Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-19

Then there is another thing which is also part of regularization what is called actually dropout. So, this dropout right what this means is this right. So suppose I have to start with right let me just draw a simple thing. So I have let us say 3 of these neurons and then I have let us say 2 neurons and let me just take a very simple case and let us say I have a 3 dimensional input. Then let me just look at this. So this goes so what you will typically have are these right.

So we will have another weight going to this and then let us say we have a third guy going to this. I am not numbering the weights and all, but we understand and then like this goes here, this goes here, this goes here and similarly this goes here, this goes here, this goes here and then let us say let us say outcomes right something. Now you of course have a large network right I mean it is not going to be as simple as this, but then a dropout right actually means that means like you know deactivating one of these guys. So for example if I drop this neuron out okay dropout means okay drop it is okay I will tell you what we actually do it is not like you drop this one neuron, but suppose you drop it right then what it effectively means is that it is so it is like hanging there okay it does not do anything okay just to illustrate right what this idea of dropout is.

So what this will effectively mean is that once you drop it right then this network will then kind of look like this. So you still have this okay, but then what is it? So the other one so okay so we just have the purple guy going and then you have these two and what was it that I used okay this color right. So this goes we had this, this and now what will happen is right so these guys will go here and similarly these two will go here and out comes this right. So dropout effectively means this so this guy is left hanging there okay it does not receive anything it does not send out anything it just sits there okay and if you are doing backward forward pass backward pass and all that will all that will all exclude this kind of say neuron. So all the paths will go from elsewhere, but through this neuron you would not find any path going through.

So all forward and backward passes will leave this neuron out okay all forward and backward passes will exclude this or leave this neuron out okay will exclude this neuron means okay this guy okay maybe I should just indicate it by some color. So this guy

exclude this neuron out, exclude this neuron that is the drop neuron out. By the way there is just one small little table that I forgot to actually write down because that table it I think you should basically keep in mind okay and every time we do now it is best to keep that in mind. So I will just draw that you know quickly okay and then we will come back to this I wanted to do that I somehow forgot okay one thing is this okay so one table is this okay so I think you know right these are two tables that you should just sort of you know keep in mind. So this has something like this so you have algorithm and then you have number of steps in one epoch, steps in one epoch okay number of steps in one epoch will actually mean that how many updates you make right on the weights correct number of steps.

So the steps is number of updates or you see iterations all those things are equivalent right iterations, updates, number of updates or iterations right that is what it means. So when I say steps what we really mean is that okay and you can have you can have three situations right that we have seen already just that right just that we should just keep them. So vanilla GD or what is called a batch gradient descent which we saw earlier vanilla GD okay so this is all about GD okay this is all gradient descent and assuming that if n is the number of training samples n is the total number of training samples and given that n is the total number of training samples and or data points right and b is the batch size and b is the batch size, b is the batch size okay. Then let us say first is this one next is stochastic GD which we saw earlier right yesterday the third is mini batch GD right now can you tell me number of steps in one epoch vanilla 1 stochastic GD n and mini batch GD n by b right n by b. So always right so from now on because we are going to be talking about batch norm and all therefore it always remember that when we say a mini batch it means that it is all happening within the epoch okay so whenever you make some updates and all and then we say that we are doing it every for every batch of samples mini batch of samples that means that right during the iteration itself it is all happening okay so that is why this is one table and the other one is regarding this was this you know because we did softmax right so today so I thought I will also write about the other day that has nothing to do with gradient descent but that is that is a different table but just for your quick reference right we will also write that down.

So that is like outputs okay what is by outputs what we mean is whether your output is a probability or whether it is real valued number for example it could be any image right in which case it will be just a number and then so let us say this again it will ask you guys then okay so this is like real values so the output can be either real valued or it could be a probability right which means the probability means you are doing a classification problem if you are doing real values it means that you are doing a regression problem okay and then we have let us say two more rows down there and then here right let us just have two guys here so one is I want to know what will be the output activation. So what will you fill in

so I have I so on my output I need I want a real value let us say so what should be my output activation be like what kind of a unit will you be will you think of should it be a sigmoid or should it be a linear unit is okay right I mean the linear means it could be a ReLU or whatever right so by I mean ReLU is strictly speaking not linear but by linear it what we mean is you know you can also have you can also have completely linear okay it does not mean that ReLU always has to be there okay if you are just saying real and it can be negative and all then it should be just fully linear if you are saying real and it should be greater than or equal to 0 then it will be ReLU. So I will write in general linear because I have not said real values but then greater than or equal to 0 I have not said so it means strictly speaking just a linear unit will do that means summation Wi Xi plus whatever not Xi Ai something right which is the previous output plus whatever the Bi for that and no more activation on that just that that comes out that is okay right you will be surprised that is okay right so just a linear unit okay does not do anything but so accordingly rate all this will change so this output activation if I say greater than or equal to 0 but real then it will become a ReLU right things like that then if it was a probability probability not sigmoid because sigmoid is okay for binomial but not for a multinomial sort of distribution then loss function so if you if you have an output that is real valued then what kind of a loss function would you would you want to do you want to say propose a mean square error right so this will be like a squared error okay so this will be like squared error and if it is a probability it will be a cross entropy okay so these so these two tables right keep in your mind and most and right these two are really not even not even the relevant because most of the time right we will have only this okay we will be doing only this and these this whole thing right is actually important because depending upon the problem the unit and everything can change you can even the loss function and you know and what kind of activation you want and so on okay now coming back to this to this dropout okay now let me just write down okay a little bit about dropout so what it actually means okay so the idea behind dropout is that right you do not want only a few neurons to take on all the load because when you when you kind of when you free when you when you make it a free for all then what happens is certain guys will will will just like us right they will also be quiet right and they will say let the other neurons do the work right and what will happen is that is exactly what is what is actually overfitting right because what will happen is some neuron will just take on all the load and then they will say okay right we will all together you know you know you know fit this cost function but what will happen is the rest of the neurons which could have ideally been you know probably they are able to do other tasks right which they are very very kind of very probably you know they need not be very strong but they are also capable of doing certain things but then they won't even do that so in a dropout what happens is what you do is you know at any one point let's say within the within a mini batch that's why again I come back to this mini batch term so what this means is within a mini batch when you have a network what you do is randomly right with some sort of a probability and this probability actually varies okay at the input it is actually

a different probability but let's not worry too much about it let's say let's just focus on the weights so what happens is so the process with a certain sort of probability that's typically it's 0.5 so with a 0.5 probability right you will actually you know drop a neuron or whatever it when you either pick or drop right either way it's 0.

5 so what will happen is at any one point of time only half the neurons are roughly active okay so so what you have is I mean so so then it means that during that mini batch when you are when you are when you are doing when you are when you are kind of computing the losses and all all the forward passes and all right they are all happening only only over over over those set of neurons that are actually active and the rest of the guys are like the one that I showed they are just floating around not doing anything then what you do is then after that particular that particular right mini batch is over and when you go to the next mini batch you gain you again bring back all of them right you bring back all of them then again again you see sample randomly so what you do so if you if you kind of it continue doing this idea is that all neurons will have a will have a say in the matter right so finally there will be because then see this generalization right when we say the generalization we are hoping that there could be other neurons right which will actually end up doing very well I mean when you give a task which is a little I mean which is like you know which is not exactly what you see in the training example but then something you know which could be a little away from that but then there are other neurons that can probably handle it so so this so the idea when dropout is not really reduce the number of neurons I mean you would know the number of weights okay that doesn't happen because eventually you will use the entire this one network I mean it's not like you will you will you will know right no drop you see some of these or something right so you don't really drop anything but the idea is that this overfitting right but again like I said it regularization is all about being able to being able to you know do a generalization being able to generalize well outside a training set and for that you need so right dropout is one way to sort of enforce that kind of regularization it's not true that you will get fewer weights but then you know the idea is that the generalization capability actually goes up okay so sorry this is what you had in summary it is so I just write it write this down okay now so so if only some of the neurons take on the entire load if only some of the neurons take on the entire load of you know of the task on hand and it can lead to overfitting okay that is the reason why you sort of left let only a few of them handle the task so really the I mean rest of the neurons really do not learn anything useful the rest of the neurons okay do not learn anything anything useful because because the because the others have anyway kind of right done the job and therefore these guys really do not learn anything do not learn anything useful anything useful means they don't really learn any you know useful features that can be that that probably will be useful when you're when you're doing you know doing during inference time then right so don't worry so right so typically okay yeah typically well almost always nodes are dropped nodes means these neurons okay nodes

are dropped only once for for a for a mini batch of samples only once for a mini batch of samples that means when you when you accumulate the cost only only those set of neurons will be there for their entire mini batch okay you don't keep changing within a mini batch once for a for a for a mini batch of samples and then before you start the next mini mini batch you sort of right you could have bring them all back on board and and nodes are dropped with a probability p nodes are dropped or selected whatever nodes are dropped with a probability p and typically p is equal to 0.5 for the hidden layers this is a this is a I mean this is roughly okay sometimes you may see that some people may use a little away but may not be exactly 0.5 and and forward and backward pass are done only through the active neurons backward pass are done only through the active neurons then finally right the okay which means that the dead neurons do not really only participate and at test time okay that means right during during training is all okay but then at the time of inference right when you are using this network all the neurons are active all the neurons are active but their weights are actually multiplied by which means their activation value right but their weights are multiplied by multiplied by p multiplied by the same p okay that you had used for for dropping the idea being that since every neuron is likely to have been picked with a probability p since every neuron is likely to have been picked with to have been okay dropped or sorry okay now okay we will just stick to dropping okay because if p is some other thing then it is not 0.5 then this can change so we will just keep it as dropped okay it is likely to have been dropped with probability p its activation should be should be reduced its activation should be reduced by p should be reduced by p okay so in a sense right this word does actually drop out and many people actually use this pretty right effectively okay so this is something that you will that you will find that in many of the papers right people ever people actually use this.