Deep Learning
Prof. Mitesh M. Khapra
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 5.8
Lecture – 05
Line Search

So, we were looking at these different variants of gradient descent. We saw that gradient descent has this problem that it finds it difficult to navigate the gentle slopes. So, we came up with tricks on momentum based gradient descent and also, Nesterov accelerated gradient descent.

The trick in momentum was that if lot of your history is telling you to move in a direction, then just continue to gain momentum in that direction. So, instead of just updating based on the current gradient, you also update based on the history, right and there we saw that this is always going to be a problem that you will end up taking u-turns and we had this analogy of how you look for directions and you just overshoot your destination and have to come back and take a u-turn and come back and so on.

So, to prevent that we realize that the update done by momentum base gradient descent is two step update. You actually the first step is based on the history and then, another step based on the gradient at the current time step, right. So, then instead of doing these two steps at one go, why not just update based on the history, see what the gradient that tells you and then, we saw this nice figure. I hope it was nice and where you saw that if you look ahead point, then you will be immediately corrected with respect to your errors, right. So, that was about nag and momentum.

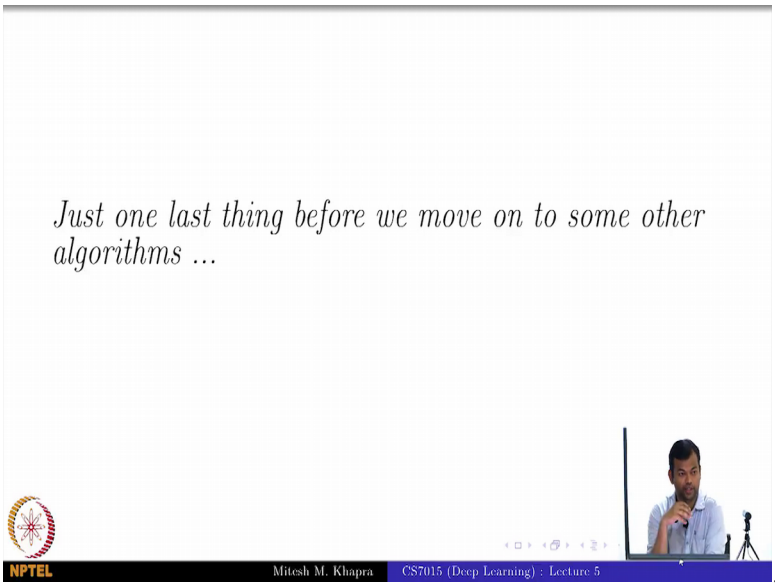
Then, we saw the stochastic versions of these algorithms, where we realize that if we do the batch version, then you go over a million points and then, make only one update which could be very slow in cases where you have large data. So, we then decided to the stochastic version where we just update for every point that again had these oscillations because we were taking greedy decisions, we were just relying on one point to tell us which was the right direction to go on and you saw that these esteem has become better as you increase the value of this k, right.

So, k equal to 1 is the most stochastic version and then, k equal to 2 you get the mini

batch version and then, you could just have different values of k, so that you have more reliable estimates of the gradients, right and in the limit if you have the entire data, then you are just doing the full batch gradient descent, right. This is the one gradient descent. Anything else did we cover, then we had some tips on the learning rate and the momentum, these are again juristic. I gave you some ideas and you could try these in your back propagation assignment and see which one works better for you. You could see you have any peculiar observations while implement the back propagation assignment, ok.

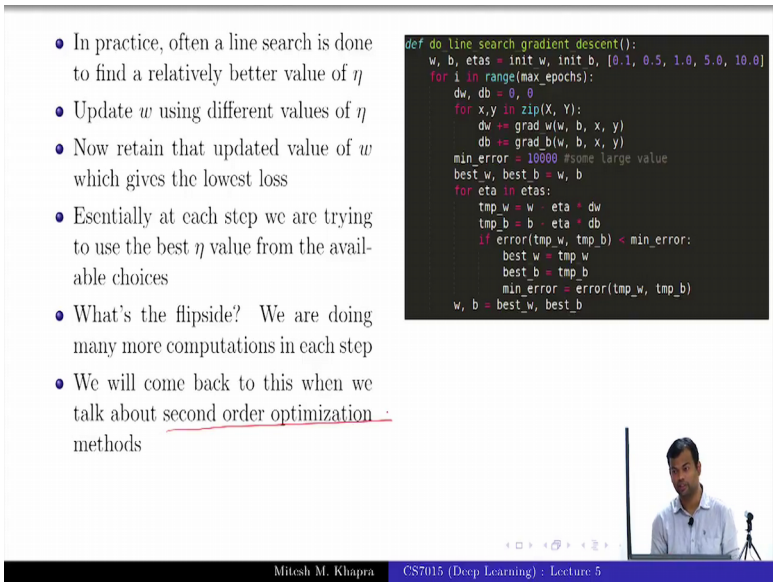
So, now there are a few more things left in this lecture. So, I will start with the line search first, ok. So, this is one more thing before you move on to some more interesting algorithms which are the current state of the art and lot of deep learning solutions, right.

(Refer Slide Time: 02:44)



So, most people that you read would look at, would have algorithms that we will see after 10 minutes, ok.

(Refer Slide Time: 02:54)



So, now this is where just to contest contextualize things, right. So, we are still trying to see, what is the light right learning rate. To use a line search is one such method where instead of just doing one learning. So, you can look at the code and just focus on this part and tell me actually what are we trying to do, how many of you get what the algorithm is trying to do, right. So far what we were doing is, we were just having a single learning rate, and we saw that this learning rate can make a lot of difference, right because if you are on the gentle part, you want larger learning rate and if you want steep part, you want smaller learning rate.

So, just fixing the learning rate to one value does not really help because then you will make, you will suffer on one of the two cases are either on the gentle case or on the steep case. Now, what line search does is, instead of just using one learning rate at every step, now whether it is vanilla gradient descent which is the batch 1 or mini batch or stochastic, right just use a bunch of learning rate. So, I have used five different learning rates here, and I have computed the gradients. That part remains the same, right. The computation of gradients does not change.

Now, you have the value. Now, you want to be conservative, you want to multiply the gradients with this eta right, but you know that you do not always want to be conservative. In fact, in some cases when you are on the gentle slope, you do not want to be conservative at all. You want actually below of the gradients, right. So, now try these

different learning rates and update w and b, ok. So, if you have five learning rates, you will get five different updated values for w, b.

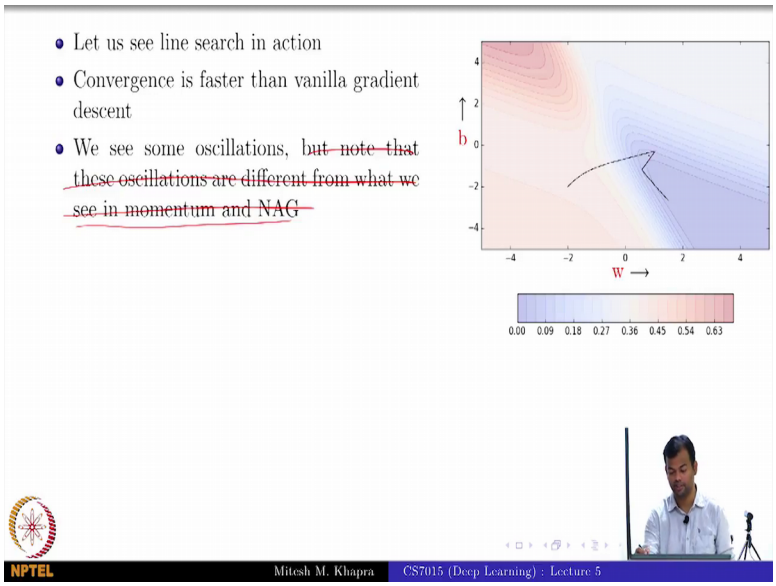
Now, plug in all these w, b values into your loss function, right and see whichever is the minimum, retain that w, b value and repeat the process. That means again you will compute the gradients with respect to this new value of w, b and the new loss function. Again try out these five different learning rates and continue. Everyone gets that, ok.

So, now are we using a fixed learning rate at every step? No and now do you see that if we are at a gentle slope, it would pick probably this as the learning rate and if we are on steep slope which should probably pick one of these as the learning rate and even lesser than that. If you have the disruption, it does not make sense. You see the advantage of this now. You are in some way heuristically trying to adapt to the slope of the error surface, right by just giving a different learning rates. So, try all of these and whichever works best, pick it up, ok. So, that is about it. We are trying different values.

Now, what is the flip side of this? Now, if you have k different learning rates that you are trying, then at every step you have now increased your computation k types, right. So, earlier you just add one learning rate u just going by that, but now I have k. So, now this is again a trade off which is you have to see. Now, I will give an example where this trade off clearly works, right.

So, now if you are at the gentle slope, now making k more computations and moving out of that slope is definitely worthwhile as compared to just sticking to that slope where even after hundred more computations, you will not really move out of that slope. So, remember that gradient descent algorithm that we have seen where you just stick to the gentle slope after hundred iterations also right, but instead if I tried five different learning rates and there is a high chance that, I could have moved out of the gentle slope. Does that make sense? You see the advantage of this ok, fine.

(Refer Slide Time: 06:31)



So, this is something that I have to talk about when I back to second order optimization which my time do not allow me to teach. So, I will see when to teach that. So, let us see line search in action. So, this is again gradient descent, this black curve, which is visible there ok. This is the one I am talking about which is run for few iterations and is just stuck on the steep curve. You know this story now and it is just get stuck there, ok.

Now, let us see what happens if I run. So, now I will start running the line search based gradients descent, ok. So, what do you expect now? So, it will just move very fast, right. So, on the first step itself, it is crossed wherever gradient descent was stuck after 50 iteration or so, I will keep moving fast, ok

Now, here is an interesting question. Would you see oscillations here? So, when you see oscillations? It is when your loss is actually increased from whatever it was currently. Will that happen in line search? The answer is always no.

It could happen, when could this happen? So, it depends on the learning rates that you have chosen, right. So, if you have chosen the learning rates, so suppose at one point to really be effective, you needed the learning rate to be 0.01, and now if 0.01 learning rate was not in your set, right that means, everything that is there in your set is faster than 0.01, so that it will again have the same problem as momentum because you will move faster than what you should actually move. So, it depends on this careful choice of the learning rate set. How many of you get this? Good, right.

So, that is all I have to say. So, there is a slight convergence would be faster than vanilla gradient descent. That is obvious and we see some oscillations, and the statement is actually wrong. We need to remove that, ok. We see some oscillations and these could be the similar wants to the once with that we see in momentum because we overshoot because we have not chosen the right set of learning rates, right.

One of the learning rates which was actually needed at a particular point right say at this point. Suppose I needed to move very slowly and that very slowly say 0.001 and that was not in my site, then any other learning rate is always going to be much faster then, right. So, you could see oscillation.