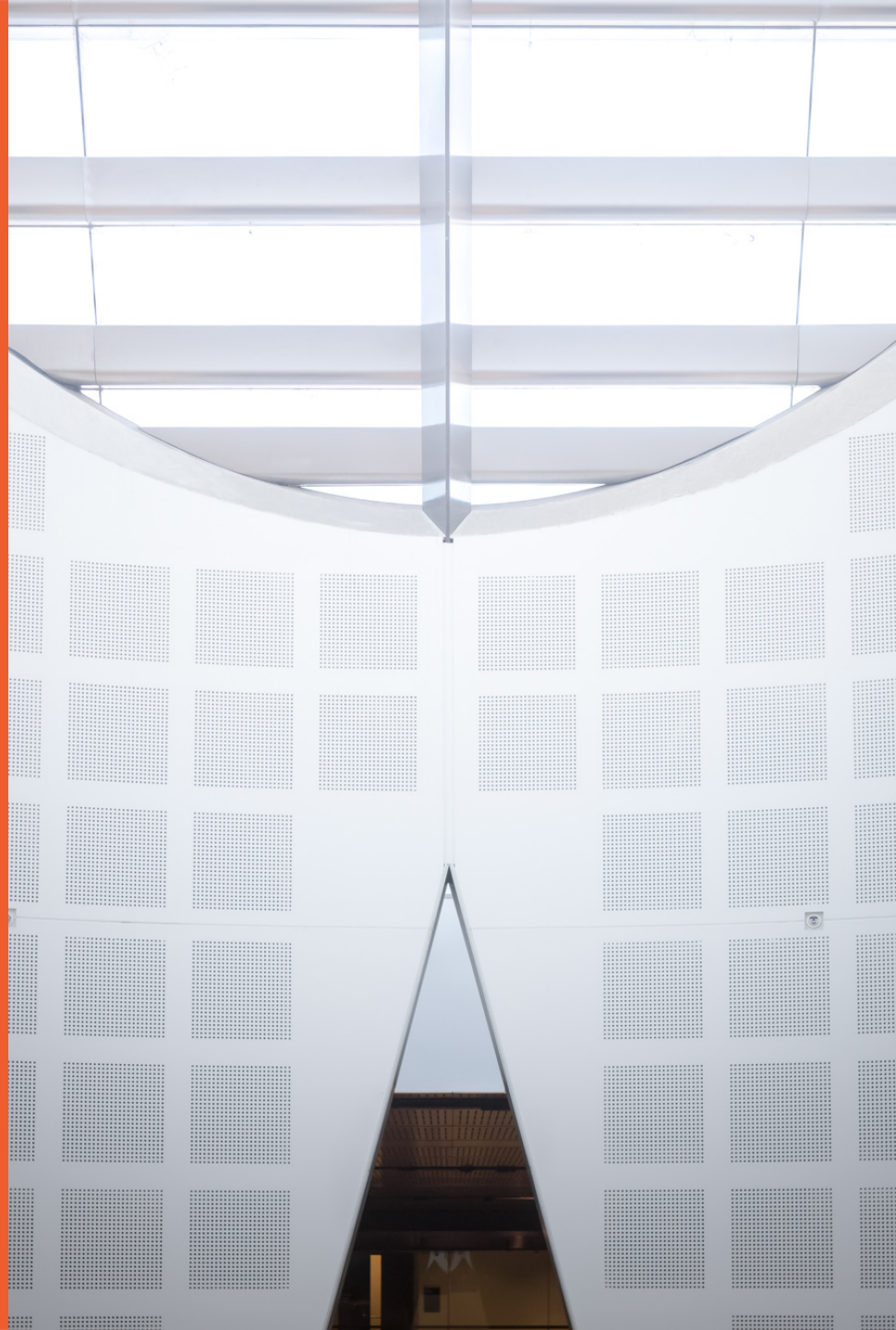
Network Structures

Dr Chang Xu
UBTECH Sydney AI Centre

Acknowledgements:
Baosheng Yu and Jiayan Qiu



THE UNIVERSITY OF SYDNEY

Quick Review

ILSVRC

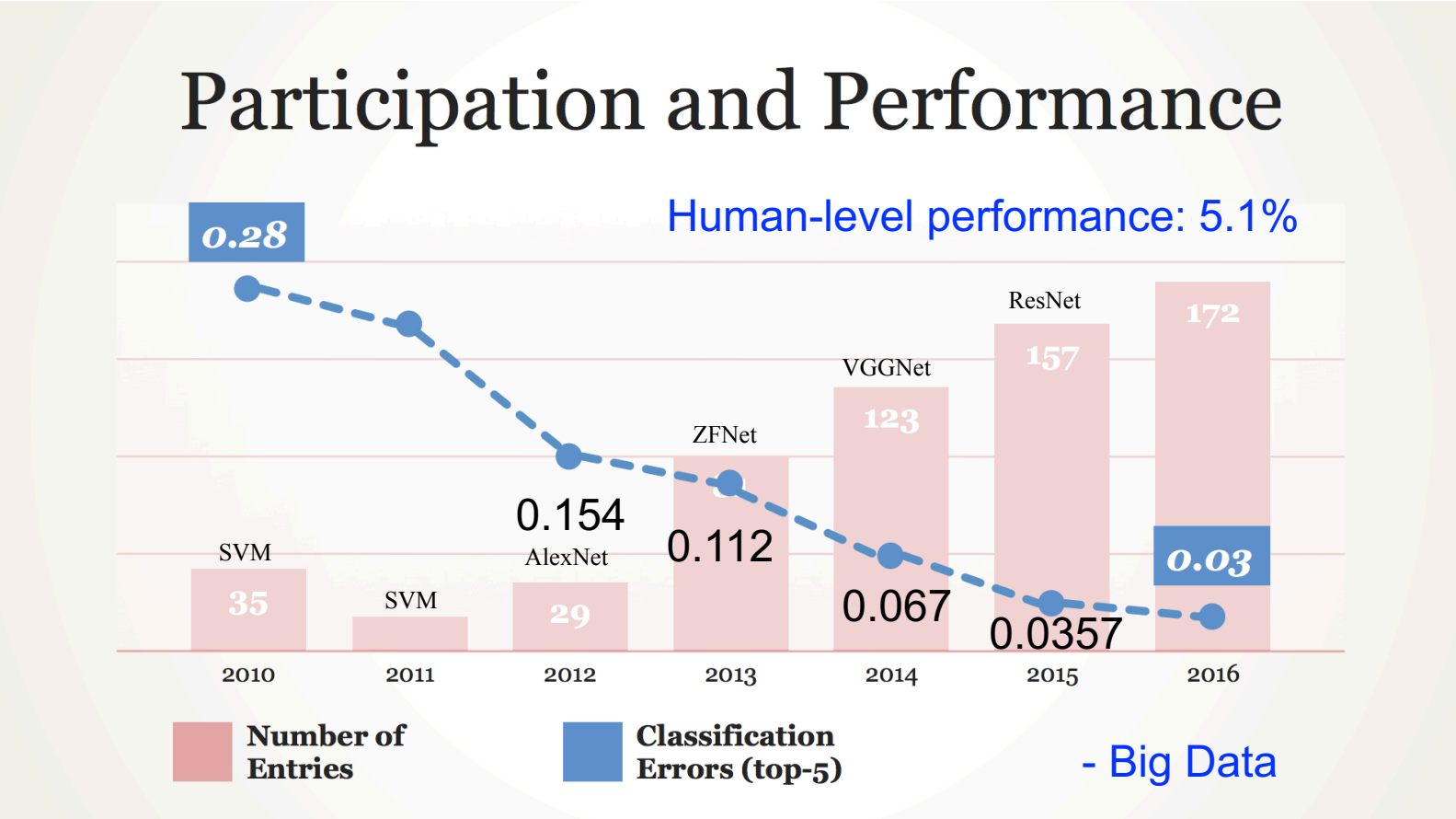
- **Image Classification**
  - one of the core problems in computer vision
  - many other tasks (such as object detection, segmentation) can be reduced to image classification

- **ImageNet Large Scale Visual Recognition Challenge**
  - 2010 - 2017
  - main tasks: classification and detection
  - a large-scale benchmark for image classification methods
    - 1000 classes
    - 1.2 million training images
    - 50 thousand verification images
    - 150 thousand test images

* *Some slides are borrowed from cs231n.stanford.edu*

ILSVRC

ILSVRC



- Big Data

- GPU

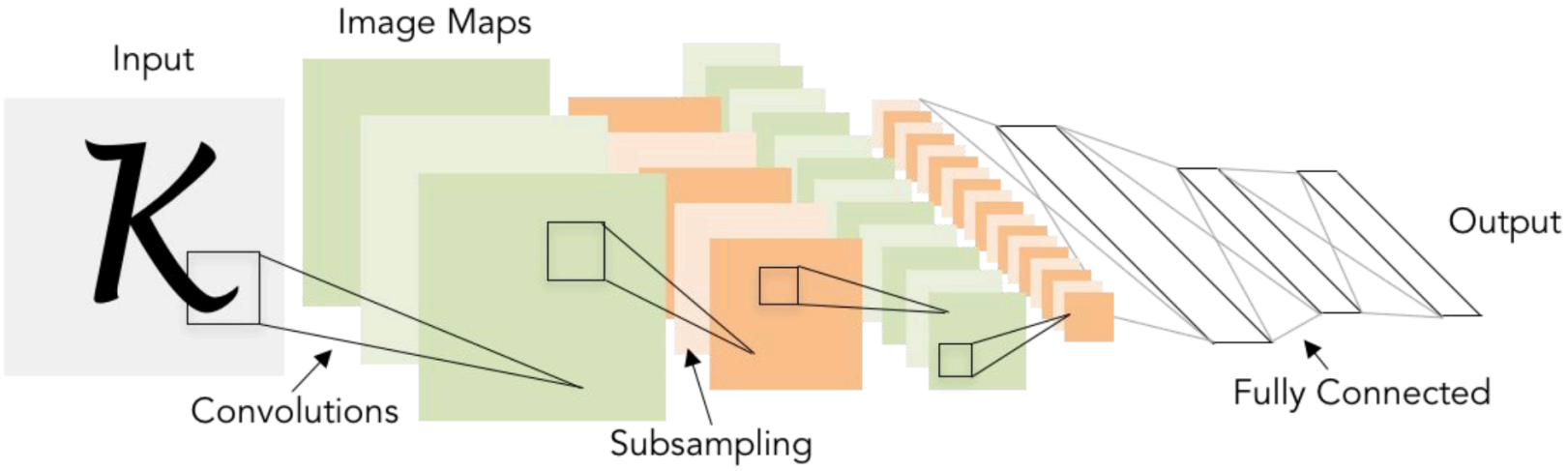
- Algorithm improvement

Winning CNN Architectures

Model	AlexNet	ZF Net	GoogLeNet	Resnet
Year	2012	2013	2014	2015
#Layer	8	8	22	152
Top 5 Acc	15.4%	11.2%	6.7%	3.57%
Data augmentation	√	√	√	√
Dropout	√	√		
Batch normalization				√

CNN Architecture: Part I

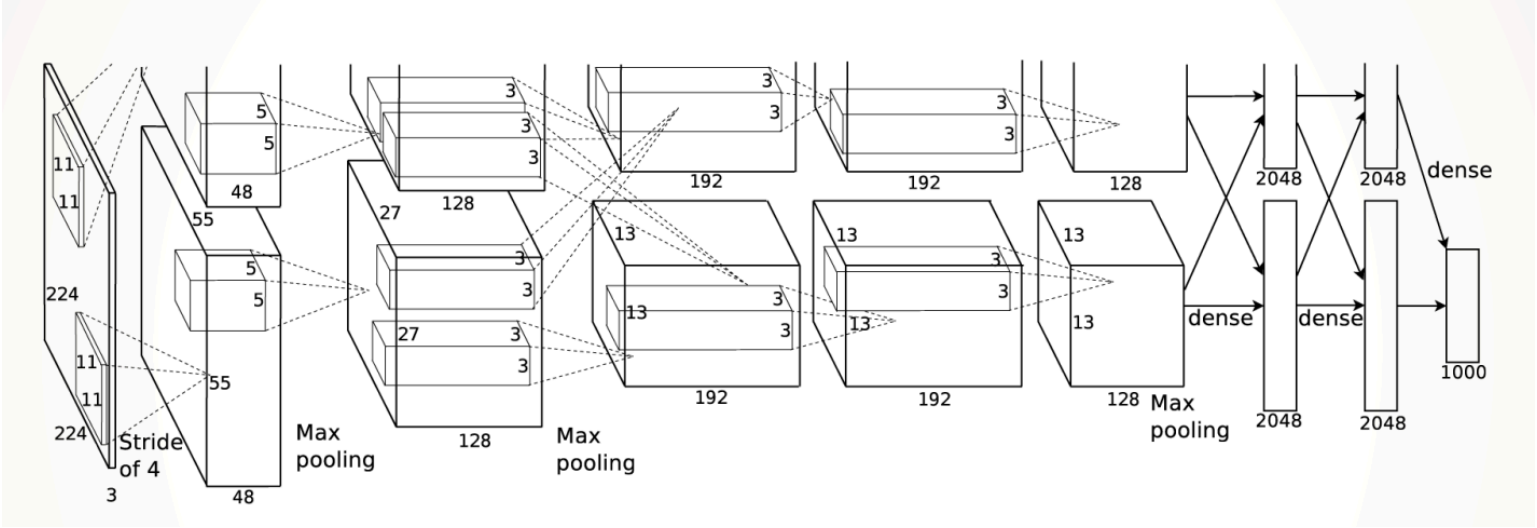
❑ **LeNet** [LeCun et al., 1998]



- Architecture is [CONV-POOL-CONV-POOL-FC-FC]
- CONV: 5x5 filter, stride=1
- POOL: 2x2 filter, stride=2

CNN Architecture: Part I

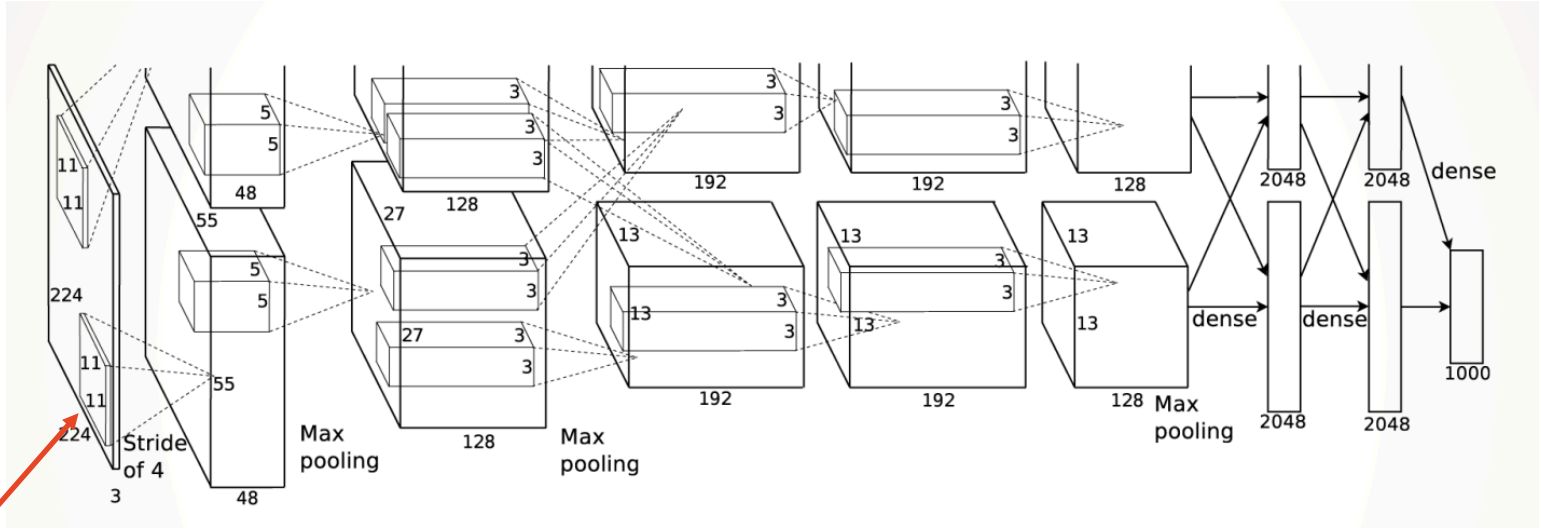
- **AlexNet** [Krizhevsky et al. 2012]



ILSVRC2010: 28.2%
ILSVRC2011: 25.8%
ILSVRC2012: 16.4% (second winner 26.2%)

CNN Architecture: Part I

❑ **AlexNet** [Krizhevsky et al. 2012] -5 conv layers, 3 fully connected layers.



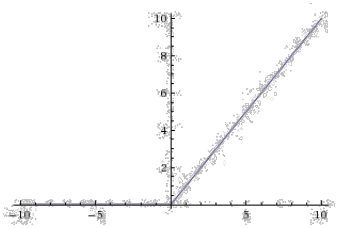
11x11 filters

- Activation function: ReLU
- Data Augmentation
- Dropout (drop rate=0.5)
- Local Response Normalization
- Overlapping Pooling

CNN Architecture: Part I

❑ **AlexNet** [Krizhevsky et al. 2012] -5 conv layers, 3 fully connected layers.

- Activation function: ReLU
- Data Augmentation
- Dropout
- Local Response Normalization
- Overlapping Pooling



$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

$$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

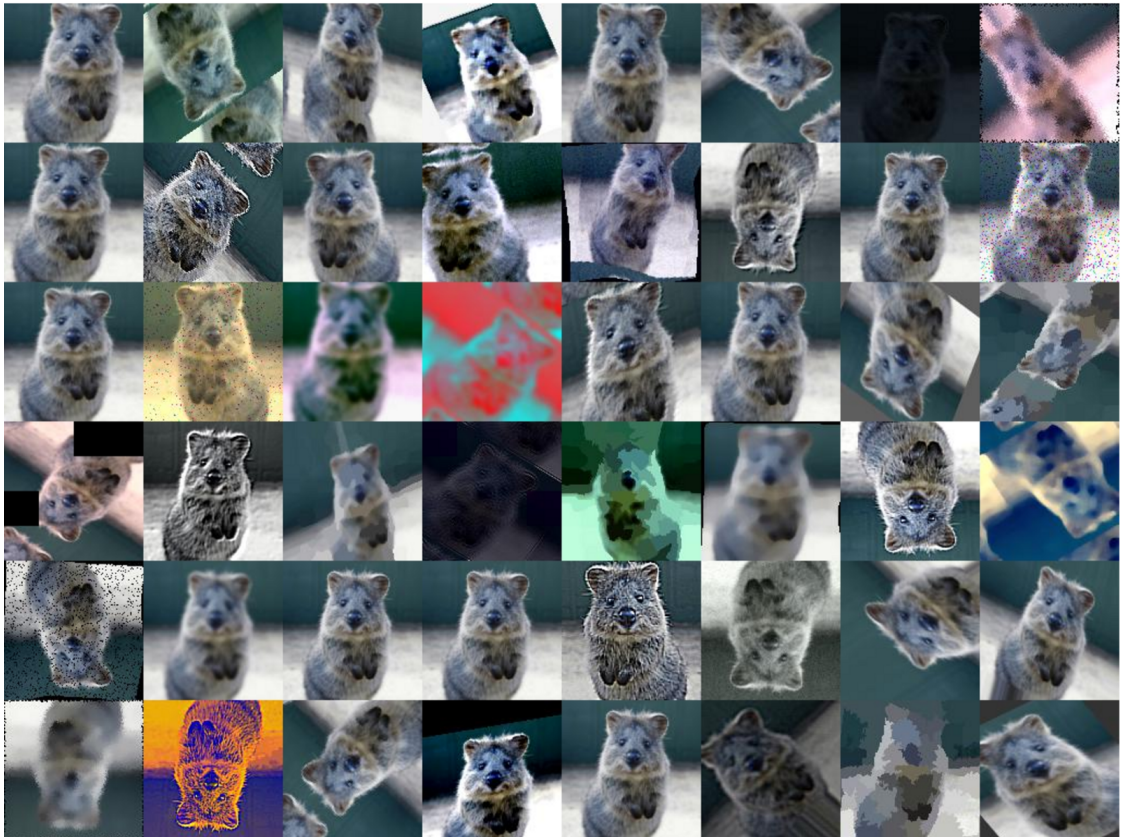
Pros:
- Easy to feed forward
- Easy to back propagate
- Help prevent saturation
- Sparse

Cons:
- Dead ReLU

CNN Architecture: Part I

❑ AlexNet [Krizhevsky et al. 2012]

- Activation function: ReLU
- Data Augmentation
- Dropout
- Local Response Normalization
- Overlapping Pooling

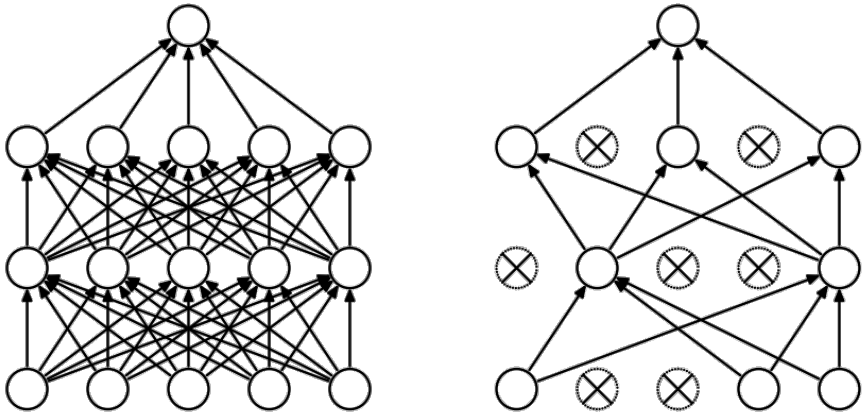


- Randomly Rotate Images
- Flip Images
- Shift Images
- Contrast Stretching
- Adaptive Equalization
- etc.

CNN Architecture: Part I

❑ **AlexNet** [Krizhevsky et al. 2012]

- Activation function: ReLU
- Data Augmentation
- Dropout
- Local Response Normalization
- Overlapping Pooling



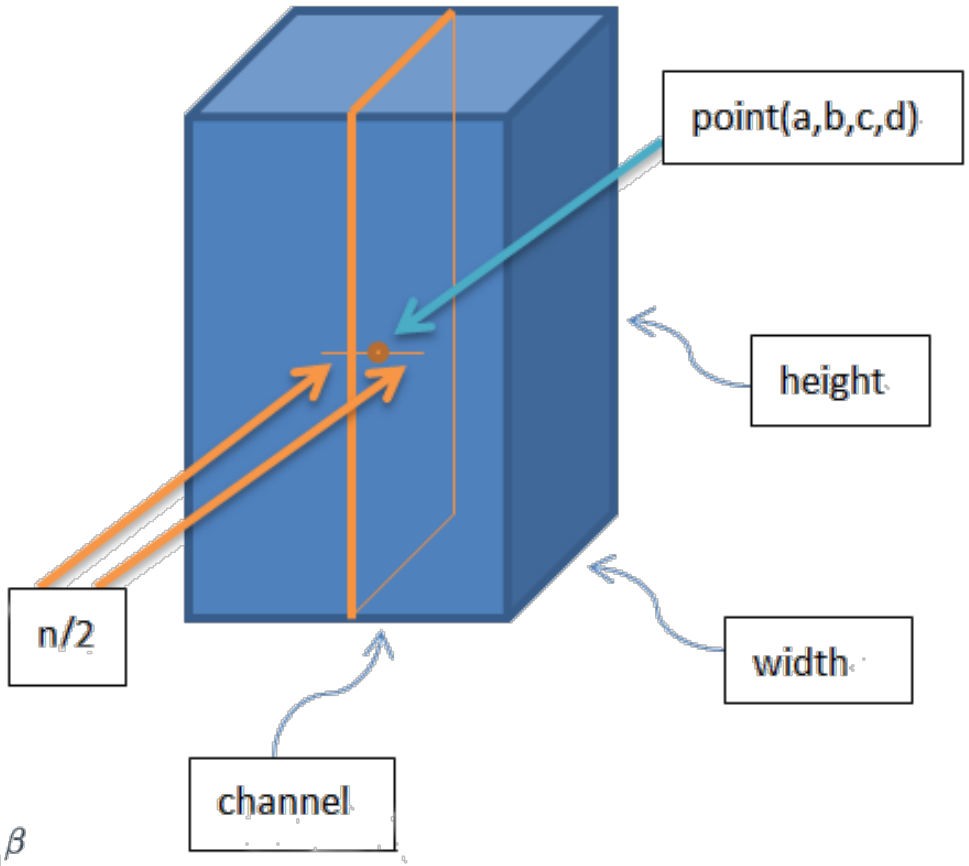
In practice, dropout trains 2^n networks (n – number of units).

Drop is a technique which deal with overfitting by combining the predictions of many different large neural nets at test time.

CNN Architecture: Part I

❑ AlexNet [Krizhevsky et al. 2012]

- Activation function: ReLU
- Data Augmentation
- Dropout
- Local Response Normalization
- Overlapping Pooling



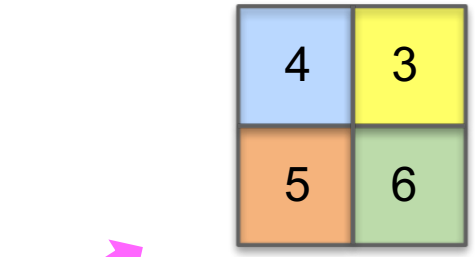
$$b^i_{x,y} = a^i_{x,y} / (k + \alpha \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} (a^i_{x,y})^2)^\beta$$

CNN Architecture: Part I

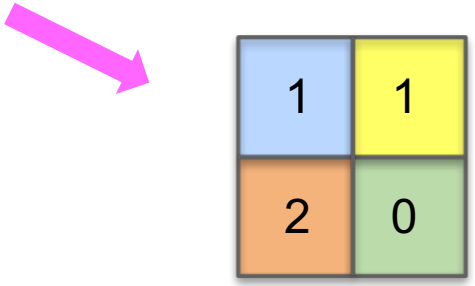
❑ AlexNet [Krizhevsky et al. 2012]

- Activation function: ReLU
- Data Augmentation
- Dropout
- Local Response Normalization
- Overlapping Pooling

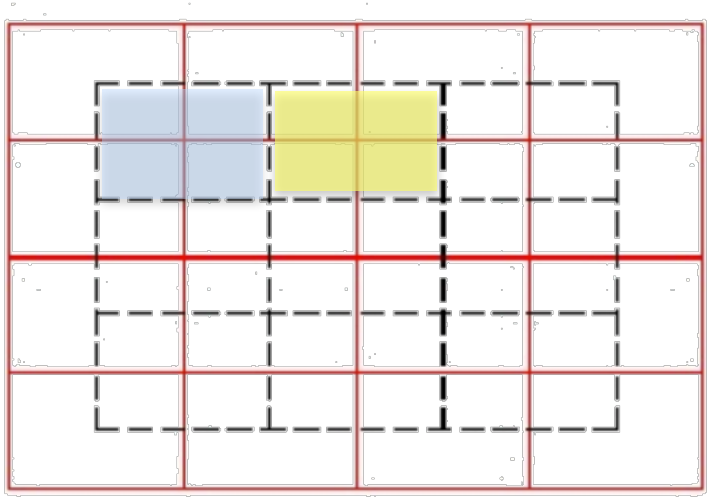
Feature map



Max pooling



Average pooling



Pooling stride < Pooling kernel size

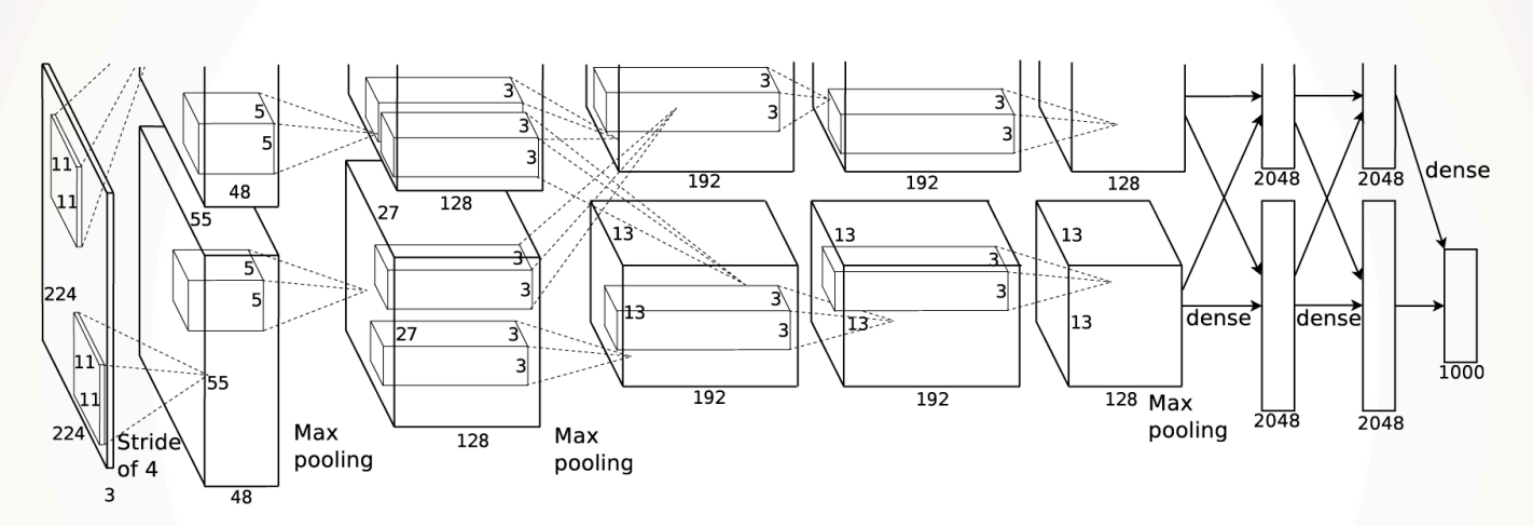
Overlapping Pooling

Standard Pooling

Pooling stride = Pooling kernel size

CNN Architecture: Part I

- **AlexNet** [Krizhevsky et al. 2012]

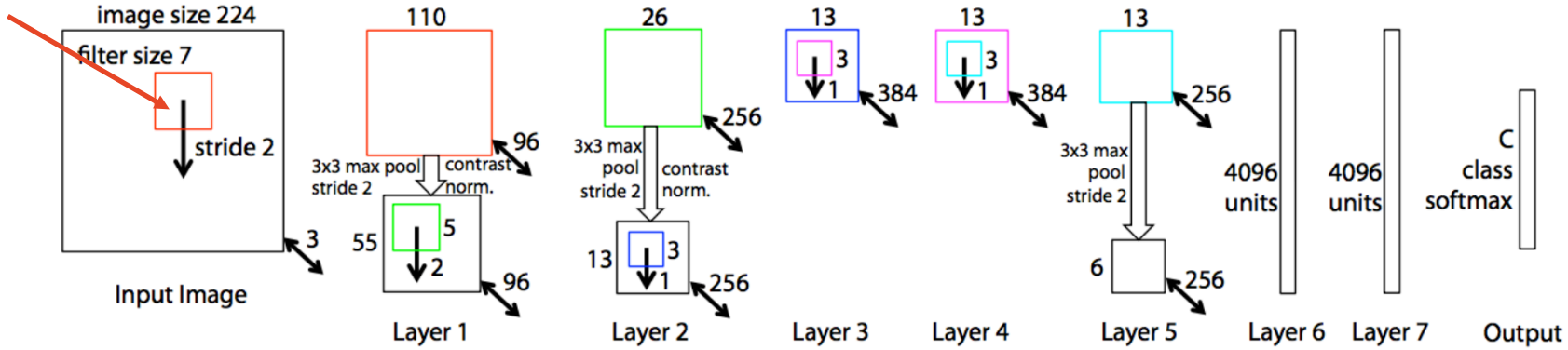


The problem set is divided into 2 parts, half executing on GPU 1 & another half on GPU 2.

CNN Architecture: Part I

❑ ZFNet [Zeiler and Fergus, 2013]

7x7 filters



- An improved version of AlexNet: top-5 error from 16.4% to 11.7%
- First convolutional layer: 11x11 filter, stride=4 -> 7x7 filter, stride=2
- The number of filters increase as we go deeper.

CNN Architecture: Part I

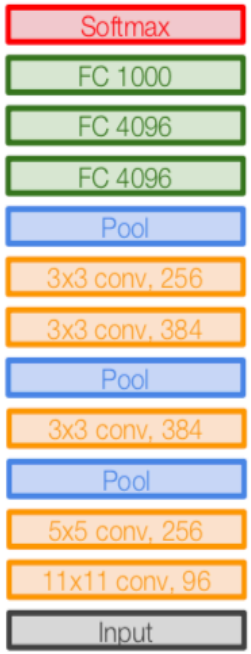
❑ **VGGNet** [Simonyan and Zisserman, 2014]

Small filters:

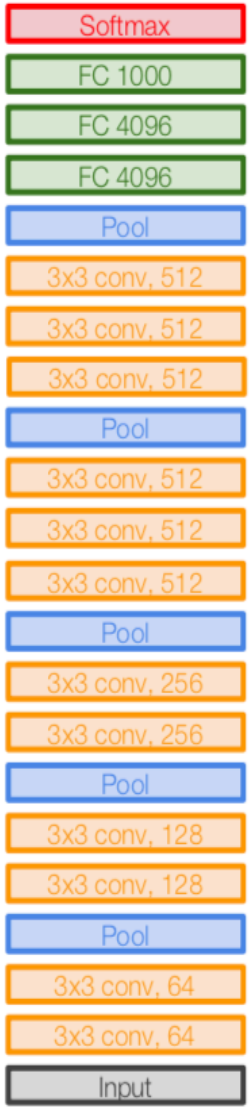
- 3x3 convolutional layers (stride 1, pad 1)
- 2x2 max-pooling, stride 2

Deeper networks:

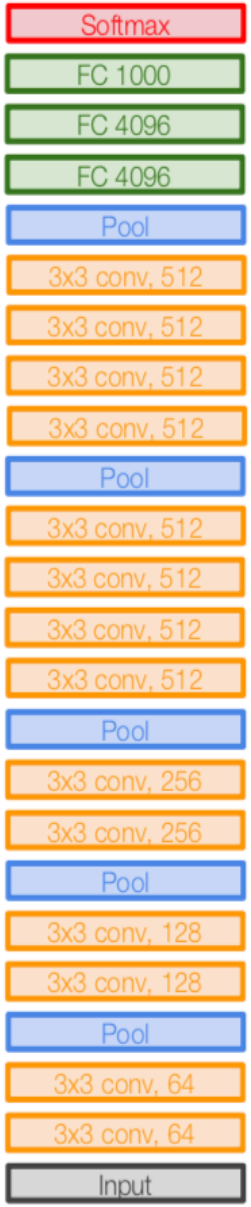
- AlexNet: 8 layers
- VGGNet: 16 or 19 layers



AlexNet



VGG16



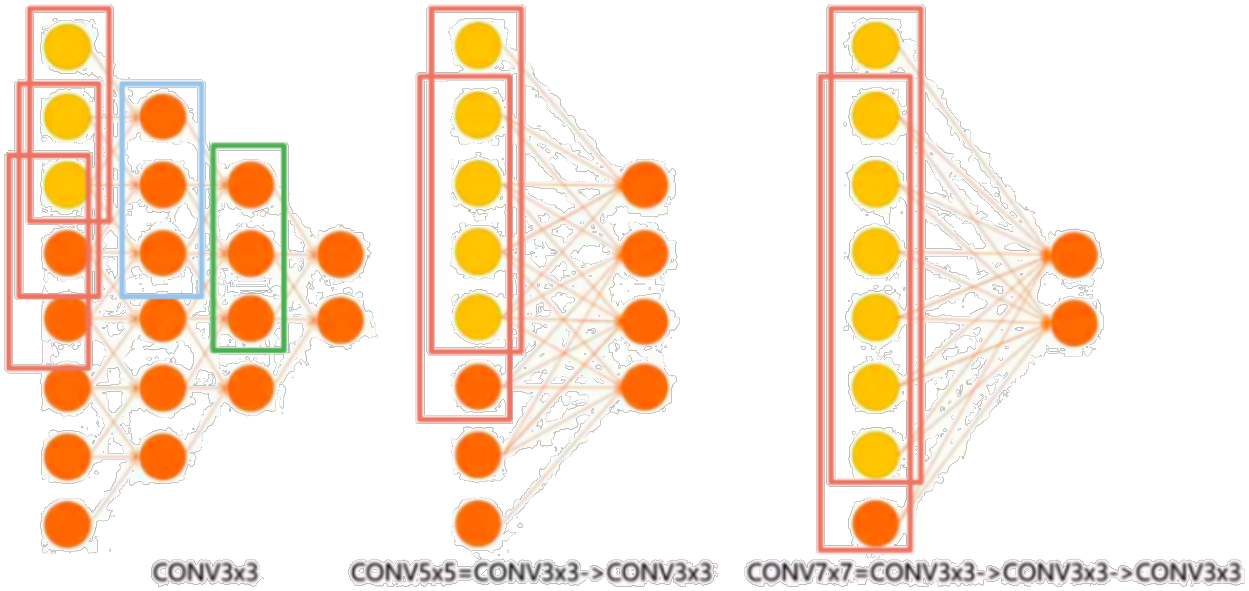
VGG19

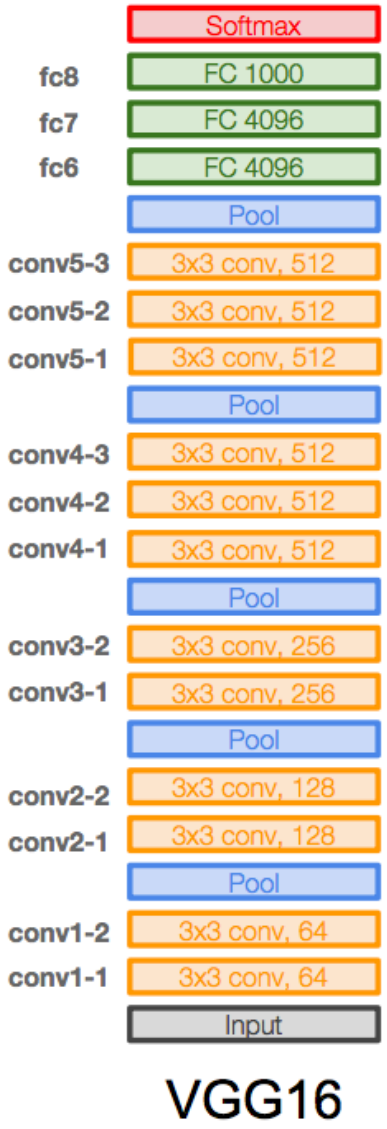
CNN Architecture: Part I

Why small filters?

- Two 3x3 layer = 5x5 layer
  Three 3x3 layer = 7x7 layer
- Deeper, more non-linearity
- Less parameters

Receptive Field (RF): the region in the input space that a particular CNN's feature is looking at (i.e. be affected by).





VGG16

Image credit to mc.ai

CNN Architecture: Part I

Why small filters?

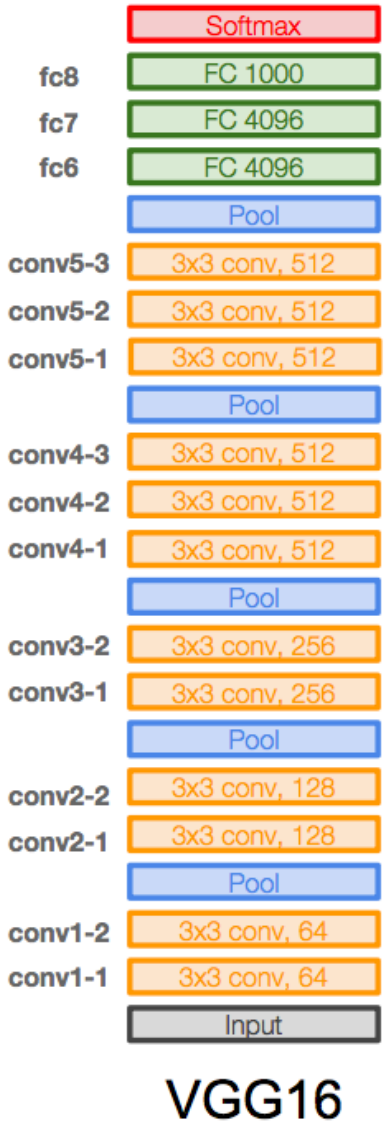
- Two 3x3 layer = 5x5 layer
  Three 3x3 layer = 7x7 layer
- Deeper, more non-linearity
- Less parameters

Why popular?

- FC7 features generalize well to other tasks
- Plain network structure

The rule for network design

- Prefer a stack of small filters to a large filter



VGG16

CNN Architecture: Part I

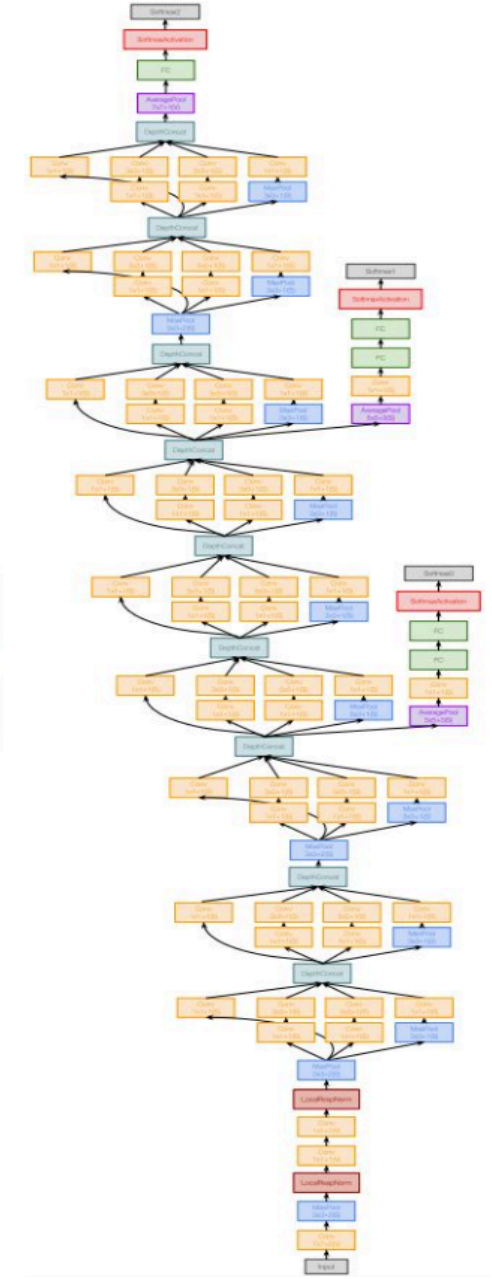
❑ GoogLeNet [Szegedy et al., 2014]

Deeper networks

- 22 layers.
- Auxiliary loss

Computational efficiency

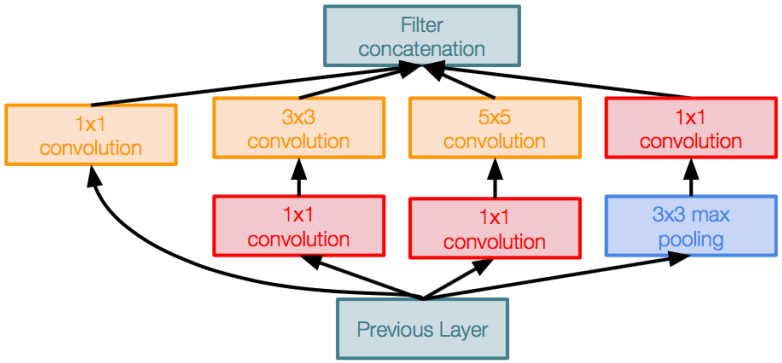
- Inception module
- Remove FC layer, use global average pooling
- 12x less parameters than AlexNet



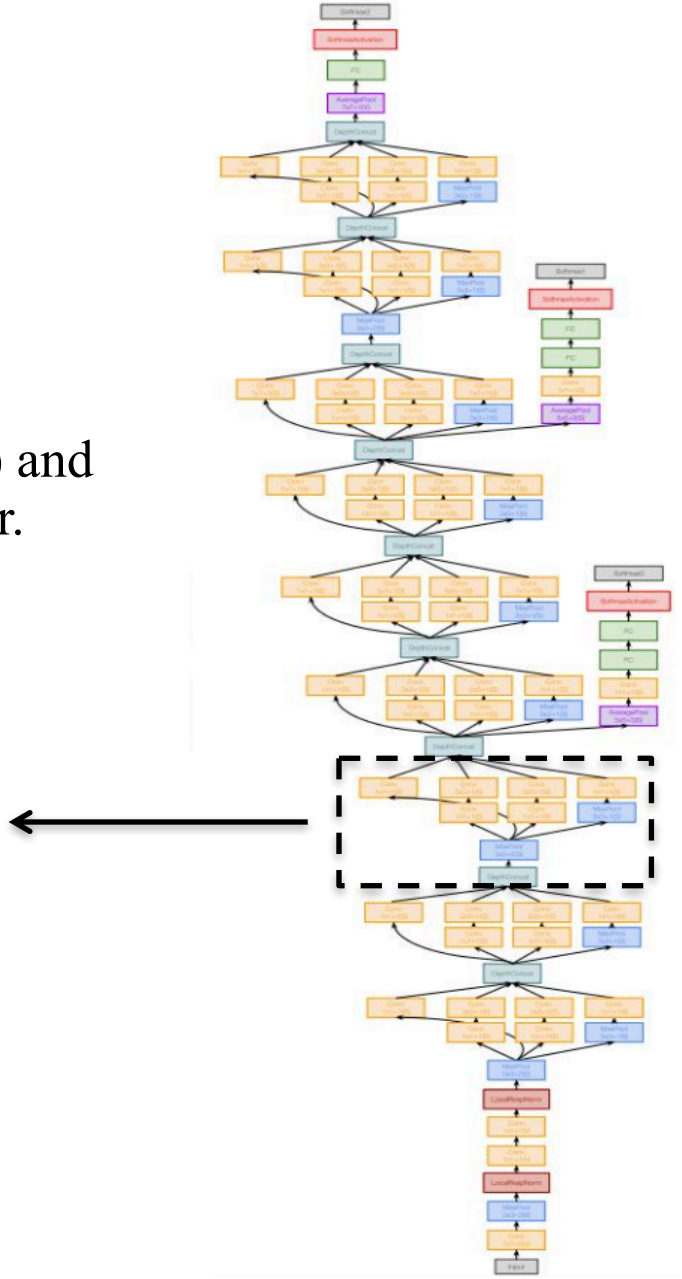
CNN Architecture: Part I

❑ GoogLeNet [Szegedy et al., 2014]

Design a good local network topology (NIN) and then stack these modules on top of each other.



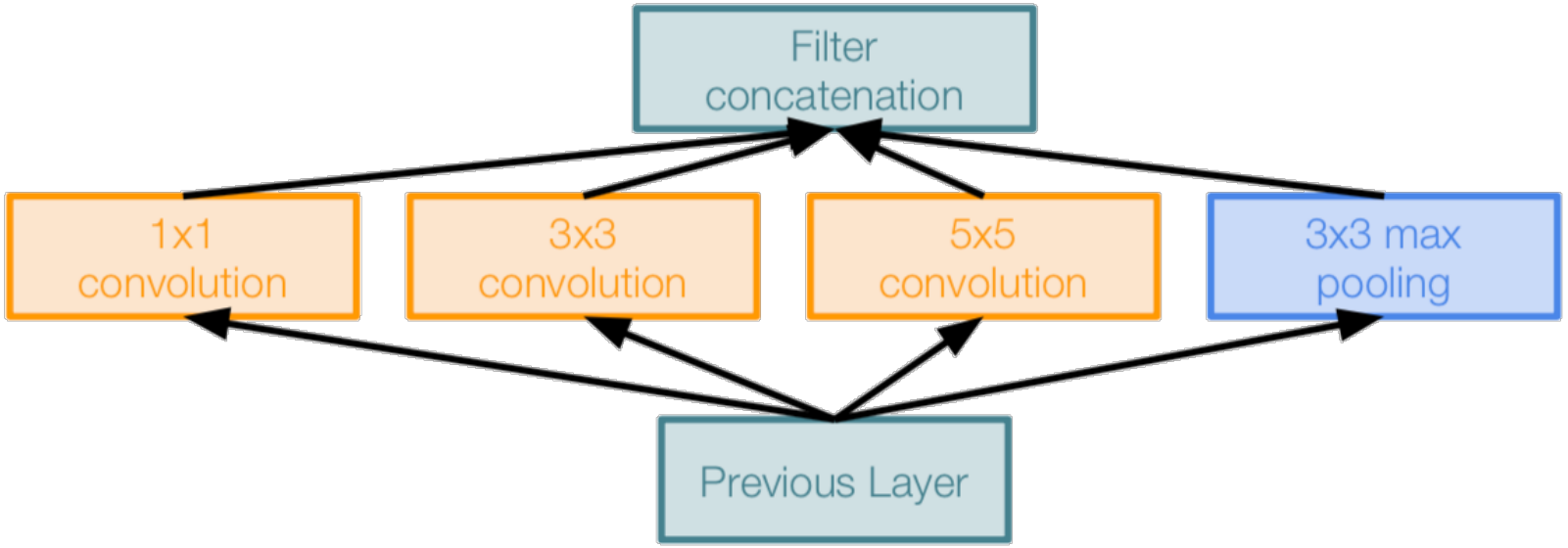
Inception module



CNN Architecture: Part I

- Different receptive fields
- The same feature dimension
- The effectives of pooling

❑ **GoogLeNet** [Szegedy et al., 2014]



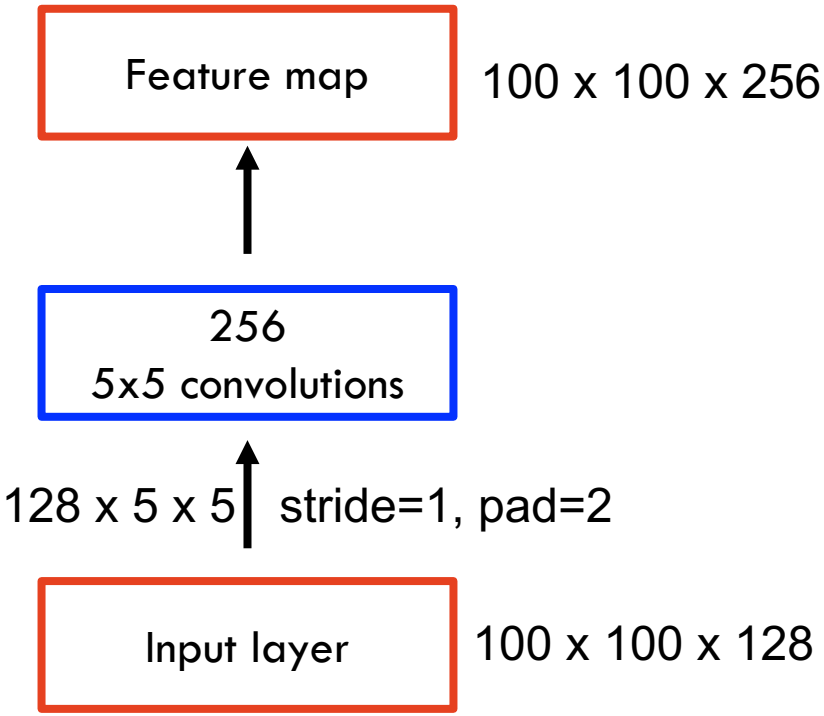
Naive Inception module

$$\text{Output Size} = \frac{N+2P-K}{S} + 1$$

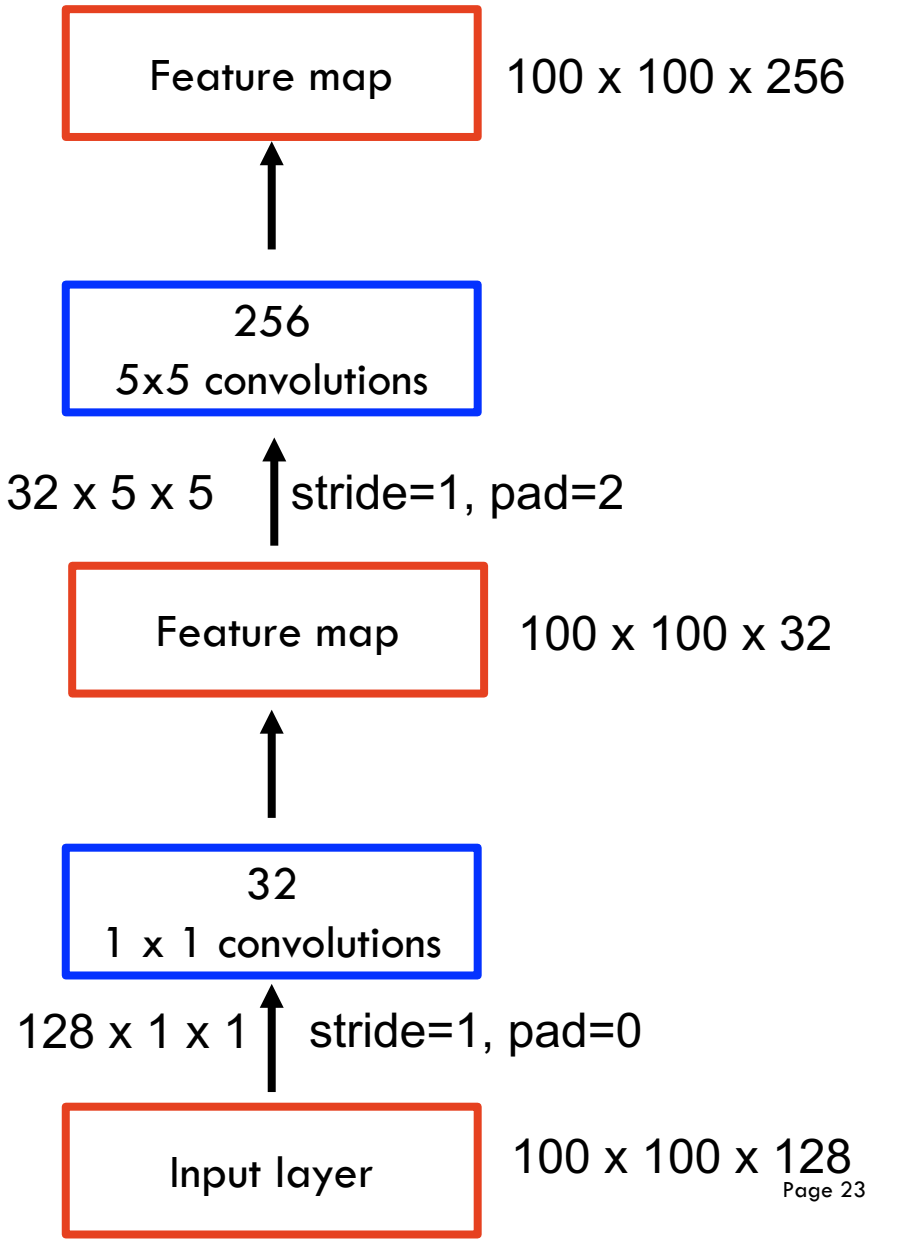
CNN Architecture: Part I

❑ **GoogLeNet** [Szegedy et al., 2014]

#parameter: 128x5x5x256

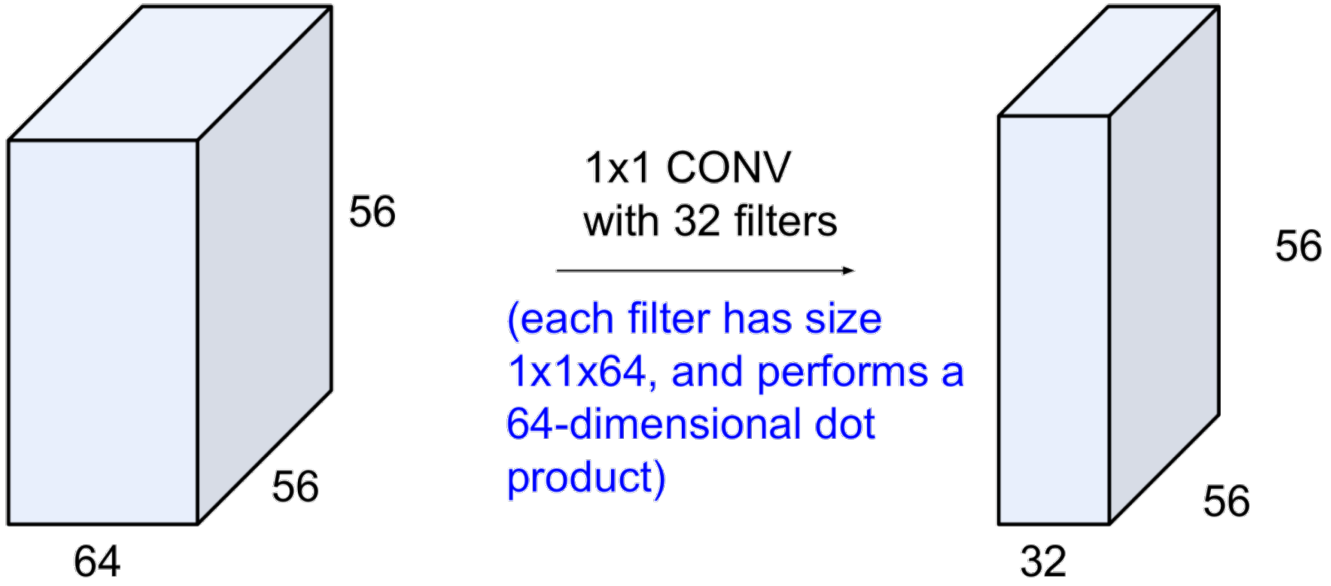


#parameter: 128x1x1x32 + 32x5x5x256



CNN Architecture: Part I

❑ **GoogLeNet** [Szegedy et al., 2014]

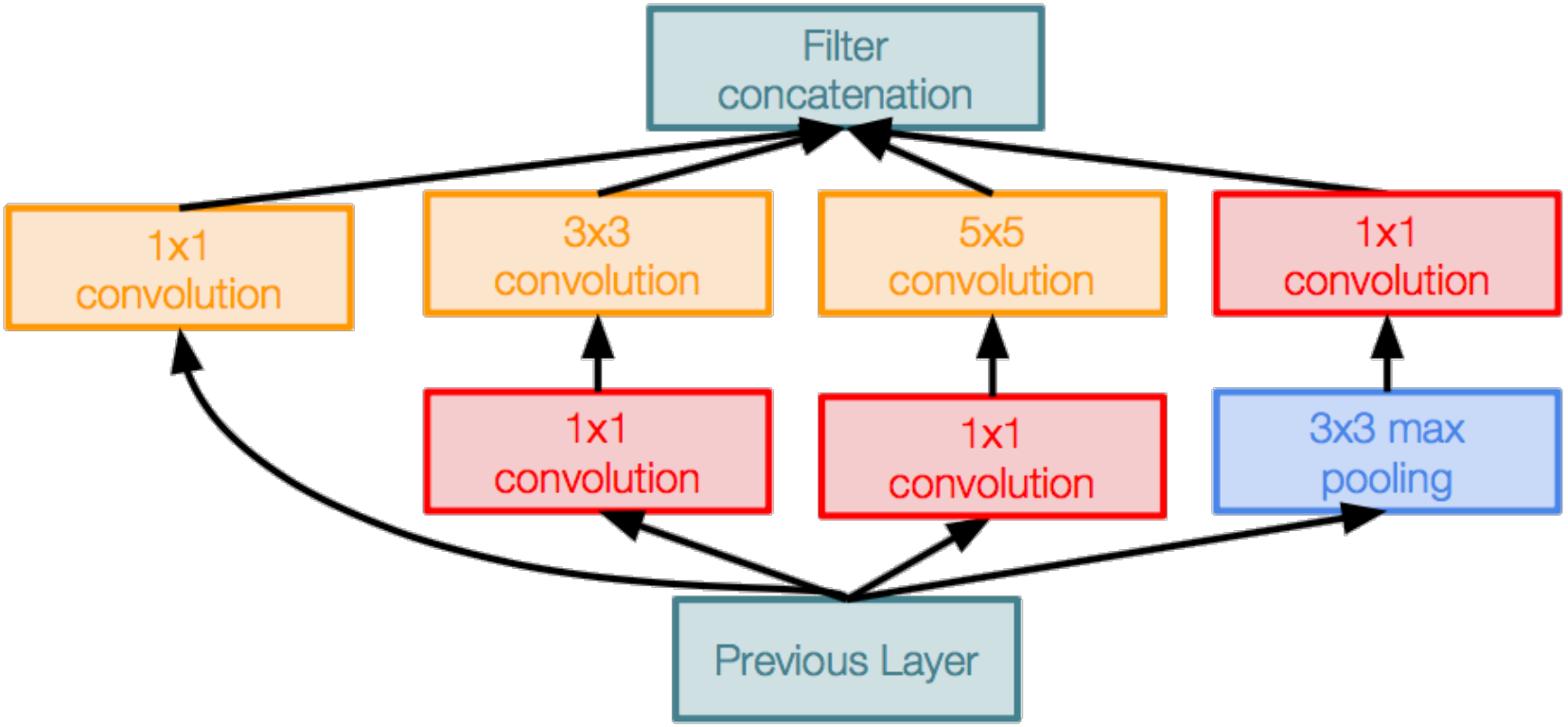


1x1 convolutional layer

- preserve spatial dimensions (56x56)
- reduces depth (64 -> 32)

CNN Architecture: Part I

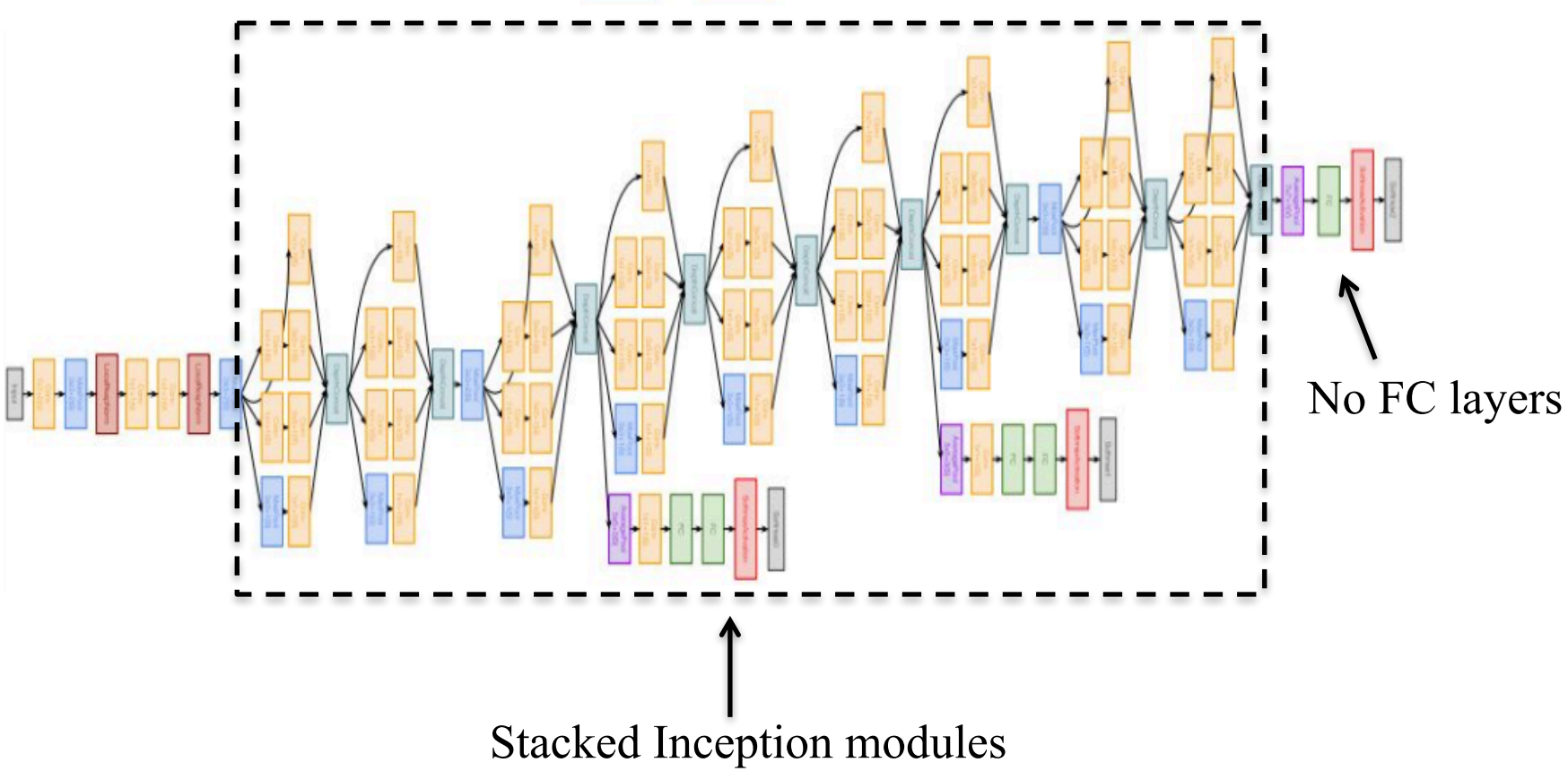
❑ **GoogLeNet** [Szegedy et al., 2014]



Inception module

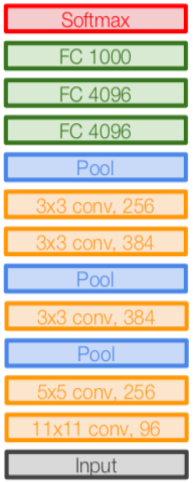
CNN Architecture: Part I

❑ **GoogLeNet** [Szegedy et al., 2014]



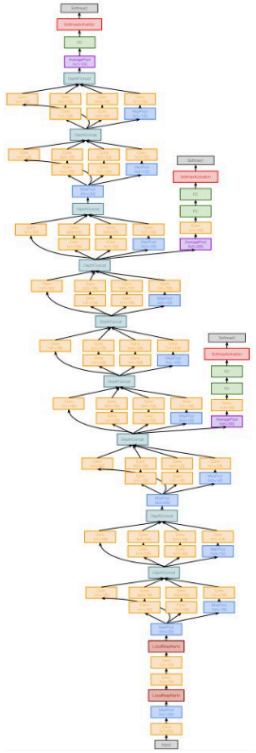
CNN Architecture: Part I

The deeper model should be able to perform at least as well as the shallower model.



AlexNet (9 layers)

VGG16 (16 layers)

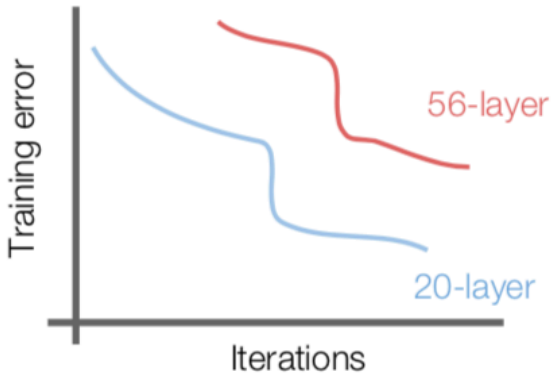


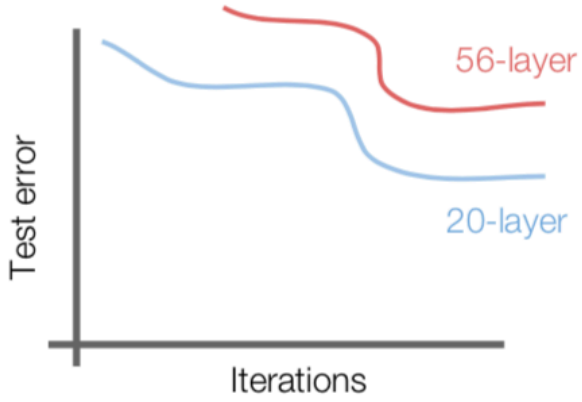
GoogLeNet (22 layers)

CNN Architecture: Part I

❑ ResNet [He et al., 2015]

Stacking deeper layers on a "plain" convolutional neural network
=> the deeper model performs worse, but not overfitting.



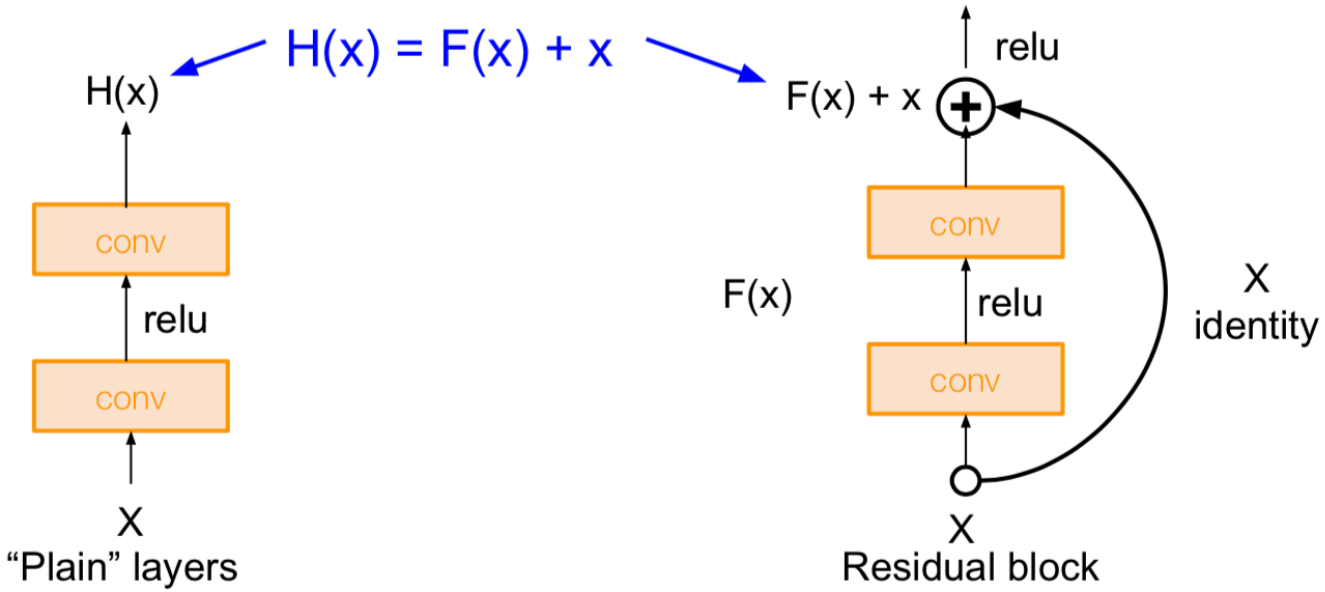


Hypothesis: deeper models are harder to optimize.

CNN Architecture: Part I

❑ ResNet [He et al., 2015]

Solution: use network layers to fit a residual mapping instead of directly trying to fit a desired underlying mapping



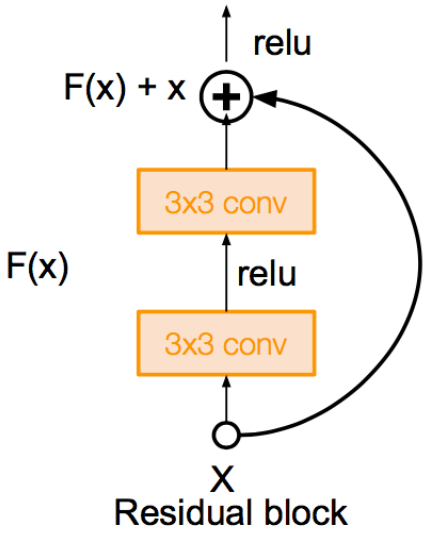
CNN Architecture: Part I

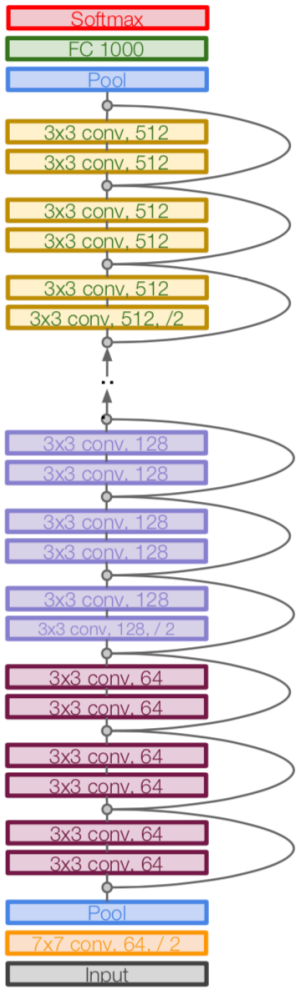
❑ ResNet [He et al., 2015]

Very deep networks using residual connections

Basic design:

- all 3x3 conv (almost)
- spatial size /2 => #filter x2
- just deep
- average pooling (remove FC)
- no dropout



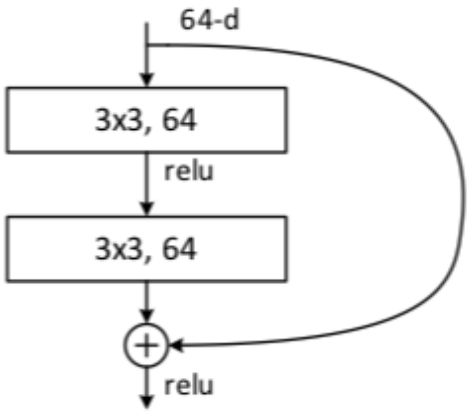


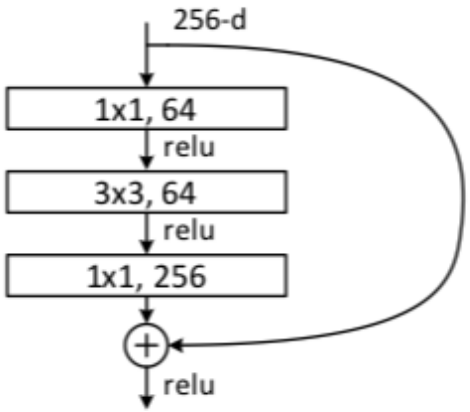
ResNet

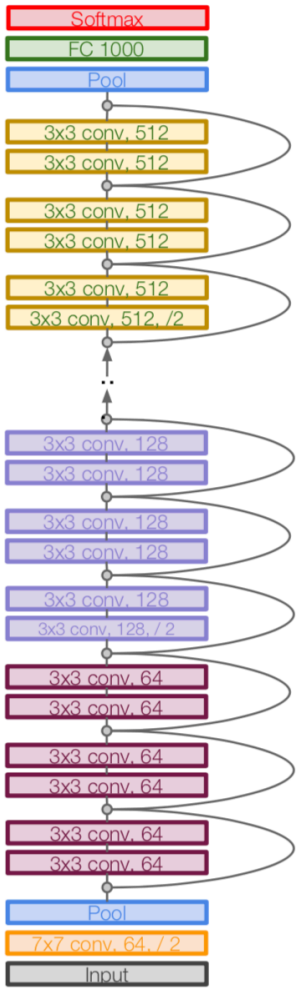
CNN Architecture: Part I

❑ ResNet [He et al., 2015]

Use "**bottleneck**" layer to improve efficiency (similar to GoogLeNet).



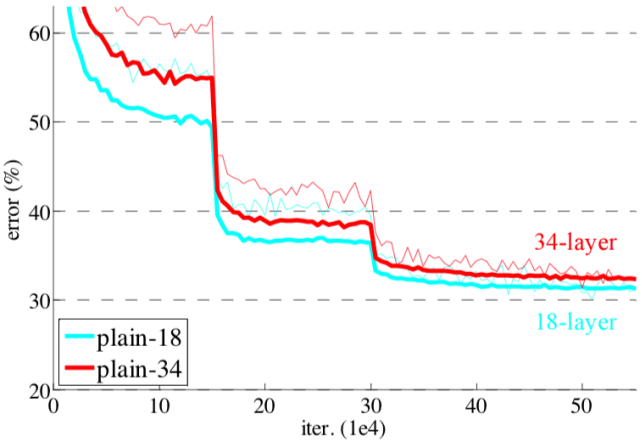


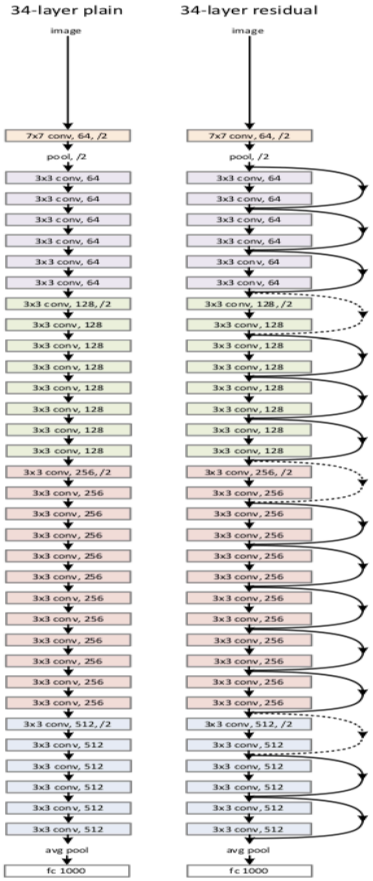


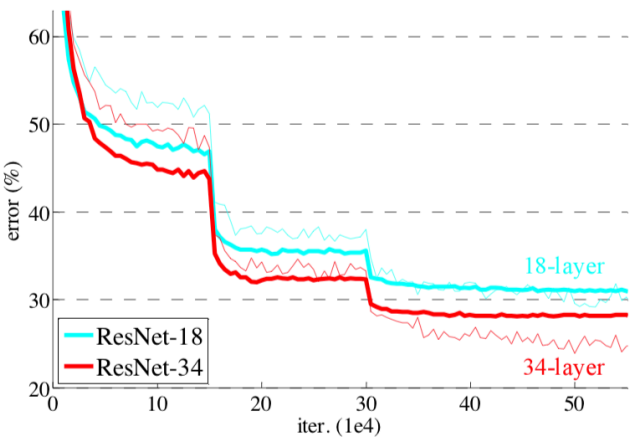
ResNet

CNN Architecture: Part I

- **ResNet** [He et al., 2015]







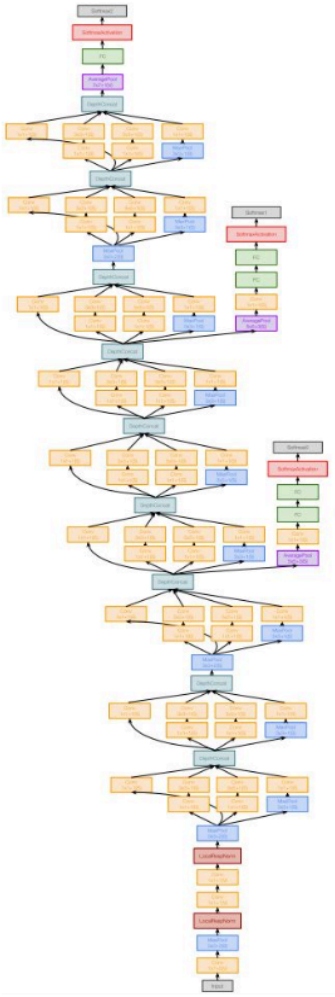
CNN Architecture: Part I

❑ **ResNet** [He et al., 2015]

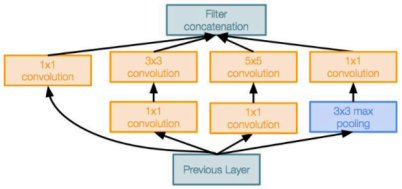
- **1st places in all five main tracks**
  - ImageNet Classification: "*Ultra-deep*" 152-layer nets
  - ImageNet Detection: 16% better than 2nd
  - ImageNet Localization: 27% better than 2nd
  - COCO Detection: 11% better than 2nd
  - COCO Segmentation: 12% better than 2nd

Features matter: deeper features are well transferrable

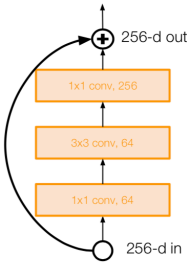
CNN Architecture: Part I



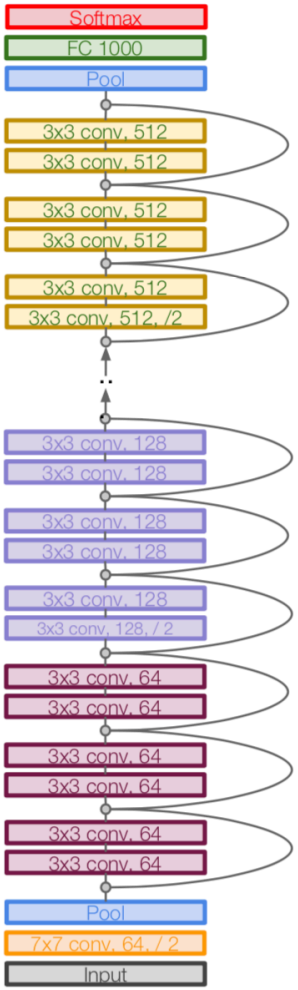
Stacked modules or blocks.



Inception module



Residual block



Advanced CNN Architectures

Advanced CNN Architectures

Inception

Inception v1,v2,v3,v4

Inception-res-v1,v2

Xception

ResNet

Wide-ResNet

ResNeXt

DenseNet

Light CNN Networks

SqueezeNet

MobileNet

ShuffleNet