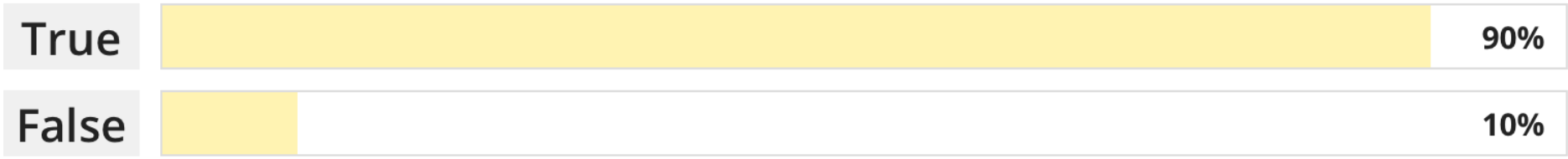
CS 190I
Deep Learning
Residual Network and other CNN variants

Lei Li (leili@cs)
UCSB

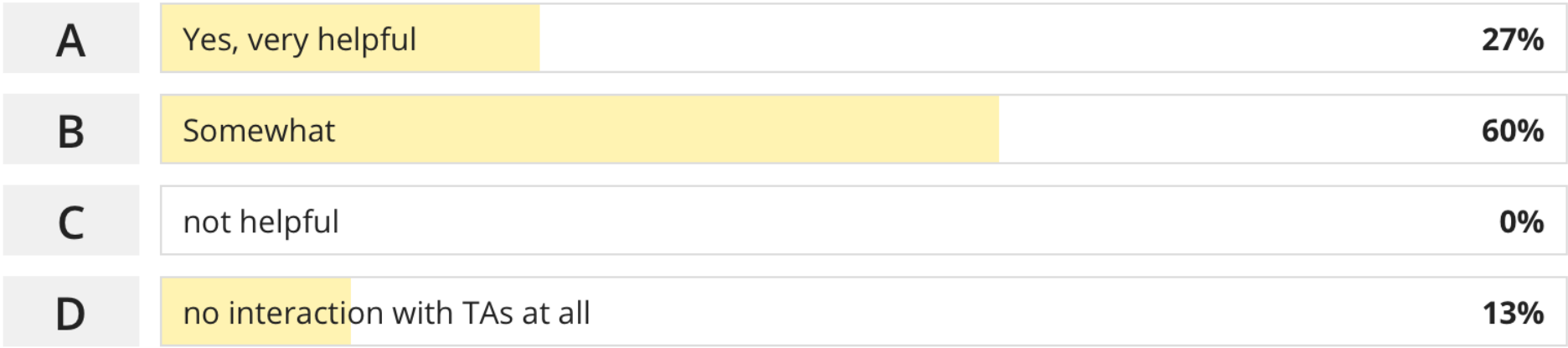
Acknowledgement: Slides borrowed from Bhiksha Raj's 11485 and Mu Li & Alex Smola's 157 courses on Deep Learning, with modification

Survey Result

Does the course content so far meet your expectation?



Are TAs helpful in your learning?



Survey Result

What are the most significant things you have learned thus far from taking this course?

I have learned how Convolutional Neural Networks works and how to use them in machine learning.

More specifics on backpropagation, basics of vector calculus

Learned about the mathematics behind deep learning. For example using argmin and understanding how the softmax activation function works. Also learning to compute backpropagation in a neural network.

L1 & L2 Regularzation / dropout

I think the most significant thing I learned right now is to calculate the gradient.

The use of PyTorch, forward and backward propagation.

all course content is fairly new to me, it is all very interesting

The most significant things that I have learned is the details of linear regression and introduction to linear descent and forward/backward propagation.

Survey Result

Does the instructor clearly explain course slides? Is it too fast or too slow? Do you have any suggestions for improvement for slides?

I like the enthuisiasm with the instructor. The pace is good and the material is good as well. However, I believe some explicit examples would help understanding the material more.

I think the course is going a bit fast right now. Also, maybe have less equations on the slides because they are kind of hard to follow. I think adding simple examples would be more beneficial to understand the concept at a high level, then introduce the equations.

Yes, the instructor explains the course slides well. I think that at times he assumes that we understand the math equations even though they are hard to grasp.

Professor is very knowledgeable in the subject and explains the concepts well. He is easily able to answer questions.

Professor Li does a fine job in explaining the course slides, but I feel the slides could be slightly more compact (somtimes there is too little on one slide, but it takes a long time to go thru it).

The slides don't contain enough information to be an adequate resource if you didn't understand during the lecture, and the lectures either go too slow on prior taught material, and goes too fast over new material. Having examples that the professor works through, especially simple ones would be amazing!

Survey Result

If I do the reading or study the slides prior to class, the lecture are easy to follow. But on days I don't, the lecture are hard to follow.

Sometimes there is a lot of content on a single slide and it is hard to understand how the details follow from each other.

Would be more helpful by going over examples of code in class to demonstrate the theory in practice

Yes, the instructor clearly explains course slides and goes the right pace

A little fast sometimes, and if one wishes to go back to the slides later it is somewhat hard to understand what's going on.

Too fast. Had a hard time to catch

It is a little fast for me, but the lectures are clear!

It is a bit too fast, I wish the slides were a bit more easy to digest instead of being very math and notation heavy, because I can read the textbook for that. The slides should be able to summarize the key concepts (conceptually) so I can gain a broad understanding of what is going on, then I can study the details on my own.

Survey Result

Sometimes I need the instructor to reword the content so that I could fully understand the material.

Maybe we went too fast. Slides could include more examples relating to the quiz or HWs.

The second slides where we talked about vector calculus were pretty fast. Everything after that is fine and I am able to catch.

Too fast when introduced math logic behind the ML, but can be solved by self-learning and math-lab

Sometimes I think the slides are a bit too dense / have too many equations, which are hard to take in all at once. It would also be helpful if the slides included more concrete examples of how to implement the problem solving techniques / concepts being taught.

Sometimes a bit too slow spending too much time on derivations but overall enjoyable lectures.

Yes.More quiz.

Survey Results

I think the speed of the course right now is forcing me to spend a lot more time than I expected. This is not a bad thing since I actually enjoy the content.

I would like to have time for the MP's and understand the basis like setting up the GPU and how to write the code. The code seems more complicated since I haven't used pytorch before. The section covered an example but just went cell by cell in a notebook.

I'm not sure if I'm prepared enough to complete MP1 as we have learned a lot of techniques but not much on how they are implemented in PyTorch.

The math can be tough

The slides/homework/etc. feel pretty disconnected with the textbook. I wish there were more resources to learn what is needed.

Too many people coughing and not wearing masks, makes it hard to focus.

I wish there was a better textbook that explained the math of the theory, not code implementations

The content is sometimes too abstract. But that’s normal for explaining DL concepts.

It's a tough topic. I need to go over the course material more.

Recap

- Convolutional layer
  - Reduced model capacity compared to dense layer
  - Efficient at detecting spatial pattens
  - High computation complexity
  - Control output shape via padding, strides and channels
- Max/Average Pooling layer
  - Provides some degree of invariance to translation

2-D Convolution Layer

$$y_{i,j} = \sum_{a=1}^{h} \sum_{b=1}^{w} w_{a,b} x_{i+a,j+b}$$

Input

0	0	0	0	0
0	0	1	2	0
0	3	4	5	0
0	6	7	8	0
0	0	0	0	0

*

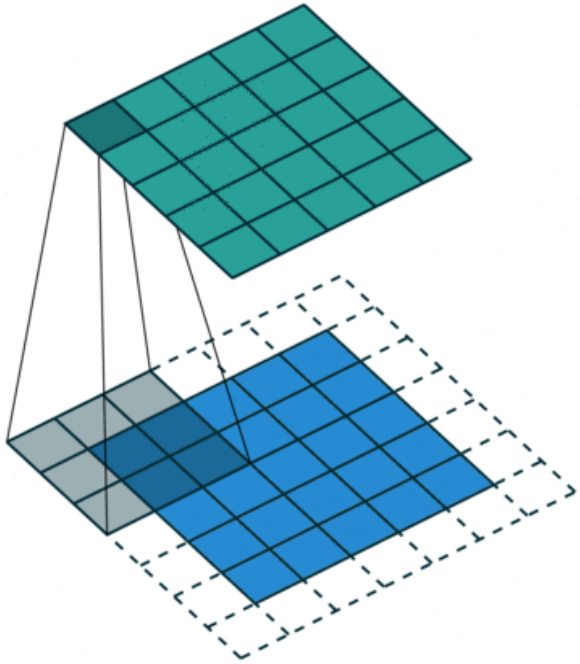
Kernel

0	1
2	3

=

Output

0	3	8	4
9	19	25	10
21	37	43	16
6	7	8	0



$$0 \times 0 + 0 \times 1 + 0 \times 2 + 0 \times 3 = 0$$

2-D Convolution Layer Summary

- Input $\mathbf{X} : c_i \times n_h \times n_w$
- Kernel $\mathbf{W} : c_o \times c_i \times k_h \times k_w$
- Bias $\mathbf{B} : c_o$
- Output $\mathbf{Y} : c_o \times m_h \times m_w$

$$\mathbf{Y} = \mathbf{X} \star \mathbf{W} + \mathbf{B}$$

- Complexity (number of floating point operations FLOP)

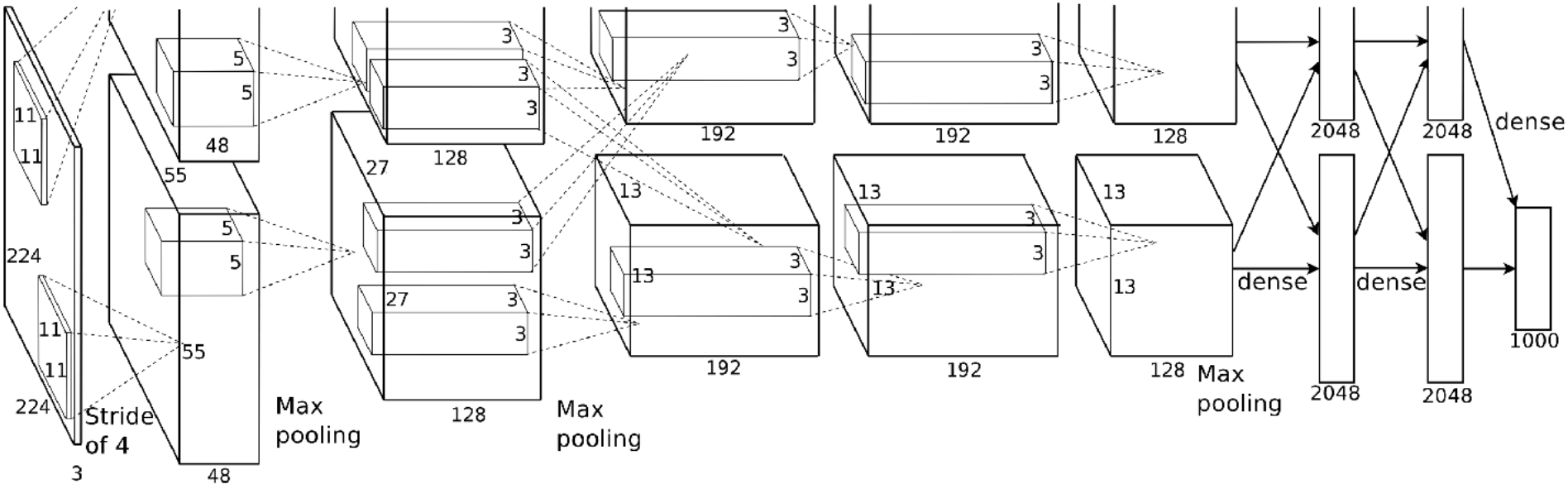
$$c_i = c_o = 100$$
$$k_h = h_w = 5$$
$$m_h = m_w = 64$$

$$O(c_i c_o k_h k_w m_h m_w)$$

1GFLOP

- 10 layers, 1M examples: 10PF
  (CPU: 0.15 TF = 18h, GPU: 12 TF = 14min)

AlexNet


224
224
3
11
11
11
11
Stride
of 4
48
55
5
5
48
55
5
5
Max
pooling
27
128
3
3
3
27
128
3
3
3
Max
pooling
192
192
128
3
3
3
3
3
3
13
13
13
13
13
13
192
192
128
3
3
3
3
3
3
Max
pooling
dense
dense
2048
2048
2048
2048
dense
1000

SVM

- In the 1990s, algorithms based on support vector machines (SVM) are developed
- Kernel methods
- There are (shallow) models
- Linear classifier with margin loss (hinge loss)

Vladimir **V**apnik

Computer Vision Pre-2012

- Extract features
- Describe geometry (e.g. multiple cameras) analytically
- **(Non)Convex** optimization problems
- Many beautiful theorems …
- Works very well in theory when the assumptions are satisfied

Feature Engineering

- Feature engineering is crucial
- Feature descriptors, e.g. SIFT (Scale-invariant feature transform), SURF
- Bag of visual words (clustering)
- Then apply SVM …

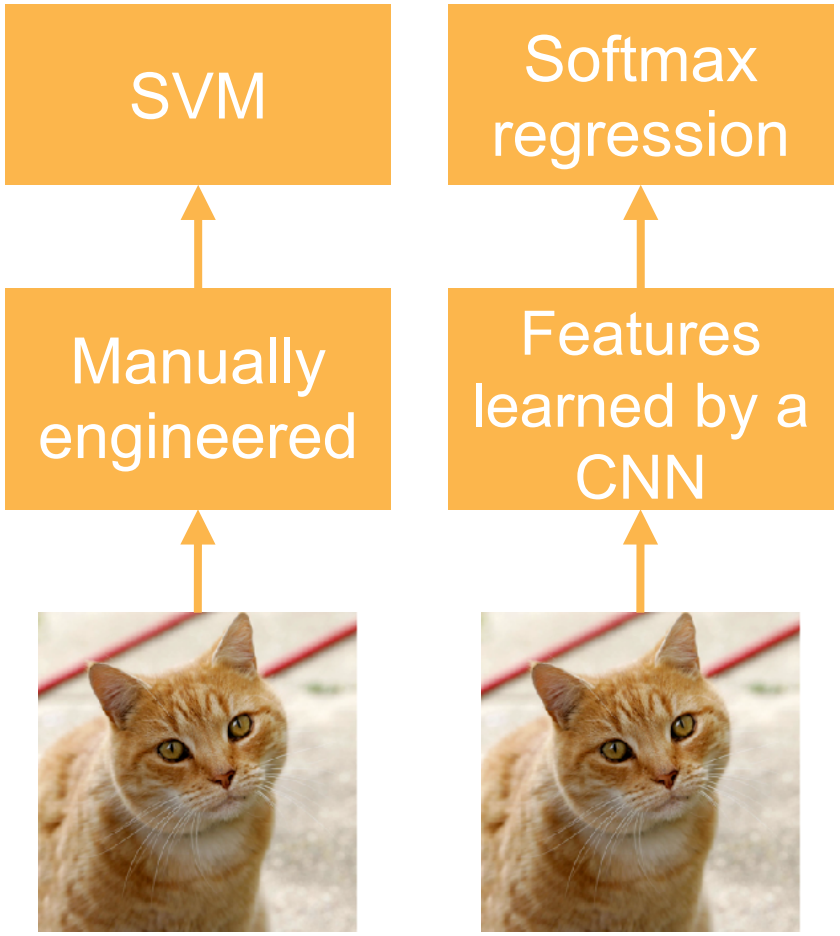
(opencv)

ImageNet (2010)

Images	Color images with nature objects	Gray image for hand-written digits
Size	469 x 387	28 x 28
# examples	1.2 M	60 K
# classes	1,000	10

AlexNet

- AlexNet won ImageNet competition in 2012
- Deeper and bigger LeNet
- Key modifications
  - Dropout (regularization)
  - ReLu (training)
  - MaxPooling
- Paradigm shift for computer vision

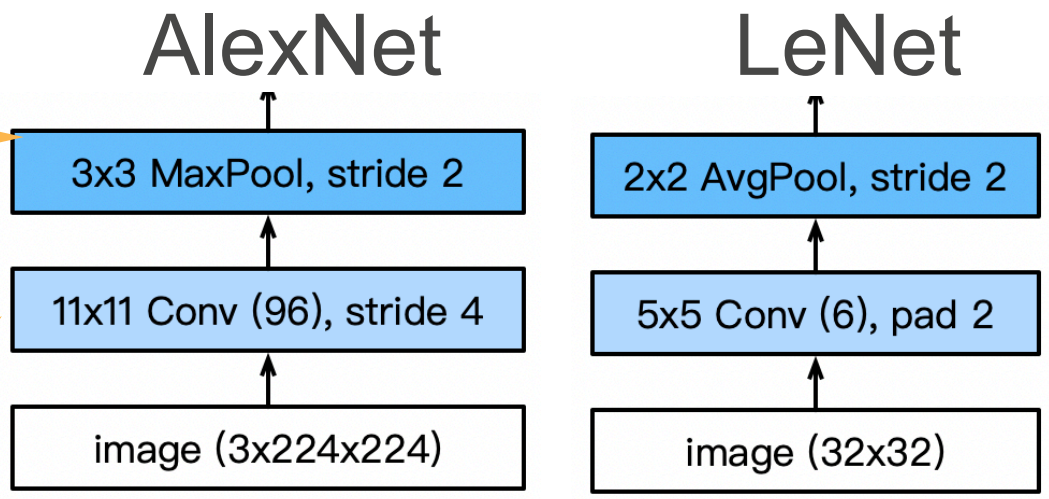


Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. 2012

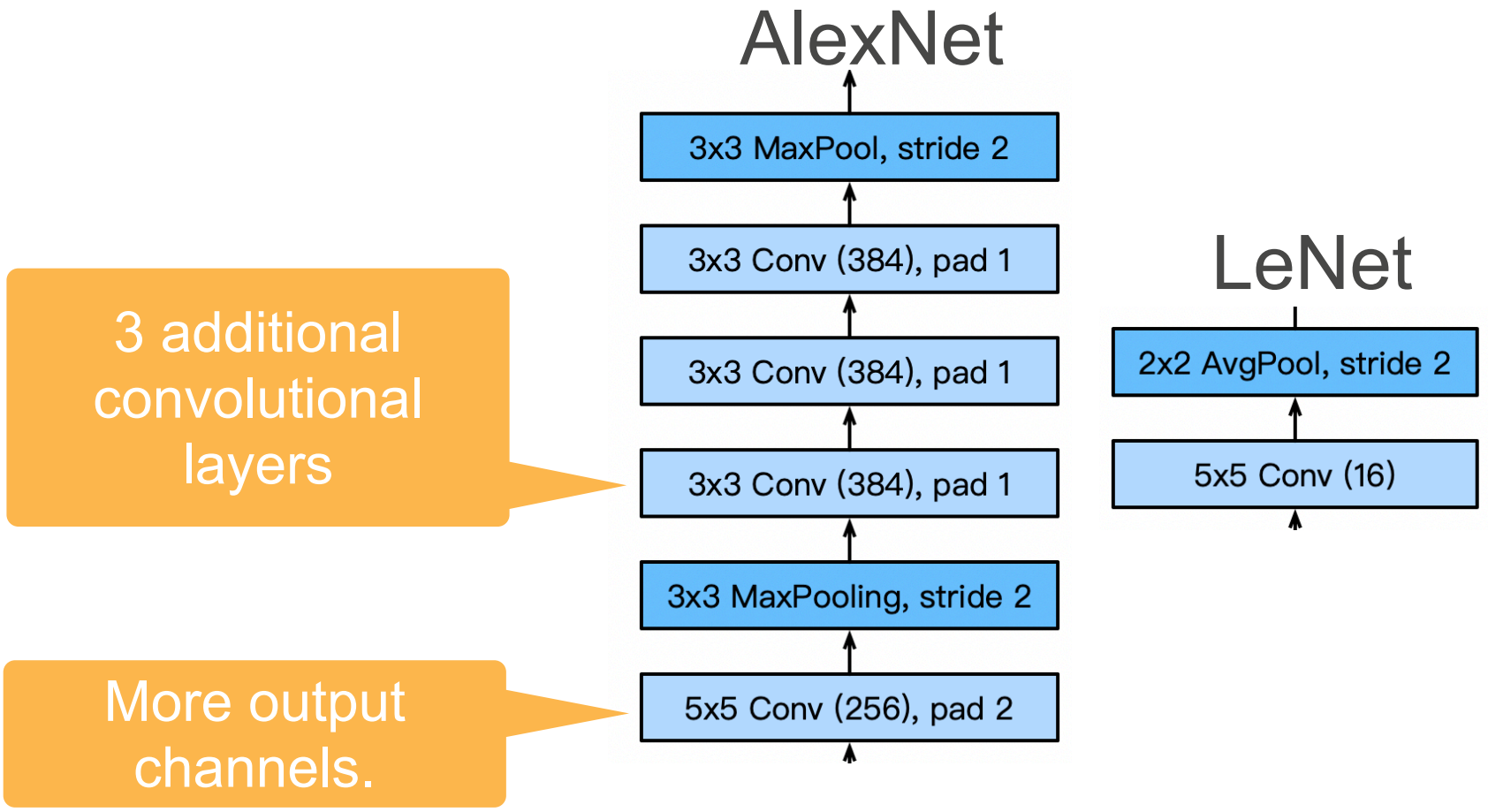
AlexNet Architecture

Larger pool size, change to max pooling

Larger kernel size, stride because of the increased image size, and more output channels.



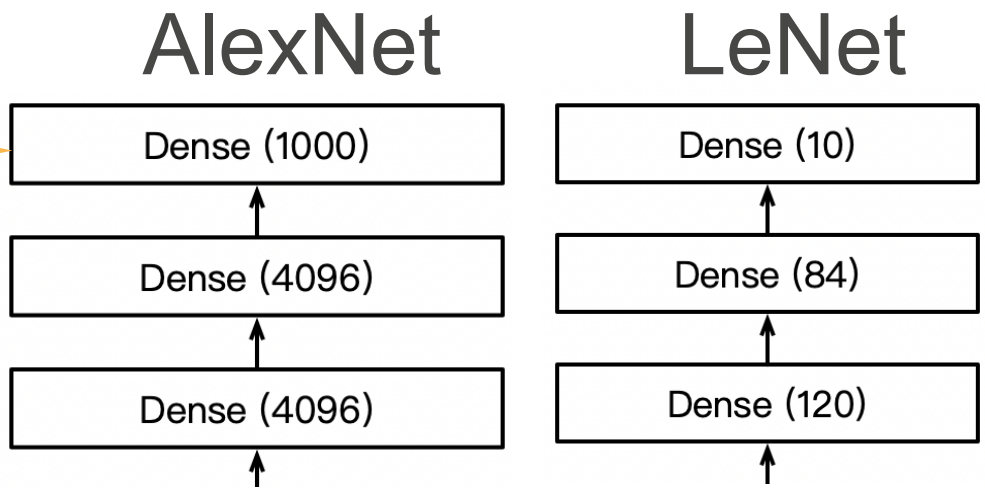
AlexNet Architecture



AlexNet Architecture

1000 classes output

Increase hidden size from 120 to 4096



More Tricks

- Change activation function from sigmoid to ReLu (no more vanishing gradient)
- Add a dropout layer after two hidden FFN layers (better robustness / regularization)
- Data augmentation

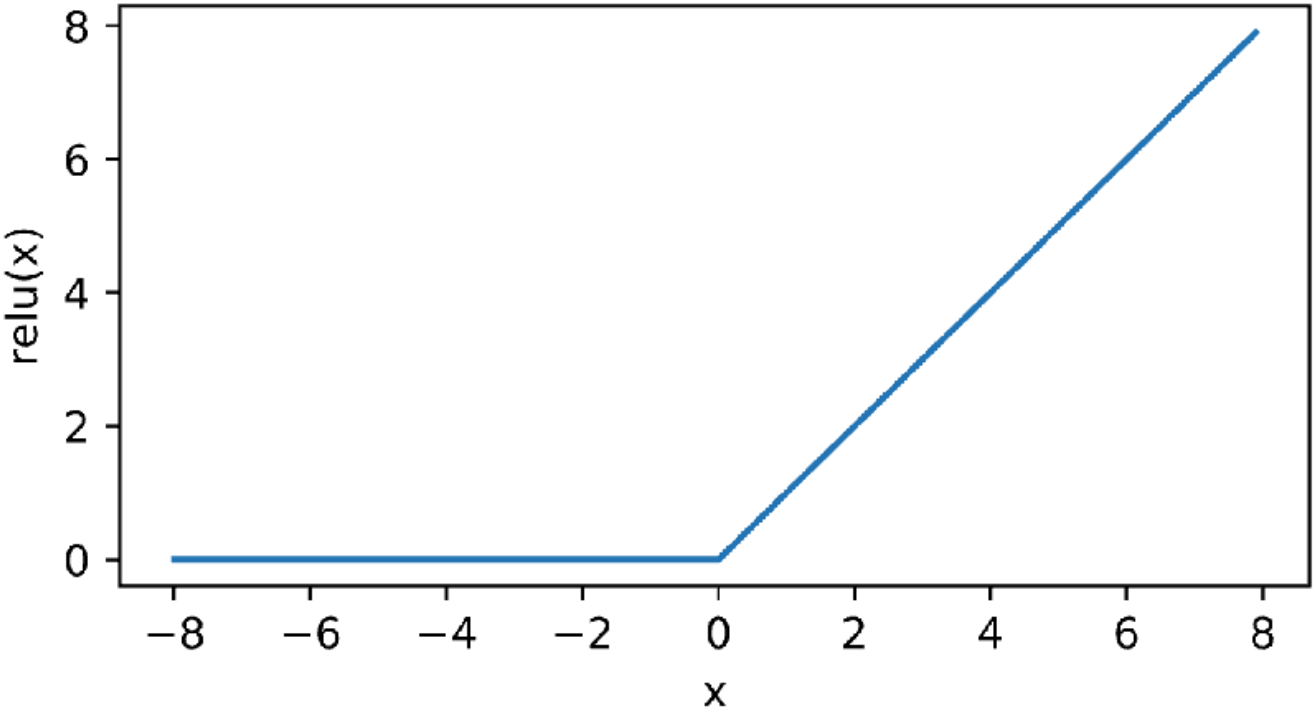
Data Augmentation

- Create additional training data with existing data

ReLU Activation

ReLU: rectified linear unit

$$\mathrm{ReLU}(x) = \max(x, 0)$$

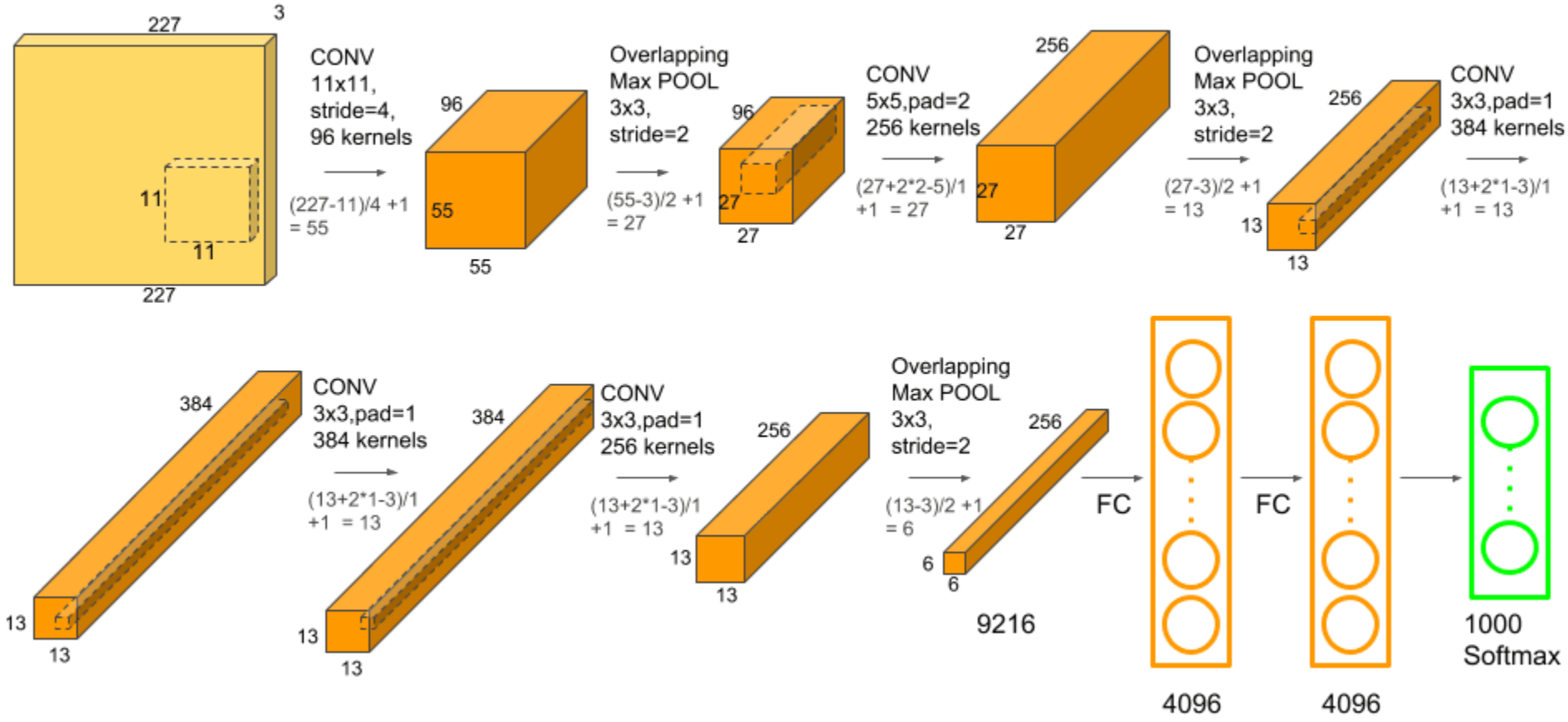


Dropout Layer

- For every input x_i, Dropout produces

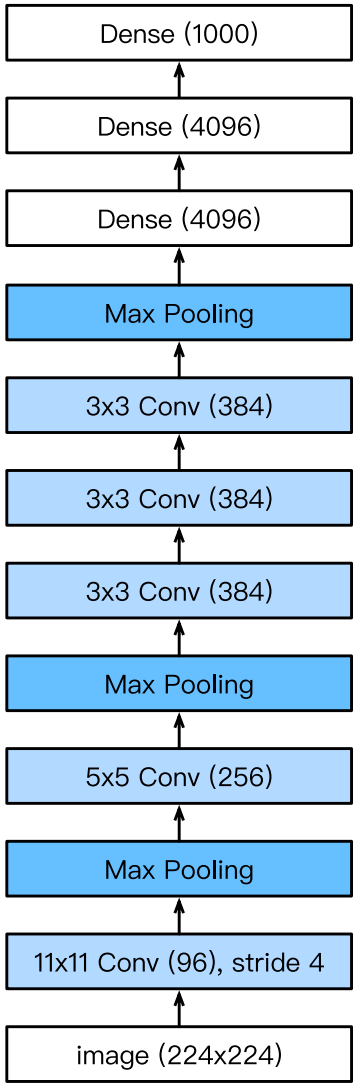
$$x_i' = \begin{cases} 0 & \text{with probablity } p \\ \dfrac{x_i}{1-p} & \text{otherise} \end{cases}$$

AlexNet

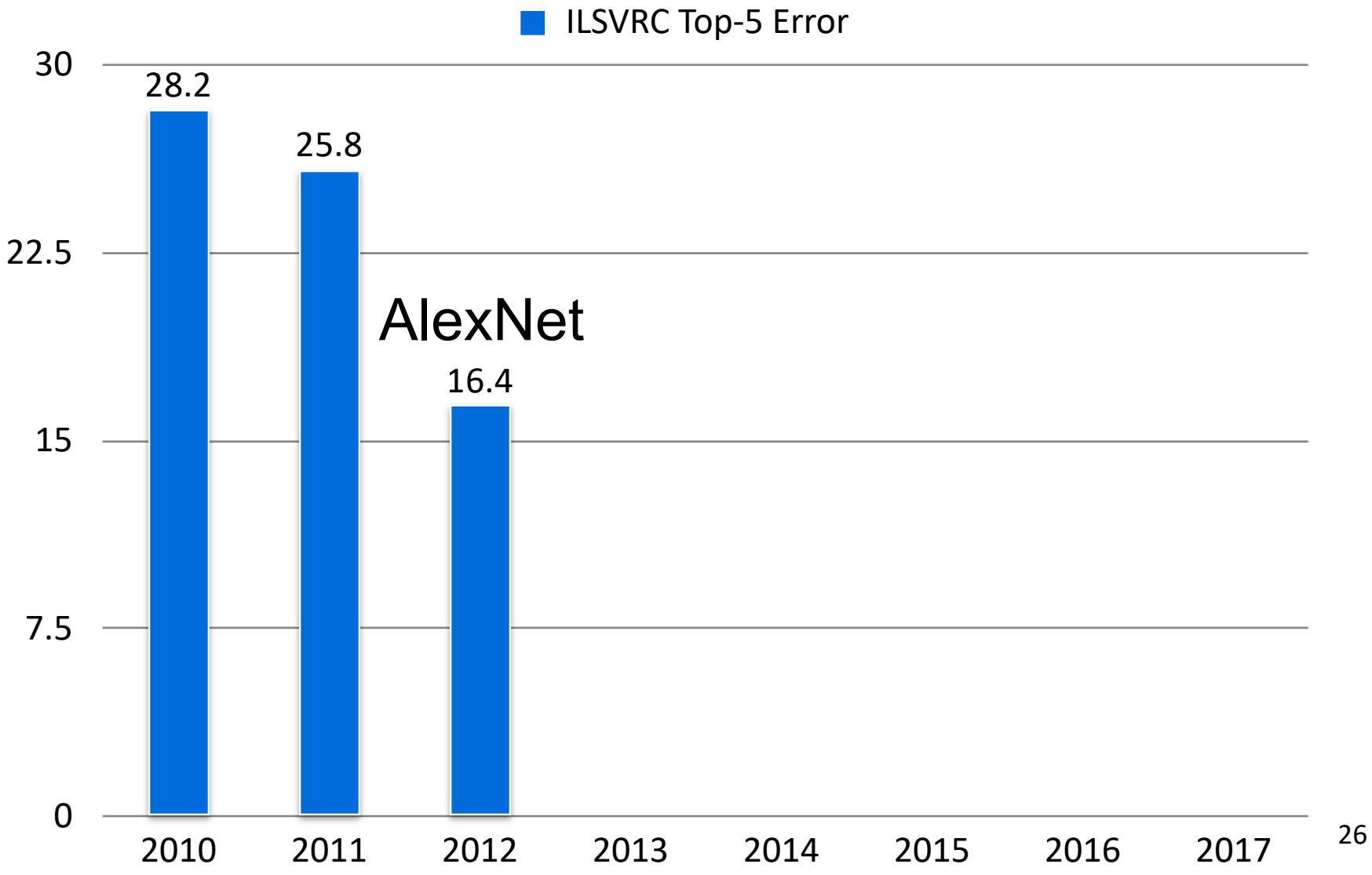

227
227
3
11
11
CONV
11x11,
stride=4,
96 kernels
(227-11)/4 +1
= 55
96
55
55
Overlapping
Max POOL
3x3,
stride=2
(55-3)/2 +1
= 27
96
27
27
CONV
5x5,pad=2
256 kernels
(27+2*2-5)/1
+1 = 27
256
27
27
Overlapping
Max POOL
3x3,
stride=2
(27-3)/2 +1
= 13
256
13
13
CONV
3x3,pad=1
384 kernels
(13+2*1-3)/1
+1 = 13
384
13
13
CONV
3x3,pad=1
384 kernels
(13+2*1-3)/1
+1 = 13
384
13
13
CONV
3x3,pad=1
256 kernels
(13+2*1-3)/1
+1 = 13
256
13
13
Overlapping
Max POOL
3x3,
stride=2
(13-3)/2 +1
= 6
256
6
6
9216
FC
4096
FC
4096
1000
Softmax

Complexity

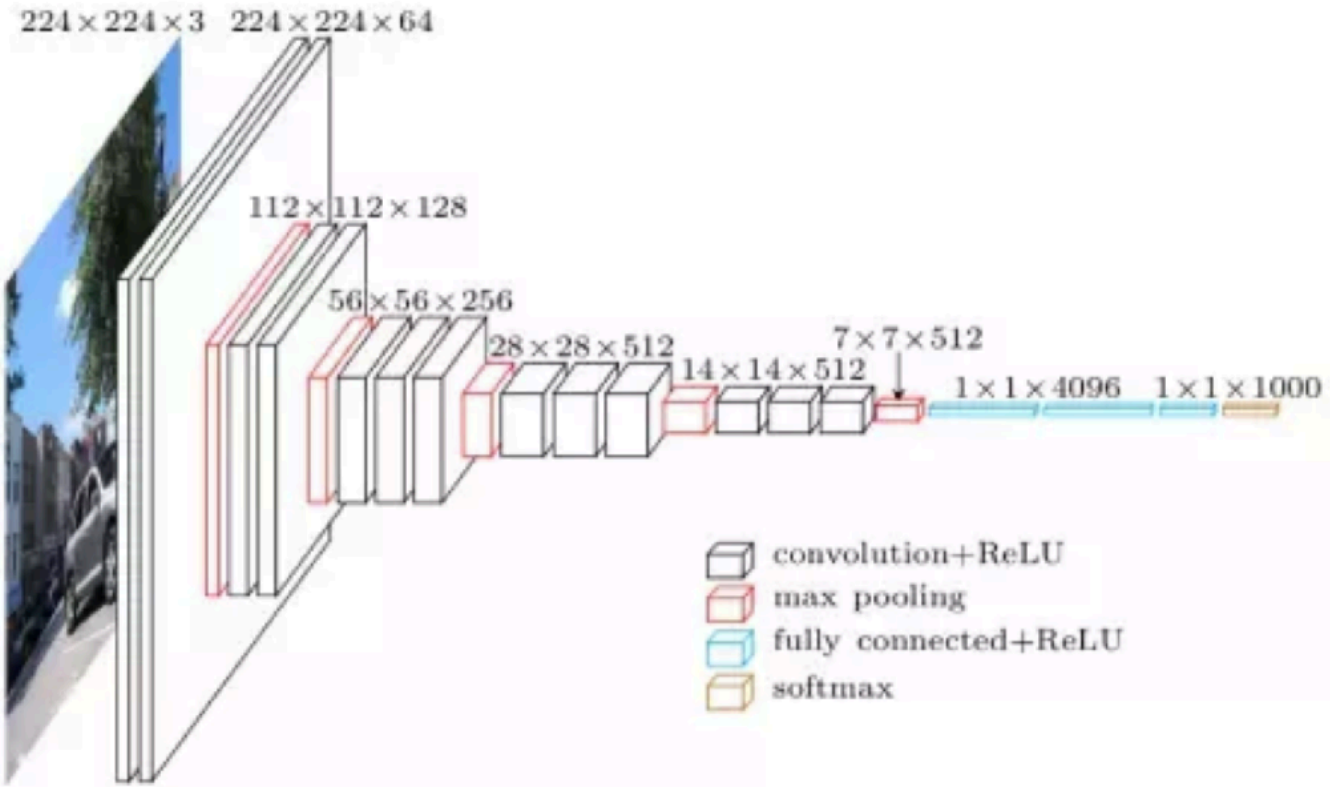
	#parameters		FLOP	
	AlexNet	LeNet	AlexNet	LeNet
Conv1	35K	150	101M	1.2M
Conv2	614K	2.4K	415M	2.4M
Conv3-5	3M		445M	
Dense1	26M	0.48M	26M	0.48M
Dense2	16M	0.1M	16M	0.1M
Total	46M	0.6M	1G	4M
Increase	11x	1x	250x	1x



ImageNet Results: ILSVRC Winners

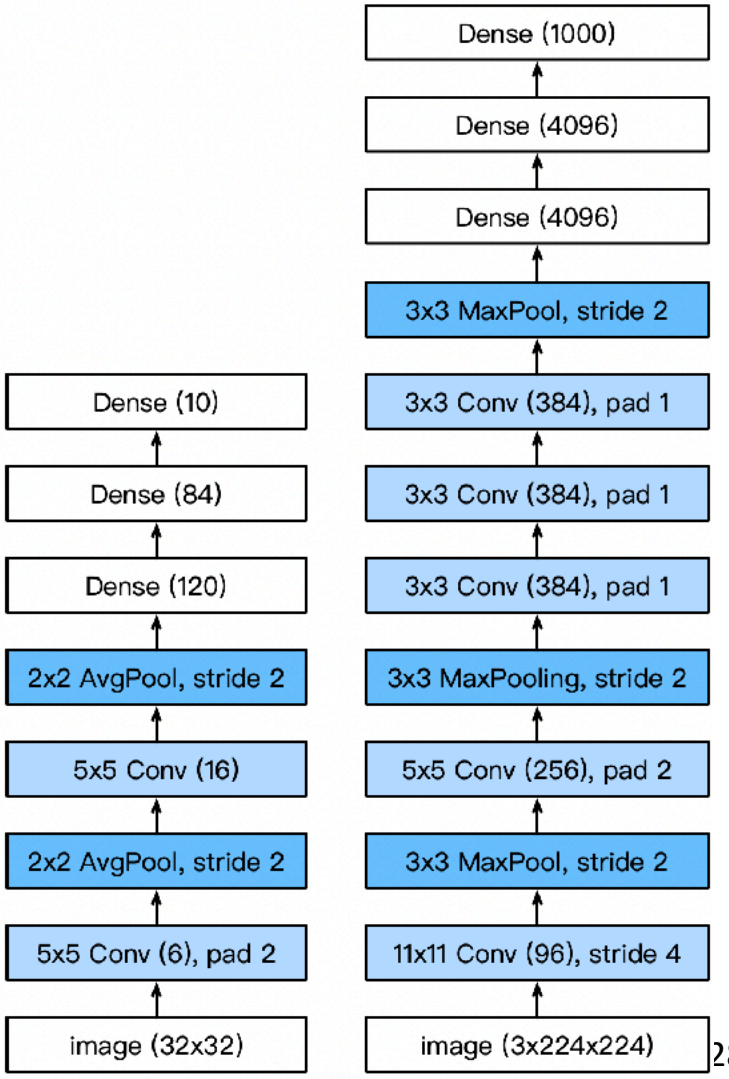


VGG


224×224×3
224×224×64
112×112×128
56×56×256
28×28×512
14×14×512
7×7×512
1×1×4096
1×1×1000
convolution+ReLU
max pooling
fully connected+ReLU
softmax

VGG

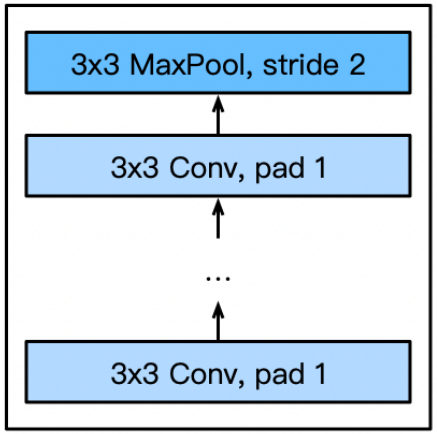
- AlexNet is deeper and bigger than LeNet to get performance
- Go even bigger & deeper?
- Options
  - More dense layers (too expensive)
  - **More** convolutions
  - Group into **blocks**



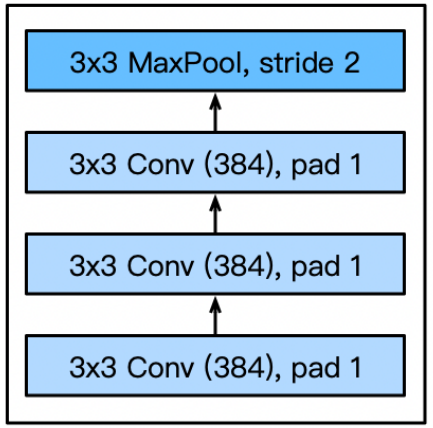
VGG Blocks

- Deeper vs. wider?
  - 5x5 convolutions
  - 3x3 convolutions (more)
  - **Deep & narrow better**
- VGG block
  - *3x3* convolutions (pad 1) **(n layers, m channels)**
  - 2x2 max-pooling (stride 2)

VGG block

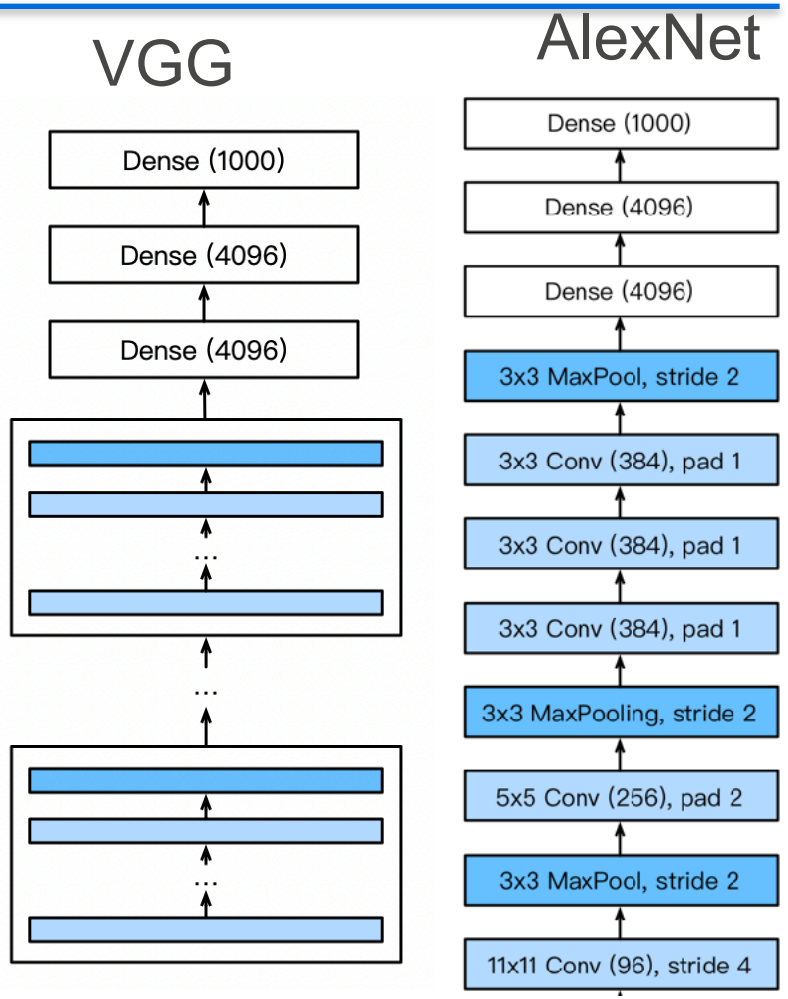


Part of AlexNet



VGG Architecture

- Multiple VGG blocks followed by dense layers
- Vary the repeating number to get different architectures, such as VGG-16, VGG-19, …



Going Deeper

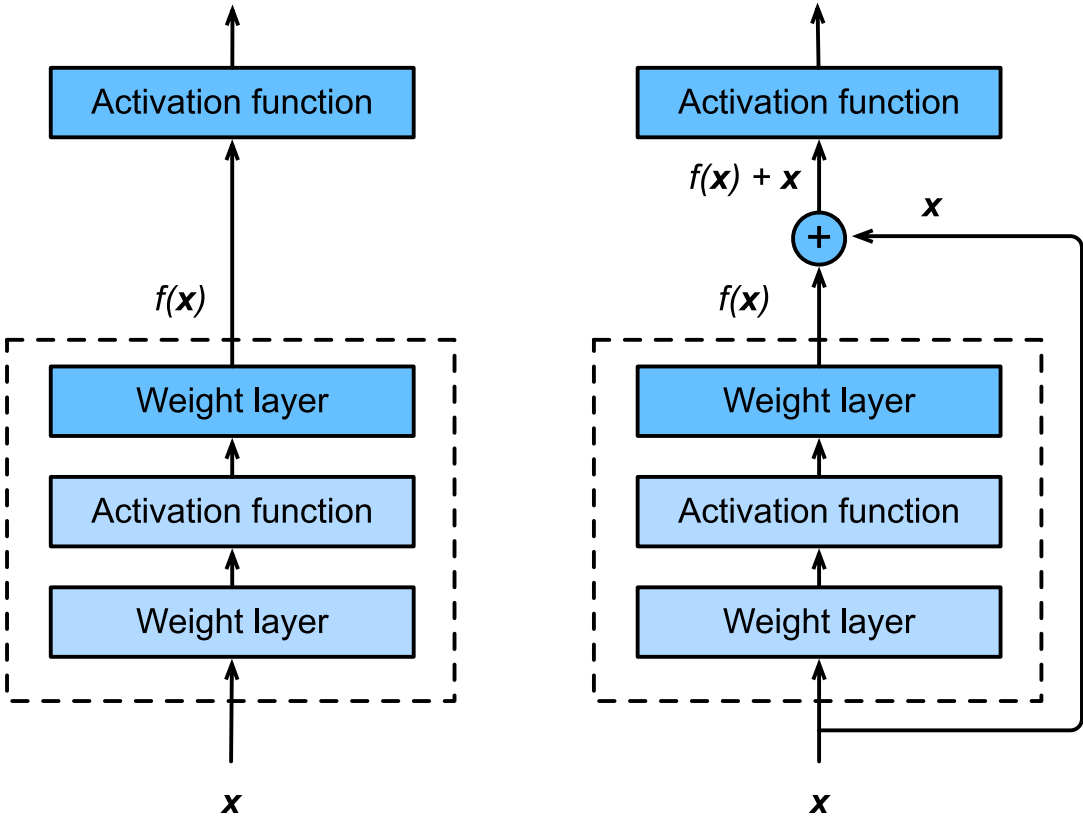
- LeNet (1995)
  - 2 convolution + pooling layers
  - 2 hidden dense layers
- AlexNet
  - Bigger and deeper LeNet
  - ReLu, Dropout, preprocessing
- VGG
  - Bigger and deeper AlexNet (repeated VGG blocks)

Residual Networks

Best paper CVPR 2016

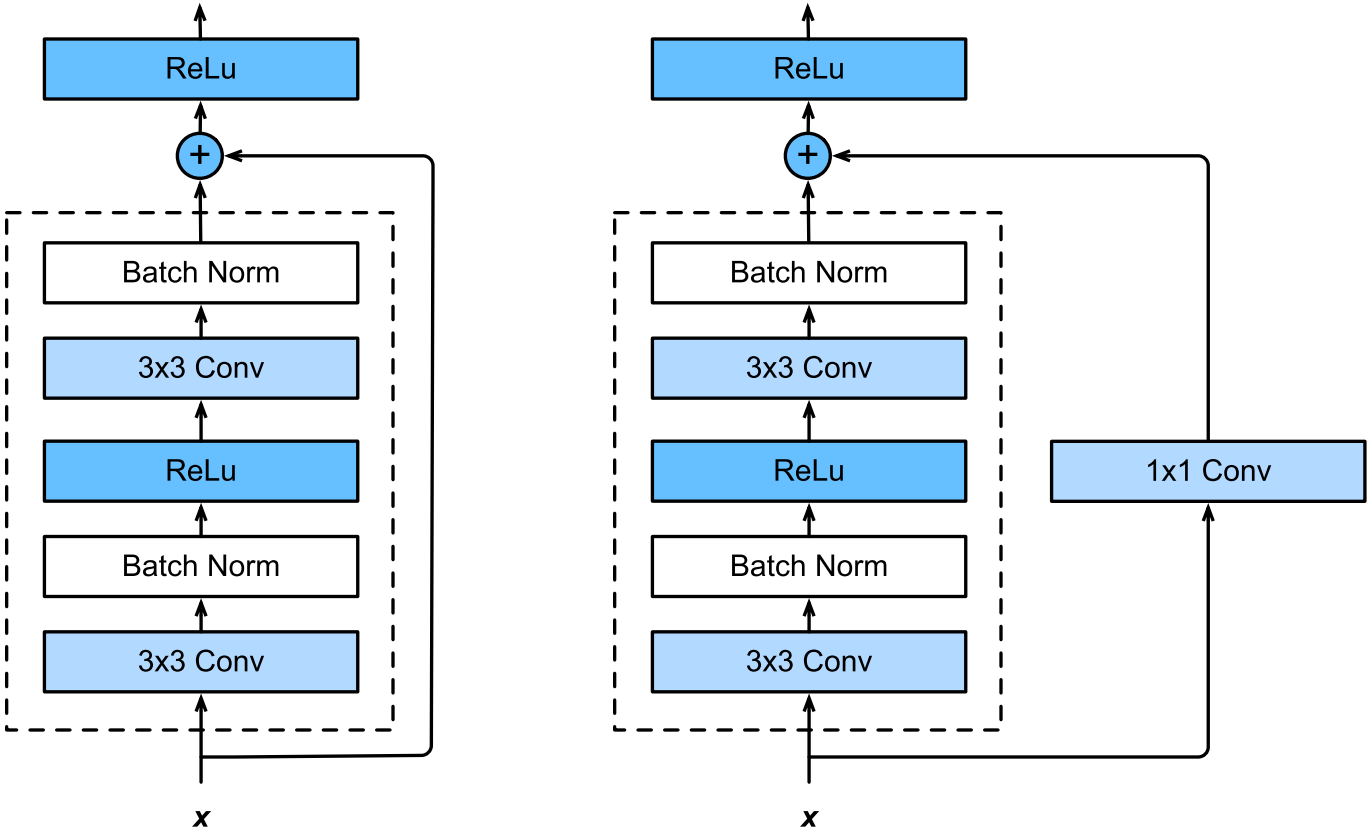
Residual Networks

- Adding a layer **changes** function class
- We want to **add to** the function class
- 'Taylor expansion' style $f(x) = x + g(x)$ parametrization



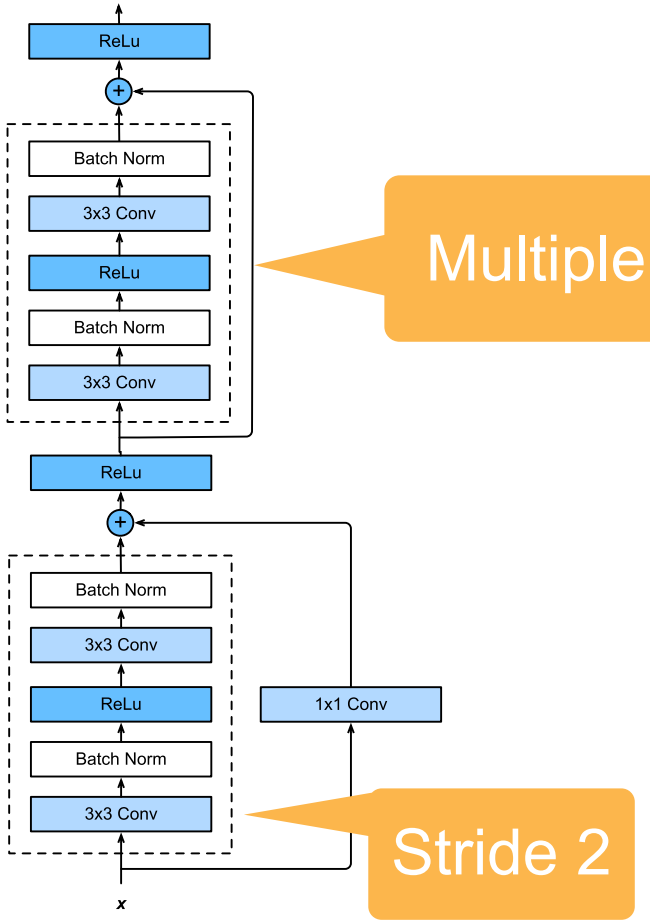
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition. 2016

ResNet Block in detail



ResNet Module

- Downsample per module (stride=2)
- Enforce some nontrivial nonlinearity per module (via 1x1 convolution)
- Stack up in blocks

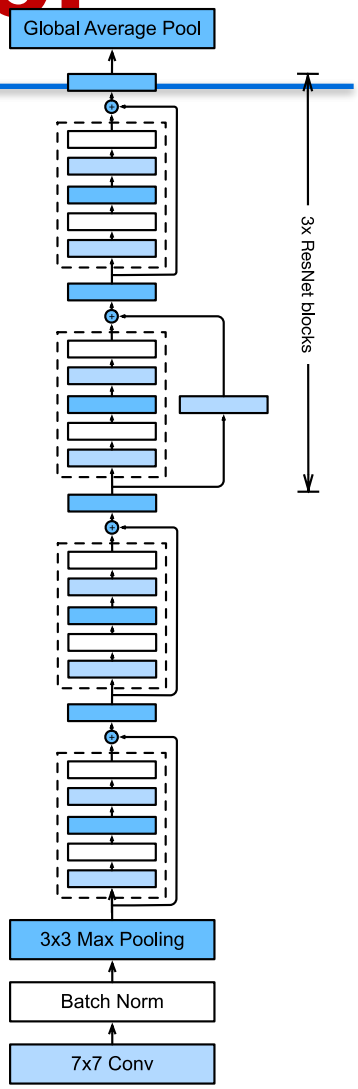


```
blk = nn.Sequential()
for i in range(num_residuals):
    if i == 0 and not first_block:
        blk.add(Residual(num_channels,
            use_1x1conv=True, strides=2))
    else:
        blk.add(Residual(num_channels))
```

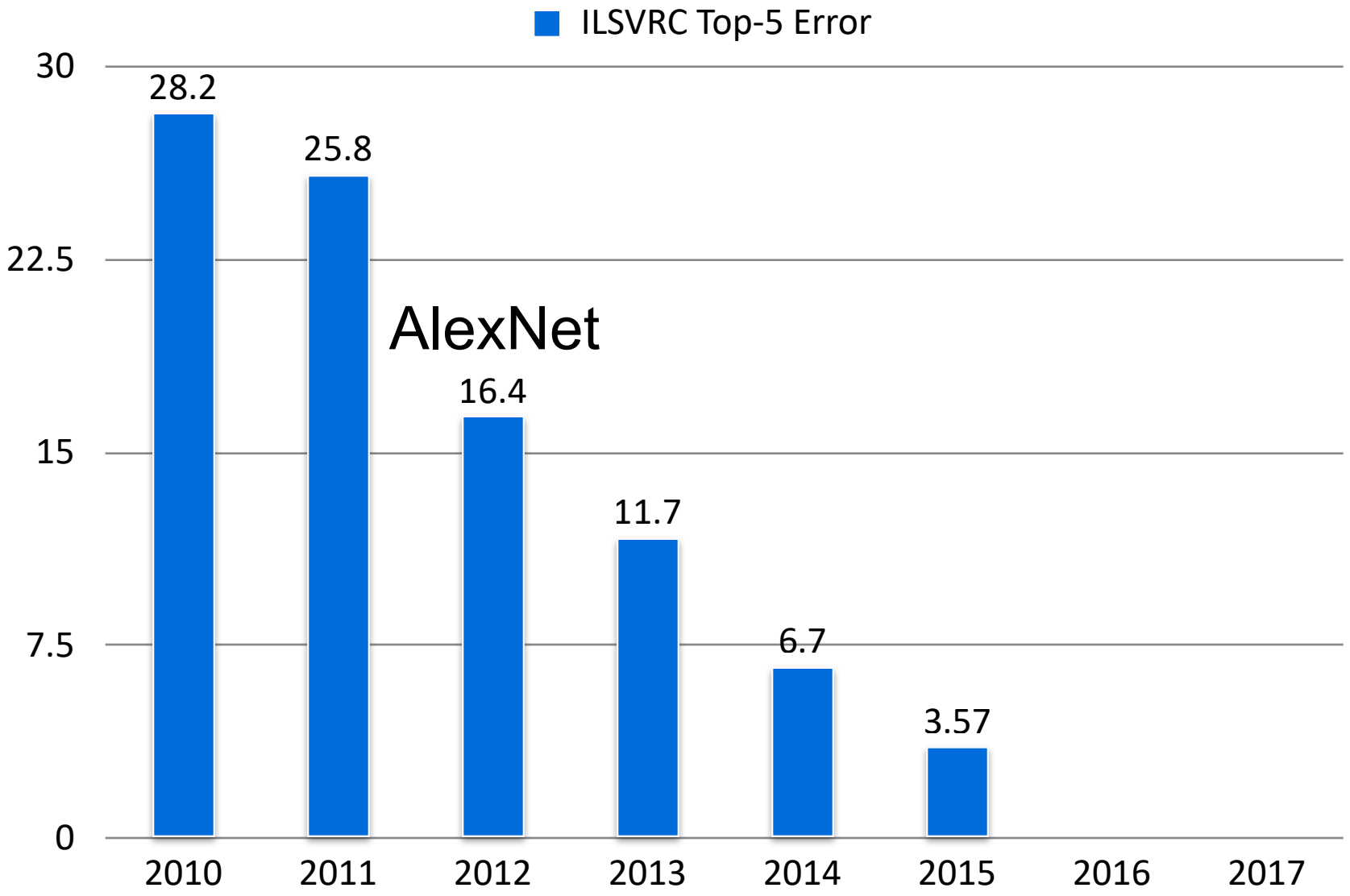
Putting it all together

- Same block structure as e.g. VGG or GoogleNet
- Residual connection to add to expressiveness
- Pooling/stride for dimensionality reduction
- Batch Normalization for capacity control

… train it at scale …



ImageNet Results: ILSVRC Winners

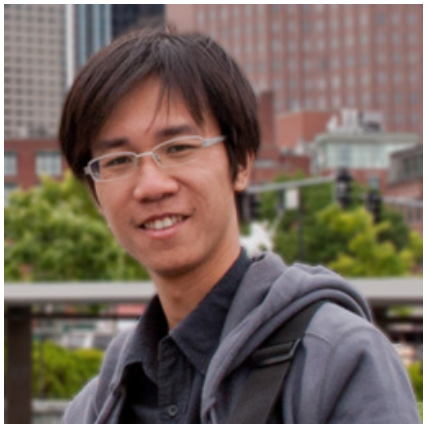


Notes

- ResNet won the champion for ILSVRC 2015
- The ResNet paper won the best paper award from CVPR 2016 (one of the leading CV conferences)
- Kaimin He won multiple best papers.

Papers of Kaimin He

- Exploring Simple Siamese Representation Learning. CVPR Best Paper Honorable Mention, 2021
- Group Normalization. ECCV Best Paper Honorable Mention, 2018
- Mask R-CNN. ICCV Best Paper Award (Marr Prize), 2017
- Focal Loss for Dense Object Detection. ICCV Best Student Paper Award, 2017
- Deep Residual Learning for Image Recognition. CVPR Best Paper Award, 2016
- Single Image Haze Removal using Dark Channel Prior. CVPR Best Paper Award, 2009



ResNext

Reducing the cost of Convolutions

- **Parameters**

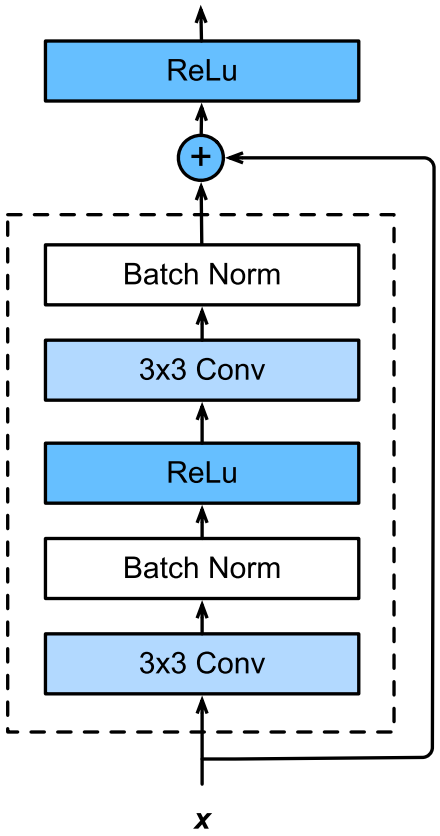
$$k_h \cdot k_w \cdot c_i \cdot c_o$$

- **Computation**

$$m_h \cdot m_w \cdot k_h \cdot k_w \cdot c_i \cdot c_o$$

- **Slicing convolutions** (Inception v4)
  e.g. 3x3 vs. **1x5** and **5x1**
- **Break up channels** (mix only within)

$$m_h \cdot m_w \cdot k_h \cdot k_w \cdot \frac{c_i}{b} \cdot \frac{c_o}{b} \cdot b$$



Reducing the cost of Convolutions

- **Parameters**

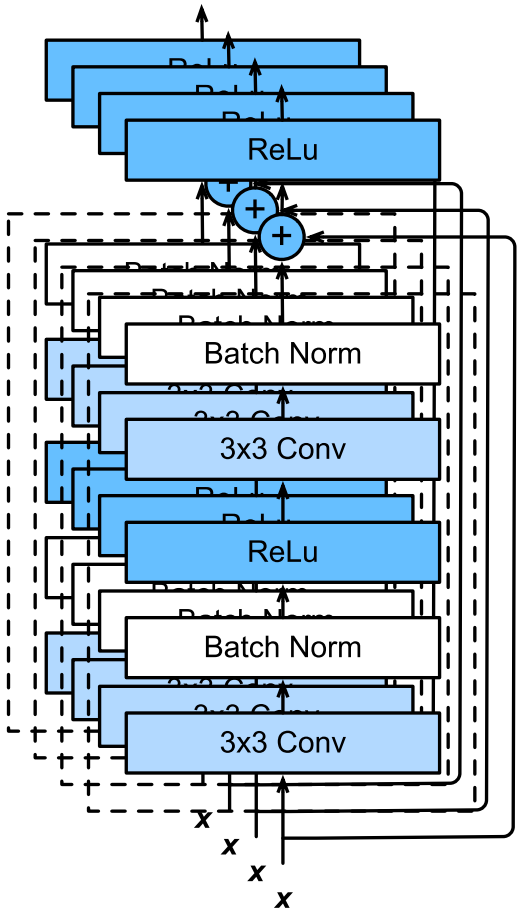
$$k_h \cdot k_w \cdot c_i \cdot c_o$$

- **Computation**

$$m_h \cdot m_w \cdot k_h \cdot k_w \cdot c_i \cdot c_o$$

- **Slicing convolutions** (Inception v4)
  e.g. 3x3 vs. **1x5** and **5x1**
- **Break up channels** (mix only within)

$$m_h \cdot m_w \cdot k_h \cdot k_w \cdot \frac{c_i}{b} \cdot \frac{c_o}{b} \cdot b$$



RexNext budget

- Slice blocks into 32 sub-blocks
- Can use more dimensions
- Higher accuracy

stage	output	ResNet-50	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2 $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times3$	3×3 max pool, stride 2 $\begin{bmatrix} 1\times1 & 128 \\ 3\times3 & 128, C=32 \\ 1\times1 & 256 \end{bmatrix} \times3$
conv3	28×28	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times4$	$\begin{bmatrix} 1\times1 & 256 \\ 3\times3 & 256, C=32 \\ 1\times1 & 512 \end{bmatrix} \times4$
conv4	14×14	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times6$	$\begin{bmatrix} 1\times1 & 512 \\ 3\times3 & 512, C=32 \\ 1\times1 & 1024 \end{bmatrix} \times6$
conv5	7×7	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times3$	$\begin{bmatrix} 1\times1, & 1024 \\ 3\times3, & 1024 \ C=32 \\ 1\times1, & 2048 \end{bmatrix} \times3$
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		**25.5**×10^6	**25.0**×10^6
FLOPs		**4.1**×10^9	**4.2**×10^9

Recap

- AlexNet
  - 11 layers, bigger convolusion
  - ReLu, Dropout, preprocessing
- VGG
  - Bigger and deeper AlexNet (repeated VGG blocks)
  - VGG-16 and VGG-19
- ResNet
  - 50 or 153 layers
  - Residual connection

Next Up

- Advanced optimization methods