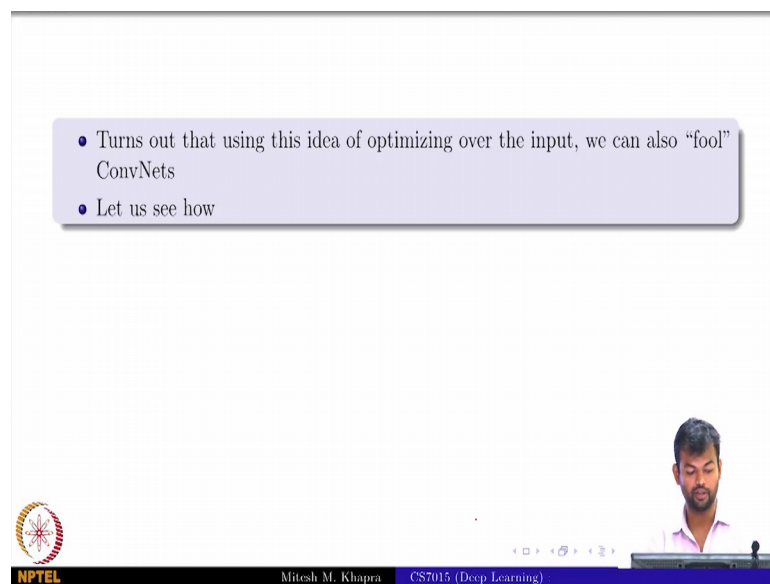


Deep Learning
Prof. Mitesh M. Khapra
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture-102
Fooling Deep Convolution Neural Networks

With that we go on to the last module which is Fooling Deep Convolutional Neural Networks.

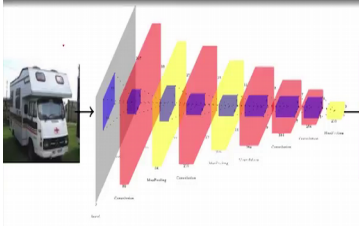
(Refer Slide Time: 00:19)



So, turns out that using this idea of optimization where, we are able to actually change the image to suit our needs right. And these needs were, one was we wanted to change the image so that, it fires for a particular class. The other was deep dream, where we wanted to change the image so that, it starts seeing patterns which were otherwise not observed in the image. And the other was d part, where we trained the image or we optimized over the image so that, we could produce some artistic images and these are the different optimization problems that we have seen.

But the same idea can actually also be used to full convolutional neural networks and I have already hinted at this earlier. So, let us see how to do that.

(Refer Slide Time: 00:55)



- Suppose we feed in an image to a Convnet.
- Now instead of maximizing the log-likelihood of the correct class (bus) we set the objective to maximize some incorrect class (say, ostrich)
- Turns out that with minimal changes to the image (using backprop) we can soon convince the Convnet that this is an ostrich.
- Let us see some examples

NPTEL
Mitesh M. Khapra CS7015 (Deep Learning) 48/51

So, now suppose, we feed an image to a convnet and I know this is the bus image right, but now what I do is, this is a trained convolutional neural network and what I do is instead of setting the cross entropy loss to maximize bus, I will set up the cross entropy loss to maximize Ostrich. And then I will back propagate through the network, I will not modify any of these weights or parameters and I am only change the image right.

So, what I am trying to do is, I know that this is the bus image, but now I am setting the objective that, it should fire for the Ostrich class. So, now, I am going to back propagate and change this image so that, the log likelihood of the Ostrich class increases. You get this set up its very straightforward and turns out that if you do this with very minimal changes to the image, you can actually fool the convolutional neural network ok.

(Refer Slide Time: 01:50)

• Notice that the changes are so minimal that the two images are indistinguishable to humans

• But the ConvNet thinks that the third image obtained by adding the first image to the second image is an ostrich

*Intriguing properties of neural networks, Szegedy et al., 2013

Mitesh M. Khapra CS7015 (Deep Learning) 49/51

So, this is the change right, you have the original image, the second image is actually the amount of change you made and the third image is the original image plus this change.

Now, to the human eye there is no distinction here right, you would all of first would still think this is a bus and in fact, I do not even see that, there is a noise in the third bus that you see. Same for some other class they have taken some bird or something like that and added some noise to it and a temple. And in all of these cases, the network actually predicts that the modified image is an Ostrich right or some very random class from the original class. So, why is this happening and before asking that question let me just finish and it need not be that you start with an original image and then try to modify it.

(Refer Slide Time: 02:34)

- We can also do this starting with random images and then optimizing them to predict some class.
- In all these cases the classifier is 99.6% confident of the class
- Let us see an intuitive explanation of why this happens

*Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images Nguyen, Yosinski, Clune, 2014

Mitesh M. Khapra CS7015 (Deep Learning) 50/51

Actually, you can start with a blank image and do the same experiment where you modify the image minimally so that, p of robin becomes 1 or close to 1. And you will get some very arbitrary noisy looking images, which no in; which to at least you and me do not look like a cheetah or robin or armadillo, but the network thinks that these are the classes that these images belong to.

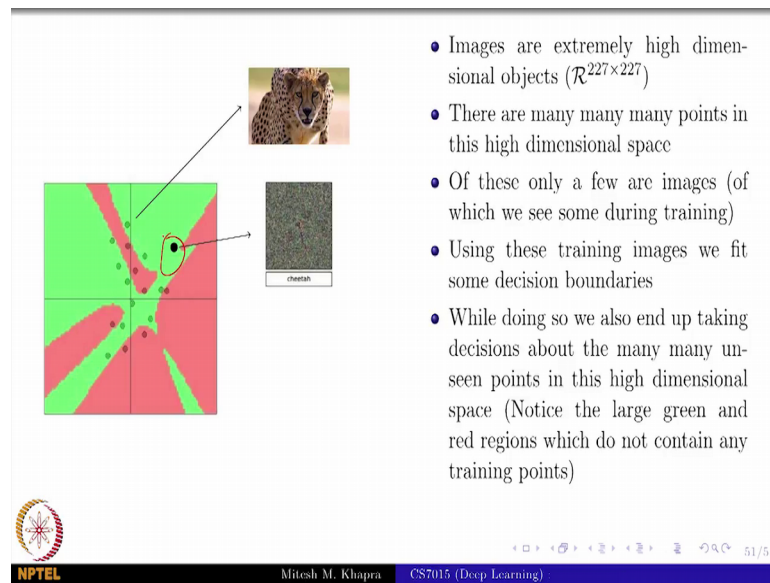
Now, this is definitely a risky, how many of you appreciate that it is bad ok. Now and a network is not just predicting, it is predicting it with a very high confidence right 99.6 percent confidence. So, why is this happening, can even think of a reason for that?

Student: (Refer Slide Time: 03:12).

No, but in that case I would have been fine if there are 1000 classes it should have given 1 by 1000 probability to all the classes right, but this is like worse than random classifier right. It is saying with 99 percent confidence that, this is a Ostrich or whatever class that is So, why is this happening and the interesting thing is that, this in some sense ties back to the universal approximation theory or at least some ideas with that. Can you think of why this is happening ok.

So, let us try to see a very intuitive explanation for this so on.

(Refer Slide Time: 03:45)



- Images are extremely high dimensional objects ($\mathcal{R}^{227 \times 227}$)
- There are many many many points in this high dimensional space
- Of these only a few are images (of which we see some during training)
- Using these training images we fit some decision boundaries
- While doing so we also end up taking decisions about the many many unseen points in this high dimensional space (Notice the large green and red regions which do not contain any training points)

NPTEL Mitesh M. Khapra CS7015 (Deep Learning) 51/51

So, this explanation is due to Andree Karpathi, we need to put the acknowledgments; this slide does not have any acknowledgments actually. So, remember that images are extremely high dimensional objects right, they are 227 cross 227 which is a very very high dimensional object, high dimensional space. And no matter how much training data you have, you see a only a small sample of this high dimensional space right because, its real numbers 227 cross 227, just imagine the number of possibilities out there, no matter you have 1 million samples 10 million samples for training, this is much much smaller than the actual number of samples which exist in this space. Of these only a few are images right.

So, now, think of all 227 cross 227 matrices that you can make and how many of them are actually going to be natural images. The probability of natural images is very very very very small, most of these are random things right, they are just matrices which do not make any sense which actually look like these images that you see here right.

Now, using the training images, we fit some decision boundaries and this is the decision boundaries that we fit right that, this is class 1, the rest of the green part is class 2 and so on. And in fact, we are doing these decision boundaries for some 1000 classes While doing so, we actually end up taking decisions for a large number of points that we have not seen. We have not seen any points in this space, but I have made a decision for them

that, all of them belong to the green class. I have not seen any point in this space, but I have ended up taking a decision for them that all of them belong to the red class right.

So, in particular what I have done is, I saw a cheetah class, image from a cheetah class. I saw a few images from the cheetah class and I drew some boundary around it to say that this is the cheetah class, but my boundary also contains images like this. Because, this is a very high dimensional space and in that boundary a lot of points actually fall in and some of these points are these random points, which have no relation to cheetah right. But I have been so aggressive in fitting to the training data, that I have drawn these boundaries which also include a lot of these points and now, all I need to do starting with these random images is that, go somewhere inside this boundary and then I am all set right it will start detecting it as cheetah because, the boundaries have been drawn by the classifier. How many if you get this explanation, good. So, that is the intuitive explanation for why this happens. So, this is where we will end the discussion on convolutional neural networks.