing L tilde omega and if by actually minimizing the delta the L omega if you had L omega as your cost just L omega as your cost and suppose you had omega star as the optimum for that okay then you want to see how is omega tilde or W tilde related to related to W star right that is the idea because something will change because of this additional factor which will get involved when you when you use this new cost right.

Now the way the way right this is arrived at is I will write the final expression but the approach is that you expand this guy in terms of a Taylor series right in terms of a Taylor series which then which then means that right so what you are what you are sort of looking at let us say suppose I call this is U okay as a difference right so let us say this is this is W minus you see right write W W star wherever I write W star is this optimum then what you can do is you can actually look at the look at the you know Taylor series expansion of this guy W star plus U okay and then if you try to if you try to write expand this then write what you will get is you know L of C W star plus whatever right U transpose gradient of gradient of L right with respect to with respect to this one right W star. So let us just write this as a gradient okay gradient L and then plus you will have a half right you will have U transpose and then H which is the Hessian of L this we have seen before right I mean so this is the Hessian of L and then you will have U right something like this is what you will

have and what is and then because you know that at omega star right this this guy is 0 right because right that is supposed to be the optimum right we have assumed that is a W star is the optimum for say L omega right when you are actually minimizing L omega therefore the second term drops out and then and then that what one can sort of look at is really examine this H so the whole thing boils down to examining H and you can show that H is actually a positive semi-definite matrix why because of the fact that right I mean see if this quantity is 0 right and if you are already at actually L omega star which is a minimum unless U transpose H U is actually greater than or equal to 0 right otherwise what will mean is you are not really at you know you are not at a local minimum right you see that right see that argument that you know H has to be P S D because if H is not a P S D then it means that L omega star is not really a local minimum right so you see you have to use arguments like that in order to be able to arrive at the fact that H is actually a P S D right under these assumptions and you can just write walk through this a little bit if you walk through this I just leave it to you as an exercise right just just show that okay and then what you have to do is you know use you know you know do an eigenvalue eigenvector decomposition on H because now it is a P S D so if you do an eigenvalue eigen decomposition on H then that can be like you know Q diagonal Q transpose right we do not want to use Hermitian and also just say right Q diagonal you know Q transpose and based upon this right and where where this does this contains the eigenvalues of H right this is a diagonal matrix containing the eigenvalues of H so this eigenvalue eigenvector decomposition you are all familiar with right so if you use this argument then what you can effectively show just you know 3 4 steps down and you can actually show that your W tilde okay can then be shown as Q diagonal plus alpha no beta okay we have used beta so beta I, I is an identity matrix inverse then diagonal Q transpose W star okay this is what you will get that means the new weight W tilde by introducing the additional regularizing term which is norm of W right the L2 norm of W by introducing that effectively what will happen is your W tilde your W star will change to W tilde so you can easily think right away if I had forced my beta to be 0 then of course diagonal inverse diagonal is identity Q Q transpose is identity and therefore you will be you will be back to say W star but given that beta is not 0 right so what will happen is we will have a term like this right and therefore right if you try to if you try to if you try to you know examine entry wise so what this so the scaling is actually happening here okay the actual scaling is happening here I mean so you know not just this so this diagonal also so what is happening is Q transpose is acting so Q transpose is a matrix that is actually rotating omega star and then after that there is a scaling going on and then after that there is a re-rotation through Q so the actual scaling of weights is happening in that is in middle term right which is this diagonal plus beta I inverse diagonal. So we try to examine any one so in that sense right if you want to examine some ith entry right then that is going to look like let us say lambda i by lambda i plus beta okay that is the kind of scaling right right you know which is being applied and these lambda i's are but the eigenvalues of H okay so what this means is and generally

right you know that eigenvalues actually right represent represent kind of variance right this is that if there is a high spread right then you will have then along that along that along that orientation which the eigenvector kind of see kind of see represents the eigenvalue sort of says that how large is that variance you guys are familiar with this right when you do a PCA or something that what do you do I mean if you take a covariance matrix you get the eigenvectors then when you want to get a principal eigenvectors what you do you go and look at the look at the most significant eigenvalues because orientations you can have in various ways but then which ones have the highest spread you start from the sigma that your yeah the singular value or the eigenvalue whichever is the highest then you kind of order them down right because because you are sort of thinking that the you know that you have a distribution that is most spread out in in the in let us say one particular direction then next most spread out is in the other direction and so on right so in the same way since lambda i's are kind of right you know so these lambda i's sort of are signifying these importance and what this means is that if lambda i is actually is actually pretty high then then you see beta and of course beta is actually a positive number so it can be greater than it can be equal to 0 if you do not want the regularizing term but when you are talking about regularization of course and it is a number greater than 0 then what you what you what you see is that right so so you see that the scaling rate is almost equal to 1 for that for that particular weight which means that it is a fairly significant sort of weight and then you do not want to sort of knock it off whereas if you have a situation right when or where so the other situation when you have lambda i's right like much less than much less than you see beta right I mean in which case this will almost go down to 0 because I mean you have you have a beta right which is much much greater than lambda i so this denominator is almost like beta and lambda m i by beta can go close to 0 that means where the spread is really not much in which case that weight is probably not even significant and you want to sort of you know you may not reduce it to 0 but you have sort of right you know you are sort of alleviating its its importance right so you sort of right bring it down so in a sense in a sense this happens automatically right and the only control parameter sort of a hyper parameters is beta so so such things are called hyper parameters ok so you had unknown parameters that you are anyway estimating so those are learnable right in within a network always right keep in mind that there are things that are learnable like the weights the biases and so on and there are things like this hyper parameters and all which you do not call them as trainable or learnable so these have to be figured out so you have to kind of say tinker with it so for a particular problem a certain beta might work well and for another problem the same beta may not work therefore these have to be arrived at by actually doing some you see brute force experiments ok there is no easy way how to tell what will be a good value for let us say beta ok so this beta is called a hyper parameter and these hyper parameters you will encounter often in fact many things are hyper parameters even the number of layers that you use in actually a neural network that is a hyper parameter nobody knows how many layers you need to use and then what should be the what should be the

right dimension all these are hyper parameters so beta is a hyper parameter and and what you can also show is that what shows that I mean effectively if I do the summation over over all the weights right I mean let us say 1 to n lambda i lambda i plus you see beta then this number right is much much much less than actual n so so so so in effect right in effect it is like saying that it is a no in effect it is like saying that you know instead of instead of dealing with with a with a value like n right you can now deal with something much less than that but again right what get a beta will work best depends upon the problem so you will have to look at your final you know accuracy and then sort of right because you cannot arbitrarily make this beta very high because if you make beta very high then the actual original cost rate will lose its importance right so you cannot just arbitrarily make l 2 norm to be of the weight to be the most important thing so you have to strike a right balance and that right balance comes out of right this one ok experimentation ok. So so this is about this is about regularization by imposing a regularizing term on the weights you can also there is there is an interesting thing right which you can do which is not very obvious another way to actually regularize is through is through noise injection is through noise is injection ok this is also again one more way to actually you know to bring in a regularization which which again means that you know something like this kind of you know well you can actually show that you know this is I mean like this ok let us just show this so for example right so what this means is that it is ok so so you have so suppose let us say right I have an output suppose y right which is I mean right at some layer right I am writing so this is let us say summation let us say right I mean the final output layer or something and then I have like summation w i ok I mean at some layer x i plus now what I would normally I have had is w i x i but now right I am going to add noise to the my input so this noise injection is to the input ok if I actually inject noise so this is e i ok by the way and e i is let us say Gaussian with some with 0 mean and right sigma variance ok let us call this as y noisy so this y original right would have been just the summation w i x i right is what I would have done but the fact that I am injecting noise into x i means that effectively what I what I will then see is not the original y but then over then y noisy and this y noisy right you can just expand it so that will become this all over i ok and and i will be like you know the right number of number of number of terms that you have and this will be like summation w i x i plus w i e i and w i x i we know is y therefore y noisy is y plus let us say summation over i w i e i and and my have what I have a target right let us say and of course this noise injection this is only for an MSE loss ok this regularization is for MSE loss ok so ok acts as a regularizer I mean injection of noise into the input acts as a regularizer provided the loss is MSE ok acts as a regularizer for MSE loss noise injection ok I mean I think I should write this here noise acts as a regularizer for MSE loss ok. Now if my if my final sort of a target value is t and if my target value is t right I mean that is that is what I that is what I actually want then what you can what you can kind of look at is the expectation of y noisy minus t square right I mean if you if you write so instead of y minus t the whole square and now what you have is really y noisy minus minus t the

whole square it is what you have.

So then we can just substitute for y noisy right into this that will be like y plus summation w i e i minus t the whole square and then this will be like expectation I can just rearrange this put this as y minus t plus summation w i e i square right and this we can write as and if ok let us just assume that y and and t are both deterministic I mean if not then of course, you will have to use an expectation there also ok nothing nothing great will change, but let us say just to make matter simple if you assume that y and the and t are both see right deterministic then this will become simply y minus t the whole square if it is not then you will replace it with expectation y minus t the whole square and then this if you expand right if you. So, because you are assuming that you know e i is 0 mean therefore, that product term goes away and because you are also you are also assuming them to be them to be uncorrelated right therefore, what will happen is you will have the summation w i square ok and then expectation e i square or in other words should be this we can simply write this as sigma square right which is the which is variance right this is over i. So, now, if you kind of look at this term right this is like this is like you know norm on norm on the weight right and of course, you know it is being scaled by the by this noise variance if you if you make sigma to be 1 then it will be you know summation w i square which is exactly what you had earlier and you had this is it right l 2 norm on c w right. So, what this is effectively saying is that another way to regularize that means right you want to sort of bring in bring in an action or begin a regularization term on the weight that is by actually adding noise to the input and that you can and for an MSC loss if you can effectively show that is equivalent to doing a regularize you know l 2 regularization on the weight ok. So, all these are tricks that let us say right people play it is not like right everything is done, but depending on problem to problem right people try various things, but these are all things that one can try.