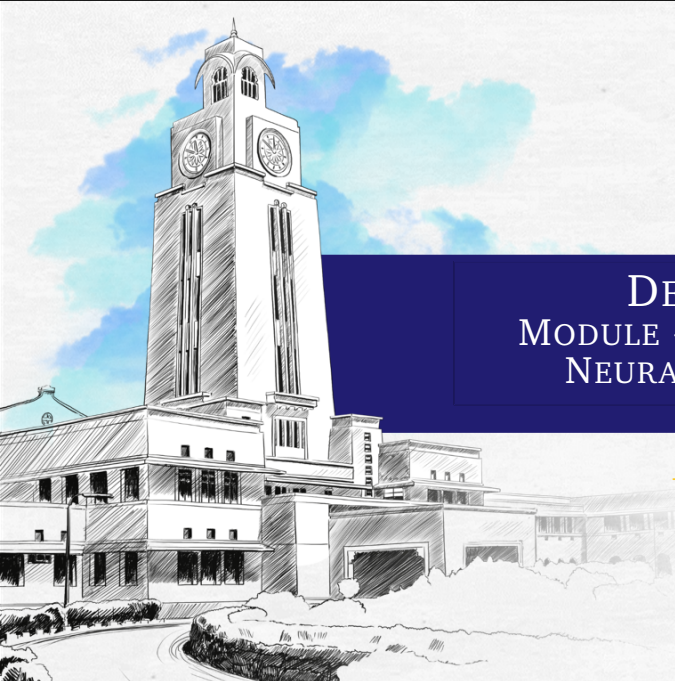


BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Deep Learning
Module # 5 : Convolutional Neural Network [CNN]

Seetha Parameswaran
Asst Prof, BITS Pilani

The instructor is gratefully acknowledging the authors who made their course materials freely available online.

In This Segment

LeNet 5

- Every convolutional layer includes three parts: convolution, pooling, and nonlinear activation functions.
- Using convolution to extract spatial features.
- Conv filters were 5x5, applied at stride 1.
- Subsampling average pooling layer. Subsampling (Pooling) layers were 2x2 applied at stride 2.
- tanh activation function.
- Using MLP as the last classifier.
- Architecture is [CONV-POOL-CONV-POOL-FC-FC]
- Sparse connection between layers to reduce the complexity of computation.

LeNet-5

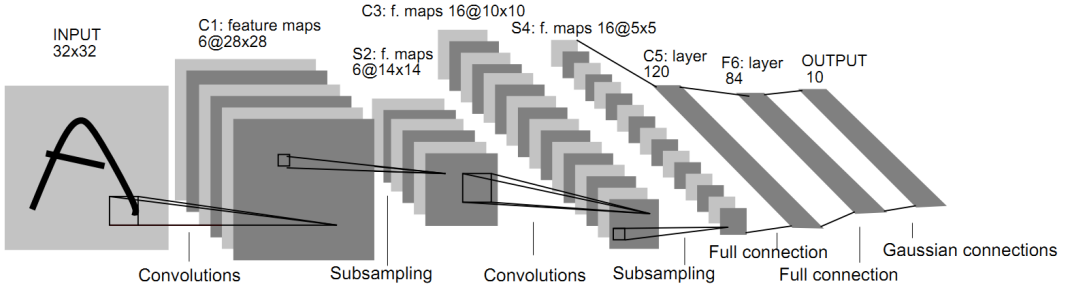
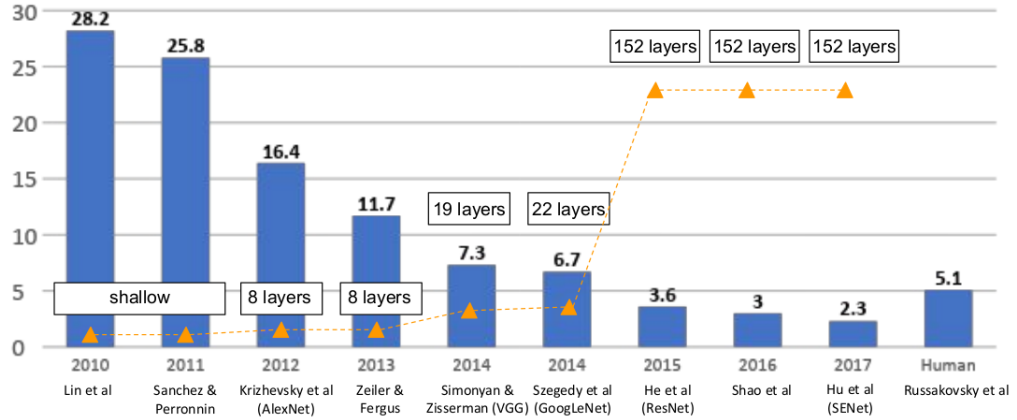


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

ImageNet Large Scale Visual Recognition



ImageNet Data

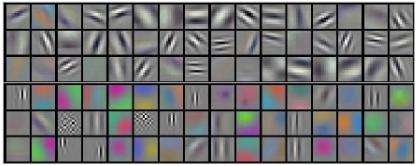


Figure 3: 96 convolutional kernels of size 11*11*3 learned by the first convolutional layer on the 224*224*3 input images. The top 48 kernels were learned on GPU 1 while the bottom 48 kernels were learned on GPU

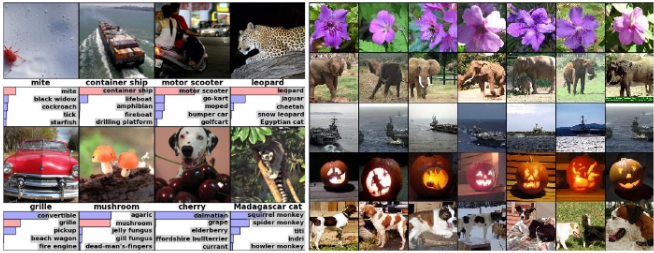
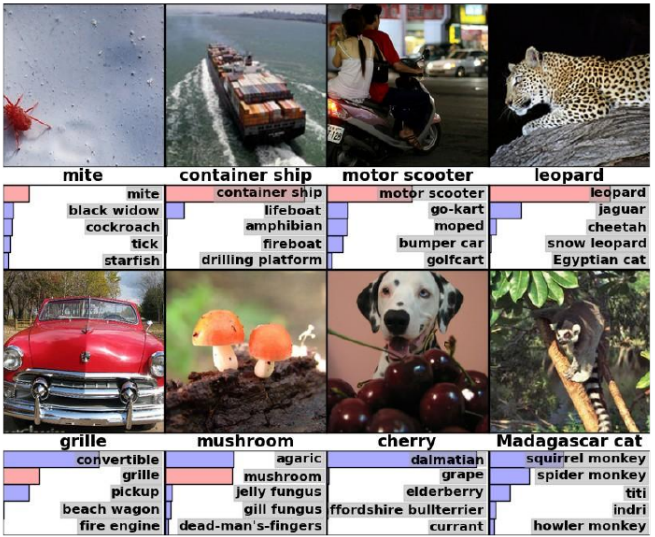


Figure 4: **(Left)** Eight ILSVRC-2010 test images and the five labels considered most probable by our model. The correct label is written under each image, and the probability assigned to the correct label is also shown with a red bar (if it happens to be in the top 5). **(Right)** Five ILSVRC-2010 test images in the first column. The remaining columns show the six training images that produce feature vectors in the last hidden layer with the smallest Euclidean distance from the feature vector for the test image.

ImageNet Data



The ImageNet set that was used has ~1.2 million images and 1000 classes

Accuracy is measured as top-5 performance: Correct prediction if the true label matches one of the top 5 predictions of the model

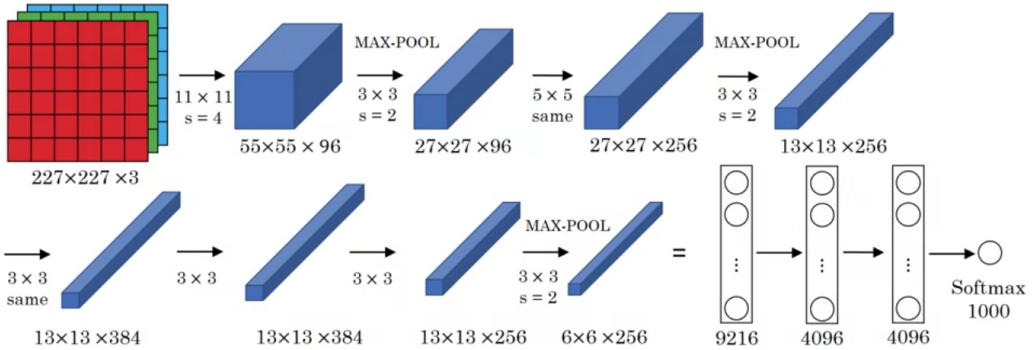
AlexNet

- AlexNet competed in the ImageNet Large Scale Visual Recognition Challenge on September 30, 2012.[3] The network achieved a top-5 error of 15.3%, more than 10.8 percentage points lower than that of the runner up.
- Details/Retrospectives:
  - first use of ReLU
  - used Local Response Normalisation (LRN) layers (not common anymore)
  - Approx. 60 million parameters are learned.
  - heavy data augmentation
  - dropout 0.5
  - batch size 128
  - SGD Momentum 0.9
  - Learning rate $1e-2$, reduced by 10 manually when val accuracy plateaus
  - L2 weight decay $5e-4$
  - 7 CNN ensemble: $18.2\% -> 15.4\%$

AlexNet Architecture

227x227x3 INPUT
55x55x96 CONV1 : 96 11x11 filters at stride 4, pad 0
27x27x96 MAX POOL1 : 3x3 filters at stride 2
27x27x96 NORM1 : Normalization layer
27x27x256 CONV2 : 256 5x5 filters at stride 1, pad 2
13x13x256 MAX POOL2 : 3x3 filters at stride 2
13x13x256 NORM2 : Normalization layer
13x13x384 CONV3 : 384 3x3 filters at stride 1, pad 1
13x13x384 CONV4 : 384 3x3 filters at stride 1, pad 1
13x13x256 CONV5 : 256 3x3 filters at stride 1, pad 1
6x6x256 MAX POOL3 : 3x3 filters at stride 2
4096 FC6 : 4096 neurons
4096 FC7 : 4096 neurons
1000 FC8 : 1000 neurons (class scores)

AlexNet



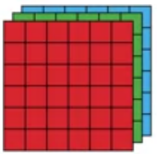
[Krizhevsky et al., 2012. ImageNet classification with deep convolutional neural networks]

Andrew Ng

VGG 16

- VGG stands for Visual Geometry Group with 16 Layers
- Plug and play in Caffe
- Deeper the better
- Details
  - ILSVRC'14 2nd in classification, 1st in localization
  - Similar training procedure as Krizhevsky 2012
  - Approx. 138 million parameters are learned.
  - No Local Response Normalisation (LRN)
  - Use VGG16 or VGG19 (VGG19 only slightly better, more memory)
  - Use ensembles for best results
  - All convolutions with a 3×3 kernel
  - All max-pooling layers with a 2×2 kernel
  - FC7 features generalize well to other tasks

VGG 16



CONV = 3×3 filter, s = 1, same MAX-POOL = 2×2 , s = 2

224×224 ×3 → [CONV 64] ×2 → 224×224×64 → POOL → 112×112 ×64 → [CONV 128] ×2 → 112×112 ×128 → POOL → 56×56 ×128

→ [CONV 256] ×3 → 56×56 ×256 → POOL → 28×28 ×256 → [CONV 512] × 3 → 28×28 ×512 → POOL → 14×14×512

→ [CONV 512] ×3 → 14×14 ×512 → POOL → 7×7×512 → FC 4096 → FC 4096 → Softmax 1000

[Simonyan & Zisserman 2015. Very deep convolutional networks for large-scale image recognition]

Andrew Ng

Network in Network

- A convolution kernel can be thought of as a generalized linear model (GLM).
- Using a sophisticated non-linear function approximator like MLP, enhance the abstraction ability of the local model.
- Replace GLM by **micro network**.
- Replace the MLP micro structure via convolutions.
- Replace fully connected layers in the last layers by global average pooling.

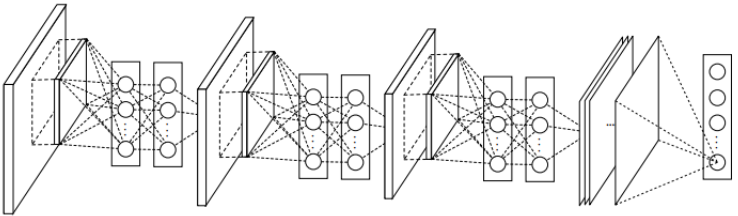
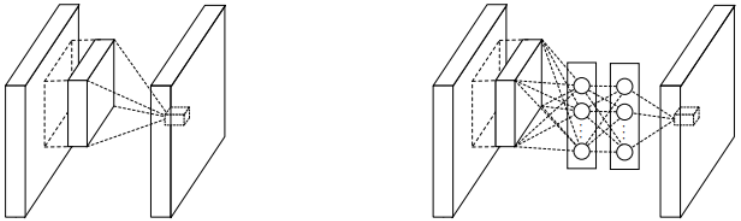


Figure 2: The overall structure of Network In Network. In this paper the NINs include the stacking of three mlpconv layers and one global average pooling layer.

Network in Network

Using micro-networks allow to extract more sophisticated features and may need fewer extractors and can avoid learning too simple or redundant features.



(a) Linear convolution layer (b) Mlpconv layer

Figure 1: Comparison of linear convolution layer and mlpconv layer. The linear convolution layer includes a linear filter while the mlpconv layer includes a micro network (we choose the multilayer perceptron in this paper). Both layers map the local receptive field to a confidence value of the latent concept.

Network in Network

Fully connected layers have a lot of learnable parameters and may cause overfiting.
Replace FC by global average pooling for better generalization.

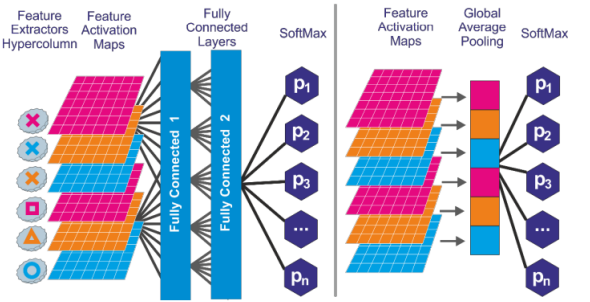


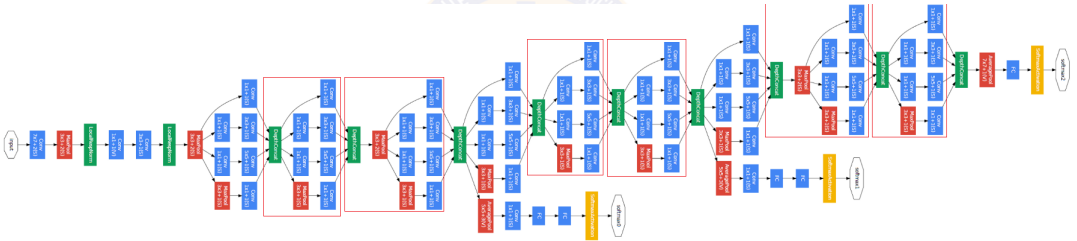
Figure 16: Global average pooling layer replacing the fully connected layers. The output layer implements a Softmax operation with $p_1, p_2, \ldots, p_n$ the predicted probabilities for each class.

Figure Source: Singh, Anshuman Vikram. "Content-based image retrieval using deep learning." (2015)

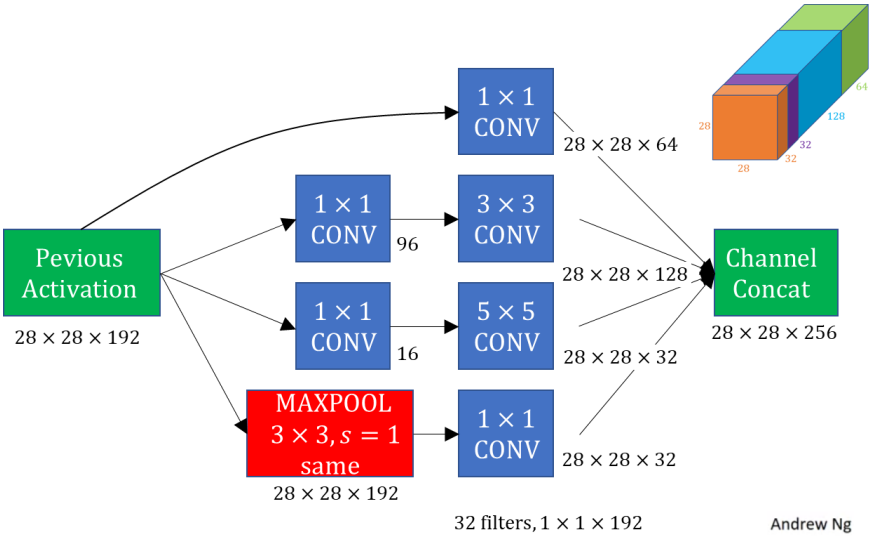
GoogLeNet / Inception Net

- ILSVRC'14 classification winner (6.7% top 5 error)
- 22 layers with weights
- Only 5 million parameters (12x less than AlexNet and 27x less than VGG-16)
- Inception Module - convolutional "blocks" – efficient
  - Design a good local network topology (network within a network) and then stack these modules on top of each other.
- Linear layers at the end
- Max pooling in between, multiple Conv layers between pooling
- Great ideas for data augmentation
- Deeper networks, with computational efficiency
- No FC layers.
- After the last convolutional layer, a global average pooling layer is used that spatially averages across each feature map, before final FC layer.

Inception Net



Inception Net Module



Inception Net Module

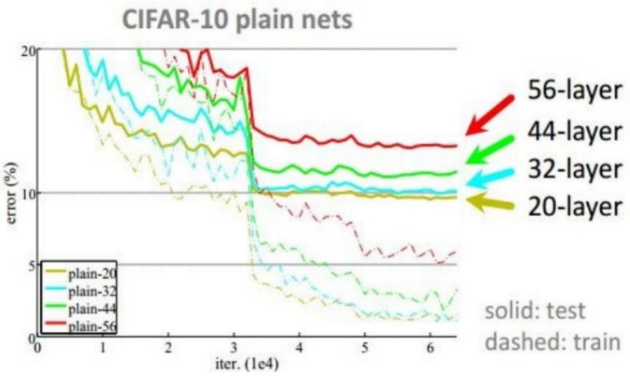
- Apply parallel filter operations on the input from previous layer.
- Multiple receptive field sizes for convolution (1x1, 3x3, 5x5) and Pooling operation (3x3).
- Concatenate all filter outputs together channel-wise.
- Very expensive compute
- 1x1 convolutions to reduce feature channel size.
  - Assume an input of WxHx64.
  - Each filter has size 1x1x64, and performs a 64-dimensional dot product.
  - Preserves spatial dimensions, reduces depth.
  - Projects depth to lower dimension (combination of feature maps).

Residual Net (ResNet)

- Very deep networks using residual connections
- 152-layer model for ImageNet
- ILSVRC'15 classification winner (3.57% top 5 error)
- Swept all classification and detection competitions in ILSVRC'15 and COCO'15.
- Add residual connections between convolution blocks
- Easy gradient flow for deep models
- Enable very deep models (from 20 layers to 150 and even 1000 layers!)
- No linear layers except classifier.
- Solve the problems of vanishing gradients using Residual connections.

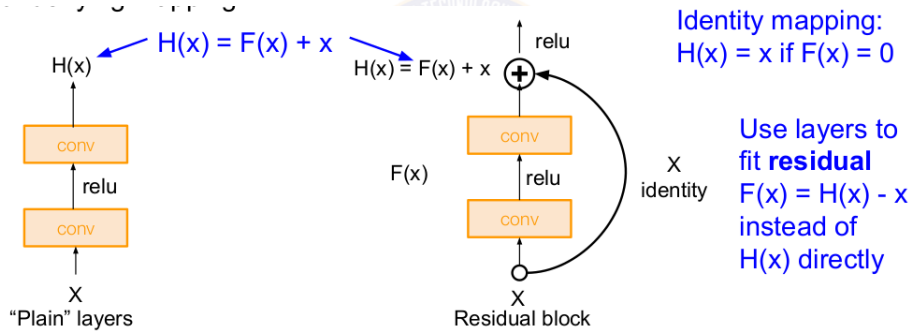
ResNet Block

- What happens when we continue stacking deeper layers on a convolutional neural network?
- The deeper model performs worse, but it's not caused by overfitting.

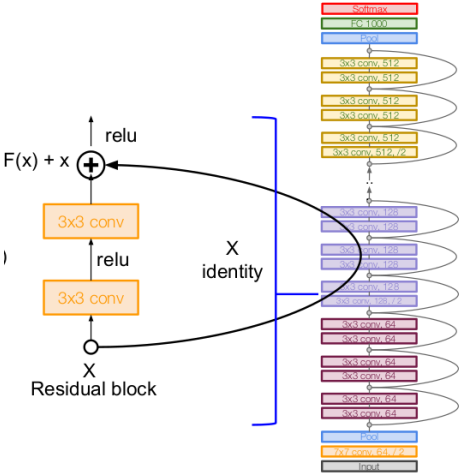


ResNet Block

- Solution: Use network layers to fit a residual mapping instead of directly trying to fit a desired underlying mapping.



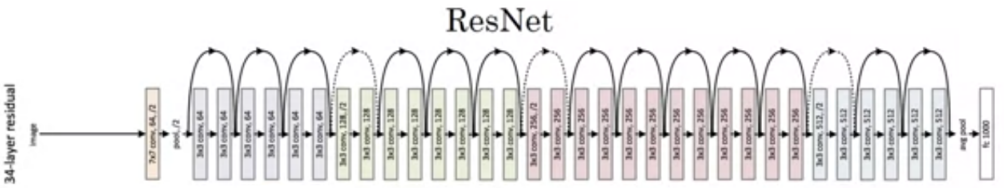
ResNet


relu
F(x) + x
3x3 conv
relu
3x3 conv
X
Residual block
X
identity
Softmax
FC 1000
Pool
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
Pool
7x7 conv, 64, /2
Input

ResNet Modules

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

ResNet



[He et al., 2015. Deep residual networks for image recognition]

Andrew Ng

ResNet architecture

- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2. Reduce the activation volume by half.
- Additional conv layer at the beginning (stem).
- No FC layers at the end (only FC 1000 to output classes)
- Total depths of 18, 34, 50, 101, or 152 layers for ImageNet.
- Training ResNet in practice:
  - Batch Normalization after every CONV layer
  - Xavier 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

Efficacy of ResNet

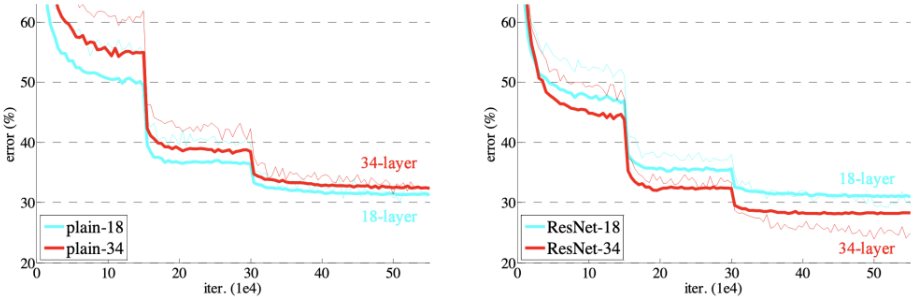
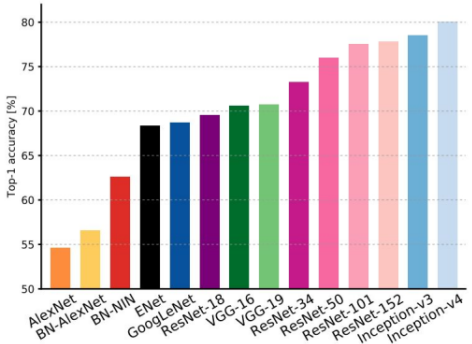
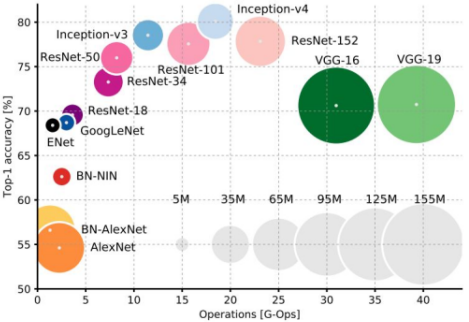


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

https://arxiv.org/abs/1512.03385

Comparing complexity





Alfredo Canziani, Adam Paszke, Eugenio Culurciello, An Analysis of Deep Neural Network Models for Practical Applications, 2017

Comparing complexity

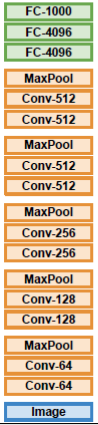
- Inception-v4: Resnet + Inception – more accuracy
- VGG: most parameters, most operations
- GoogLeNet: most efficient
- AlexNet: Smaller compute, still memory heavy, lower accuracy
- ResNet: Moderate efficiency depending on model, highest accuracy

In This Segment

Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014

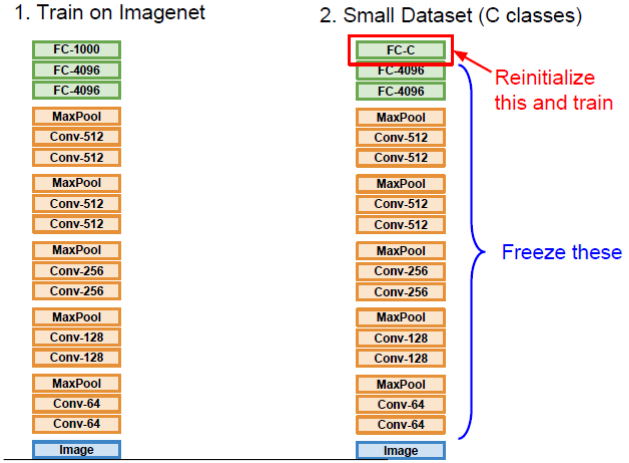
1. Train on Imagenet



Transfer Learning

Transfer Learning with CNNs

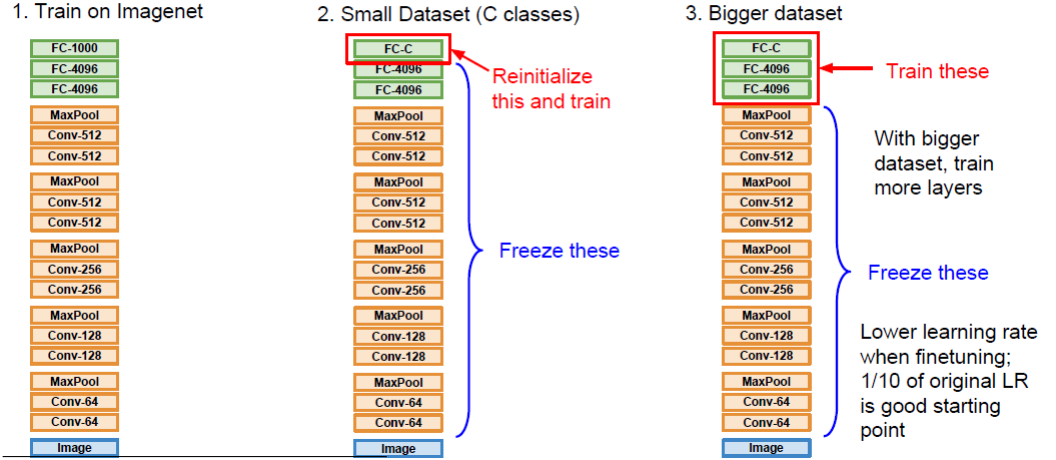
Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014



Transfer Learning

Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014



Transfer Learning for Image Data

- Use a deep learning model that is pre-trained on large dataset like ImageNet or MS Coco.
- Oxford VGG Model
- Google Inception Model
- Microsoft ResNet Model

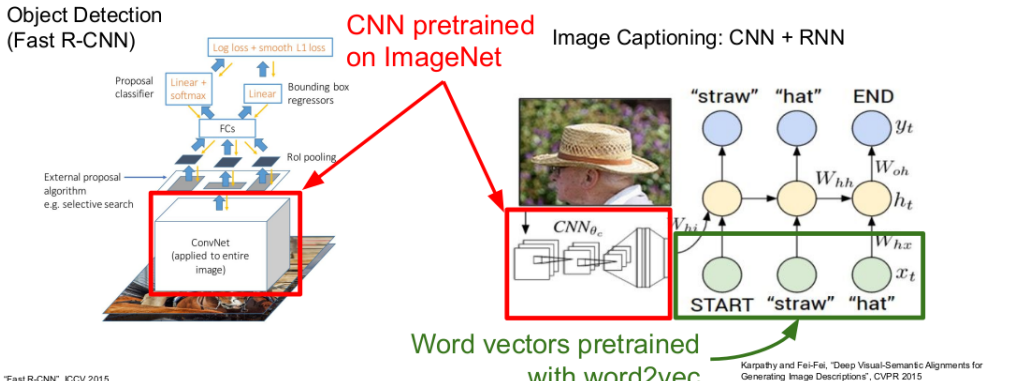
Transfer Learning for Text Data

- Embedding is the mapping of words to a high-dimensional continuous vector space where different words with similar meanings have similar vector representations.
- Google's word2vec Model
- Stanford's GloVe Model
- FastText
- Gensim

Transfer Learning - when to use?

- You need a lot of a data if you want to train/use CNNs / RNNs.
- Task A and Task B have the same type of input. Eg: Input is images for both tasks.
- We have lot of data for training Task A and relatively low data for training in Task B.
- Low level features obtained from Task A could be more helpful for learning Task B.

Transfer Learning Example



Girshick, "Fast R-CNN", ICCV 2015
Figure copyright Ross Girshick, 2015. Reproduced with permission.

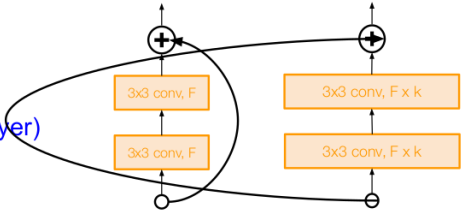
Karpathy and Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015
Figure copyright IEEE, 2015. Reproduced for educational purposes.

In This Segment

Wide Residual Networks

[Zagoruyko et al. 2016]

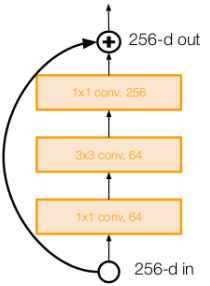
- Argues that residuals are the important factor, not depth
- User wider residual blocks (F x k filters instead of F filters in each layer)
- 50-layer wide ResNet outperforms 152-layer original ResNet
- Increasing width instead of depth more computationally efficient (parallelizable)

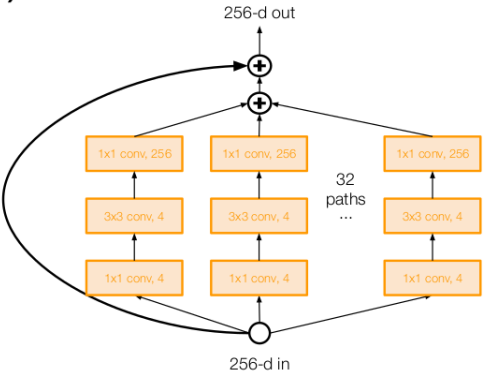


Aggregated Residual Transformations for Deep Neural Networks (ResNeXt)

[Xie et al. 2016]

- Also from creators of ResNet
- Increases width of residual block through multiple parallel pathways ("cardinality")
- Parallel pathways similar in spirit to Inception module

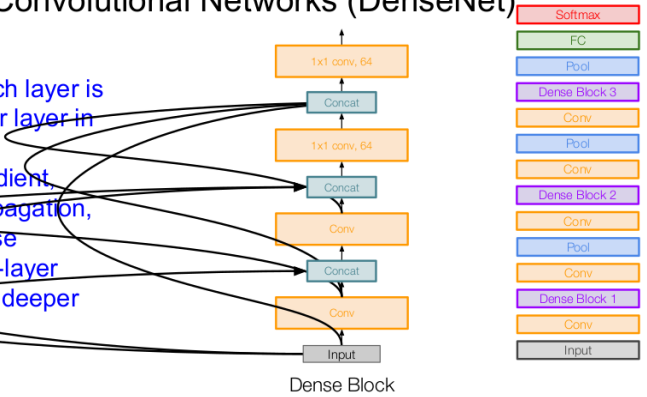




Densely Connected Convolutional Networks (DenseNet)

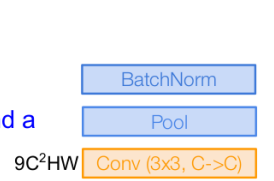
[Huang et al. 2017]

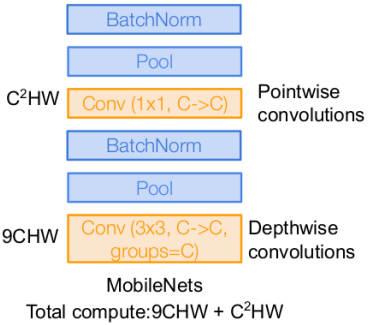
- Dense blocks where each layer is connected to every other layer in feedforward fashion
- Alleviates vanishing gradient, strengthens feature propagation, encourages feature reuse
- Showed that shallow 50-layer network can outperform deeper 152 layer ResNet



MobileNets: Efficient Convolutional Neural Networks for Mobile Applications *[Howard et al. 2017]*

- Depthwise separable convolutions replace standard convolutions by factorizing them into a depthwise convolution and a 1x1 convolution
- Much more efficient, with little loss in accuracy
- Follow-up MobileNetV2 work in 2018 (Sandler et al.)
- ShuffleNet: Zhang et al, CVPR 2018



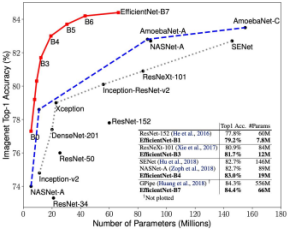


EfficientNet: Smart Compound Scaling

[Tan and Le. 2019]

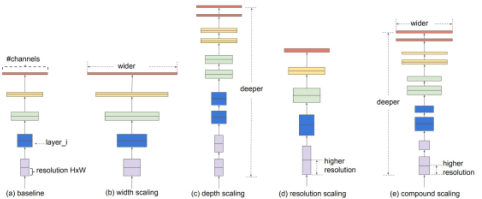
- Increase network capacity by scaling width, depth, and resolution, while balancing accuracy and efficiency.
- Search for optimal set of compound scaling factors given a compute budget (target memory & flops).
- Scale up using smart heuristic rules

$$\text{depth: } d = \alpha^\phi$$
$$\text{width: } w = \beta^\phi$$
$$\text{resolution: } r = \gamma^\phi$$
$$\text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$$
$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.2%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.7%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**83.0%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

† Not plotted



References

Refer Chapter 8 of Dive into Deep Learning book.

LeNet5 : Yann Lecun and Leon Bottou and Yoshua Bengio and Patrick Haffner, Gradient-based learning applied to document recognition, IEEE-1998

AlexNet : Imagenet classification with deep convolutional neural networks, Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey E, In: Advances in neural information processing systems, NIPS 2012

VGG16 : Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition."

Inception v1 : Going deeper with convolutions, Szegedy, Christian et.al CVPR 2015

ResNet : Deep residual learning for image recognition, He et.al CVPR-2016

Thank You!