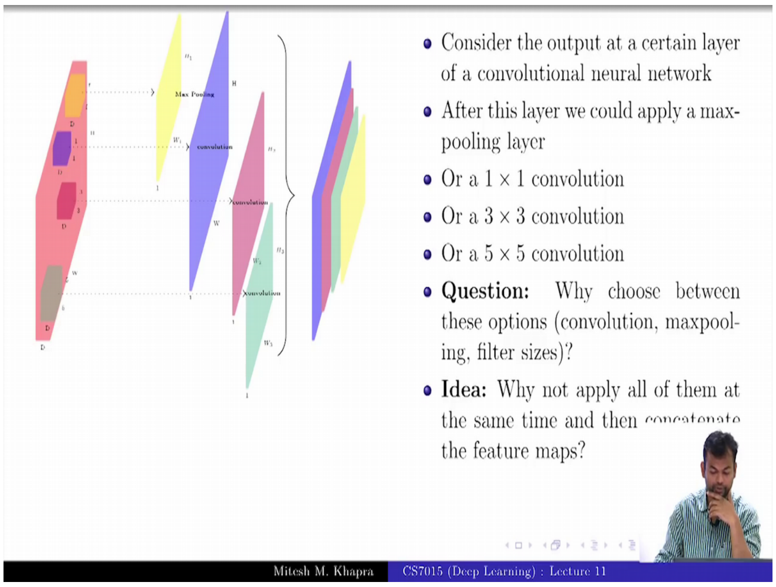
Deep Learning

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

Lecture - 11
Image Classification continued
(GoogLeNet and ResNet)

So, we will go to the next module where you wanted to look at two more architectures for image classification, these are GoogLeNet and ResNet.

(Refer Slide Time: 00:22)



- Consider the output at a certain layer of a convolutional neural network
- After this layer we could apply a max-pooling layer
- Or a 1×1 convolution
- Or a 3×3 convolution
- Or a 5×5 convolution
- **Question:** Why choose between these options (convolution, maxpooling, filter sizes)?
- **Idea:** Why not apply all of them at the same time and then concatenate the feature maps?

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

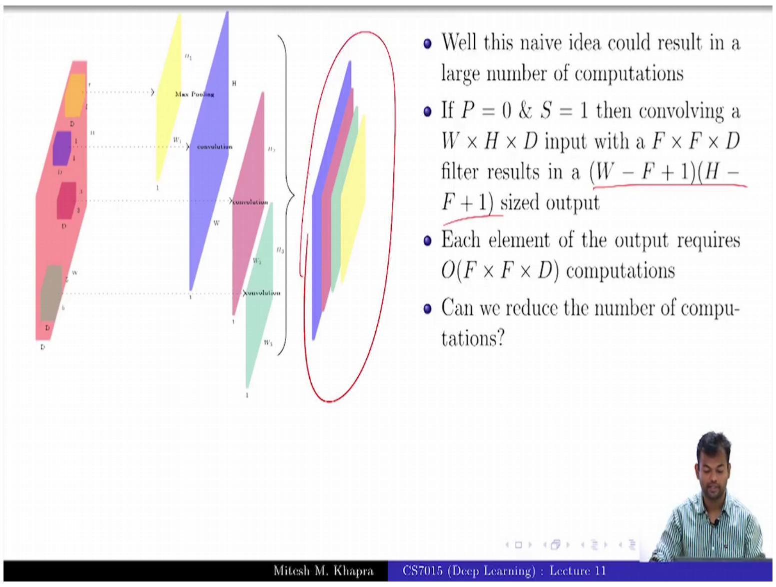
So, here is a question right, so, consider the output at a certain layer of a convolutional neural network. So, you have this layer after layer of convolutions and max pooling and so on and, you are at somewhere in the middle and this is what your volume looks, like this is what your output volume looks like. Now, after that you could apply a max pooling layer, you could apply a 1 cross 1 convolution, you could apply a 3 cross 3 convolution or you could apply a 5 cross 5 convolution right.

And so, far we saw that all these architectures they do one of these things. They either do a max pooling or they do a 3 cross 3 convolution or they do a 5 cross 5 convolution or a 7 cross 7, 11 cross 11 right any convolution. But, they are all uniform they are all either 3 cross 3 all either 5 cross 5 or either 7 cross 7 right. So, why choose between these options? Why not do all of this at every layer? Do you get the question that I am trying to ask right?

So, far what we were doing is that you have this volume, this volume at a certain layer of the convolutional neural network and after that you are either using all 3 cross 3 filters. So, you are using 256 3 cross 3 filters or 5 into 3 cross 3 filters or using 7 cross 7 or using 5 cross 5, you are never using a mix of all these right. So, why not use a mix of all these? Why the why take a decision on that I only want 3 cross 3? Because, it is possible that you want to capture interactions at different levels so, you should have varied size filters at every layer.

So, how many of you get the question and the intuition that I am trying to ask . So, the idea here in GoogLeNet or in the inception net is that why not apply all of them at the same time and then concatenate the feature maps right. So, I will also do max pooling, I will also do 3 cross 3 feature maps, I will also create 5 cross 5 11 cross 11 and then just concatenate all of these together so, let us see how to do that right.

(Refer Slide Time: 02:14)



- Well this naive idea could result in a large number of computations
- If $P = 0$ & $S = 1$ then convolving a $W \times H \times D$ input with a $F \times F \times D$ filter results in a $\underline{(W - F + 1)(H - F + 1)}$ sized output
- Each element of the output requires $O(F \times F \times D)$ computations
- Can we reduce the number of computations?

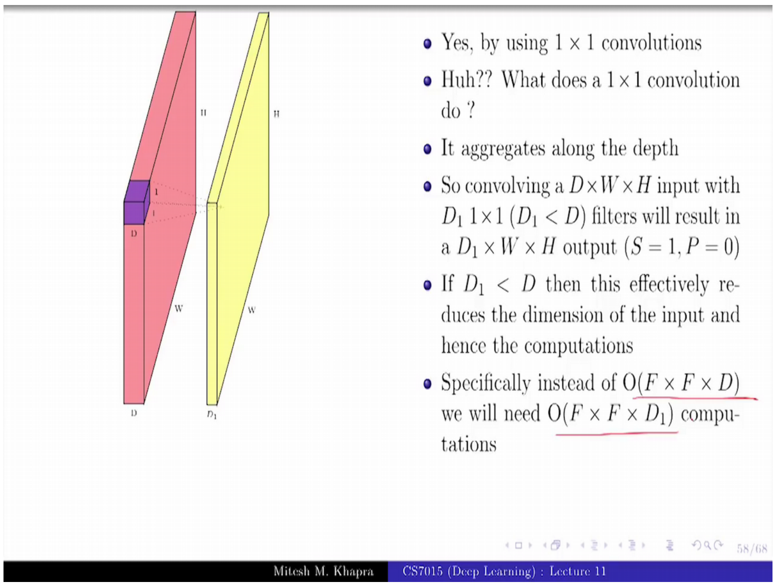
Now, one problem with this naive idea is that it could result in a large number of computations so, let us see what I mean by that. So, suppose the padding is 0, the stride is 1 then, if you have a W cross H cross D input as the volume and if you have an F cross F cross D filter, then the output would be of this size we all agree with this, this is the formula that we have been looking at. So, this is the size of the output volume. Now, every entry in this output volume requires how many computations? To get a single entry in this output volume how many computations do I have to do?

Student: F cross F cross D.

F cross F cross D so, how do I get a single value? I apply a convolution at that value and then I do those many computations. And, here the number of computations is that I am going over this block of F cross F cross D, doing a weighted multiplication and then adding them up right. So, you need that many computations, everyone is clear with this fine.

So, each element of the output requires these many computations and we have so many elements in the output right. So, you are doing really those many number of computations right. So, can we do something to reduce the number of computations right so, that is the key idea that we need to focus on. So, all of us buy the idea that doing this multi granular or multi sized filters is a good idea because, you are capturing interactions are different layer, but I showed you that this is a problem with this; you guys just apply multiple filters so, let us see what we do.

(Refer Slide Time: 03:53)


H
H
1
1
D
W
W
D
D_1

- Yes, by using 1×1 convolutions
- Huh?? What does a 1×1 convolution do ?
- It aggregates along the depth
- So convolving a $D \times W \times H$ input with D_1 1×1 ($D_1 < D$) filters will result in a $D_1 \times W \times H$ output ($S = 1, P = 0$)
- If $D_1 < D$ then this effectively reduces the dimension of the input and hence the computations
- Specifically instead of $O(\underline{F \times F \times D})$ we will need $\underline{O(F \times F \times D_1)}$ computations

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

58/68

So, we what we do is 1 cross 1 convolutions. What is the 1 cross 1 convolution do? What does it make sense?

Student: (Refer Time: 04:02).

How does the 1 cross 1 convolution make sense? You have a pixel, I fit a 1 cross 1 convolution on that what will I get? I will get back the pixel.

Student: (Refer Time: 04:12).

Along the depth right so, remember it is not 1 cross 1, it is 1 cross 1 cross depth.

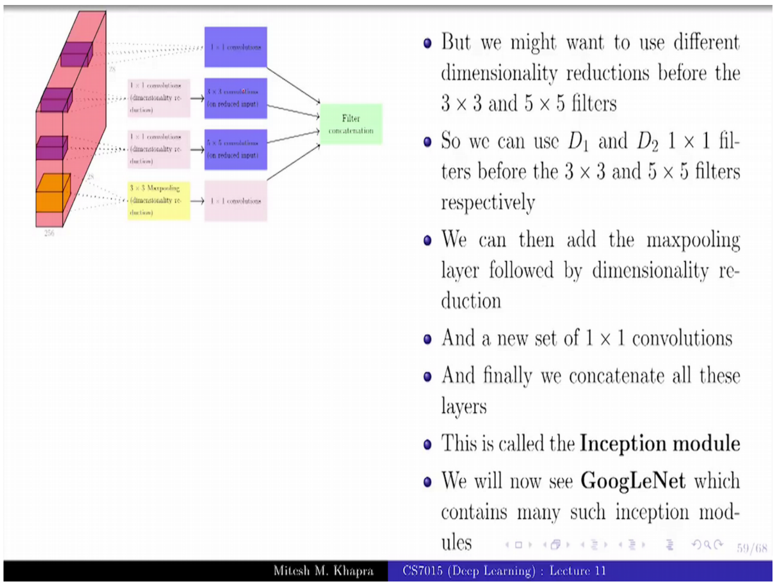
Student: Depth.

Right so, what does a 1 cross 1 convolution do? It actually aggregates along the depth. So, this is what my 1 cross 1 convolution looks like it is 1 cross 1 cross D. So, I just fit that block on that pixel and do everything along the depth and get a single value right. So, from a 3 D output using a 1 cross 1 convolution, I can go to a 2 cross 2 D feature map, everyone gets this ok.

Now, I could use several of these 1 cross 1 operations 1 cross 1 convolutions. In fact, I could use D 1 of these such that D 1 is less than D. So, what effectively happens? The depth of the output reduces. So, I take a certain output volume whose original depth was D, now I take D 1 1 cross 1 convolutions right. So, I get an output whose depth is smaller than the depth of the original output, is that fine everyone gets this ok. .

And, you see how this will save computations right because, remember that this was F cross F cross D and now I have reduced D to D 1 ok. So, it is going to reduce the number of computations. So, that is what the idea is you reduce it from F cross F cross D to F cross of cross D 1 right. So, that is this particular network or this paper introduced the idea of this 1 cross 1, actually it did not introduce it used it, but it made it popular probably.

(Refer Slide Time: 05:41)


28
1 × 1 convolutions
1 × 1 convolutions (dimensionality reduction)
3 × 3 convolutions (on reduced input)
1 × 1 convolutions (dimensionality reduction)
5 × 5 convolutions (on reduced input)
28
3 × 3 Maxpooling (dimensionality reduction)
1 × 1 convolutions
Filter concatenation
256

- But we might want to use different dimensionality reductions before the 3×3 and 5×5 filters
- So we can use D_1 and D_2 1×1 filters before the 3×3 and 5×5 filters respectively
- We can then add the maxpooling layer followed by dimensionality reduction
- And a new set of 1×1 convolutions
- And finally we concatenate all these layers
- This is called the **Inception module**
- We will now see **GoogLeNet** which contains many such inception modules

59/68

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

Now, once you have done this so, this is how I am going to proceed now. I have a certain volume I have applied 1 cross 1 convolutions to it, using that I have reduced the number of dimensions. Now, I am going to apply 3 cross 3 convolutions as well as apply 5 cross 5 convolutions on top of that right because, that is the motivation that I had started with that I want to apply kernels of multiple granularity.

Now, can you think of some refinement to this? You see this branching over here, why use the same 1 cross 1 convolutions before feeding to 3 cross 3 as well as 5 cross 5. I could use a different set of 1 cross 1 convolutions and feed it to a 5 cross 5 and use a different set of 1 cross 1 convolutions and feed it to 3 cross 3, is that fine. What is the problem with this?

Student: Again, increasing the number of computations.

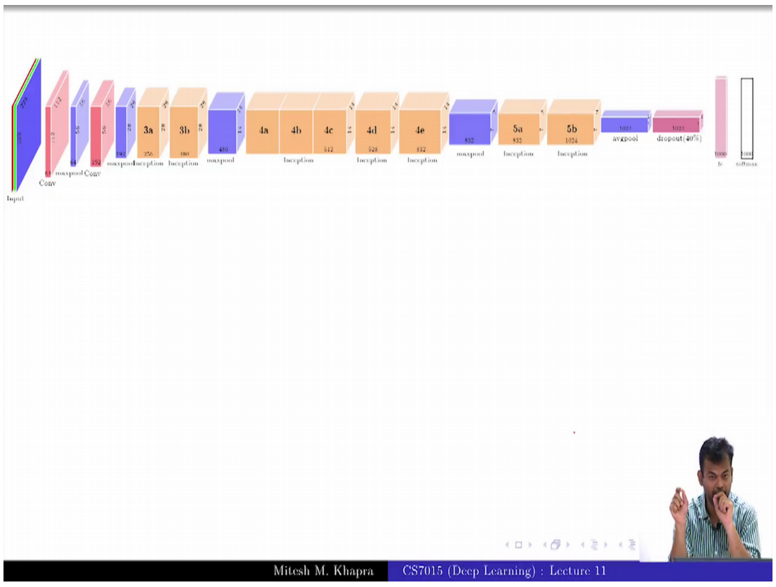
Again, increasing the number of computations right, but they found out that the tradeoff between this is fine. Even if you are doing more 1 cross 1 operations it still is ok, the number of computations are still manageable ok. And, then you could also do a max pooling because, we were choosing between these things right 5 cross 5 3 cross 3 7 cross 7 and max pooling. So, we will do all of these in parallel and we also do some 1 cross 1 convolutions. So, how many different types of operations we have done? We have done 1 cross 1 3 cross 3 5 cross 5 and max pooling followed by 1 cross 1 convolutions.

Now, all of these outputs that we have got, we have got a bunch of feature maps now. This is one set of feature maps, this is another feature maps, this is another and this is another. All of these 4 we are going to concat together to get a single output volume. Do you see what is happening right? It is not very mechanical there is nothing really profound about what is being done. The only two profound ideas are, one is apply multiple kernels of different sizes and the other is to use 1 cross 1 convolutions to make the whole computation manageable; that these are the only to main ideas. The rest of it is not very different from what we have been doing. How many if you get this operation completely?

So, this block is called the Inception module ok, this entire thing is called an inception module. So, in subsequent slides when I put an inception module then you know that these parallel operations are happening right. So, far whenever we had seen a convolutional neural network, it was all serial right so, you started with one operation then another operation and so on. Now, you have an output or an input volume, you apply multiple operations in parallel and get one single output right so, it is a parallel serial combination ok.

So, you will now see the full GoogLeNet architecture so, his question was basically 3 cross 3 would result in a different sized feature map right because of an 5 cross 5 would result in a different sized feature map. So, I will use appropriate padding so, that all of this becomes equal ok.

(Refer Slide Time: 08:26)



Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So, this is how GoogLeNet looks like so, you have the input again RGB and same 227 cross 227 or 229 cross 229. Then you apply a convolution layer followed by a max pooling layer, convolution, max pooling then you have this inception module with a very specific configuration. So, they have 96 1 cross 1 convolutions before feeding to 128 3 cross 3 convolutions, 16 1 cross 1 convolutions before feeding to 32 5 cross 5 convolutions and so on. And, I do not really see much point in going into the details of these numbers, there is hardly any intuition behind them.

I again guess that it's you try a bunch of things and this is the one which probably gave the best output. So, the key idea is that of course, you have this inception module which is a parallel module which does a lot of operations in parallel. This is again followed by another inception module which has a different configuration followed by max pooling, then again a few inception modules. In fact, 5 of them again max pooling then inception and this is the other interesting idea that they came up with.

So, at this point remember in VGG net at the final layer you had an output of size 512 cross 7 cross 7 right. And, we said that this was a problem how many of you remember this? Why was this a problem?

Student: (Refer Time: 09:54).

Because, I need to connect this to a.

Student: Fully connected.

Fully connected layer right and that fully connected layer was of size 4096 right. So, what they said is that what you could do is instead of taking 512 cross 7 cross 7, for each of these 512 feature maps that you have take the 7 cross 7 and just do an average pooling from there. What does what do I mean by that?

Student: Average.

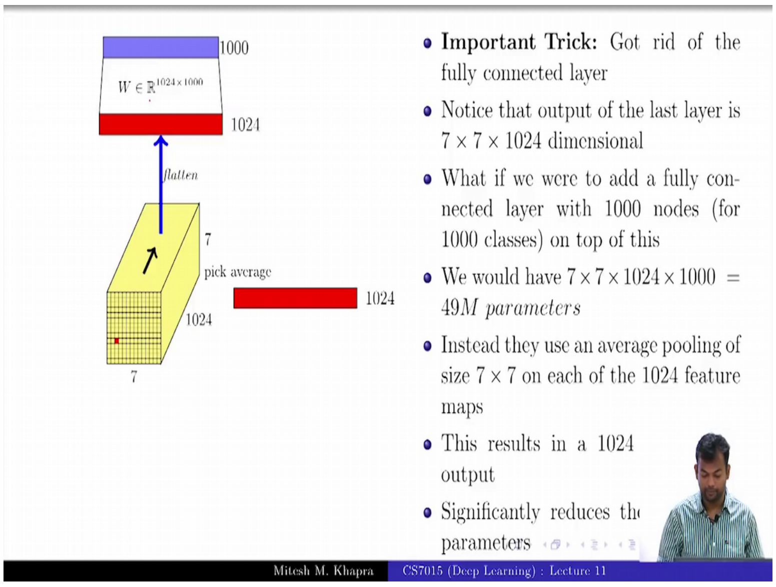
Take these 7 cross 7 values, take an average of that. So, now instead of 512 cross 7 cross 7, how many values will you end up with?

Student: 512.

Just 512 and in their case instead of 1024 cross 7 cross 7 you will just end up with 1024 values right. So, instead of looking at these dense connections with every pixel in the output volume, you just take the average of those pixels and then do a dense connection from there. So, from this volume you just go to a vector of size 1024 which is exactly this vector shown here and from there on life becomes easier right because, you have done a 50 percent sorry 50 times reduction in the volume. So, this was 1024 cross 49, now we just have 1024 cross 1 so, you have a 49 times reduction in the size and that is a huge parameter reduction.

And, that actually worked very well in practice they of course, add these dropouts and other things and then you have your fully connected layers. And finally, the soft max layer at the output to predict one of the thousand classes right. So, this is the full structure of GoogLeNet or inception net or with multiple inception modules right. So, just remember that key takeaways are three: one is half filters at multiple granularity applied in parallel, the other is use 1 cross 1 convolutions to reduce the number of computations. And, the final one is to use this average pooling to make sure that you do not have this blow up of parameters at the output ok. So, these are the three main ideas that you need to do right ok.

(Refer Slide Time: 11:45)

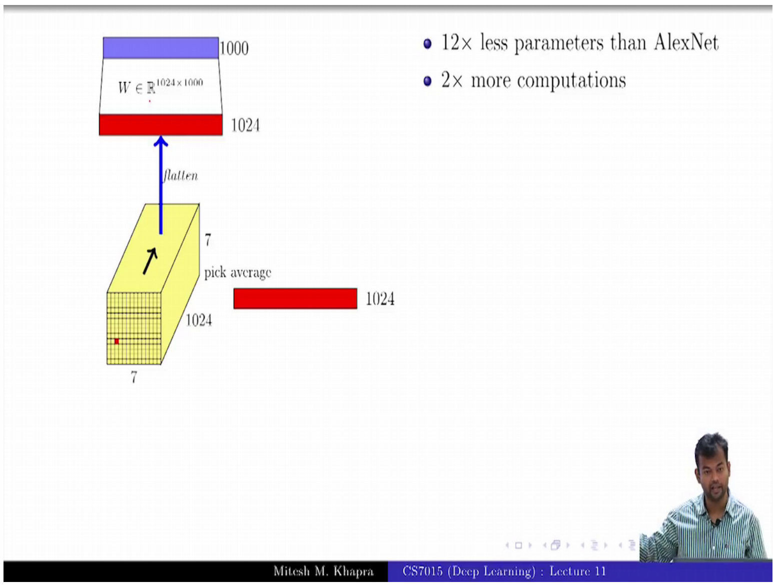


- **Important Trick:** Got rid of the fully connected layer
- Notice that output of the last layer is $7 \times 7 \times 1024$ dimensional
- What if we were to add a fully connected layer with 1000 nodes (for 1000 classes) on top of this
- We would have $7 \times 7 \times 1024 \times 1000 = 49M$ *parameters*
- Instead they use an average pooling of size 7×7 on each of the 1024 feature maps
- This results in a 1024 output
- Significantly reduces th parameters

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So, this is exactly what I explained so, instead of having this nasty looking connection which would have been 50176 cross 1000, you just take the average from this grid. And, just get a 1024 dimensional vector which results in a much smaller weight matrix at the output; everyone gets this so, this is fine.

(Refer Slide Time: 12:08)



- $12\times$ less parameters than AlexNet
- $2\times$ more computations

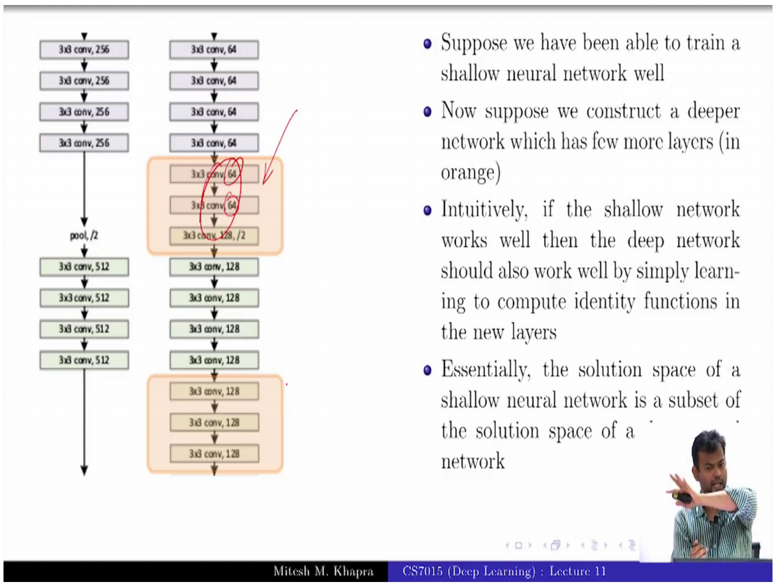
Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So, this has 12 times less parameters than AlexNet, it has 2 times more computations right. So, that is what I meant by the tradeoff so, the number of parameters has reduced significantly of course, a large amount of this savings happen in the fully connected layer. It is not the ingenuity in the inception module which led to the fewer number of parameters that actually leads to more number of parameters right.

But, the reason they could afford more number of parameters in the convolution layers is because, they reduced a lot of parameters in the fully connected layer. Do you get that? So, they did this tradeoff and it has 2 times more computations then AlexNet, but it is still acceptable because you see that there are many layers as compared to AlexNet right. So, let us actually count the number of layers that we had here so, 1 2 3 4 5 6 7 8 9 10 11 12 13 14 right so, it has 14 layers and each of these inception modules is again like split layer right it has this parallel components there.

So, having 2 times more computations was still an acceptable tradeoff. And, it of course, led to much better accuracy as compared to AlexNet or ZF Net or VGG Net right that we had seen in the original trend graph ok. So, now we will go on to the last architecture that we will discuss for image classification which is ResNet.

(Refer Slide Time: 13:24)



- Suppose we have been able to train a shallow neural network well
- Now suppose we construct a deeper network which has few more layers (in orange)
- Intuitively, if the shallow network works well then the deep network should also work well by simply learning to compute identity functions in the new layers
- Essentially, the solution space of a shallow neural network is a subset of the solution space of a network

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So, here is the idea behind ResNet or here is the motivation right. Now, suppose we have been able to train a shallow neural network well, now again my definitions of shallow are relative this is by no means shallow; there are many layers here right. So, you have some 8 layers here.

Now, if I have been able to do this properly; that means, what I mean by that is that using this network at least I was able to reduce the training error to 0 or close to 0 some acceptable value. And, I was able to get some reasonable generalization performance; that means, on the test that I was able to get some reasonable accuracy that is what I mean by I was able to make this network work well.

Now, suppose I add a few layers to this network and I have carefully added some layers in between here and in between here or over there right. Now, intuitively I could argue that if the shallow network was working well right. Then for the deep network this is exist at least one solution which can directly come from the shallow network. Can you tell me what that solution is?

Student: (Refer Time: 14:29).

I want all of you to kind of digest that idea. What the deep network could have done is, I know that this shallow network works why not I just behave like that and I learn these parameters in such a way that I just end up copying from here to here. How many if you get this? Right so, there is a case for the deep network to do at least as well as the shallow network and it could do the same thing at this point. All of you get this idea right?

So, in other words the solution space of the deep network or rather the solution space of the shallow network is actually a subset of the solution space of the deep network. There was one solution for the shallow network which could have been used as it is for the deep network. Of course, for the deep network there are several other solutions because, instead of the identity here you could learn different things there, but at least that one solution exists. So, I should at least if I do use this in practice ex I should expect that this would work as well as the shallow network right. Is that argument fine with everyone?

Student: (Refer Time: 15:29).

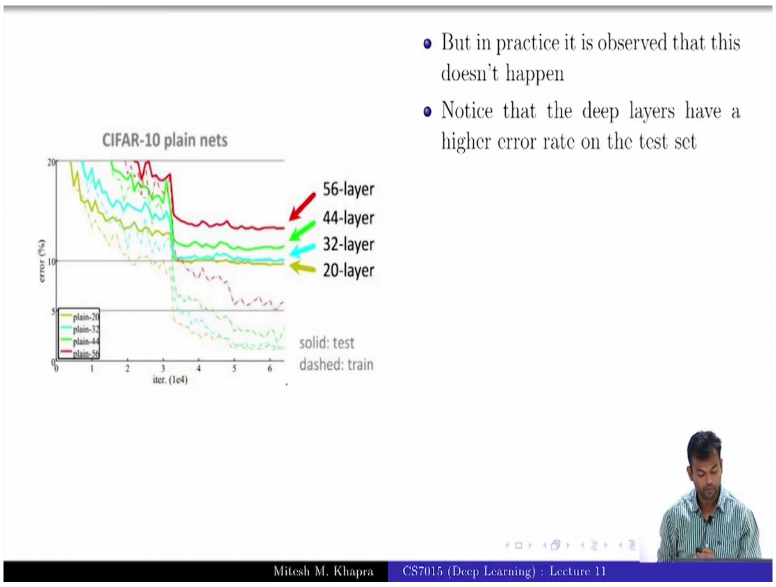
Or which has only one of course.

Student: Yes.

I mean so it those arbitrary things would not work, but here what there it is the for the explanation intuition right you are using some reasonable things. And, you are just trying to make it compatible with whatever you have so far. So, the argument is valid right so, I cannot expect that I had a volume whose depth was 128 and then I suddenly decide to use only 1 filter in the next layer right.

That means, I have compressed everything and now I expect it to be able to recover from there that is not going to happen right. So, that is a fair argument, but the argument which I was trying to make or at least for the illustration purpose is that, if you do reasonable things and that is what people were trying out right. These this is the exact network that someone was trying out and this did not work with well; I will tell you what it is. So, do you get his doubt and my clarification on that is that fine.

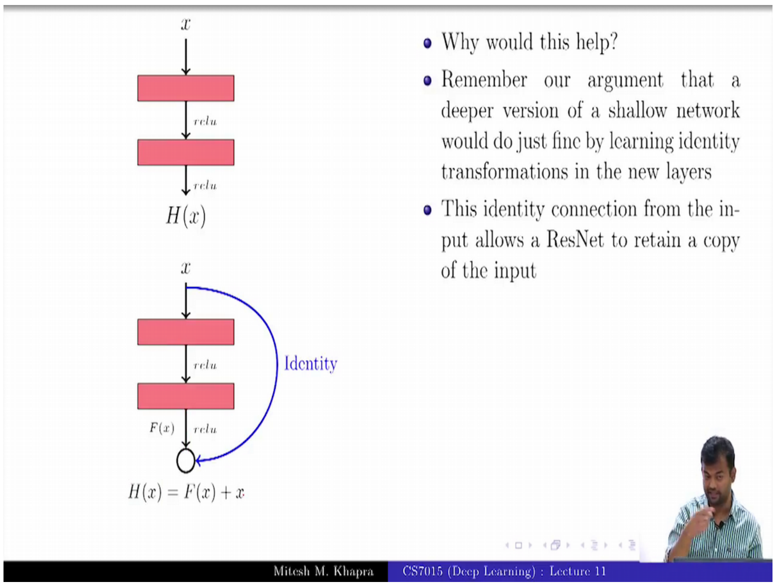
(Refer Slide Time: 16:16)


- But in practice it is observed that this doesn't happen
- Notice that the deep layers have a higher error rate on the test set

So, this is what was happening in practice right. So, you have a 20 layer network or 32 layer network or 44 layer network and a 56 layer network. And, you see that the training error of deeper networks is much higher than the training error of the shallow networks. That means, this argument which I was trying to make that the deep network should at least do as well as the shallow network was not working well in practice right. And, it is if you think about it is not very surprising because, this argument hinged on the fact that it should be able to learn this identity mapping. But, this identity mapping is one of many solutions right.

So, for it to be able to narrow down on that solution, it is easy for you and me to think about it, but for the network it does not have this intuition right, that I can just copy it from there to here. Do you get that? The solution space is really large and like finding that needle in a haystack right, you have these many solutions possible. And, I am trying you to arrive at a solution where you end up with the identity solution is that clear and it is not so, easy for the network to do that everyone gets this? How many of you get this idea?

(Refer Slide Time: 17:25)



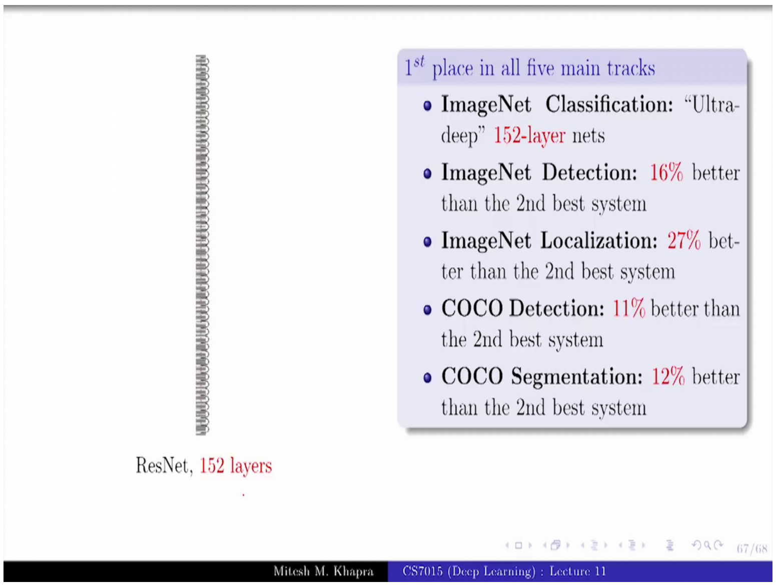
- Why would this help?
- Remember our argument that a deeper version of a shallow network would do just fine by learning identity transformations in the new layers
- This identity connection from the input allows a ResNet to retain a copy of the input

So, why not explicitly try to do something of this sort, where the network can actually learn some kind of an identity function. So, now consider any 2 layers, you know by stack layers I mean this is a convolution layer and followed by a convolution layer right so, these are 2 convolution layers back to back. So, from I what do I actually end up doing here I had a certain input x and I am learning some transformation of x, through these convolution layers right. I am trying to learn x and then I sorry I was given x, I am passing it to convolution layer so, I will run some transformation of x.

And, my argument was that if it could learn to directly copy x here, the deep network should at least work as well as the shallow network. So, why not I explicitly ensure this? So, why not I do this, that in addition to these connections I also explicitly connect x to the output. Do you get this? So, now, what I am trying to do this, is H x is equal to F of x which is the transformation that I learned for x and in addition I also add x. So, what am I doing? I am explicitly feeding it the identity function right. How many if you get the intuition for this? So, what I am trying to do is I have a sense that if I could have transferred this x as it is across layers, then there is a chance that I should be able to do as well as the shallow network right.

So, now I am going to explicitly ensure that that you learn these transformations, but I will also feed you x at every stage at a reasonable time right. So, these are known as skip connections. So, after every 2 layers I will feed back the x or you could try after every 3 layers I will feed back the x. So, I am trying to maintain the original copy of x after every interval fine. So, why would this help so, this follows back from our argument and it is the same thing which I said before.

(Refer Slide Time: 19:17)


1^{st} place in all five main tracks

- **ImageNet Classification:** "Ultra-deep" 152-layer nets
- **ImageNet Detection:** 16% better than the 2nd best system
- **ImageNet Localization:** 27% better than the 2nd best system
- **COCO Detection:** 11% better than the 2nd best system
- **COCO Segmentation:** 12% better than the 2nd best system

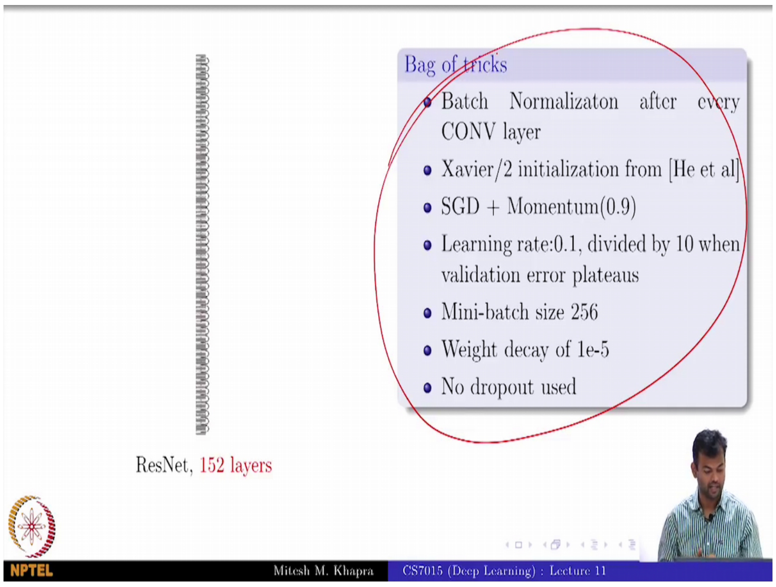
ResNet, 152 layers

67/68

Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So, using this idea of using these skip connections, these authors were able to train a really deep network of 152 layers right. And, this gave on multiple vision challenges right one being ImageNet, it gave 16 percent better results and the best network. And, this is the one which reads that near human performance. Then ImageNet localization is another challenge, where you need to localize the object. So, there they get 27 percent better than the best results and there are these bunch of other vision tasks detection segmentation and all of them. And, in all of them this significantly outperformed the best system using a very deep network. Of course, the downside is that you have a very deep network; it will take its own time to train and so on. But of course, if you have a Microsoft or Google you can afford to do that.

(Refer Slide Time: 20:08)


Bag of tricks

- Batch Normalizaton after every CONV layer
- Xavier/2 initialization from [He et al]
- SGD + Momentum(0.9)
- Learning rate:0.1, divided by 10 when validation error plateaus
- Mini-batch size 256
- Weight decay of 1e-5
- No dropout used

ResNet, 152 layers

NPTEL Mitesh M. Khapra CS7015 (Deep Learning) : Lecture 11

So, that is the current theme right I mean the one with the largest computational resources wins everything right. So, and they also are there is some other bag of tricks which is not I mean it is not very difficult to understand. So, they used batch normalization every after every conversation layer. Have you heard of batch normalization ever in your life? Ok good, they used Xavier by 2 initialization, ever heard of that? Xavier by 2 was the same as?

Student: He initialization.

He initialization right, then they use SGD not any of the fancy Adam or Adagrad or anything with a momentum of 0.9. Learning rate was set to 0.1 and divided by 10 whenever the validation error plateaus. The mini batch size was 256; they use a weight decay of 1 ht. What is weight decay? Weight decay is in the context of which regularization?

Student: l 2.

L 2 and what does this mean, weight decay of 1 e raised to minus 5.

Student: Lambda (Refer Time: 21:05).

The lambda was set to 1 e raised to minus 5; all of you remember these things right. We did it in some previous course, in some previous life and no drop out was used right so, since I have this here I will just say something more on this. So, in your reading papers on deep learning right focus on the experimental section, where all these hyper parameters are described so, these are known as the hyper parameters. These are not related to the parameters of the model, these are related to hyper parameters which is what the batch size is whether you used l 2 regularization. What was the learning rate? What was the optimization and so on?

So, turns out in many cases if you do not stick to this you will not be able to reproduce the results of the paper right. So, you might be wondering that this network I understand, this is just 152 layers and I can just keep adding skip connections; I can easily code this up. But, I am not getting the same results as the original authors of the paper.

So, this is where you need to dig up them right, you need to look at the experimental section where most good authors provide these details of how they have trained the network. How many epox that they use, what was the patients set and all that. If you follow those the chances of reproducing are much higher still not guaranteed, but definitely much higher ok. So, that is where we end the lecture on convolutional neural networks and ImageNet classification.