

LECTURE 11-2

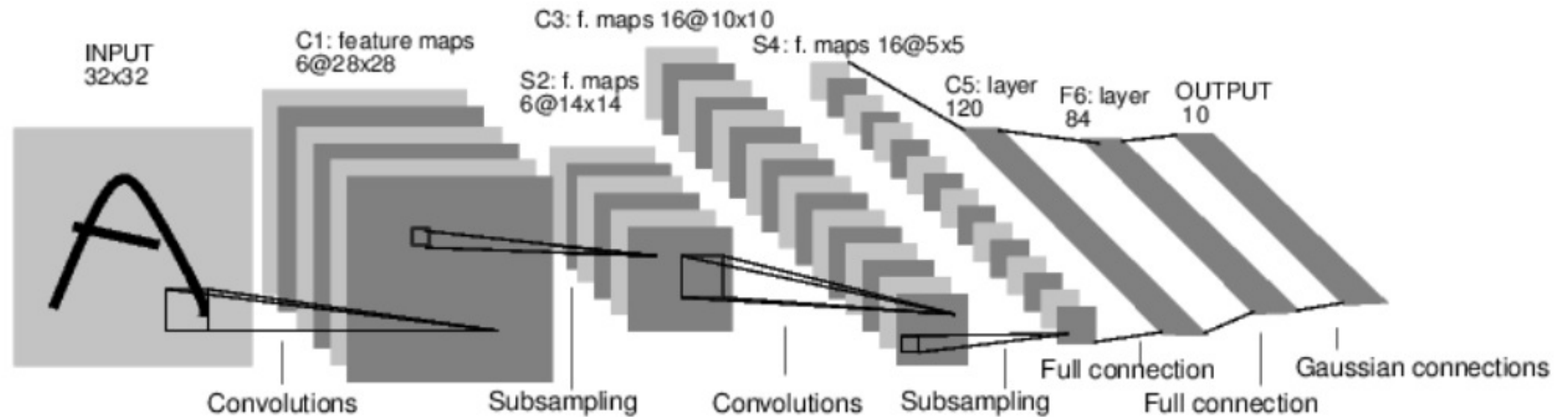
CNN CASE STUDY

Sung Kim <hunkim+ml@gmail.com>
<http://hunkim.github.io/ml>

Convolutional Neural Networks

Case Study : LeNet-5

LeCun et al., 1998



Conv filters were 5x5, applied at stride 1
Subsampling (Pooling) layers were 2x2 applied at stride 2
i.e architecture is [CONV-POOL-CONV-POOL-CONV-FC]

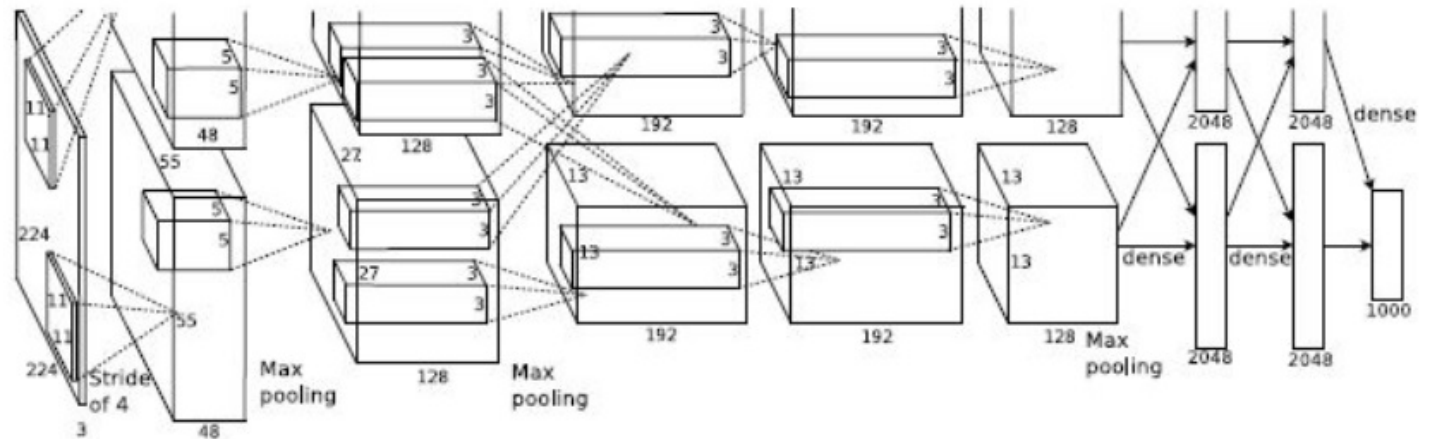
Case Study : AlexNet

Krizhevsky et al. 2012

Input:
227 X 227 X 3 images

First Layer (CONV1):
96 11x11 filters applied at stride 4

-> Output Volume [55x55x96]



Q.

What is the total number of parameters in this layer?

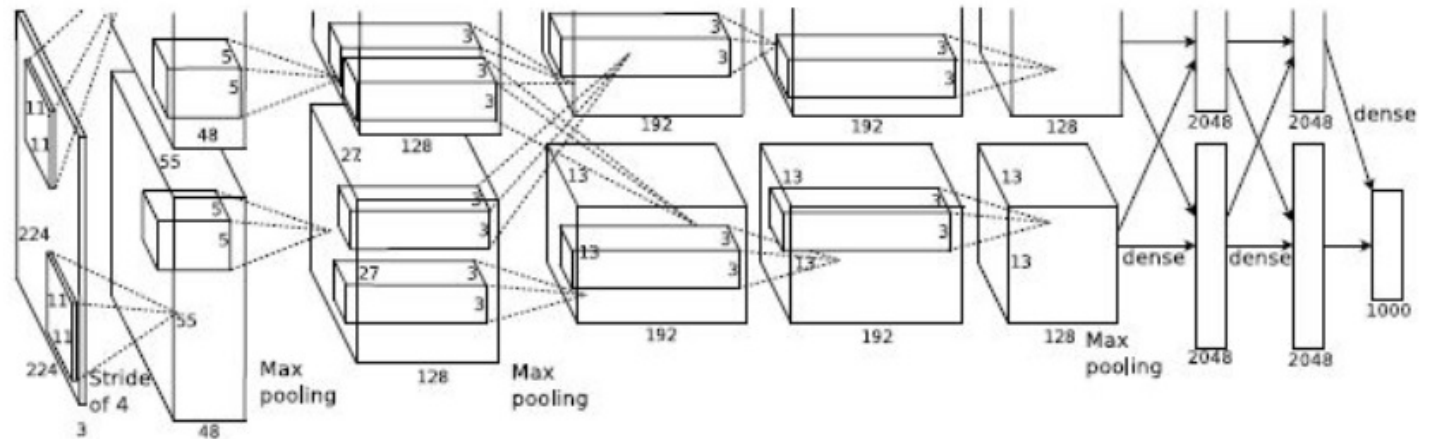
Case Study : AlexNet

Krizhevsky et al. 2012

Input:
227 X 227 X 3 images

First Layer (CONV1):
96 11x11 filters applied at stride 4

->



Q.

What is the output volume size?

Hint: $(227-11)/4+1=55$

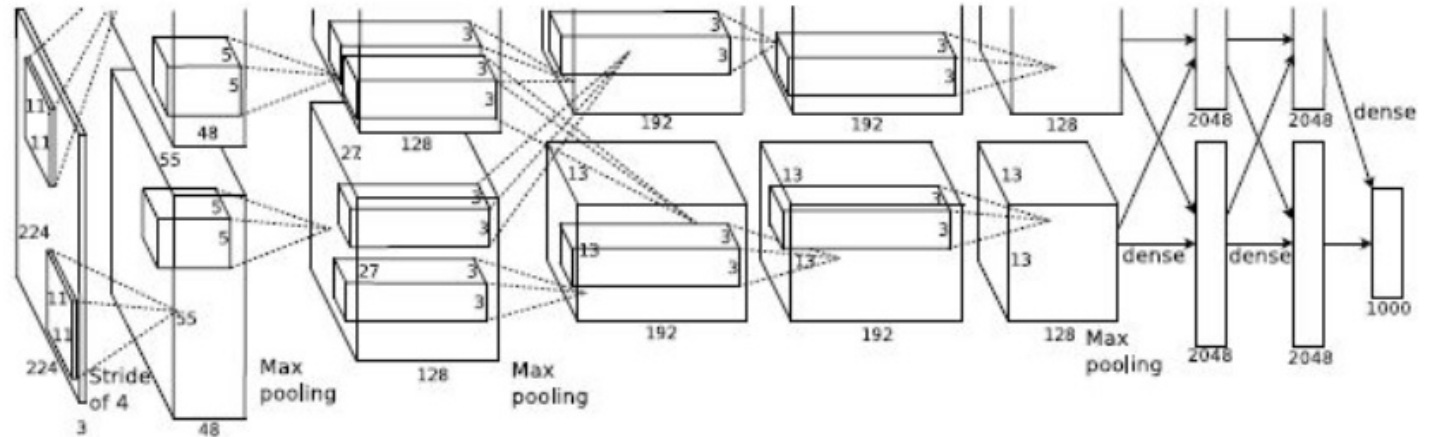
Case Study : AlexNet

Krizhevsky et al. 2012

Input:
227 X 227 X 3 images

First Layer (CONV1):
96 11x11 filters applied at stride 4

-> Output Volume [55x55x96]
Parameters: $(11 \times 11 \times 3) \times 96 = 35k$



Case Study : AlexNet

Krizhevsky et al. 2012

Input: $227 \times 227 \times 3$ images

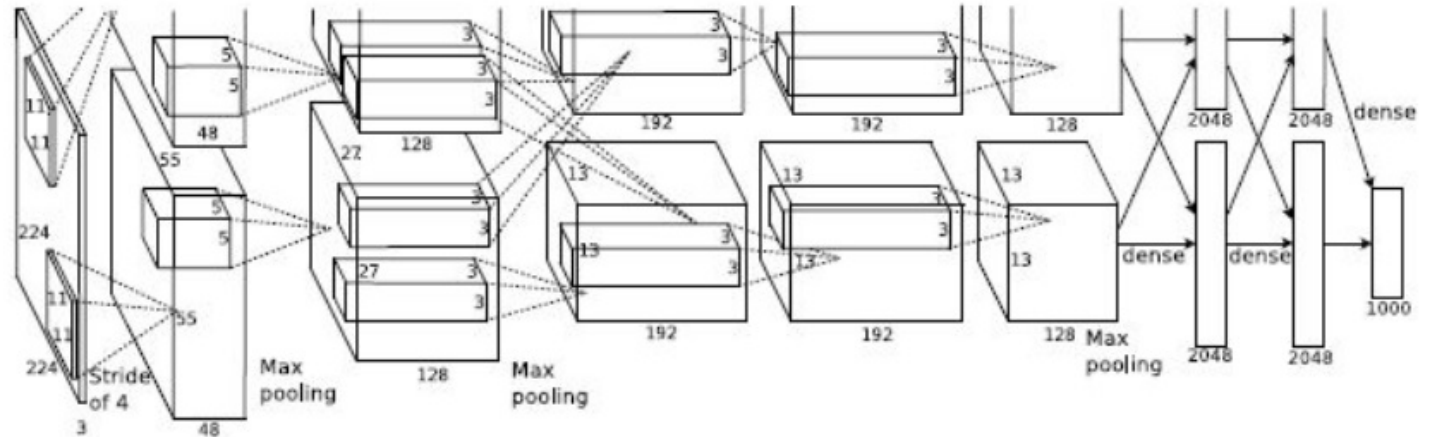
After CONV1: $55 \times 55 \times 96$

Second Layer (POOL1):

3x3 filters applied at stride 2

-> Output Volume: $27 \times 27 \times 96$

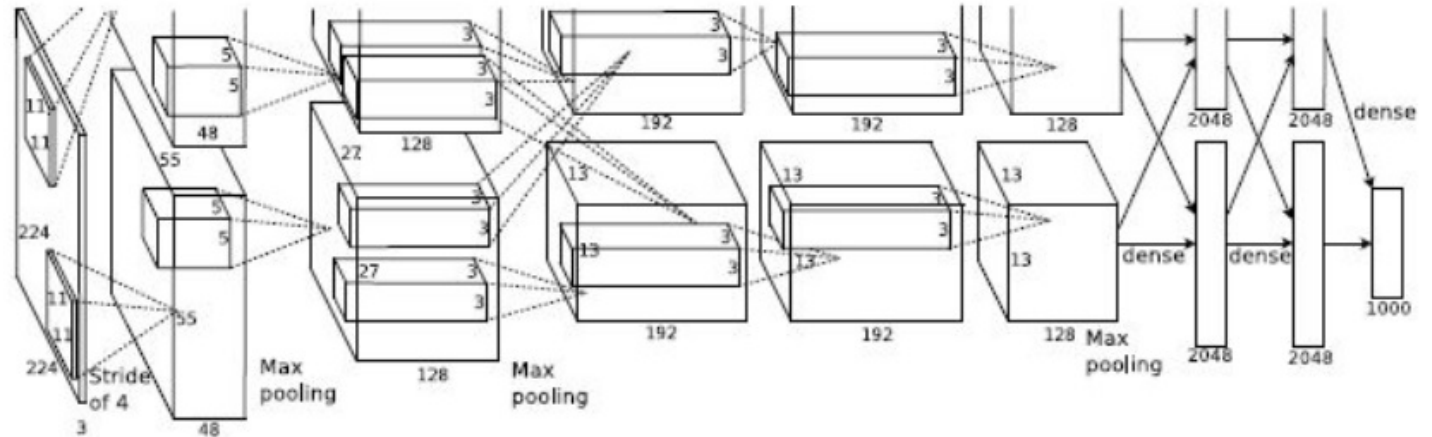
Parameters: 0!



Case Study : AlexNet

Krizhevsky et al. 2012

Input: $227 \times 227 \times 3$ images
After CONV1: $55 \times 55 \times 96$
After POOL1: $27 \times 27 \times 96$
...

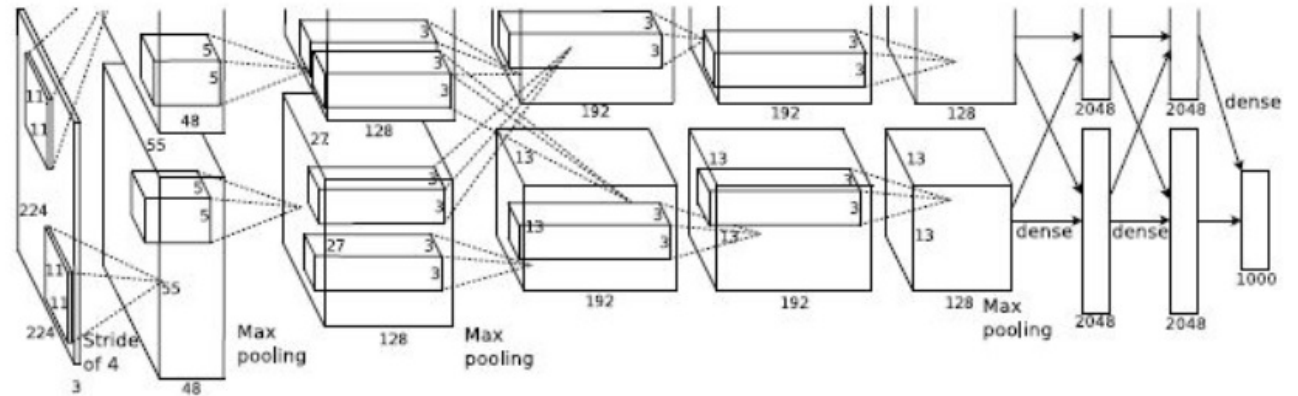


Case Study : AlexNet

Krizhevsky et al. 2012

Full (simplified) AlexNet architecture:

[227 × 227 × 3] INPUT
[27 × 27 × 96] CONV1: 96 11 × 11 filters at stride 4, pad 0
[27 × 27 × 96] MAX POOL1: 3 × 3 filters at stride 2
[27 × 27 × 96] NORM1: Normalization layer
[27 × 27 × 256] CONV2: 256 5 × 5 filters at stride 1, pad 2
[13 × 13 × 256] MAX POOL2: 3 × 3 filters at stride 2
[13 × 13 × 256] NORM2: Normalization layer
[13 × 13 × 384] CONV3: 384 3 × 3 filters at stride 1, pad 1
[13 × 13 × 384] CONV4: 384 3 × 3 filters at stride 1, pad 1
[13 × 13 × 256] CONV5: 256 3 × 3 filters at stride 1, pad 1
[6 × 6 × 256] MAX POOL3: 3 × 3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)



Case Study : AlexNet

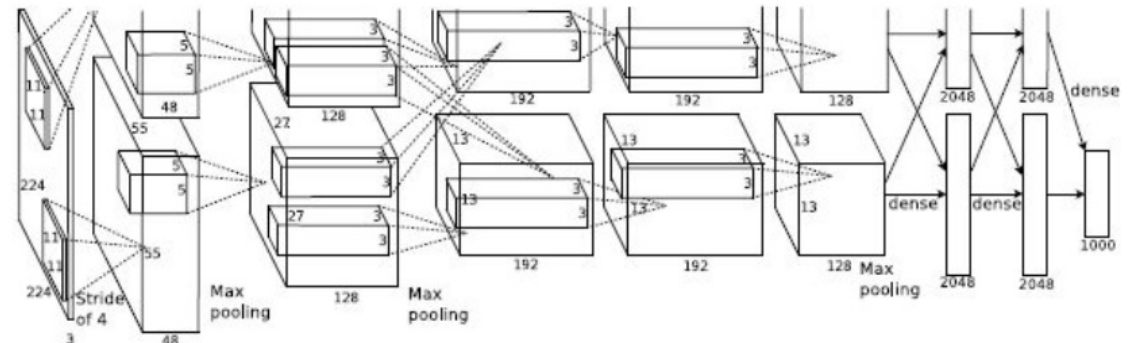
Krizhevsky et al. 2012

Full (simplified) AlexNet architecture:

[227 × 227 × 3] INPUT
[27 × 27 × 96] CONV1: 96 11 × 11 filters at stride 4, pad 0
[27 × 27 × 96] MAX POOL1: 3 × 3 filters at stride 2
[27 × 27 × 96] NORM1: Normalization layer
[27 × 27 × 256] CONV2: 256 5 × 5 filters at stride 1, pad 2
[13 × 13 × 256] MAX POOL2: 3 × 3 filters at stride 2
[13 × 13 × 256] NORM2: Normalization layer
[13 × 13 × 384] CONV3: 384 3 × 3 filters at stride 1, pad 1
[13 × 13 × 384] CONV4: 384 3 × 3 filters at stride 1, pad 1
[13 × 13 × 256] CONV5: 256 3 × 3 filters at stride 1, pad 1
[6 × 6 × 256] MAX POOL3: 3 × 3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)

Details / Retrospectives:

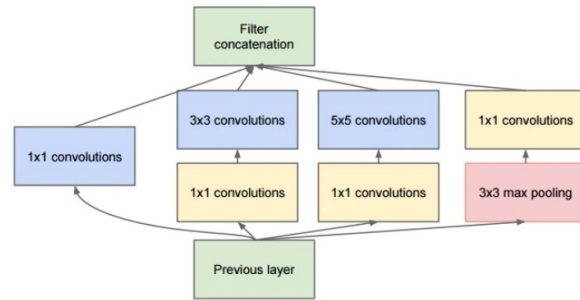
- First use of ReLU
- Used Norm layers (not common anymore)
- Heavy data augmentation
- Dropout 0.5
- Batch size 128
- SGD Momentum 0.9
- Learning rate 1e-2, reduced by 10 manually when val accuracy plateaus
- L2 weight decay 5e-4
- 7 CNN ensemble: 18.2% → 15.4%



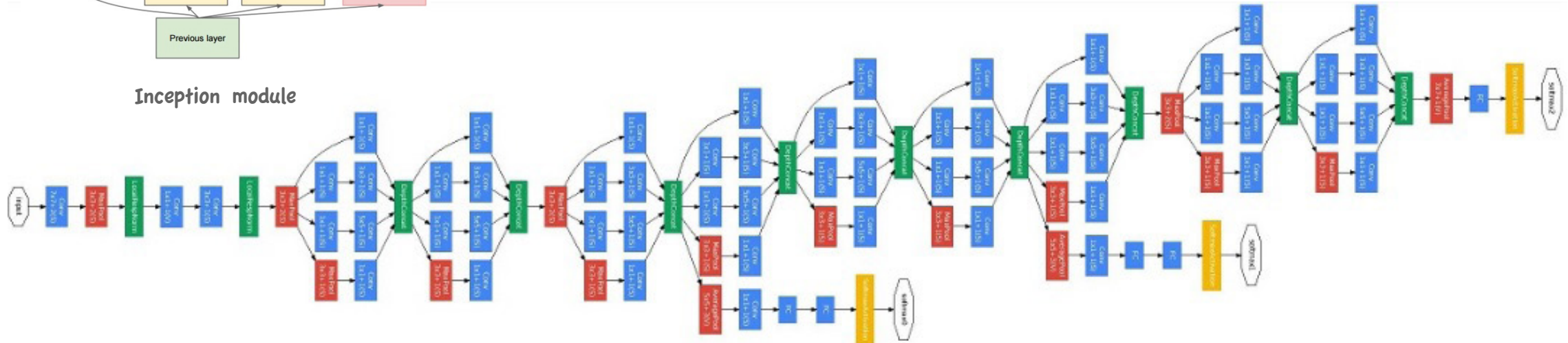
Case Study : GoogLeNet

Szegedy et al., 2014

ILSVRC 2014 winner (6.7% top 5 error)



Inception module



Case Study : ResNet

He et al. 2015

ILSVRC 2015 winner (3.6% top 5 error)

MSRA @ ILSVRC & COCO 2015 Competitions

- 1st places in all 5 main tracks
 - ImageNet Classification : "Ultra-deep" (quote Yann) 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

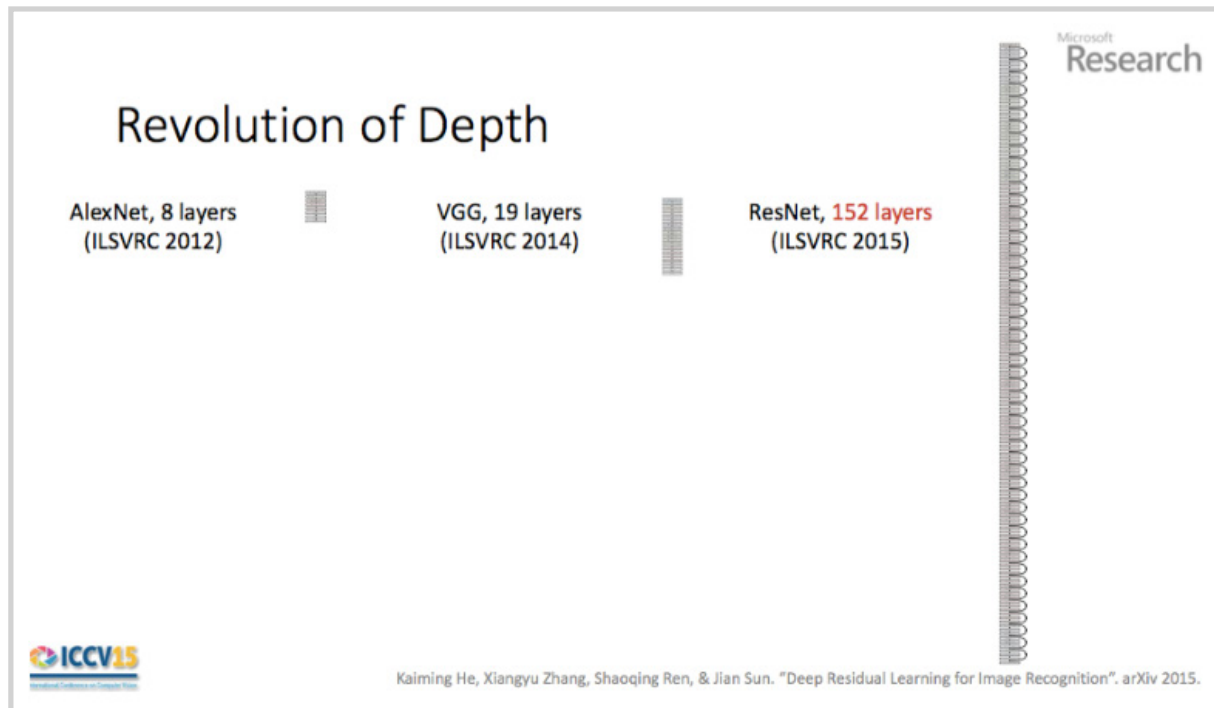
*improvements are relative numbers

Kaiming, Xiangyu Zhang, Shaoqing Ren, & Jian Sun, "Deep Residual Learning for Image Recognition". arXiv 2015.
Slide from Kaiming He's recent presentation <http://www.youtube.com/watch?v=1PGLj-uKT1w>

Case Study : ResNet

He et al. 2015

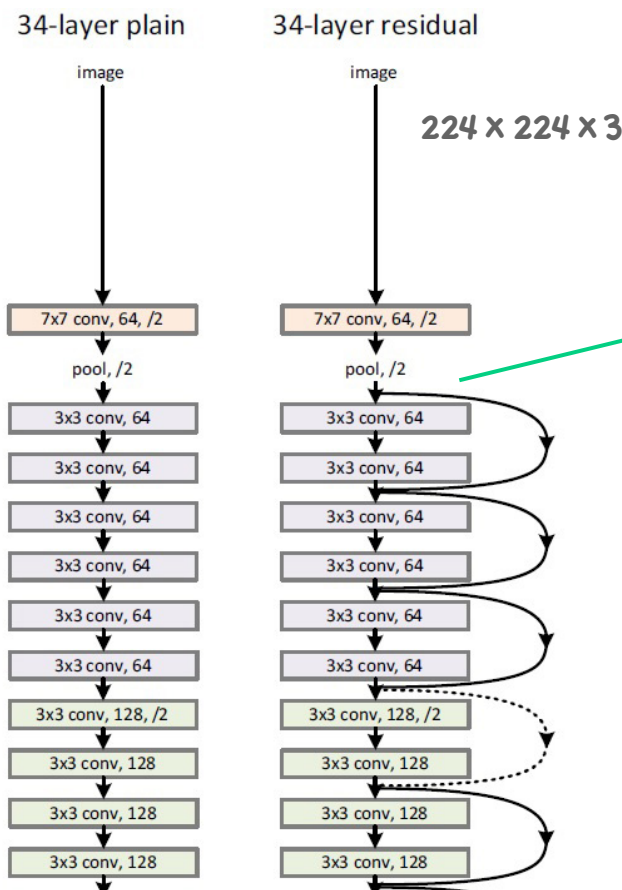
ILSVRC 2015 winner (3.6% top 5 error)



- 2-3 weeks of training on 8 GPU machine
- at runtime : faster than a VGGNet! (even though it has 8x more layers)

Case Study : ResNet

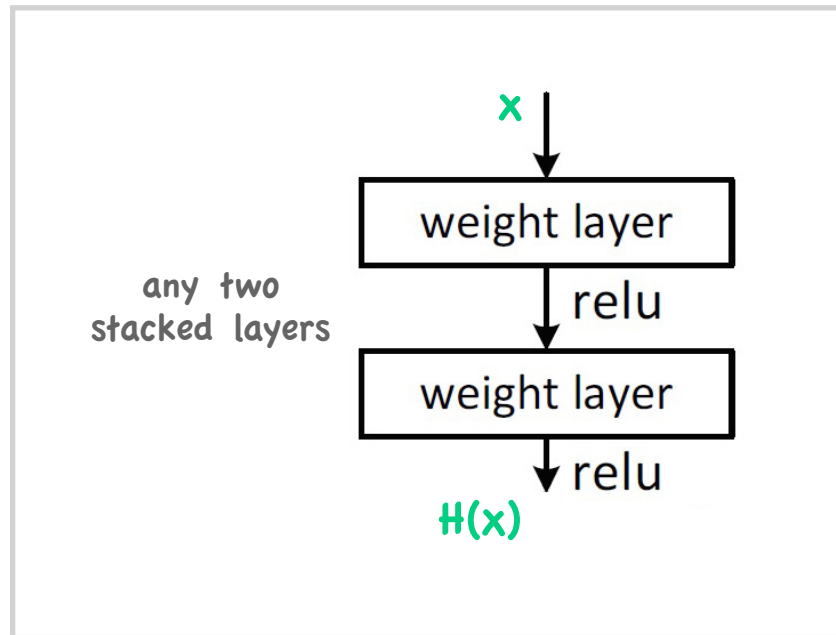
He et al., 2015



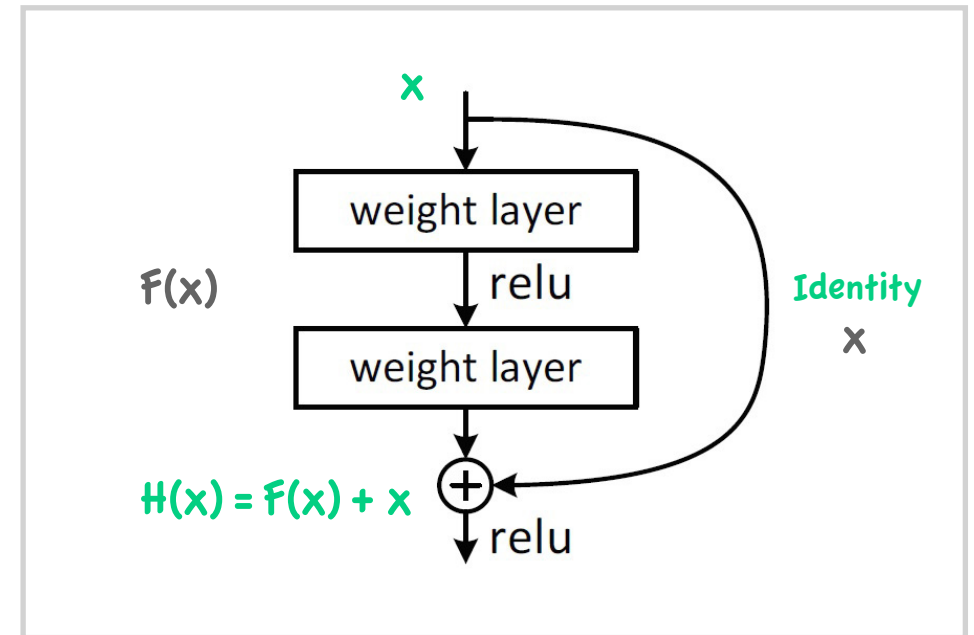
Spatial
dimension
only 56x56!

Case Study : ResNet

He et al., 2015



< Plaint net >



< Residual net >

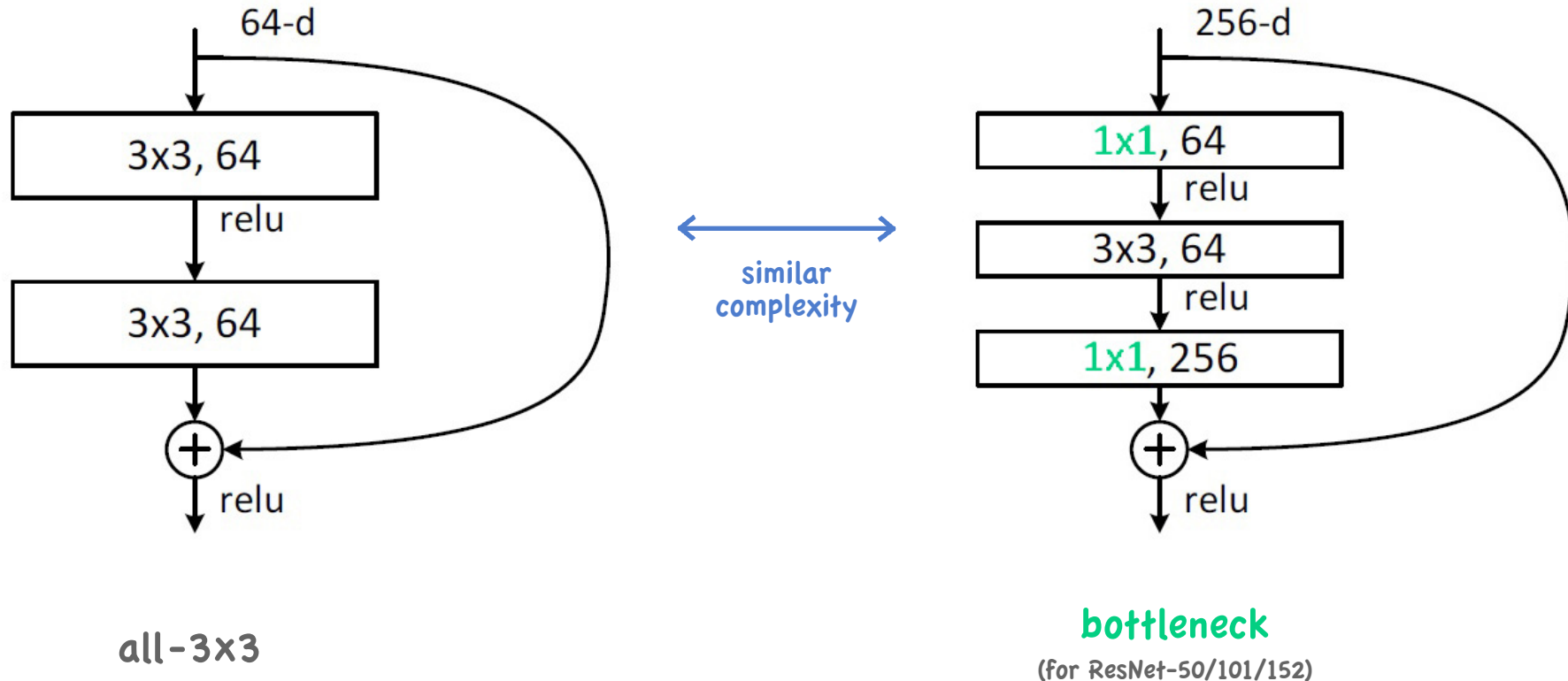
Case Study : ResNet

He et al. 2015

- Batch normalization after every CONV layer
- Xavier/2 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

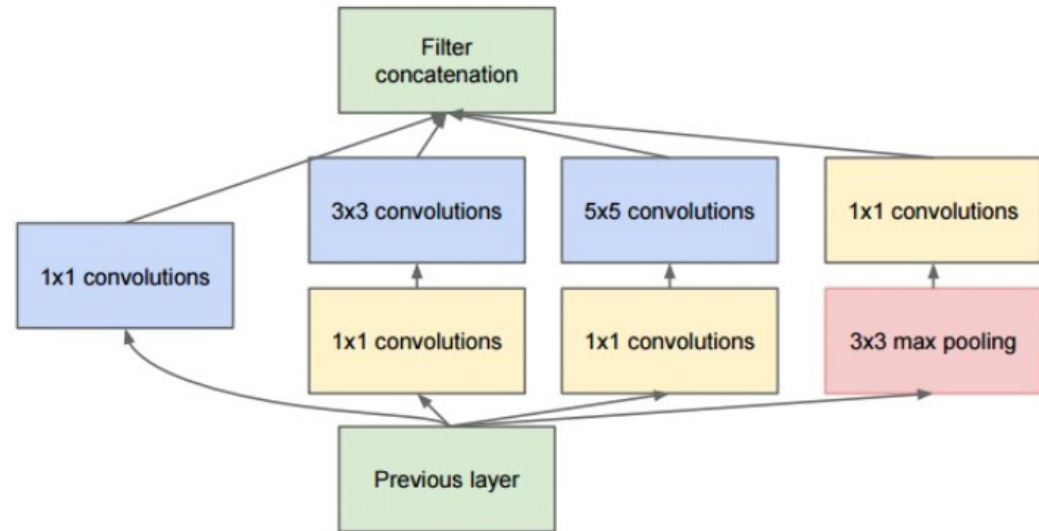
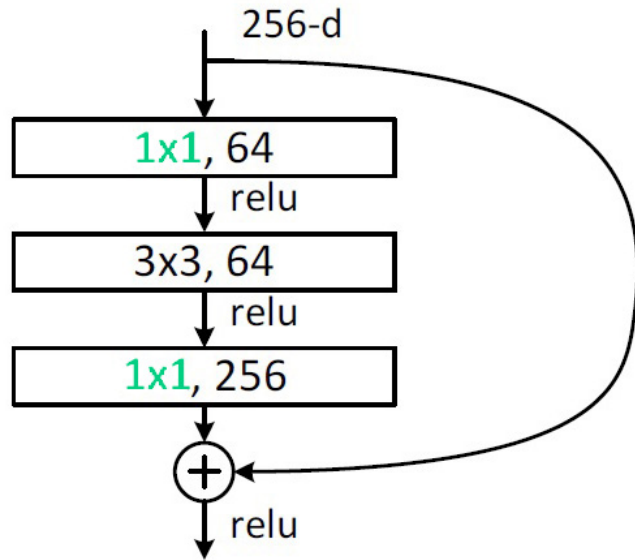
Case Study : ResNet

He et al., 2015



Case Study : ResNet

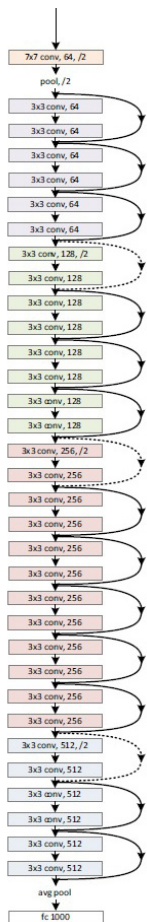
He et al., 2015



(this trick is also used in GoogLeNet)

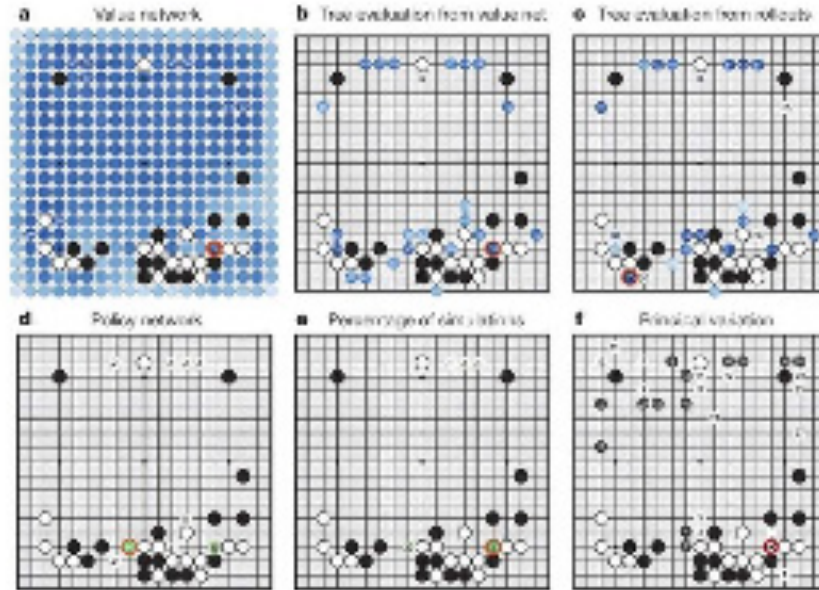
Case Study : ResNet

He et al., 2015



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	$7 \times 7, 64, \text{stride } 2$				
conv2_x	56×56	$3 \times 3 \text{ max pool, stride } 2$				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	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

Case Study Bonus : DeepMind's AlphaGo



The input to the policy network is a $19 \times 19 \times 48$ image stack consisting of 48 feature planes. The first hidden layer zero pads the input into a 23×23 Image, then convolves k filters of kernel size 5×5 with stride 1 with the input image and applies a rectifier nonlinearity. Each of the subsequent hidden layers 2 to 12 zero pads the respective previous hidden layer into a 21×21 image, then convolves k filters of kernel size 3×3 with stride 1, again followed by a rectifier nonlinearity. The final layer convolves 1 filter of kernel size 1×1 with stride 1, with a different bias for each position, and applies a softmax function. The match version of AlphaGo used $k=192$ filters; [Fig.2b](#) and [Extended Data Table 3](#) additionally show the results of training with $k=128, 256$ and 384 filters.

Policy Network :

[$19 \times 19 \times 48$] Input

CONV1 : $192 \times 5 \times 5$ filters, stride1, pad2 => [$19 \times 19 \times 192$]

CONV2...12 : $192 \times 3 \times 3$ filters, stride1, pad1 => [$19 \times 19 \times 192$]

CONV : 1×1 filter, stride1, pad0 => [19×19] (probability map of promising moves)

Convolutional Neural Networks for Sentence Classification

Yoon Kim., 2014

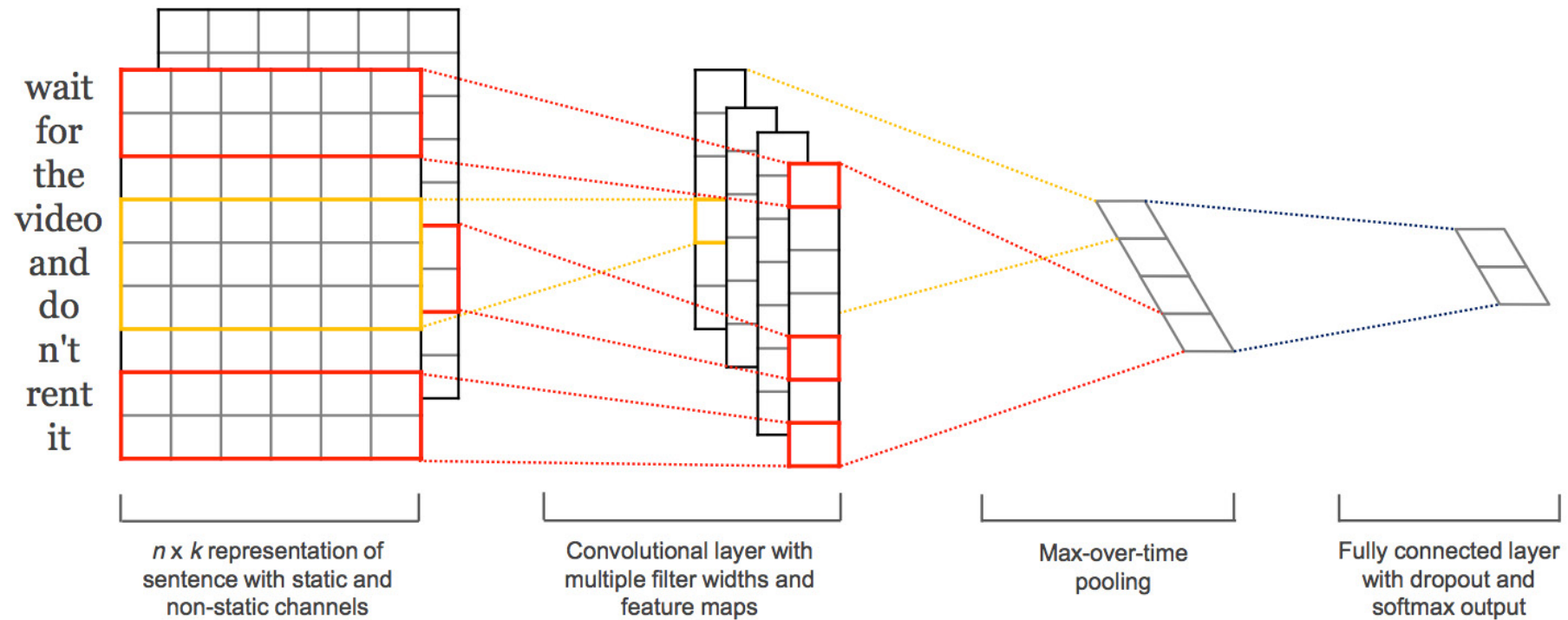


Figure 1: Model architecture with two channels for an example sentence.

NEXT LECTURE

RECURRENT NEURAL NETS (RNN)