Going deeper into Convolutions

Course: ML4DM

Instructor: Prof. Bakul Gohel

Naitik Dodia - 201501177

Raj Jakasaniya - 201501408

Introduction

- Image Classification
- Classification using Neural networks
- Classification using Convolutional Neural Networks (CCN)
- Classification using Deep CNNs.

A Convolutional Neural Network (CNN, or ConvNet) are a special kind of multi-layer neural networks, designed to recognize visual patterns directly from pixel images with minimal preprocessing.

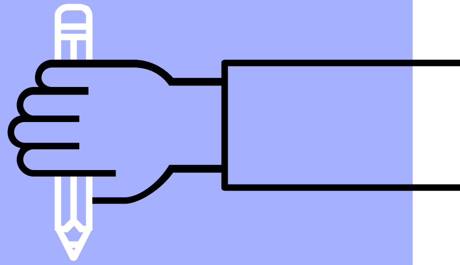
DataSet

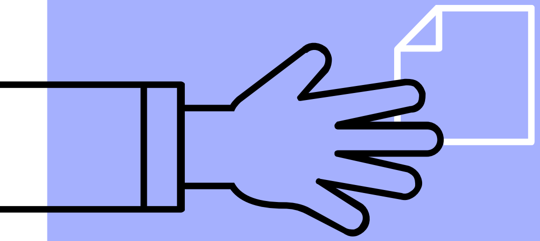
- Cifar100
  - Image dimensions - 32x32x3
  - Training samples - 50000
  - Classes - 100
  - Test samples - 10000

- Cifar10
  - Same as cifar100 but with only 10 classes

ResNet - Microsoft

ILSVRC 2015 classification task





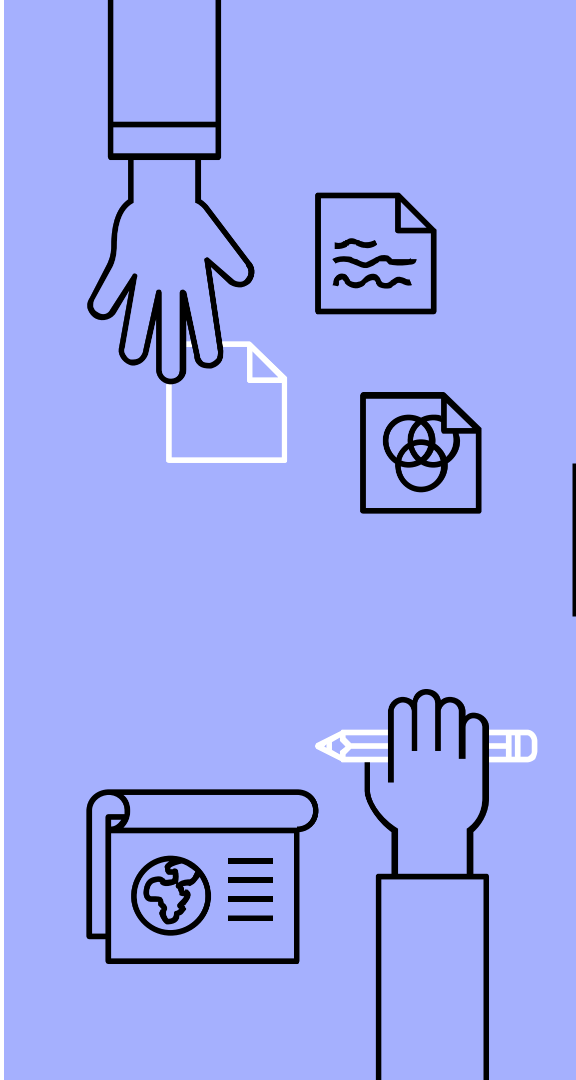
LeNet v1 - Google

ILSVRC 2014 ImageNet Large-Scale Visual Recognition Challenge

Problem with simple deep CNN architectures

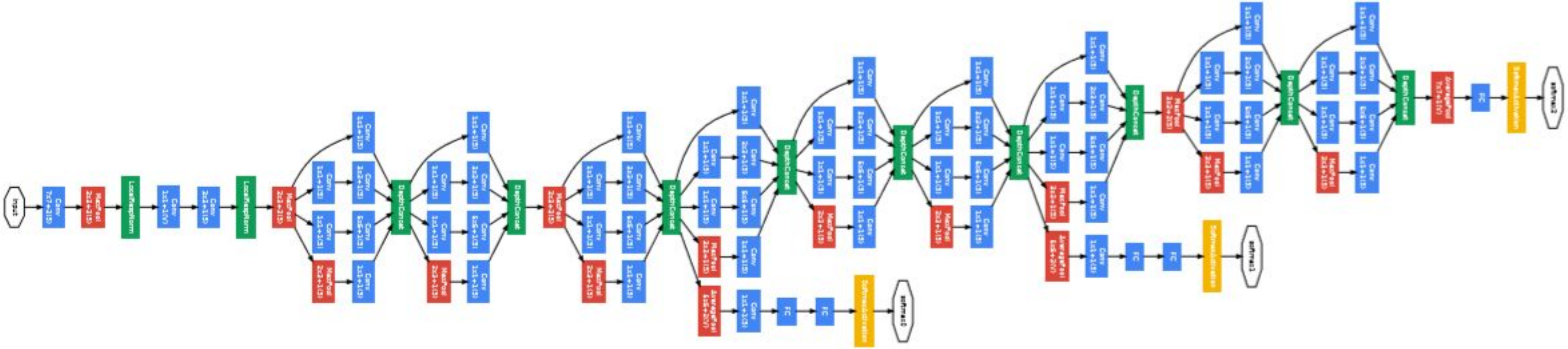
Simple deep CNN architecures (AlexNet and VGG net)

- Overfitting
- Vanishing gradient problem
- More non linearity
- As depth increases - #local minimas and saddle points increase.
- Degradation Problem - Accuracy gets saturated and then degrades rapidly (not due to overfitting)




WE NEED TO GO
DEEPER

LeNet - Google

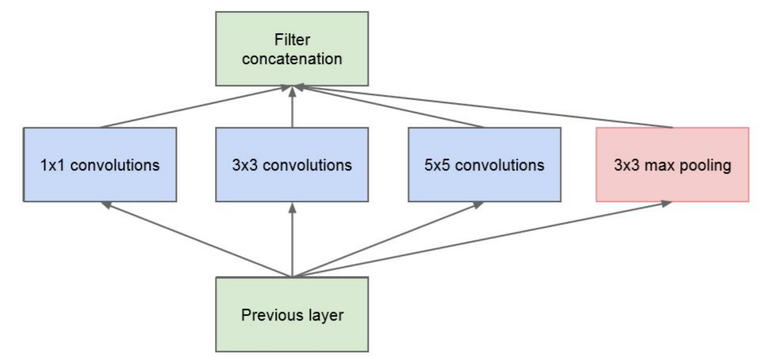


Total params: 10,612,348
Trainable params: 10,597,404
Non-trainable params: 14,944

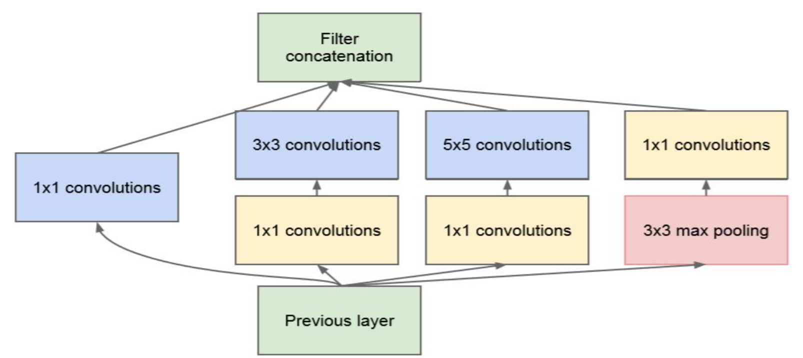
LeNet - Google

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Inception module



(a) Inception module, naïve version



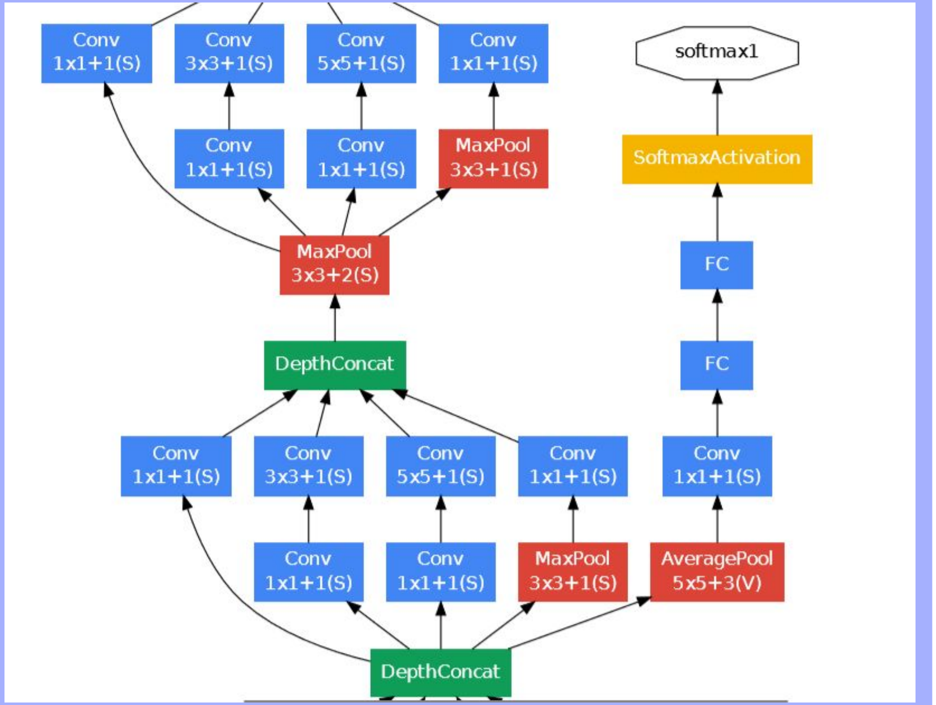
(b) Inception module with dimension reductions

- Has 3 types of convolution filter sizes (1x1, 3x3, 5x5) and a branch with no convolution.
- Intuition of learning the filter size
- Output - Concatenation of the output of individual branches.

- Same concept as of naive version but with use of 1x1 convolutions for dimension reduction

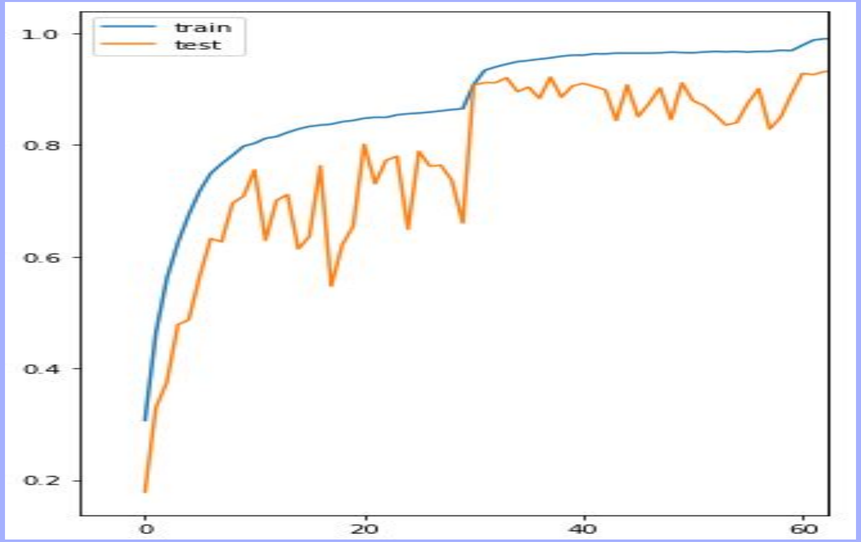
Auxilary - output

- An extra FC network from intermediate layers.
- If some intermediate layer has high discriminative power
- And for Vanishing gradient problem of deep neural networks
- Error weighed by 0.3



Accuracy

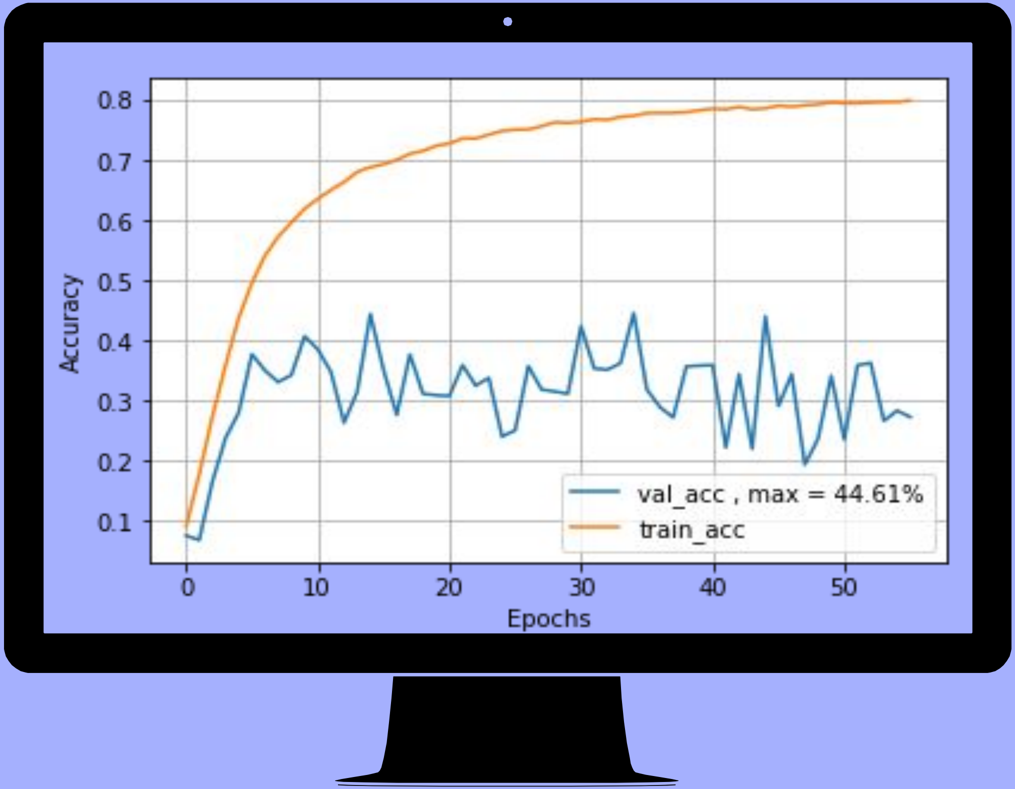
- High variance
- Saturation after very long time
- Increases using big jumps
- Computational limitations



Accuracy plot for Cifar10

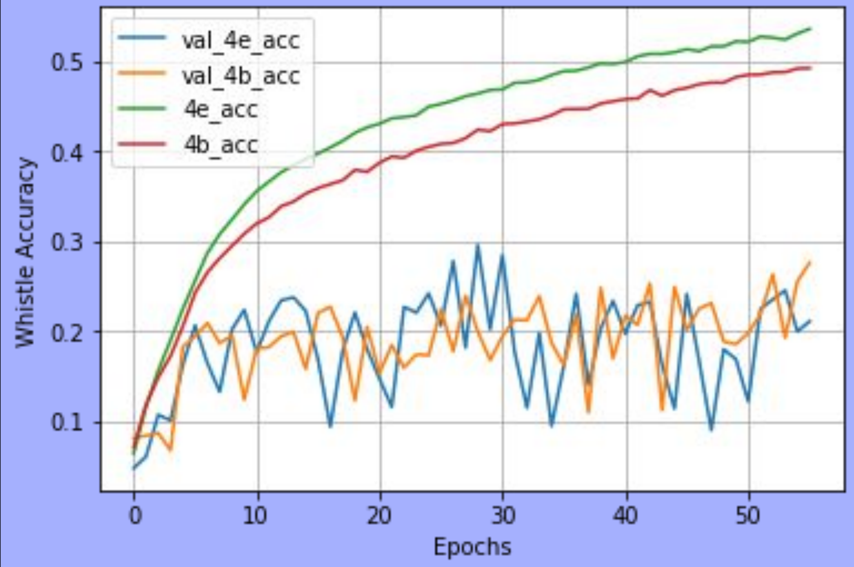
Accuracy

- High variance
- Saturation after very long time
- Computational limitations



Accuracy plot for Cifar100

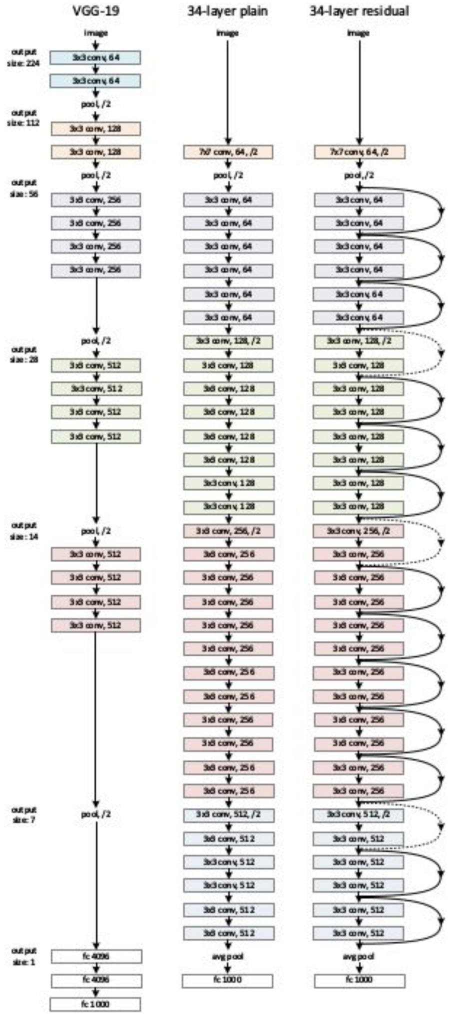
Accuracy for auxilary outputs



Accuracy plot for Cifar100

ResNet - Microsoft

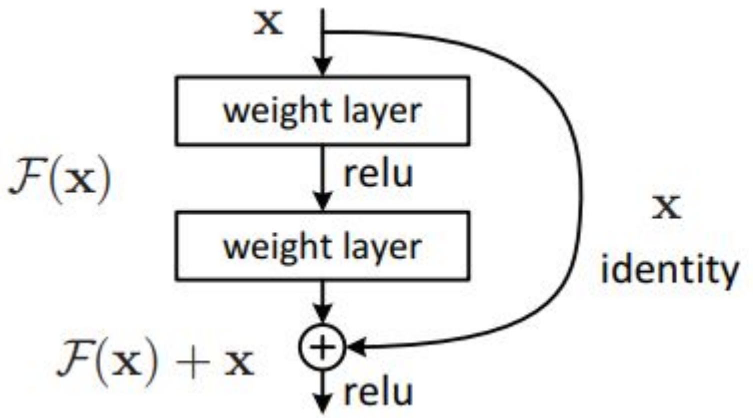
layer name	output size	18-layer
conv1	112×112	
conv2_x	56×56	
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$
	1×1	



Identity Mapping by Shortcuts

y = F(x, {Wi}) + x.

The operation F + x is performed by a shortcut connection and element-wise addition.



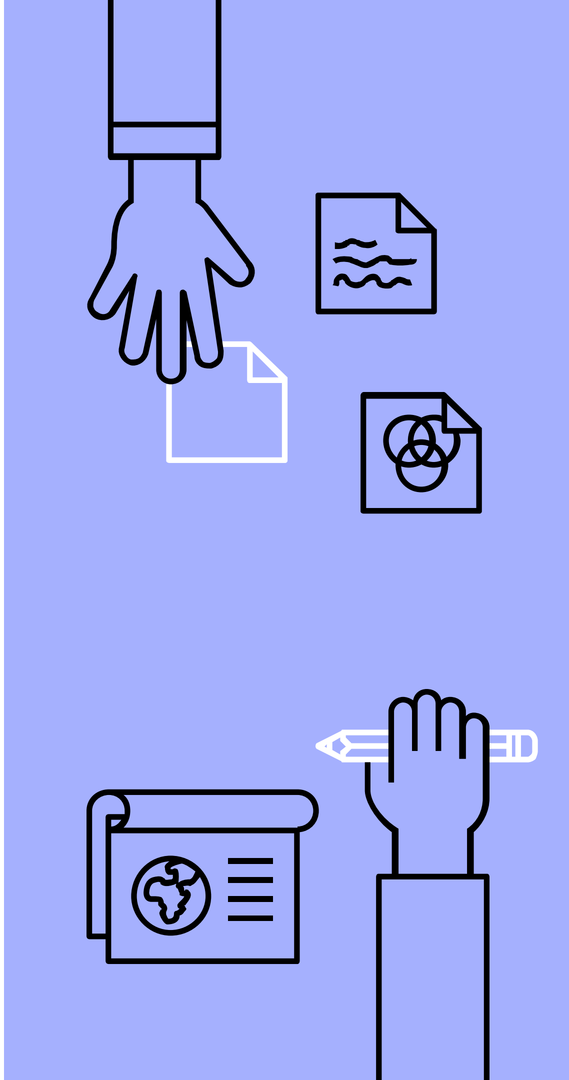
Addressing the degradation problem by introducing a deep residual learning framework. Instead of hoping each few stacked layers directly fit a desired underlying mapping, we explicitly let these layers fit a residual mapping. This reformulation is motivated by the counterintuitive phenomena about the degradation problem.

Accuracy

- Computational Limitations
- Randomness
- High Variance

Conclusion

- Both networks show high variance in validation accuracy during training.
- Because of change in internal structure
- Learns internal structure
- LeNet - Learns filter size and importance of layers
- ResNet - Learns which layers to skip in the training procedure.



Thank You

References:

- Going deeper with convolutions - Google
- Deep Residual Learning for Image Recognition - Microsoft Research
- Know your meme: We need to go deeper. http://knowyourmeme.com/memes/we-need-to-go-deeper.