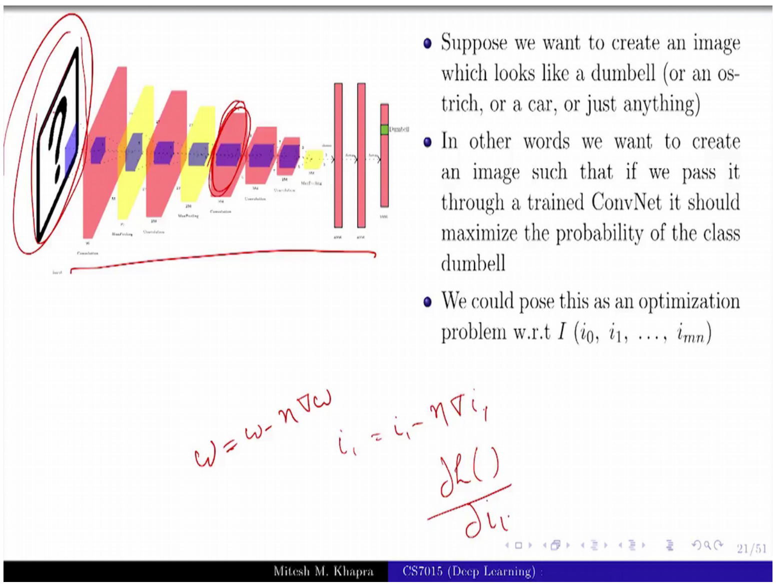
Deep Learning
Prof. Mitesh M. Khapra
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture-98
Optimization over images

Ok. The next thing that we are going to do is Optimization over images. So, this is again interesting and it eventually led to this whole field of adversarial deep learning or adversarial machine learning in general right. So, we will see what this is.

(Refer Slide Time: 00:26)



- Suppose we want to create an image which looks like a dumbell (or an ostrich, or a car, or just anything)
- In other words we want to create an image such that if we pass it through a trained ConvNet it should maximize the probability of the class dumbell
- We could pose this as an optimization problem w.r.t I $(i_0, i_1, \ldots, i_{mn})$

$w = w - \eta \nabla w$

$i_1 = i_1 - \eta \nabla i_1$

$\frac{\partial \mathcal{L}()}{\partial i_1}$

21/51

Mitesh M. Khapra CS7015 (Deep Learning)

Suppose I have a trained convolutional neural network and now, I want to figure out what kind of image should I pass through this, so that, it gets recognized as a dumbbell ok. Why where want to do I would not want to have such a weird objective, can you think of a reason, why would want such a weird objective? I know there is a convolutional neural network, which can distinguish k classes these classes could be anything.

Now, I want to deliberately create images, which get passed as the dumbbell class. Why would I want to do this? Ok you are going into your details. So, I will give you a application right suppose, this network is supposed to do face detection and the k classes which are there are k people right. Now, you want to see what kind of image should I feed to this so that, I get recognized as Amitabh Bacchan right. So, now, that could have certain benefits and various high places and so, on it is I would want to do that right.

So, that is the whole idea behind adversarial learning. So, now, I am asking this question that, I want and here it is in of course, a toy setup there is no reason, I why I would want to generate dumbbells, but say if I am going to if it is an automatic verification whether, my product looks like a dumbbell or not I might want to do this right. So, you could think of all sorts of reasons why you want to do this. So, what we will do is, the question that we are interested in is that, I have a blank slate with me, it just contains some pixels. I want to be able to modify this pixel so that, my class dumbbell class gets fired.

Now, we have done enough gradients, enough back propagation everything in this class. So, I will ask you to give me a solution for this. And the hint is, treat the image itself as a parameter matrix. The second hint is assume that all of this is going to remain constant, you are not going to change any of this. And you have initialized your parameters, which is the image pixels to 0s; that means, you are started with a gray image.

Now I will change the question a bit, only a bit and all of you will be able to answer this ok. Suppose my network is strained and now, I want to change the weights in this layer so that, my accuracy improves. So, that when it is a dumbbell class, it predicts dumbbell. How will you do that, it will pass the same image, what will you do, how will you change the weights in this layer, back propagation what is the update rule, say the gradient descent update rule, say that the gradient descent update rule.

Student: (Refer Time: 03:01).

W is equal to ok, you guys actually unanimously said gradient is an update rule ok. So, w is equal to w minus and eta into.

Student: (Refer Time: 03:16).

That is what you will do. Now if I ask you the question for this you can answer it, but if I ask you the same question here, why cannot you answer it.

Student: (Refer Time: 03:27).

So, here what were you doing, computing the gradients of the loss with respect to the weights what will you do here?

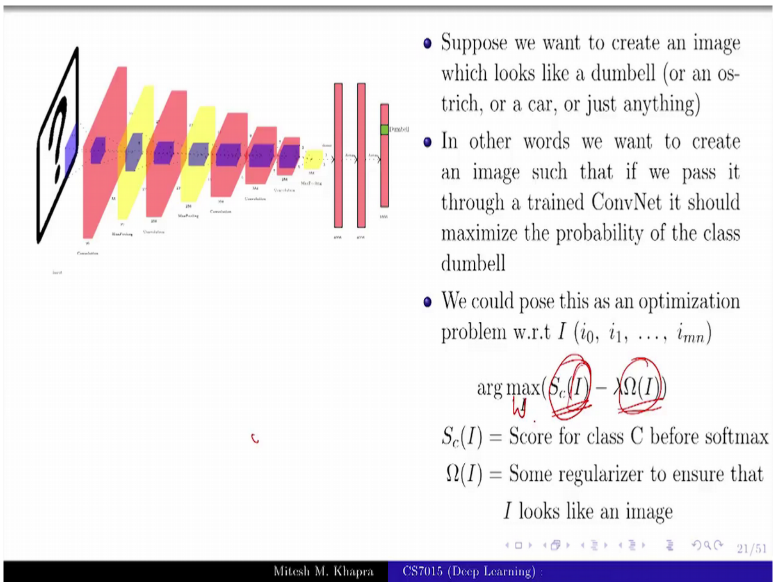
Student: (Refer Time: 03:33).

It put respect to each of these pixels and then, update this pixel by using what formula?

Student: (Refer Time: 03:39).

I 1, that is the first pixel is equal to I 1 minus eta gradient I 1 where, what is gradient I 1 actually, even gets the intuition right you can do it now.

(Refer Slide Time: 03:59)



- Suppose we want to create an image which looks like a dumbell (or an ostrich, or a car, or just anything)
- In other words we want to create an image such that if we pass it through a trained ConvNet it should maximize the probability of the class dumbell
- We could pose this as an optimization problem w.r.t I $(i_0, i_1, \ldots, i_{mn})$

$$\underset{W}{\arg\max}(S_c(I) - \lambda\Omega(I))$$

$S_c(I)$ = Score for class C before softmax

$\Omega(I)$ = Some regularizer to ensure that I looks like an image

21/51

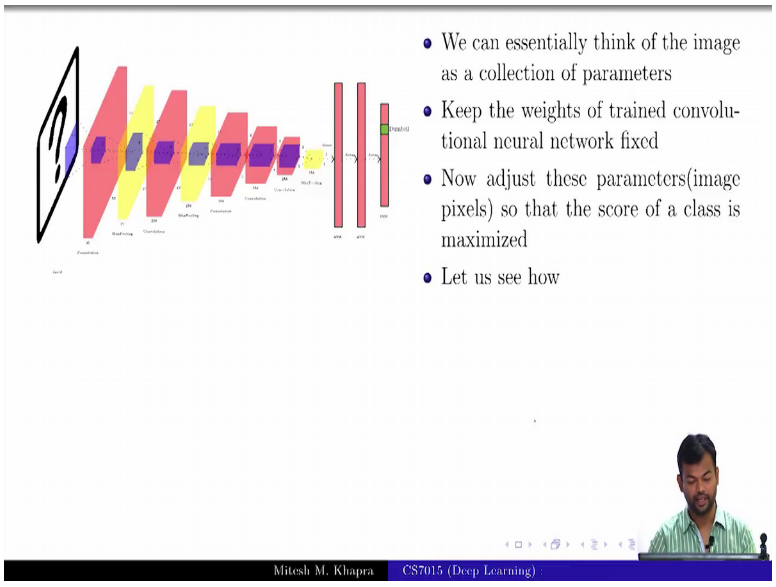
Mitesh M. Khapra CS7015 (Deep Learning)

So, we could pose this as an optimization problem where what we want to do is, given an image, we want to maximize the score of the output class. And I also want some regularization because; whatever I get I want it to look like an image right.

So, we will see different types of regularization for doing this, some very simple regularizations but this is the overall idea right. So, any generic loss function is always the training loss plus the regularization So, I have just kept both, the training loss as well as the regularization. What is my training loss? The score for the class that I am interested in. And what are the parameters of this object of this optimization problem, the input pixels right.

So, far we had already be always been doing w comma b but instead of w comma b you now, have I as the parameters of your optimization problem, is that fine.

(Refer Slide Time: 00:46)

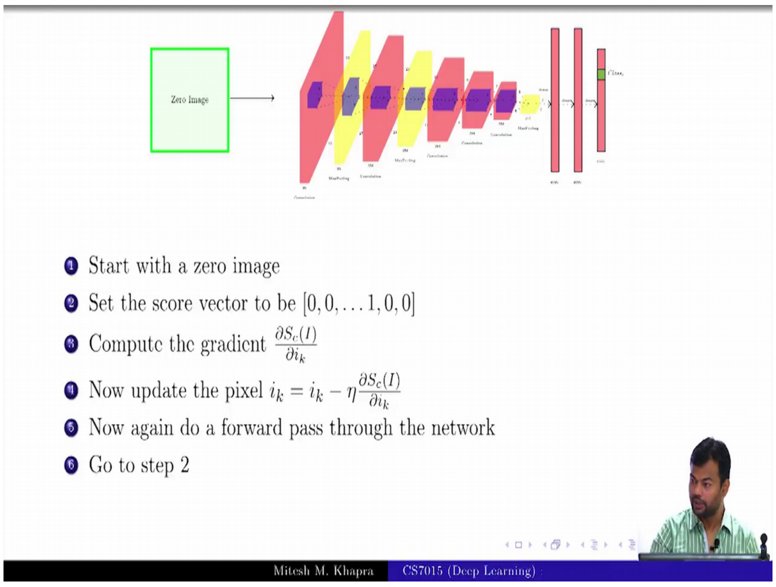


- We can essentially think of the image as a collection of parameters
- Keep the weights of trained convolutional neural network fixed
- Now adjust these parameters(image pixels) so that the score of a class is maximized
- Let us see how

Mitesh M. Khapra CS7015 (Deep Learning)

And now, we can just think of the entire image as a collection of parameters, and we can now update the weights of this matrix, which is the image matrix ok.

(Refer Slide Time: 04:58)



1. Start with a zero image
2. Set the score vector to be $[0, 0, \dots 1, 0, 0]$
3. Compute the gradient $\frac{\partial S_c(I)}{\partial i_k}$
4. Now update the pixel $i_k = i_k - \eta \frac{\partial S_c(I)}{\partial i_k}$
5. Now again do a forward pass through the network
6. Go to step 2

Mitesh M. Khapra CS7015 (Deep Learning)

So, let us see how we will do it. So, we start with a 0 image as I said. Set the score vector to all zeros and 1 for the class that I am interested in ok.

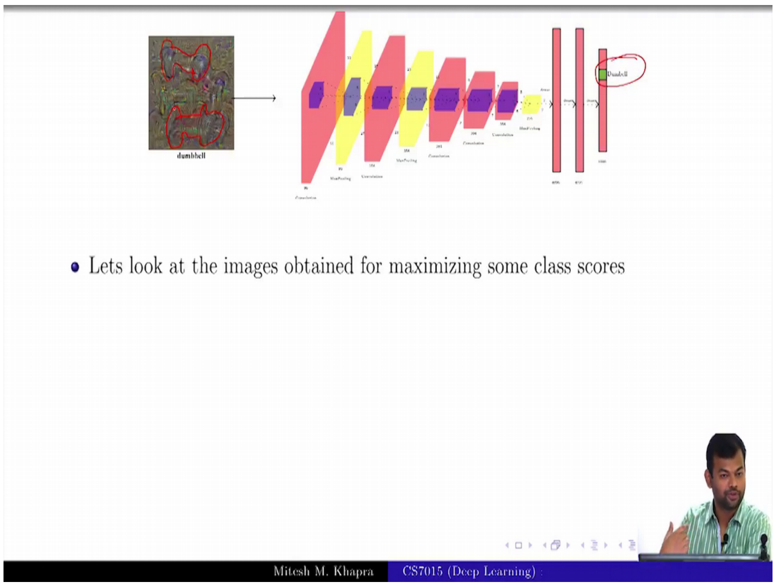
Now, compute the gradient of this score vector with respect to ik, it is I want this quantity to be maximized everything else to be 0. So, that is what my loss function is. So, I am going to compute the gradient of each of the pixels with this. Now I am going to update the pixel using my gradient descent rule, which I just explained brief previously.

Now I again do a forward so, now instead of this 0 image I have a modified image, slightly modified image because, the pixels I have moved away from 0 update based on the gradients. Now this image I will pass back through the network and what will I do now? Again change, so this is the same as the weight matrix right, so you should be able to visualize it exactly the same way as you would have visualized this. You had certain weights here, you change them a bit, again did the forward pass, again did the backward pass change them a bit and keep doing this till.

Student: (Refer Time: 05:59).

Till convergence right, whatever is your definition for convergence till you are satisfied and instead of score of 1 you are at least getting a score of 0.9 or 0.95 or something like that right. So, we will keep doing this right till convergence. At the end you will you all of you can imagine that this image will keep getting modified ok.

(Refer Slide Time: 06:20)

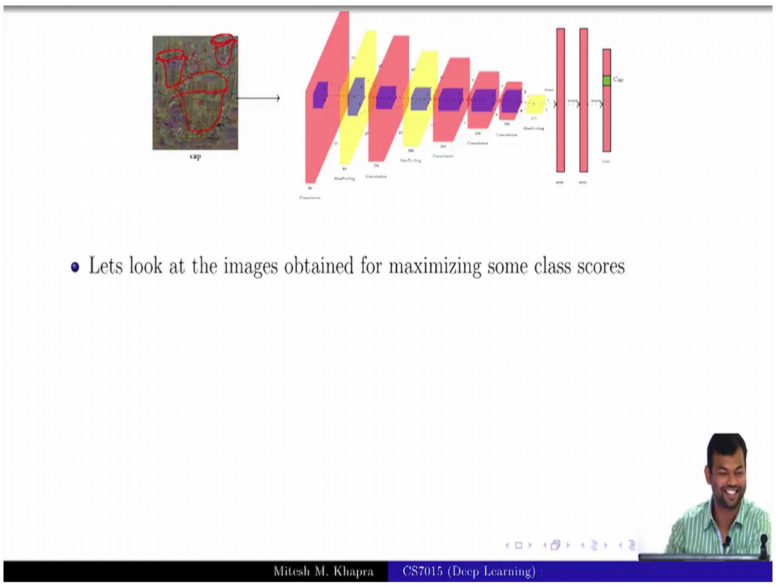


- Lets look at the images obtained for maximizing some class scores

So, now let us see if, we learn run this score or the run this code for certain classes. So, I mean interested in the dumbbell class and I have ran that algorithm starting with the 0 image and this is the kind of image that I end up with. You see a dumbbell here, without me drawing it right, if you go back and look at it, you will see that there are a lot of these dumbbell like shapes which have actually appeared here. The colour is of course, very much different I do not think dumbbells are of these colours ever, but you can see that it is actually trying to produce that shapes, which will cause the dumbbell output to fire.

Now, what is interesting is that, it is being very redundant. So, it is not trying to generate a single dumbbell, a generating a lot of dumbbells of different orientations. So, i just keeping its basis covered so that, some of this should actually fire and cause a dumbbell output to be maximized ok.

(Refer Slide Time: 07:14)

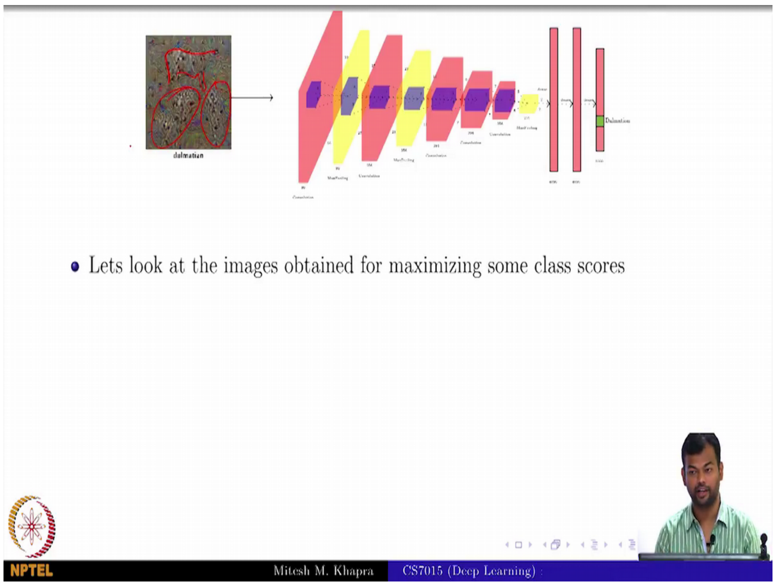


- Lets look at the images obtained for maximizing some class scores

Mitesh M. Khapra CS7015 (Deep Learning)

Now, let us see if we take a cup and this is like the trophy cup I believe. So, this is what is appearing here, there is one more cup here and there is one more cup here. It is a generating these cups so that, you cannot be, you would not be able to see it, it is different oh it really looks like I am manipulating it, but I am not, you can go back in check it, those cups are there ok.

(Refer Slide Time: 07:41)

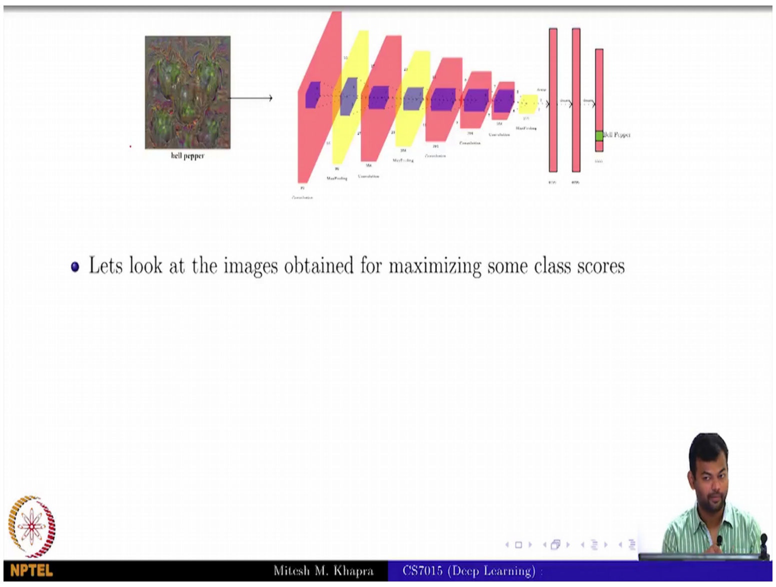


- Lets look at the images obtained for maximizing some class scores

And then, for dalmatian actually, this at least you can see some white and black spots right at least that is fine.

So, dalmatians are these dog which have these white and black spots. So, and you can also see some kind of a shape here right, which with my drawing. So it is actually producing that doglike shape and it is producing multiple of those .So, it is being redundant I am trying to compute that right.

(Refer Slide Time: 08:04)

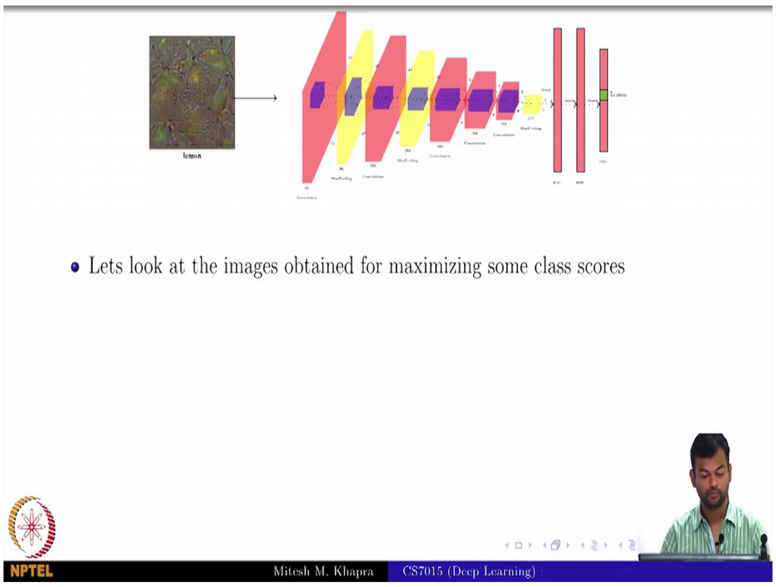


- Lets look at the images obtained for maximizing some class scores

And now you see right with these very arbitrary images, which to you and me do not know nowhere close to we will fire will classify this as dalmatian. But for the machine and is classifying this as a dalmatian and this is bad right, this is not good, there is nothing to be impressed about this is actually bad because, I can give it these horrible images and still get away by something called as a dalmatian.

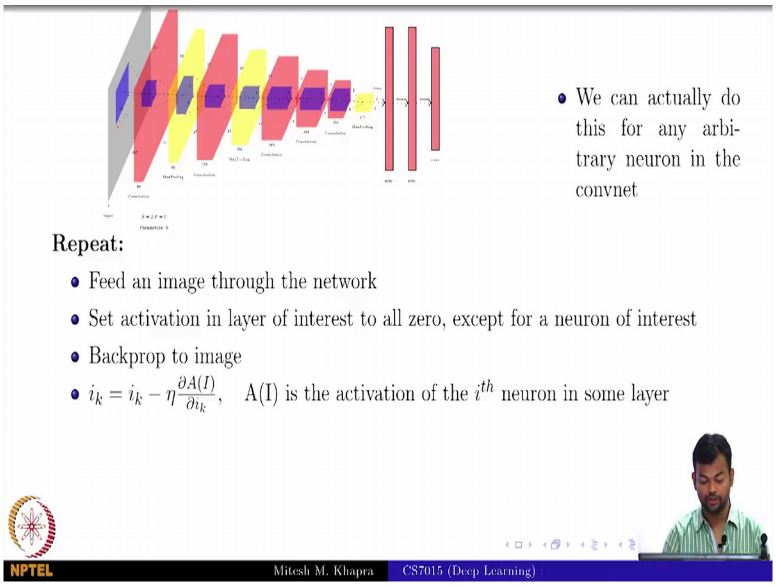
So, if I want to sell some a dalmatian on OLX, this is what I can do right. I can upload this image and a machine would trigger it and some one would buy it ok. So, and this is a bell paper so you can go back and see you see a lot of bell papers here and similar for lemon and so on right.

(Refer Slide Time: 08:41)



- Lets look at the images obtained for maximizing some class scores

(Refer Slide Time: 08:45)


- We can actually do this for any arbitrary neuron in the convnet

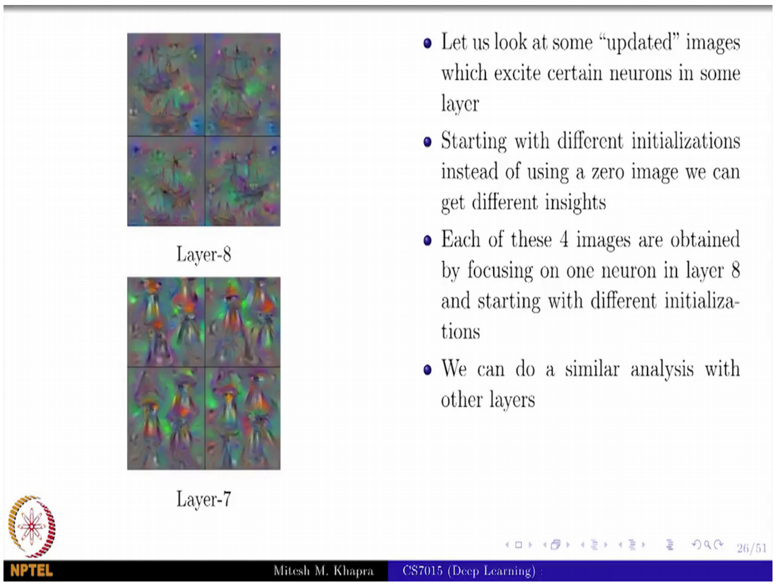
Repeat:

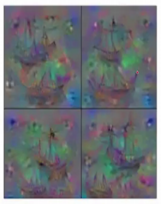
- Feed an image through the network
- Set activation in layer of interest to all zero, except for a neuron of interest
- Backprop to image
- $i_k = i_k - \eta \frac{\partial A(I)}{\partial i_k}$, A(I) is the activation of the i^{th} neuron in some layer

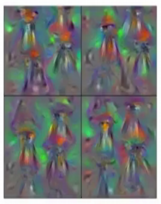
NPTEL Mitesh M. Khapra CS7015 (Deep Learning)

So, various classes you can see that, it is actually trying to produce those shapes, but it is nowhere actually producing a clear image, which is undoubtedly of that object right, is generating something which can later on be used to fool the network right, which is not a good thing ok. And we can actually do this for any arbitrary neuron, so I was trying to actually fire this neuron, which was the output layer, but maybe I want something else to fire here so, I want to actually see what is, it that causes this neuron to fire. So, I could repeat the same algorithm by setting something here as high and then again back propagating the gradients only from here, and reconstructing the image every time so that, this neuron then 5 is right.

(Refer Slide Time: 09:25)




Layer-8


Layer-7

- Let us look at some "updated" images which excite certain neurons in some layer
- Starting with different initializations instead of using a zero image we can get different insights
- Each of these 4 images are obtained by focusing on one neuron in layer 8 and starting with different initializations
- We can do a similar analysis with other layers

So, these are what the updated images look like, which excite certain neurons and some layers. So, what does this look like? It is actually like a pirates ship if it is not very clear, you have these multiple layers of things and something like this ok. So, it is some neurons are actually firing for this kind of a pattern; there are some other neurons which are firing for different kinds of patterns and so on right.

So, you can just create images which cause certain neurons to fire and all these are lot of fun to do. So, you should I would encourage you to do this, I will get more insights into what your network is.