Modern Computer Vision

Prof. A.N. Rajagopalan

Department of Electrical Engineering

IIT Madras

Lecture-20

Then the next thing right that I wanted to talk about was actually. So, this is all about the regularization and the other thing right is actually preprocessing. This is the last thing right within this is something called preprocessing. So, preprocessing you know right. So, the preprocessing could include both the weights as well as the as well as the you know the input value. So, this you could preprocess the input or you could preprocess the weights or generally preprocess both.

So, what this actually means is that right I mean suppose I have my examples right. Do you see do I simply push them in just the way they are and similarly weights right how do I actually you know initialize them or should I take care of what is happening to the weights and the input. The reason being that because you have got like millions of guys getting involved and even if your weights are typically small, but then because so many right the summation is happening so many terms are getting involved right you do not want things to blow up inside. I mean so for example, I mean you can think about a sigmoid right if you think about a sigmoid you know that if you if you go too high or if you go too low then then you have a gradient value which will which will it will fall off very fast which means that your weight updates right will all get sort of say affected.

So, this is so the preprocessing is done to sort of you know maintain some kind of a balance right I mean throughout the network so that things do not explode or you know things do not vanish right either way neither should happen. So, as far as the input is concerned right most of the time what is done is you kind of you kind of convert your data to 0 mean unit variance 0 mean unit variance right. This is this is standard which I think you know you would have seen seen you know elsewhere also. So, suppose you have a random variable X right and then you know if it has mean mu and you know variance sigma then it is simply to X minus mu by sigma right. This this in a sense will make sure that an expectation of Y is 0 and then expectation of the the variance of Y right which is let us say you know sigma sigma of Y if I call that as a as a variance or where of Y right where of Y is simply equal to 1 right because expectation X minus mu square that is sigma square sigma square by sigma square.

And so this is the simplest thing right which you can do, but what typically happens is at the input right you you might be able to do this and you typically do this ok. So, as to sort of keep them you know within check, but what can happen is as you traverse from see layer to layer right there can be what is called a covariate shift. What this means is that the that the that this is statistics which in this case we are only interested in up to the second order statistics, but that statistics can actually change right because of because of the fact that you have got your weights and all involved. And what can happen is you see go from layer to layer to layer these statistics that is appearing at the input of every layer can keep changing. When ideally right you would want to say that at the input I am actually controlling things, but then you know what control do I have after it has passed through a layer.

Do I still have a control about what is happening right at the input to the next layer then that layer does something something to this and then there is an activation then at the output something comes in which becomes the input of the next layer right. So, what can happen is you can have what is called you know a covariate shift this is called an internal covariate shift. It is like saying that you know a distribution suppose you suppose you have a network which you have trained on a distribution of examples. If that distribution changes tomorrow then this network would not work well because it is only it is only seen a distribution samples coming from a distribution right. If you just go far away from that kind of distribution draw samples from elsewhere and ask this network to see perform it will struggle.

Let us say an example could be like you know good light and dark light you have trained it all on you know right good lit well lit images and then you suddenly start showing dark images it would not understand. So, that is a distribution that is like a domain change right that is what we call that that is that we call this you know domain change problem. Here there is something similar, but then it is happening internally. So, that is why it is called an internal covariate shift internal covariate shift internal covariate shift. And in order to address this issue right what is done is there is something called batch batch normalization.

There are some people who actually who actually it is not that it is always a must, but then many times it is used. And this comes prior to the activation that means, before you apply the activation in that layer before you apply that this batch normalization comes. And the way this batch normalization right works is actually very simple I mean exactly the way that that you had and it is called batch because it is actually a mini batch ok. It is for it is for a mini batch I mean this you can in fact, right you can even do it only if you had a great mini batch situation. So, that is where this batch comes ok.

So, this batch is really a mini batch and what this means is that if you had m samples in actually a mini batch. So, so right no no no right imagine that imagine that I give you I give you sample 1, I give you sample 2, I give you sample 3 and I have m number of samples right. And let us say it has a it has a dimension I do not know what dimension I have taken here ok. Let us say right if it has if it has a you know a dimension d then what you are sort of saying is that see when you when you want to when you want to do this 0 mean unit variance right you have to go like this right you have to go like that right you have to go along along each element you have to find out because it is that same element appearing again in another another example again appearing in another you do not see the only thing that you do not consider is kind of you you consider it everything to be a sort of you know a de-correlated. So, you do not take the correlations for example, you do not worry about what correlation this pixel where this entry might have with another and so on.

So, you do not you do not take that into account. So, you just assume everything to be de-correlated which is an approximation that they are making, but what is done is at the kth entry right if you are looking at this to be the kth element then what you do is you find out mu B at k which is the mean ok the batch norm to be 1 by m that means, you are summing across the row right. So, I wanted to wanted to read you know if you want to think about to visualize it like that 1 by m summation x i k i going from 1 to m right this will be the mean and the variance will be sigma square B and let me say k it is not power k this is the kth entry will be 1 by m summation now x i k minus mu B k which is of course, a constant does not depend on i mu B k i equal to 1 to m and oh sorry square right and then you have and then and then you do you do a normalization which is what we did here right same way you do x i k minus mu B k by sigma B k right exactly as we as we do that except that now we are now we are doing it doing it for a say every element the idea being that if you keep on doing it at every input layer then you are sort of making sure that right things are sort of you know within bounds and right this will ensure that you are right that your mean is mean is this k again x i hat k. So, this x i hat k is 0 it will ensure that variance of x i hat x i hat k is 1 and also what is done is you know as a sort of a learnable parameter right people do not just leave it at this because the problem is if the if the network actually did not want this let us say you are forcing it now right you are saying that this is what I want you to I will force, but to just give an allowance on top of this what is typically done is right we actually let the network learn 2 parameters per element you know per. So, if you have if you have you know d dimension that means 2 d additional parameters which come in the form like you will have like gamma k x i hat k plus beta k that is gamma k and beta k are again are again learnable what this means is that see for example, right.

So, what you are saying is if for example, right that this guy this network was actually with x i hat k itself then it will learn gamma k to be 1 and then the beta k to be to be 0 it will automatically find it out to be that because then it will figure out that x i hat k is probably the most ideal thing, but suppose it figures that x i k the original x i k was actually good and then we tampered with it then what gamma k and you see beta k will actually give you back will you back x k the variance of what should be gamma k gamma k should be sigma b k and I have sigma b k right and then beta k should be mu v right. So, which means that you are you are letting the network figure it out because you are forcing something believing that there could be an internal covariate shift and therefore, I want to right bring you know things into bound, but these extra parameters are added as learnable these are again learned and there is a there is a there is a back prop for batch norm which we would not do here in this course we do not do that, but there is actually a formal way to do back propagation when you have batch norm because these are become these are also now these have to be learned now gamma and beta, but the idea behind using gamma and beta is that if the network believes that x i hat is the best it will take that if it believes that x i is the best it will take that if it believes something else is best it will then find gamma and beta k according ok. So, that is the that is the that is the thing about batch norm this is one thing remaining that is about initialisation of weights that would not that is a very small thing. So, I will talk about it in our next class ok. So, that then we can switch over to CNNs ok in the next class we will be doing CNNs.