Week 2

Computer Vision

What we have learned in this week's videos?

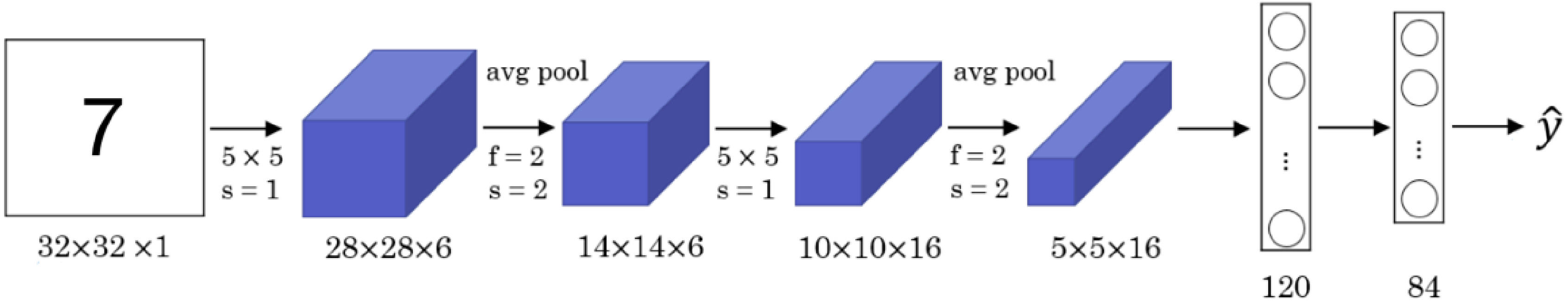
- CNN Architectures
- CPU vs GPU
- Transfer learning

Session agenda

- CNN Architectures
- Spectrum of depth
- Network in network
- Transfer learning
- Case study
- Questions

CNN Architectures

LeNet-5



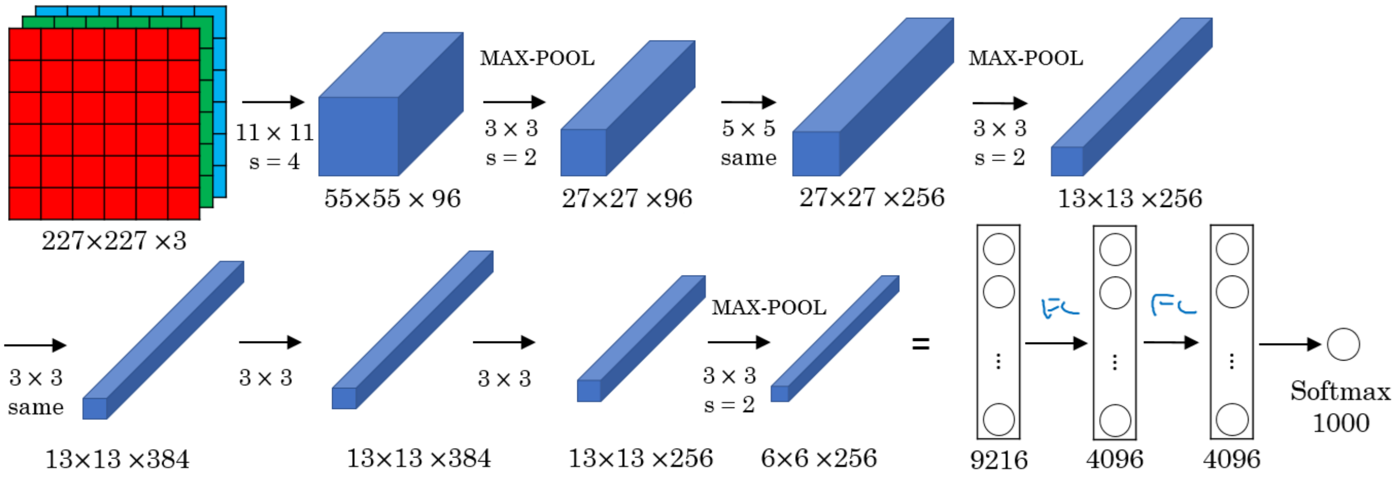
- The goal for this model was to identify handwritten digits in a 32x32x1 gray image.
- This model was published in 1998. The last layer wasn't using softmax back then.
- It has 60k parameters.
- The dimensions of the image decreases as the number of channels increases.
- Conv ==> Pool ==> Conv ==> Pool ==> FC ==> FC ==> softmax this type of arrangement is quite common.
- The activation function used in the paper was Sigmoid and Tanh. Modern implementation uses RELU in most of the cases.

Paper source: *stanford.edu*

LeNet

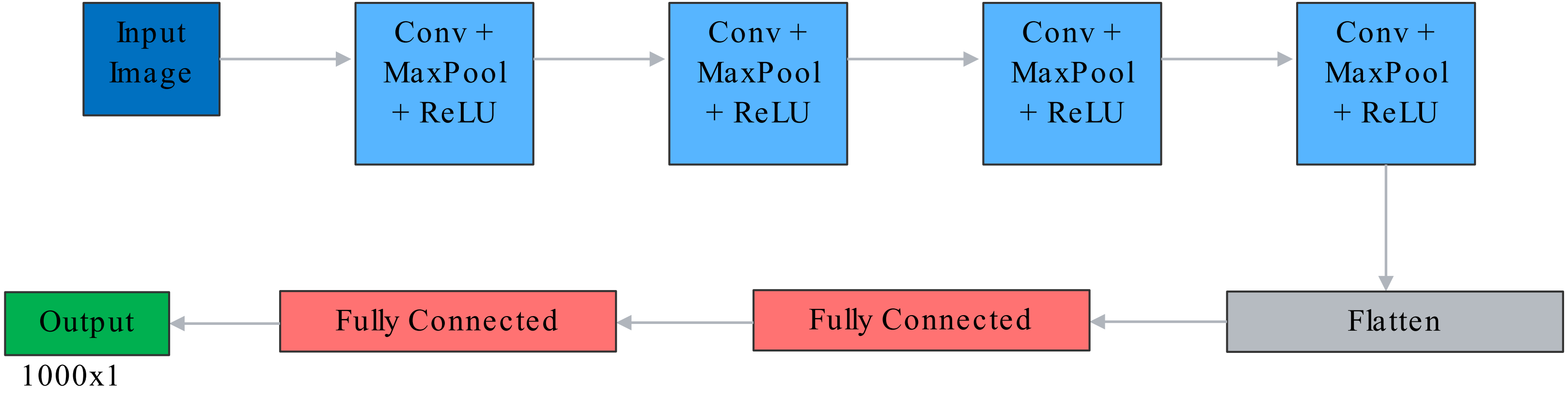
- 10 way neural network classifier
- Handwritten digits as an input
- Tolerant of various transformations like rotation and scale
- Was used by banks to recognize handwritten numbers on digitized checks
- 4 weight layers

AlexNet



- The goal for the model was the ImageNet challenge which classifies images into 1000 classes.
- Has 60 Million parameter compared to 60k parameter of LeNet-5.
- It used the RELU activation function.
- This paper convinced the computer vision researchers that deep learning is so important.
- Conv => Max-pool => Conv => Max-pool => Conv => Conv => Conv => Max-pool ==> Flatten ==> FC ==> FC ==> Softmax

AlexNet

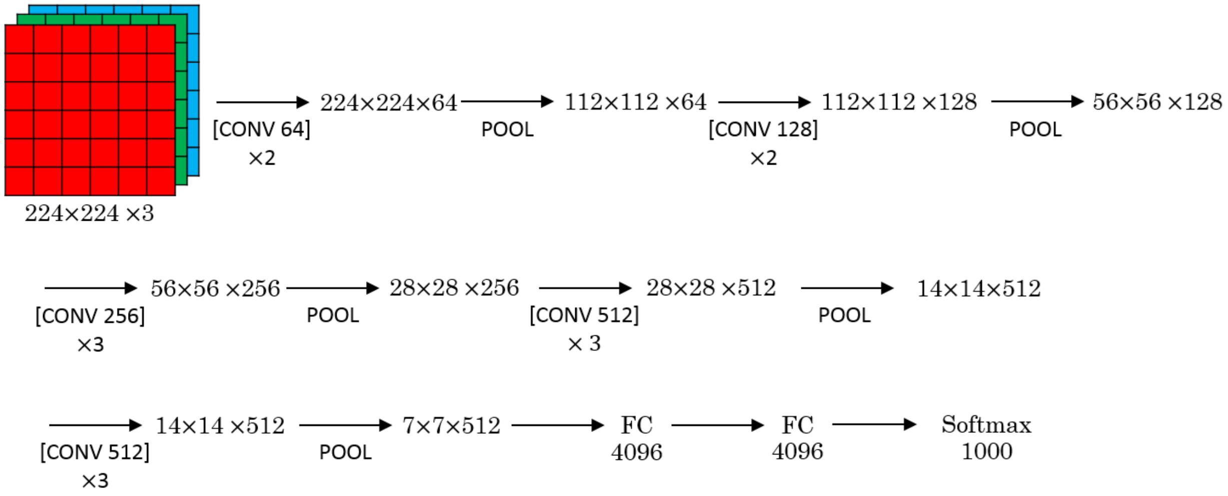


Input Image
Conv + MaxPool + ReLU
Conv + MaxPool + ReLU
Conv + MaxPool + ReLU
Conv + MaxPool + ReLU
Flatten
Fully Connected
Fully Connected
Output
1000x1

What made AlexNet successful?

1. AlexNet architecture
2. Deep dive block by block
3. Overlapping max pooling
4. ReLu
5. Dropouts
6. Cropping
7. Data Augmentation
8. Inference Augmentation

VGG-16



- A modification for AlexNet.
- Instead of having a lot of hyperparameters lets have some simpler network.
- Focus on having only these blocks:
  - CONV = 3 X 3 filter, s = 1, same
  - MAX-POOL = 2 X 2 , s = 2

Different VGG architectures

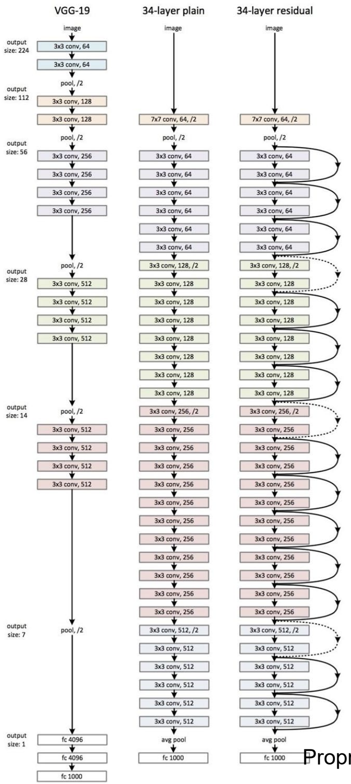
ConvNet Configuration					
A 11 weight layers	A-LRN 11 weight layers	B 13 weight layers	C 16 weight layers	D 16 weight layers	E 19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					

Architectures used in the VGG work

VGG-16

- This network is large even by modern standards. It has around 138 million parameters.
    - Most of the parameters are in the fully connected layers.
- It has a total memory of 96MB per image for only forward propagation!
    - Most memory are in the earlier layers.
- Number of filters increases from 64 to 128 to 256 to 512. 512 was made twice.
- Pooling was the only one who is responsible for shrinking the dimensions.
- There are another version called VGG-19 which is a bigger version. But most people uses the VGG-16 instead of the VGG-19 because it does the same.
- VGG paper is attractive it tries to make some rules regarding using CNNs.
- https://arxiv.org/abs/1409.1556

Residual Networks (ResNets)

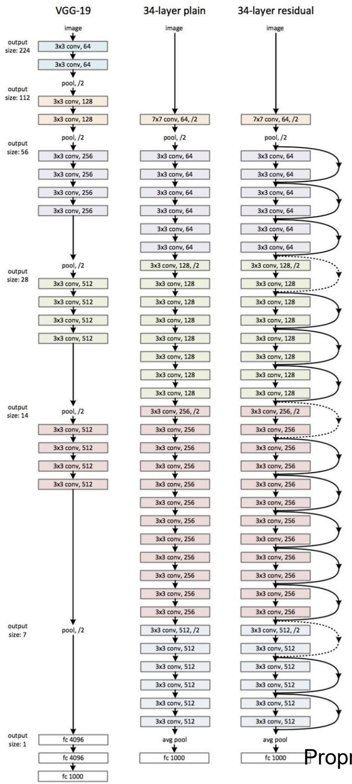


- Very, very deep NNs are difficult to train because of vanishing and exploding gradients problems.
- In this section we will learn about skip connection which makes you take the activation from one layer and suddenly feed it to another layer even much deeper in NN which allows you to train large NNs even with layers greater than 100.

- **Residual block**
  - ResNets are built out of some Residual blocks.
  - They add a shortcut/skip connection before the second activation.
  - The authors of this block find that you can train a deeper NNs using stacking this block.

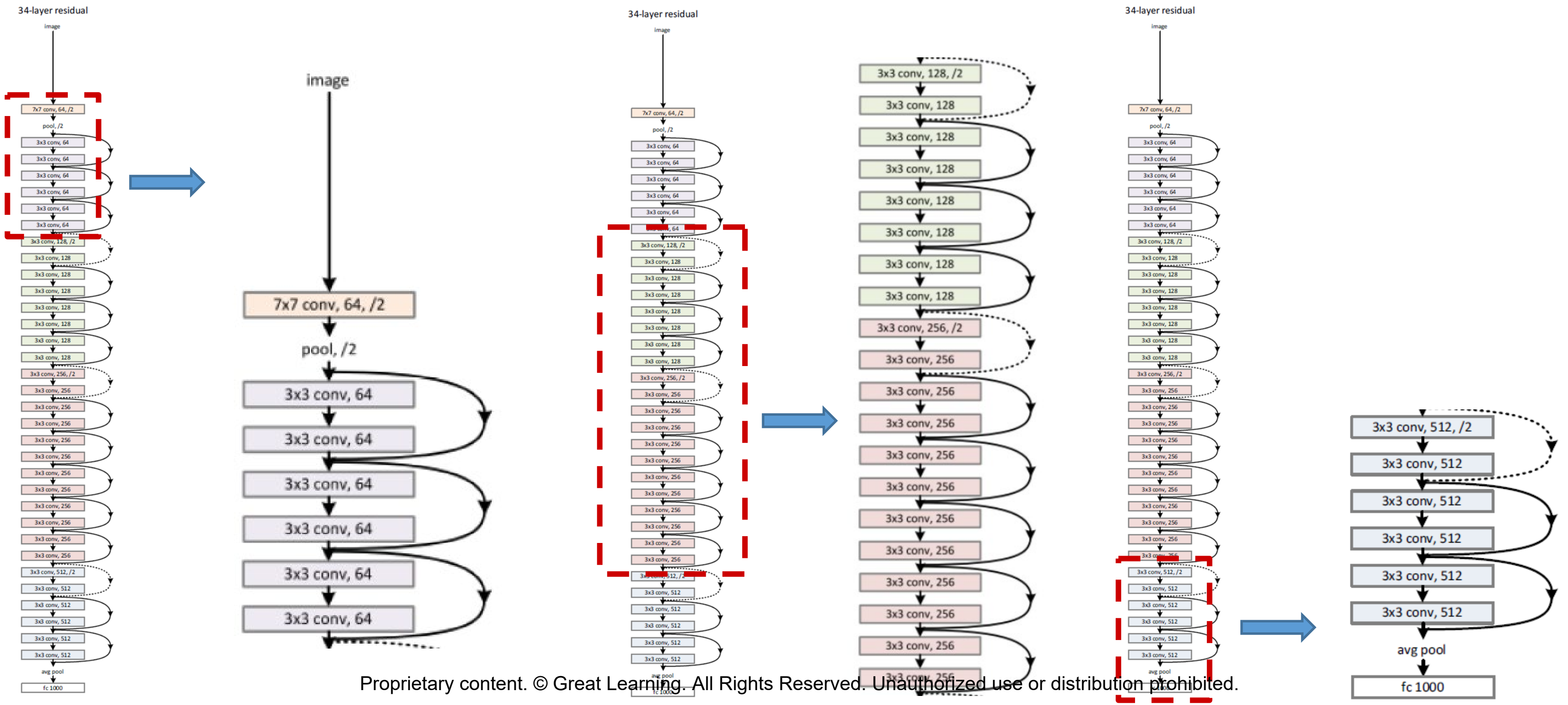
- **Residual Network**
  - Are a NN that consists of some Residual blocks.
  - These networks can go deeper without hurting the performance. In the normal NN - Plain networks - the theory tell us that if we go deeper we will get a better solution to our problem, but because of the vanishing and exploding gradients problems the performance of the network suffers as it goes deeper.

ResNet-34



- The architecture of **ResNet-34**
- All the 3x3 Conv are same Convs.
- Keep it simple in design of the network.
- spatial size /2 => # filters x2
- No FC layers, No dropout is used.
- Two main types of blocks are used in a ResNet, depending mainly on whether the input/output dimensions are same or different. You are going to implement both of them.
- The dotted lines is the case when the dimensions are different. To solve then they down-sample the input by 2 and then pad zeros to match the two dimensions. There's another trick which is called bottleneck.

Basic architecture: ResNet-34



Spectrum of depth

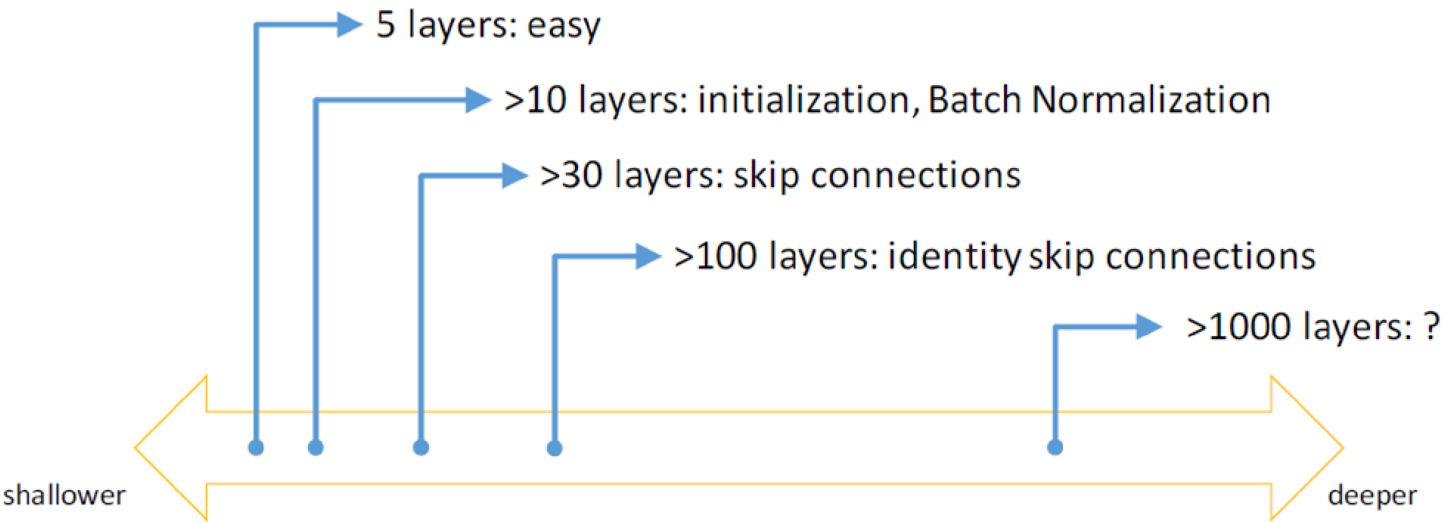


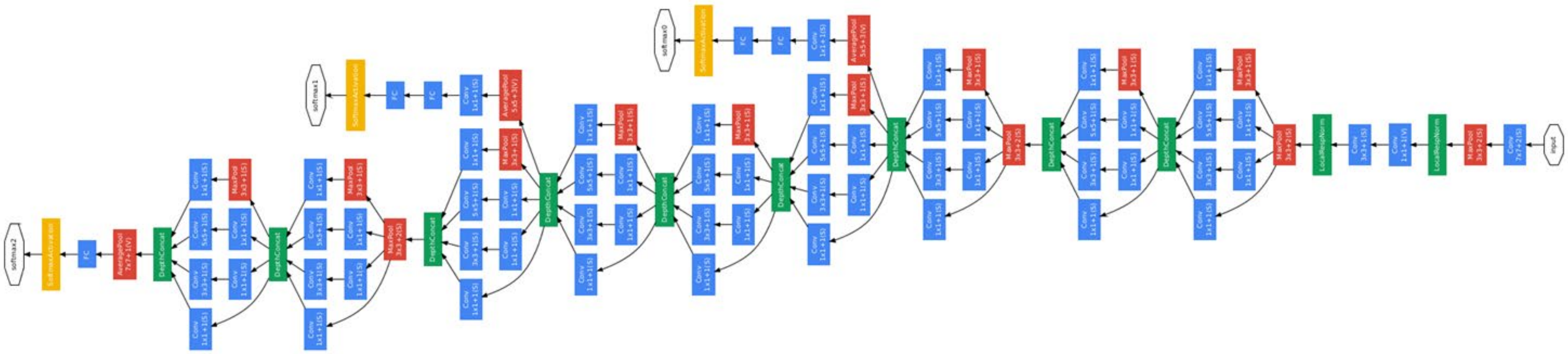
Image source:[icml.cc]

greatlearning
Power Ahead

Network in network & 1 x 1 convolutions

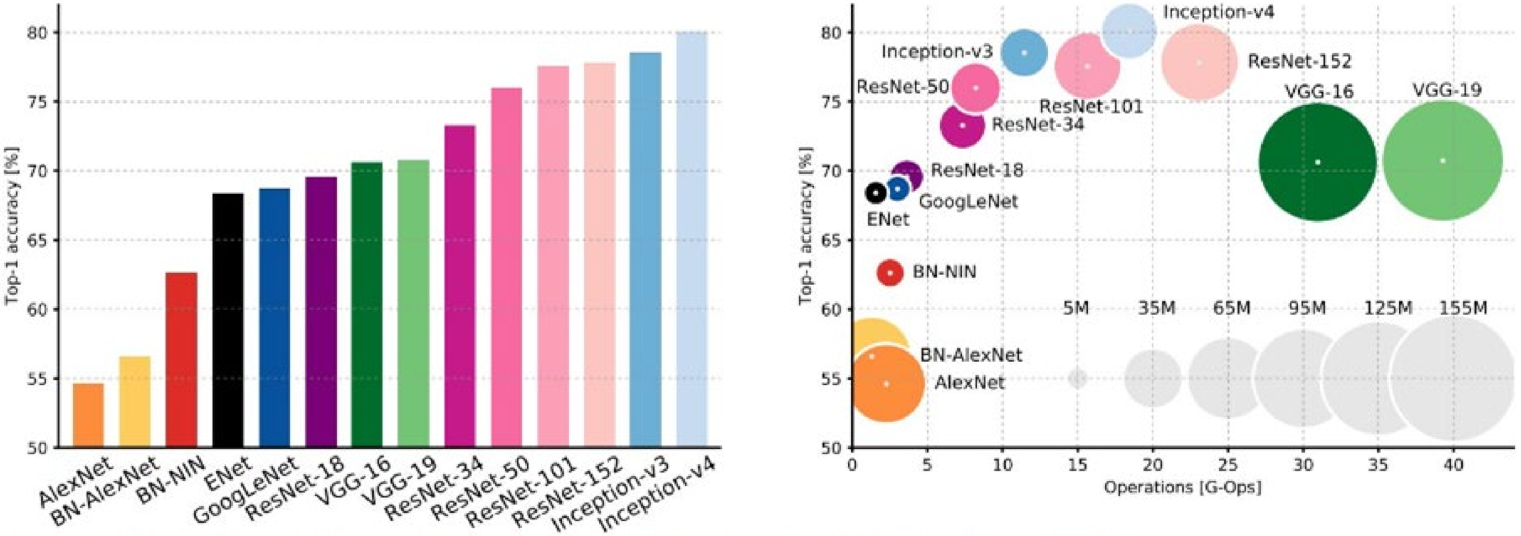
- A 1 x 1 convolution - We also call it Network in Network- is so useful in many CNN models.
- It has been used in a lot of modern CNN implementations like ResNet and Inception models.
- A 1 x 1 convolution is useful when:
  - We want to shrink the number of channels. We also call this feature transformation. Shrinking can save a lot of computations.
- If we have specified the number of 1 x 1 Conv filters to be the same as the input number of channels then the output will contain the same number of channels. Then the 1 x 1 Conv will act like a non linearity and will learn non linearity operator.

Inception network (GoogleNet)



- The inception network consist of concatenated blocks of the Inception module.
- The name inception was taken from a meme image which was taken from Inception movie.
- Some times a Max-Pool block is used before the inception module to reduce the dimensions of the inputs.
- There are a 3 Sofmax branches at different positions to push the network toward its goal. and helps to ensure that the intermediate features are good enough to the network to learn and it turns out that softmax0 and sofmax1 gives regularization effect.

State of the art CNN architectures



An Analysis of Deep Neural Network Models for Practical Applications, 2017.

Transfer learning

greatlearning
Power Ahead

Transfer learning

- If you are using a specific NN architecture that has been trained before, you can use this pretrained parameters/weights instead of random initialization to solve your problem.
- It can help you boost the performance of the NN.
- The pretrained models might have trained on a large datasets like ImageNet, Ms COCO, or pascal and took a lot of time to learn those parameters/weights with optimized hyperparameters. This can save you a lot of time.
- If you have enough data, you can fine tune all the layers in your pretrained network but don't random initialize the parameters, leave the learned parameters as it is and learn from there.

Why transfer learning?

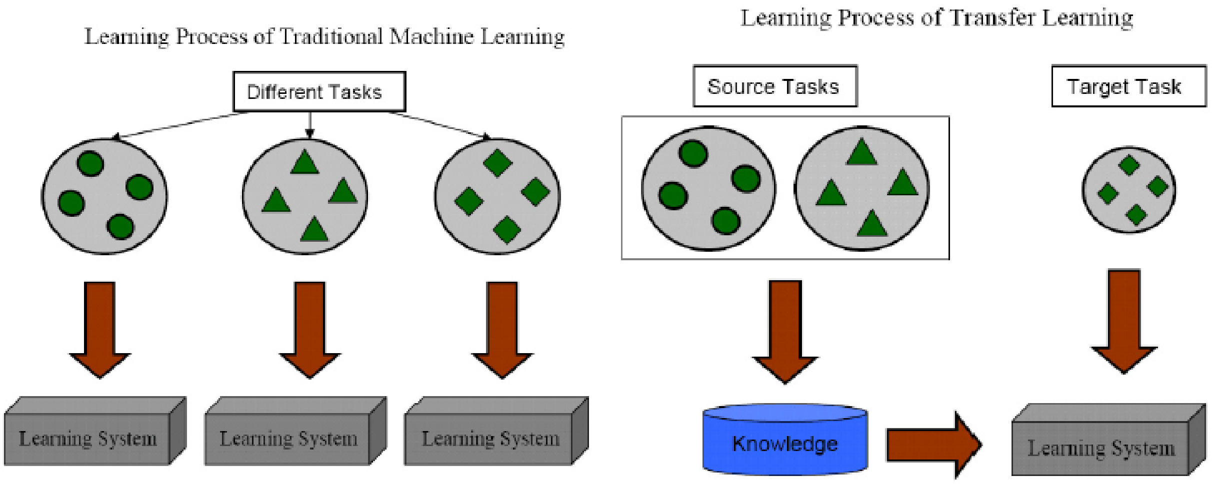
Given a new application, one looks at opportunities for re-using knowledge (e.g. architectures and weights) from similar learning problems which were trained with large amounts of data

Transfer Learning!

Humans are great at transfer learning

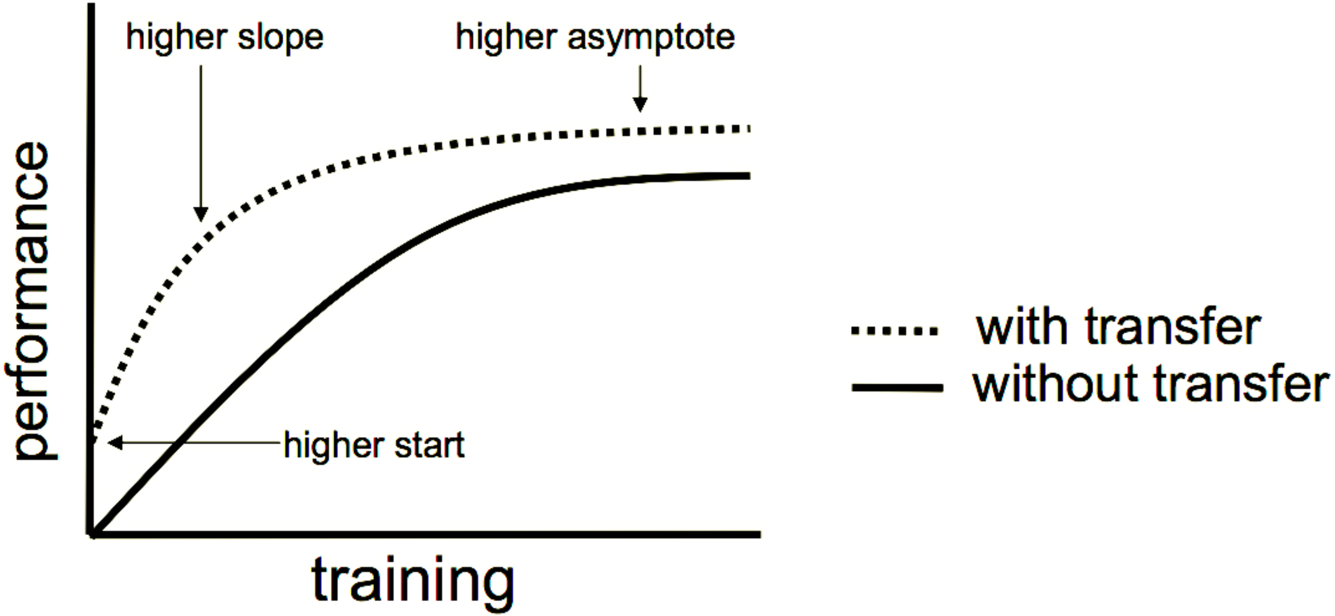
(e.g. Bicycle bike, Tennis, Badminton, Language skills)

Transfer learning: how it works?



(a) Traditional Machine Learning (b) Transfer Learning

Advantages of transfer learning



Summary

Summary

- We learned about various CNN Architectures.
- LeNet, AlexNet, ResNet, GoogleNet.
- We now know what is spectrum of depth.
- Network in network and 1 x 1 convolutions.
- We learned about transfer learning and it's applications.

greatlearning
Power Ahead

Thank you! :)

Questions are always welcome