

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

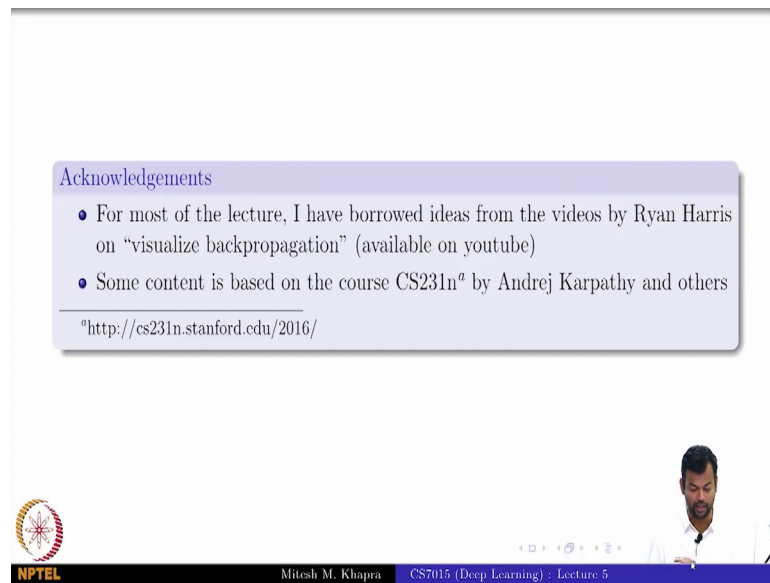
Lecture - 05
Gradient Descent (GD), Momentum Based GD, Nesterov Accelerated GD,
Stochastic GD, AdaGrad, RMSProp, Adam

Welcome to lecture 5 of the course on Deep Learning. So, today we look at some variants of gradient descent. So, we will just quickly do a recap of gradient descent and then look at some variants of it, or some ways of improving it, which is momentum based gradient descent, Nesterov of accelerated gradient descent, stochastic gradient descent, AdaGrad RMSProp and Adam.

So, just to set the context. So, we started with this gradient descent algorithm for a single sigmoid neuron, and then we saw how to extend to network of neurons with back propagation. So, we realized that all we need is the gradients or the partial derivatives, with respect to all the weights and biases. Once we compute that we can just use the gradient descent update rule.

Now, today what we are going to see is, are there better update rules which lead to faster conversion or better performance in various ways. So, that is why we are going to look at all these different variants or methods of improving on gradient descent ok. So, that is the context.

(Refer Slide Time: 01:18)



Acknowledgements

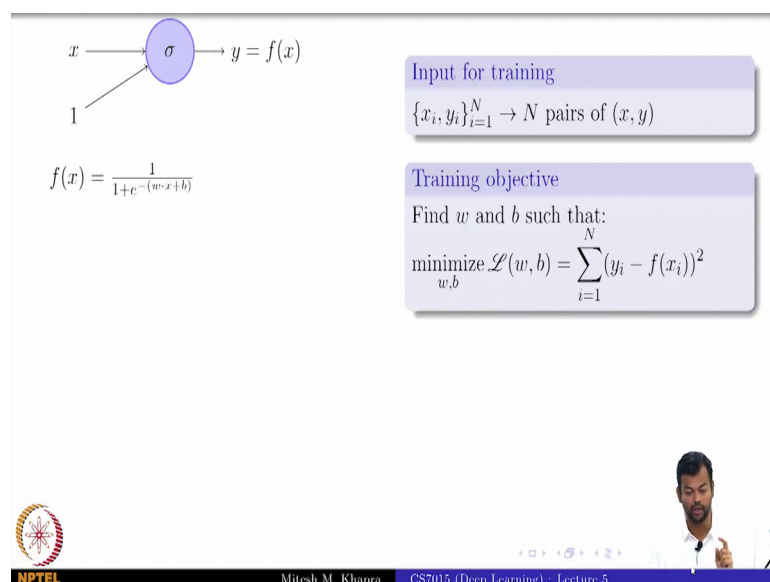
- For most of the lecture, I have borrowed ideas from the videos by Ryan Harris on “visualize backpropagation” (available on youtube)
- Some content is based on the course CS231n^a by Andrej Karpathy and others

^a<http://cs231n.stanford.edu/2016/>

NPTEL Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5

I will just quickly rush through. So, for most of the lecture, I have borrowed ideas from the videos by Ryan Harris on visualize back propagation and some content is based on this course by Andrej Karpathy and others, when I talk about some tips for learning rate and so on. So, you can just look at those also. So, we will just quickly rush through the first two modules which we have already done.

(Refer Slide Time: 01:46)



x → σ → $y = f(x)$

1

$$f(x) = \frac{1}{1+e^{-(w \cdot x + b)}}$$

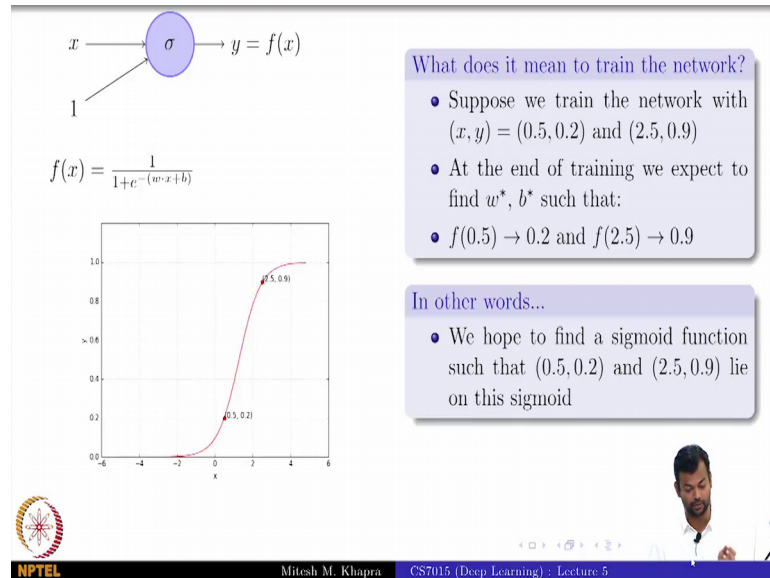
Input for training
 $\{x_i, y_i\}_{i=1}^N \rightarrow N$ pairs of (x, y)

Training objective
Find w and b such that:
minimize $\mathcal{L}(w, b) = \sum_{i=1}^N (y_i - f(x_i))^2$
 w, b

NPTEL Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5

Which was, we were interested in learning the weights and biases for this very toy network, with just 1 input and 1 output, and we started by doing something known as guesswork where we were just trying to adjust these weights and biases by hand.

(Refer Slide Time: 01:52)



The slide shows a diagram of a single neuron with input x and bias 1 entering a sigmoid function σ to produce output $y = f(x)$. The sigmoid function is defined as $f(x) = \frac{1}{1 + e^{-(w \cdot x + b)}}$. A graph plots the sigmoid function with two target points: $(0.5, 0.2)$ and $(2.5, 0.9)$. The slide includes a list of training goals and a note about finding a sigmoid function that passes through these points.

What does it mean to train the network?

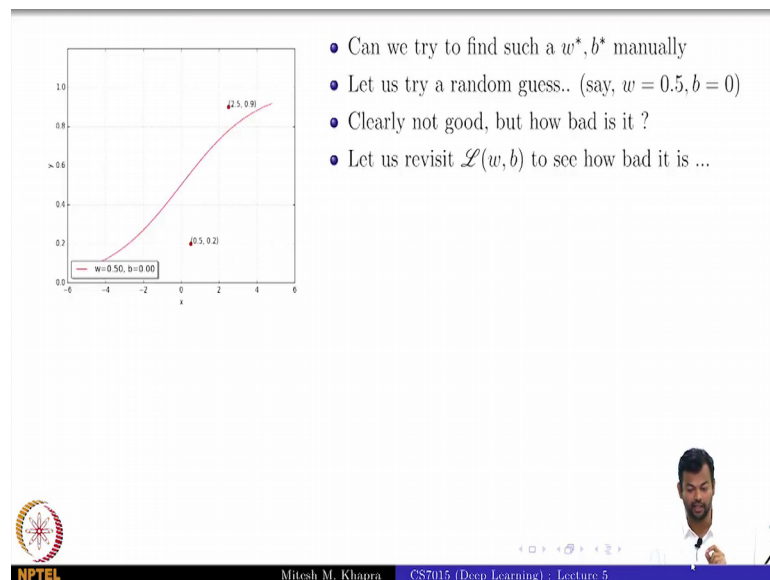
- Suppose we train the network with $(x, y) = (0.5, 0.2)$ and $(2.5, 0.9)$
- At the end of training we expect to find w^*, b^* such that:
- $f(0.5) \rightarrow 0.2$ and $f(2.5) \rightarrow 0.9$

In other words...

- We hope to find a sigmoid function such that $(0.5, 0.2)$ and $(2.5, 0.9)$ lie on this sigmoid

NPTEL Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5

(Refer Slide Time: 01:56)

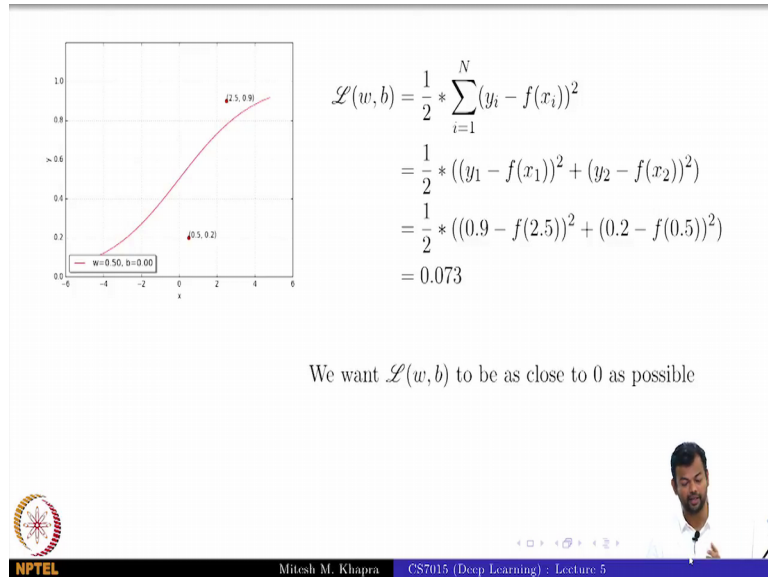


The slide shows the same sigmoid function graph as the previous slide, but with a legend indicating $w=0.50, b=0.00$. A list of questions is posed regarding the manual search for weights and biases.

- Can we try to find such a w^*, b^* manually
- Let us try a random guess.. (say, $w = 0.5, b = 0$)
- Clearly not good, but how bad is it ?
- Let us revisit $\mathcal{L}(w, b)$ to see how bad it is ...

NPTEL Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5

(Refer Slide Time: 01:58)



$$\mathcal{L}(w, b) = \frac{1}{2} * \sum_{i=1}^N (y_i - f(x_i))^2$$

$$= \frac{1}{2} * ((y_1 - f(x_1))^2 + (y_2 - f(x_2))^2)$$

$$= \frac{1}{2} * ((0.9 - f(2.5))^2 + (0.2 - f(0.5))^2)$$

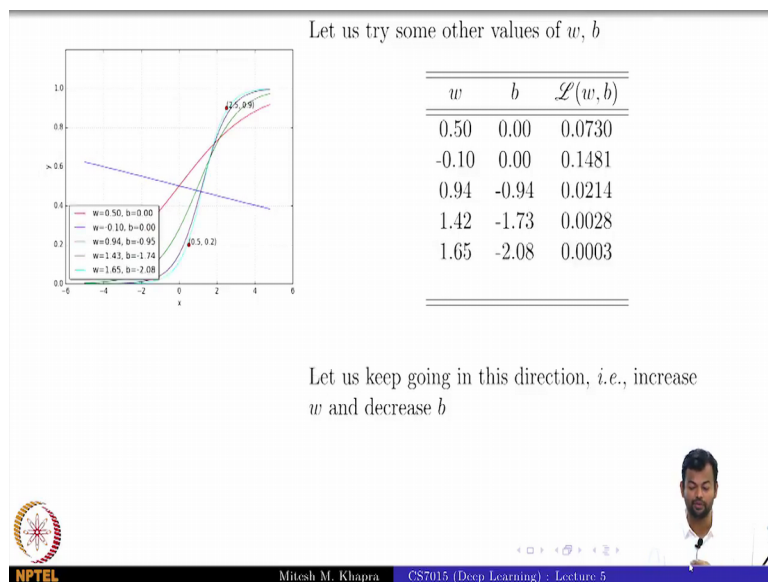
$$= 0.073$$

We want $\mathcal{L}(w, b)$ to be as close to 0 as possible

NPTEL Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5

And we realized that its clearly not good and, but we still try to do a very smart guess work, where we were driven by this loss function, which was telling us whether this guess, the current guess is better than the previous guess or not. And we just kept following our guess work and try to reach to some solution, and for this toy network it was very easy to do that.

(Refer Slide Time: 02:06)



Let us try some other values of w, b

w	b	$\mathcal{L}(w, b)$
0.50	0.00	0.0730
-0.10	0.00	0.1481
0.94	-0.94	0.0214
1.42	-1.73	0.0028
1.65	-2.08	0.0003

Let us keep going in this direction, *i.e.*, increase w and decrease b

NPTEL Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5

(Refer Slide Time: 02:17)

Random search on error surface

- Since we have only 2 points and 2 parameters (w, b) we can easily plot $\mathcal{L}(w, b)$ for different values of (w, b) and pick the one where $\mathcal{L}(w, b)$ is minimum
- But of course this becomes intractable once you have many more data points and many more parameters !!
- Further, even here we have plotted the error surface only for a small range of (w, b) [from $(-6, 6)$ and not from $(-\infty, \infty)$]

NPTEL Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5

And what we were actually doing is, there is this error surface which exists, which can be plotted for all possible values of w comma b . And what we were trying to do with this guesswork is, trying to find path over the cellar surface, so that we enter into the better regions. So, red is bad, blue is good; the darker the shade of blue the better. And this of course, becomes intractable when you have many parameters and so on.

(Refer Slide Time: 02:38)

Let us look at the geometric interpretation of our "guess work" algorithm in terms of this error surface

NPTEL Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5

So, we wanted to have a better way of navigating the error surface. So, this is exactly what we were doing with the guesswork algorithm.

(Refer Slide Time: 02:48)

vector of parameters, say, randomly initialized $\theta = [w, b]$

change in the values of w, b $\Delta\theta = [\Delta w, \Delta b]$

We moved in the direction of $\Delta\theta$

Let us be a bit conservative: move only by a small amount η

$\theta_{new} = \theta + \eta \cdot \Delta\theta$

Question: What is the right $\Delta\theta$ to use?

The answer comes from Taylor series

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5 14/87

So, then this better way actually we realized that we could arrive at it from a very principled solution from, starting from Taylor series.

(Refer Slide Time: 02:55)

For ease of notation, let $\Delta\theta = u$, then from Taylor series, we have,

$$\begin{aligned} \mathcal{L}(\theta + \eta u) &= \mathcal{L}(\theta) + \eta * u^T \nabla \mathcal{L}(\theta) + \frac{\eta^2}{2!} * u^T \nabla^2 \mathcal{L}(\theta) u + \frac{\eta^3}{3!} * \dots + \frac{\eta^4}{4!} * \dots \\ &= \mathcal{L}(\theta) + \eta * u^T \nabla \mathcal{L}(\theta) \quad [\eta \text{ is typically small, so } \eta^2, \eta^3, \dots \rightarrow 0] \end{aligned}$$

Note that the move (ηu) would be favorable only if,

$$\mathcal{L}(\theta + \eta u) - \mathcal{L}(\theta) < 0 \quad [\text{i.e., if the new loss is less than the previous loss}]$$

This implies,

$$u^T \nabla \mathcal{L}(\theta) < 0$$

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5 15/87

And we went to this derivative, where we finally came up with this rule that move in the direction opposite to the gradient.

(Refer Slide Time: 02:58)

Okay, so we have,

$$u^T \nabla \mathcal{L}(\theta) < 0$$

But, what is the range of $u^T \nabla \mathcal{L}(\theta)$? Let's see...

Let β be the angle between u^T and $\nabla \mathcal{L}(\theta)$, then we know that,

$$-1 \leq \cos(\beta) = \frac{u^T \nabla \mathcal{L}(\theta)}{\|u\| * \|\nabla \mathcal{L}(\theta)\|} \leq 1$$

Multiply throughout by $k = \|u\| * \|\nabla \mathcal{L}(\theta)\|$

$$-k \leq k * \cos(\beta) = u^T \nabla \mathcal{L}(\theta) \leq k$$

Thus, $\mathcal{L}(\theta + \eta u) - \mathcal{L}(\theta) = u^T \nabla \mathcal{L}(\theta) = k * \cos(\beta)$ will be most negative when $\cos(\beta) = -1$ i.e., when β is 180°

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5 16/87

(Refer Slide Time: 03:03)

Gradient Descent Rule

- The direction u that we intend to move in should be at 180° w.r.t. the gradient
- In other words, move in a direction opposite to the gradient

Parameter Update Equations

$$w_{t+1} = w_t - \eta \nabla w_t$$
$$b_{t+1} = b_t - \eta \nabla b_t$$

where, $\nabla w_t = \frac{\partial \mathcal{L}(w, b)}{\partial w}$ at $w = w_t, b = b_t$, $\nabla b_t = \frac{\partial \mathcal{L}(w, b)}{\partial b}$ at $w = w_t, b = b_t$

So we now have a more principled way of moving in the w - b plane than our "guess work" algorithm

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5

So, that is the rule that we have been sticking to since then. And we also along the way realize some of these things which we defined carefully which was, what is, what exactly this quantity means, which is the partial derivative with respect to w evaluated at a particular weight comma bias configuration. And because this is an iterative process, you are at a certain value of weight and bias and you need to change it from there.


(Refer Slide Time: 03:30)

- Let's create an algorithm from this rule ...

Algorithm 1: gradient_descent()

```
t ← 0;
max.iterations ← 1000;
while t < max.iterations do
    | wt+1 ← wt - η∇wt;
    | bt+1 ← bt - η∇bt;
end
```

- To see this algorithm in practice let us first derive ∇w and ∇b for our toy neural network



Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5

And we then created an algorithm out of this and when we ran this, we actually derived the full derivative and so on.

(Refer Slide Time: 03:38)

```
X = [0.5, 2.5]
Y = [0.2, 0.9]

def f(w,b,x) : #sigmoid with parameters w,b
    return 1.0 / (1.0 + np.exp(-(w*x + b)))


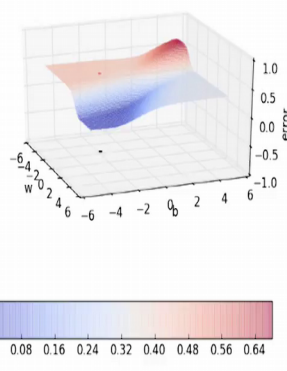
def error(w,b) :
    err = 0.0
    for x,y in zip(X,Y) :
        fx = f(w,b,x)
        err += 0.5 * (fx - y) ** 2
    return err

def grad_b(w,b,x,y) :
    fx = f(w,b,x)
    return (fx - y) * fx * (1 - fx)

def grad_w(w,b,x,y) :
    fx = f(w,b,x)
    return (fx - y) * fx * (1 - fx) * x

def do_gradient_descent() :
    w, b, eta, max_epochs = -2, -2, 1.0, 1000
    for i in range(max_epochs) :
        dw, db = 0, 0
        for x,y in zip(X, Y) :
            dw += grad_w(w, b, x, y)
            db += grad_b(w, b, x, y)
        w = w + eta * dw
        b = b + eta * db
```

Gradient descent on the error surface



Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 5

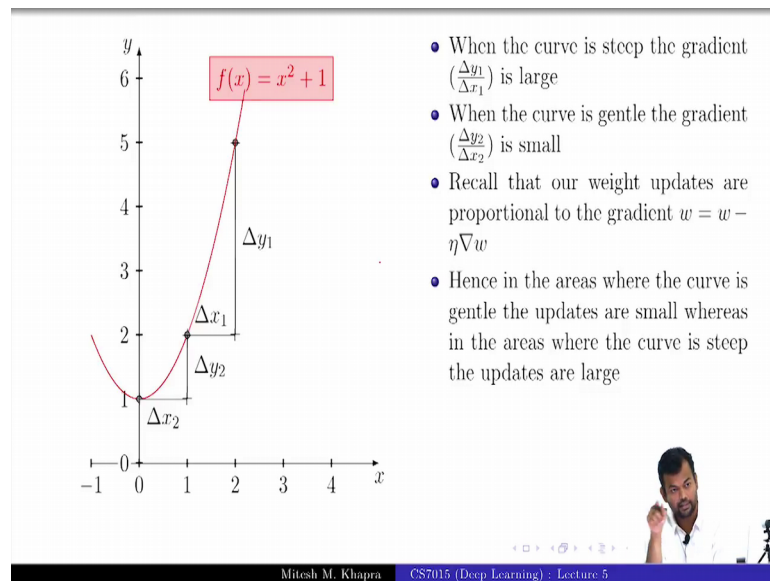
And then when we finally, ran this algorithm. So, this is where, now I will slow down. So, when we ran this algorithm. So, let us see what was happening here right. So, I will just start the algorithm from the beginning.

So, we are now going to run this code and you tell me something that you observe ok. So, I am just clicking. So, there is no change in the phase at which I am clicking this

right. So, every click of this is one time step and I am just continuously clicking this I will start now, do you observe something [FL] ok. Do you observe something?

It was initially slow then suddenly picked up and then it again became slow. Why did this happen? The slope is small why ok. How many of you completely understand why this slow and fast moment was there, please raise your hands good. So, that is what we will focus on now right. So, we will try to see this.

(Refer Slide Time: 04:30)



So, we will, I hope this has been fixed ok. So, let us take a simple function which is f of x equal to x square plus 1 right, this is how it will look like. Now in these portions of the curve, the curve is actually very steep right and in these portions the curve is a bit gentle and of course, it becomes very gentle over here right. All of you can see the pen marks properly.

So, now let us see what this means; this steep and fast and small. So, let us look at a region which is steep ok. Now what I am going to do is, I am going to change my x by 1, I move my x from 1 to 2. How much did my y change. All you need to do is just substitute in this formula right for 2 it evaluates to 5, for 1 it evaluates to 2. So, when you move from 1 to 2, your function changed from 2 to 5. So, there is a large change in the function for 1 unit change in your value of x , everyone sees that.

Now, let me do the same at a gentle portion of the curve, I will do it here. Now when I changed the x by 1 unit, again 1 unit right, it is the same change which I did earlier. I changed from zero to 1, how much did my y change.

Student: 1.

1. Now actually what is this quantity; Δy by Δx .

Student: Slope.

It is the slope, it is the derivative at that point. So, what are you inferring from this. What happens to the derivative when you are at steep slopes.

Student: It is high.

Derivative is high, because the change in y is much faster than the change in x . What happens to the derivative when you are at the gentle slopes.

Student: Smaller.

Smaller, because the change in y is small or relatively smaller as compared to the change in x or it could also missing, but just these two are relatively different, is what I am trying to impress upon right. And so; that means, the derivatives at the steep slopes are larger in magnitude, whereas, for the gentle slopes they are smaller in magnitude.

Now, can you relate it to the observation that you had on the previous slide. When we were at the plateau it was a very dash slope, gentle slope what would the derivatives be

Student: Small.

Small now what are our updates, you have w is equal to w minus the derivative. Now the derivative is small what will happen to the updates.

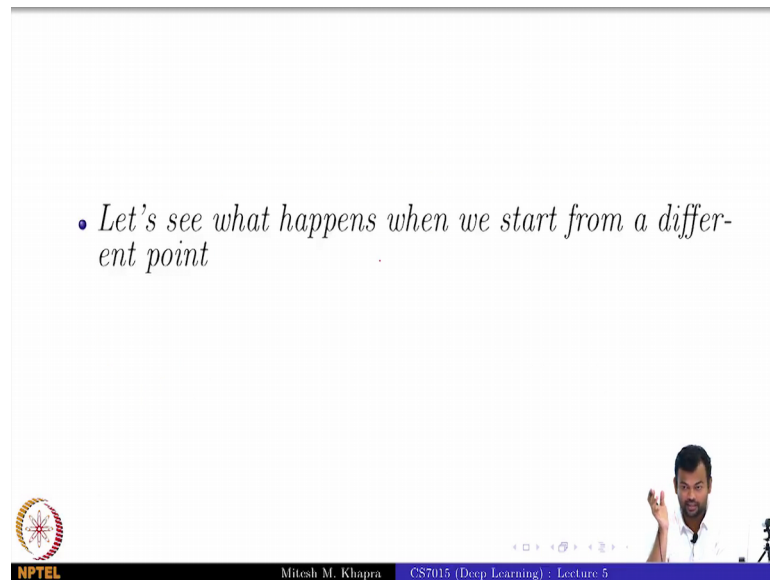
Student: Small.

They will be small. What would happen if the derivative is large.

Student: The updates would be large.

The updates would be large. Therefore, in the gentle areas you are moving slowly and in the steep areas you are moving fast ok. You get this picture very clearly. Now this is going to be the basis of a lot of things that we do today. So, it is very essential to that you understand this perfectly ok. All of you get this properly; good

(Refer Slide Time: 07:32)

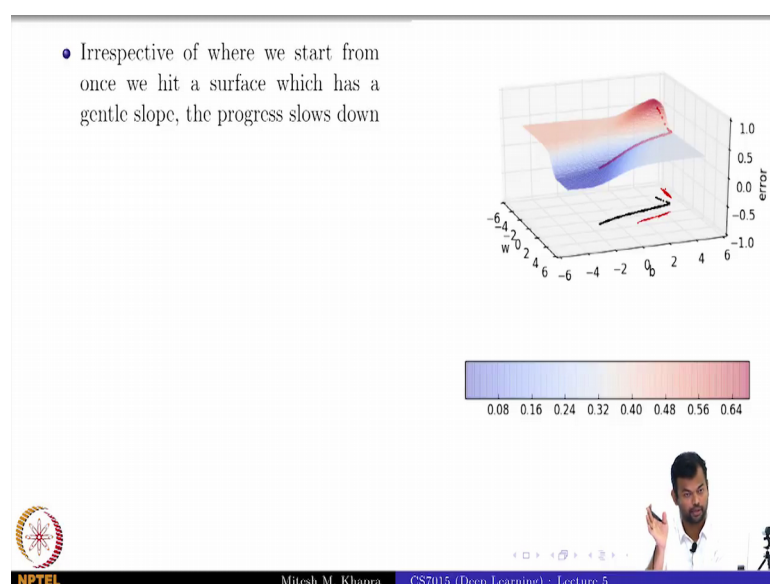


• *Let's see what happens when we start from a different point*

The slide features a white background with a single bullet point in a dark blue font. At the bottom, there is a video player interface with a small inset of the speaker, navigation icons, and a footer containing the NPTEL logo, the name 'Mitesh M. Khapra', and the text 'CS7015 (Deep Learning) : Lecture 5'.

Now, now you might say that this was only that special point again and I always get those questions. So, let us see what happens, if you start from a different point.

(Refer Slide Time: 07:40)



• Irrespective of where we start from once we hit a surface which has a gentle slope, the progress slows down

The slide contains a 3D surface plot on the right side. The plot shows a surface with a color gradient from blue (low values) to red (high values). The axes are labeled w_0 , b_0 , and z . Below the plot is a color bar legend with numerical values: 0.08, 0.16, 0.24, 0.32, 0.40, 0.48, 0.56, 0.64. At the bottom, there is a video player interface with a small inset of the speaker, navigation icons, and a footer containing the NPTEL logo, the name 'Mitesh M. Khapra', and the text 'CS7015 (Deep Learning) : Lecture 5'.

So, now again the same gradient descent algorithm I am going to run, but instead of starting at this point which was my random initialization, I just happened to choose a very different random initialization which is here ok, everyone sees that ok

Now, let us see what happens, what do you expect initially fast movement, because the steep, the slope is a bit steep. Now what would happen? It will become slow because you have entered a gentle slope region and then again fast right. So, and then again it will become slow

So, see in this gentle region right, the changes in w are so small that all your black points are actually indistinguishable from each other, it is almost like a snakes body whereas, in these steep slopes, you can see a large change in the w . You can see gaps between the values of w . So, this is irrespective of where you start from. Gentle means slow movement, steep means fast movement that is the basis.