VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan & Andrew Zisserman

presented by: Pradeep Karuturi

Recap - Krizhevsky’s work(2012)

- 1.2 million ImageNet LSVRC-2010 dataset with 1000 classes
- ReLU based non-linearity
- 5 convolutional, 3 fully connected layers
- 60 million parameters
- GPU based training
- established once for all that deep models do work for computer vision!

What next from here?

1. Apply deep models to different domains
2. Come up with more efficient training(which is what krizhevsky did next)
3. Come up with ad-hoc tricks to prevent overfitting(like Dropout)
4. Different convolution strategies(Ziegler & Fergus - 2013)
5. Try “deeper” architectures

Simoyan & Zisserman’s work

- Deals with the problem of deeper architectures, building upon the work of Krizhevsky(2012) and Ziegler(2013).
- Very “experimental” paper.


WE NEED TO GO
DEEPER

How deep are we talking about?

- 11 to 19 layers!
- 3 fully connected and the rest are convolutional

Convolution & pooling

- 3x3 convolutional filters with stride 1
- five 2x2 max-pooling layers with stride 2

Network Architecture

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	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					
soft-max					

Parameters(millions)

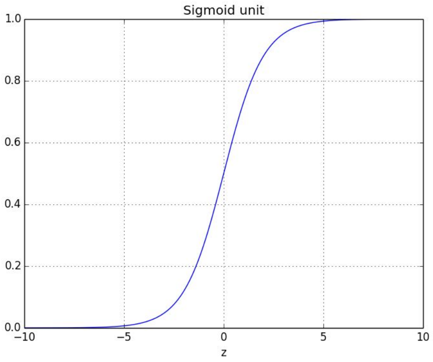
A,A-LRN	B	C	D	E
133	133	134	138	144

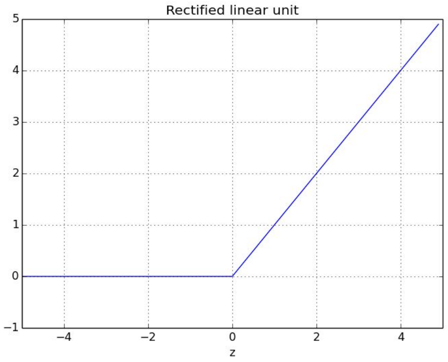
Parameter size - network D

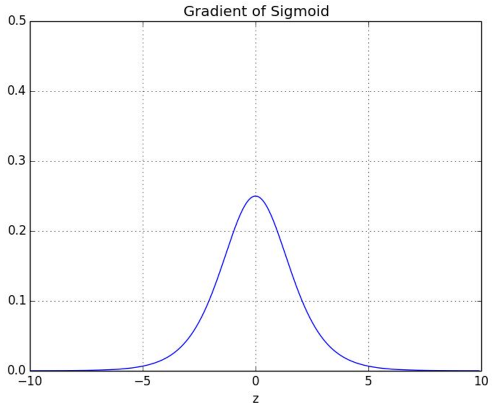
conv3-64, conv3-64	3*3*3*64+3*3*64*64+64*2	38720
conv3-128,conv3-128	3*3*64*128+3*3*128*128+128*2	221440
conv3-256,conv3-256,conv3-256	3*3*128*256+3*3*256*256*2+256*3	1475328
conv3-512,conv3-512,conv3-512	3*3*256*512+3*3*512*512*2+512*3	5899776
conv3-512,conv3-512,conv3-512	3*3*512*512*3 + 512*3	7079424
fc-1	512*7*7 * 4096 + 4096	102764544
fc-2	4096*4096 + 4096	16781312
fc-3	4096*1000+1000	4097000
	Total parameters	**138,357,544**

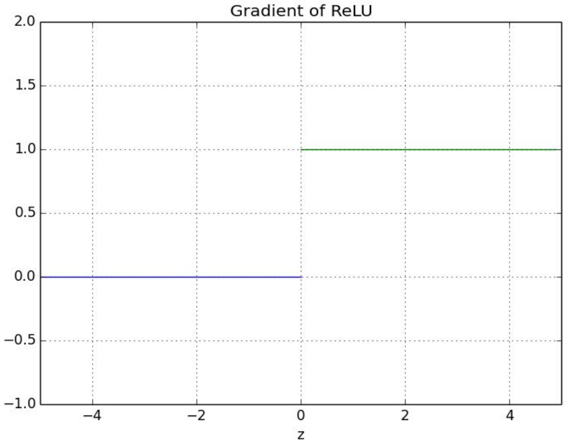
Problems with very deep networks

- Vanishing gradient problem









Problems with very deep networks

- Vanishing gradient problem
- Overfitting
- Enormous training time

Problems - solutions

- Vanishing gradient problem - partly handled by ReLUs
- Overfitting - augmentation, dropout
- Enormous training time - smart initialization of the network, GPU & ReLUs

Dataset & Metrics

ILSVRC 2012 - 1000 classes

Training	1.3 million
Validation	50,000
Test	100,000

Top-1 & Top-5 as error metrics

Training details

- rescale the given image to a training scale(S - for the smallest side)
- 224x224 RGB input(one random crop per image per SGD iteration)
- multinomial logistic regression objective
- back-prop with momentum
- mini-batch(size 256) gradient descent
- use the weights from trained shallow networks as the initialization weights for deeper networks

Experiments

- Single scale evaluation - singe scale of a test image
- Multi scale evaluation - several rescaled versions of a test image and averaging the resulting class posteriors
- Multi crop evaluation
- ConvNet Fusion - ensemble

Results - single test scale

config	top-1 error	top-5 error
A	29.6	10.4
A-LRN	29.7	10.5
B	28.7	9.9
C	27.3	8.8
D	25.6	8.1
E	25.5	8.0

Results - multiple test scale

config	top-1 error	top-5 error
B	28.2	9.6
C	26.3	8.2
D	24.8	7.5
E	24.8	7.5

Multi-crop & Fusion

- Multi-crop gives slightly better results, but at the expense of computation time.
- Ensembling two best models(D&E) gave top-5 error of 7%

Does this work for other datasets?

Method	VOC-2007 (mean AP)	VOC-2012 (mean AP)	Caltech-101 (mean CR)	Caltech-256 (mean CR)
Zeiler & Fergus (Zeiler & Fergus, 2013)	-	79.0	86.5 ± 0.5	74.2 ± 0.3
Chatfield et al. (Chatfield et al., 2014)	82.4	83.2	88.4 ± 0.6	77.6 ± 0.1
He et al. (He et al., 2014)	82.4	-	93.4 ± 0.5	-
Wei et al. (Wei et al., 2014)	81.5	81.7	-	-
VGG Net-D (16 layers)	89.3	89.0	91.8 ± 1.0	85.0 ± 0.2
VGG Net-E (19 layers)	89.3	89.0	92.3 ± 0.5	85.1 ± 0.3
VGG Net-D & Net-E	89.7	89.3	92.7 ± 0.5	86.2 ± 0.3

Representations learnt over ImageNet generalize well to other smaller datasets.

Kaggle CIFAR-10 challenge

Leaderboard

1. DeepCNet
2. jiki
3. Anil Thomas
4. Frank Sharp
5. nagadomi
6. Phil & Triskelion & Kazanova
7. Daniel Nouri
8. Terry
9. Luca Massaron
10. Gil Levi

CIFAR-10 dataset

From publicly available information, it’s clear that 7 out of the top 10 teams used deep models borrowed from this paper.

The other three might also have used it!

Discussion

- “Representational” depth benefits classification accuracy?
- Alternate deep architectures?

Alternate architectures

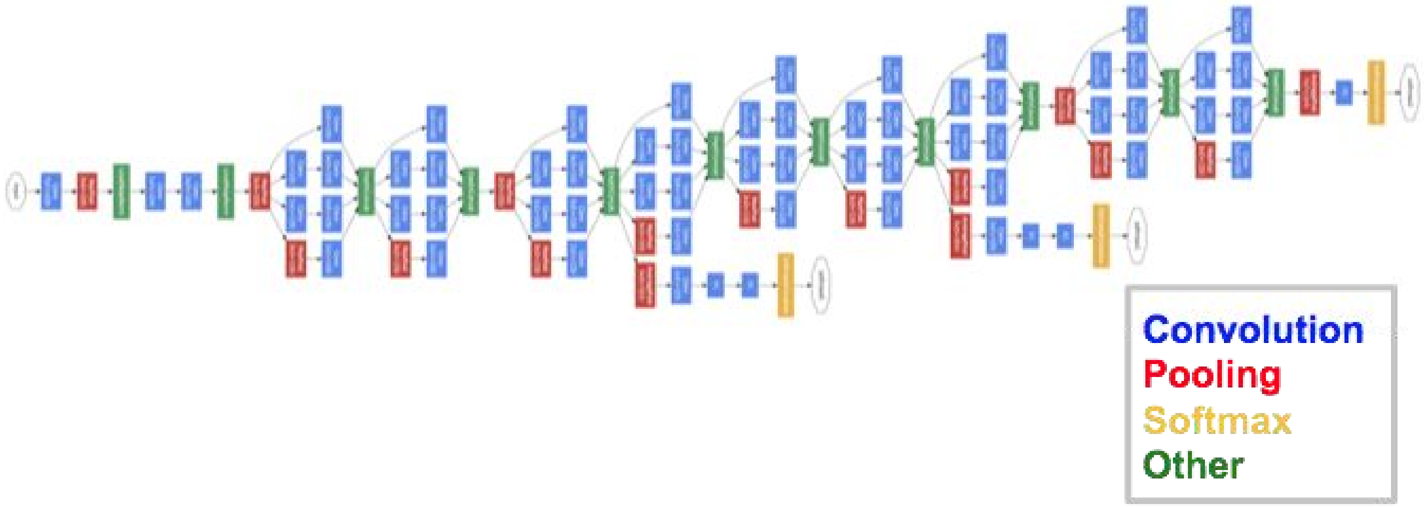
Going deeper with convolutions - Szegedy et.al

- Deeper than Zisserman et.al's work. 22 layers!
- Focus is on "efficient" architecture - in terms of computation
- Compact model (about 5 million parameters)
- Computational budget of 1.5 billion multiply-adds

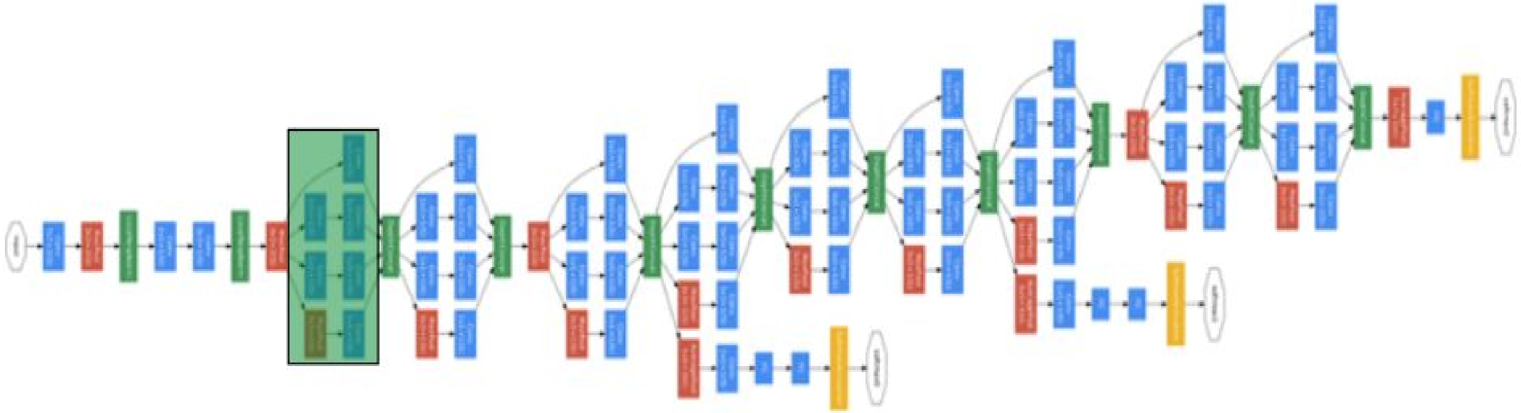
Szegedy’s work - GoogLeNet

- ReLU units
- max-pool, avg-pool
- Compact model(about 5 million parameters)
- Problems of gradient, overfitting, efficient training apply here as well!

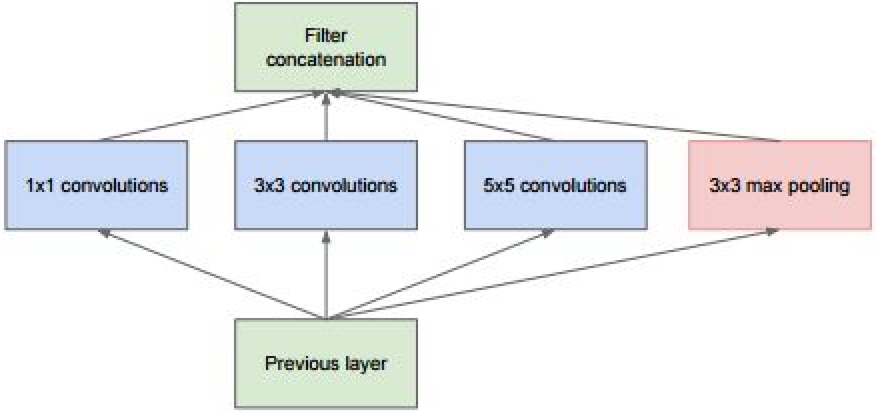
GoogleLeNet



Inception module - Basic building block

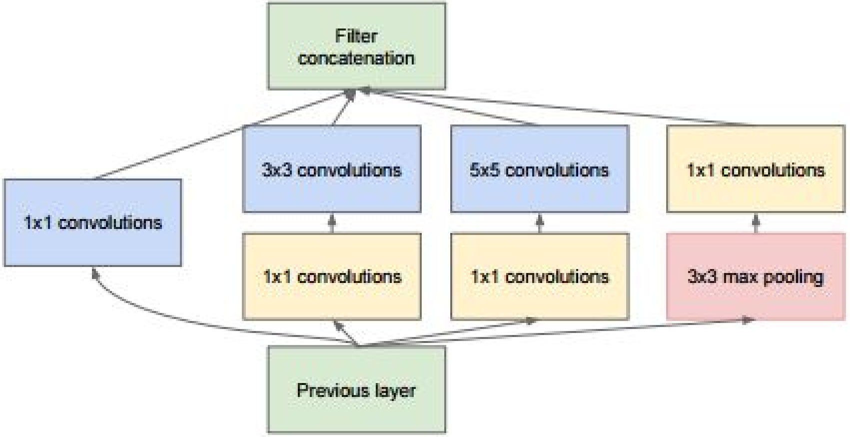


Inception module - Naive



This explodes the number of parameters! what do we do?

Inception module - Dimensionality reduction



Summary & Results

- Multi-scale architecture to mirror correlation structure in images.
- Dimensional reduction to constrain representation along each spatial scale.

Performs better than VGG(Zisserman) model - 6.67 vs 7%

Questions?