CS 291A: Deep Learning for NLP

Convolutional Neural Networks

William Wang
UCSB Computer Science
william@cs.ucsb.edu

Slides adapted from V. Chen and M. Chang.

Midterm Report

- Due today to reader Ke Ni at ke00@ucsb.edu.
- By now you should have obtained some preliminary results for your project.

Talk: Adaptive Adversarial Learning for a Diverse Visual World

- **Judy Hoffman (UC Berkeley)**
- **Date:** Thursday, February 22, 2018 - 3:30pm
- **Location:** HFH 1132
- **Speaker:**
- Judy Hoffman

Convolutional Neural Networks

- We need a course to talk about this topic
  - http://cs231n.stanford.edu/syllabus.html
- However, we only have one lecture.

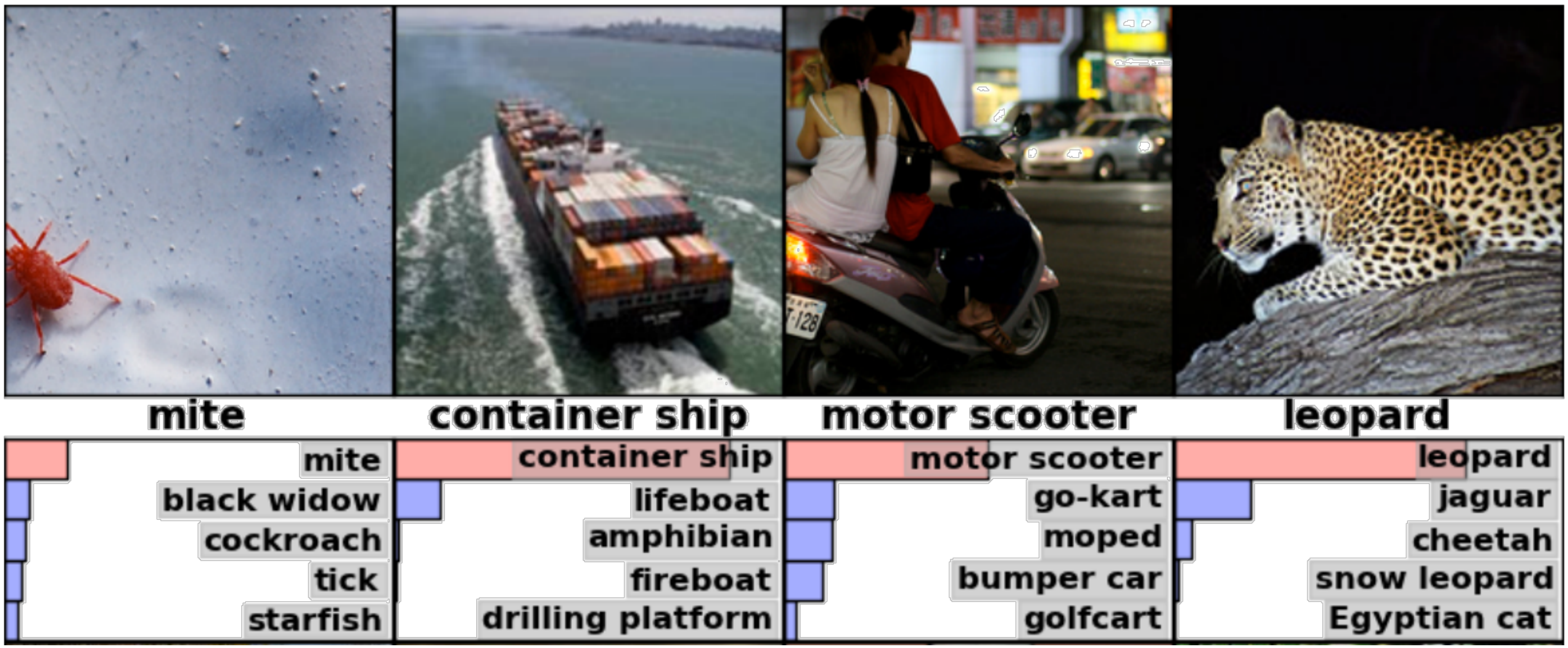
Outline

- CNN(Convolutional Neural Networks) Introduction
- Evolution of CNN
- Visualizing the Features
- Sentiment Analysis with CNN

Outline

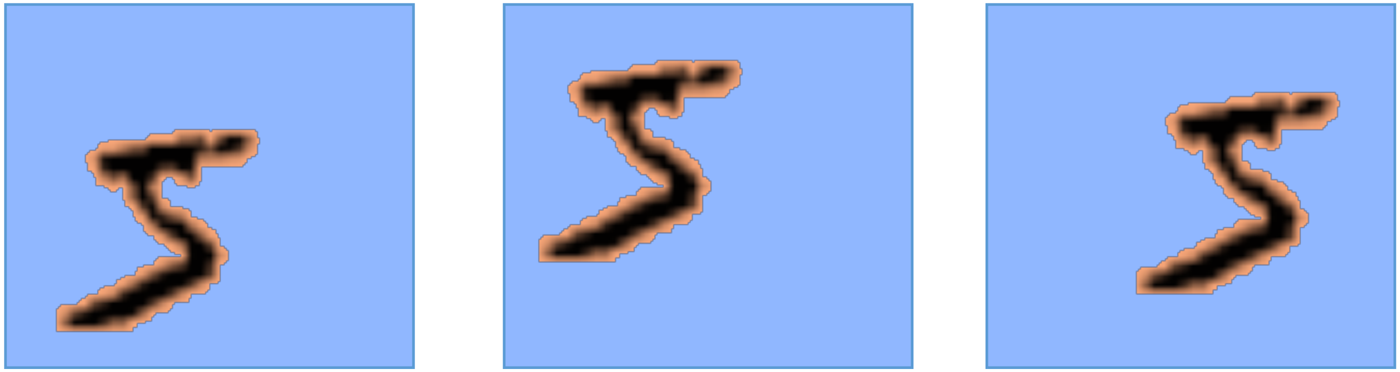
- CNN(Convolutional Neural Networks) Introduction
- Evolution of CNN
- Visualizing the Features
- Sentiment Analysis by CNN

Image Recognition

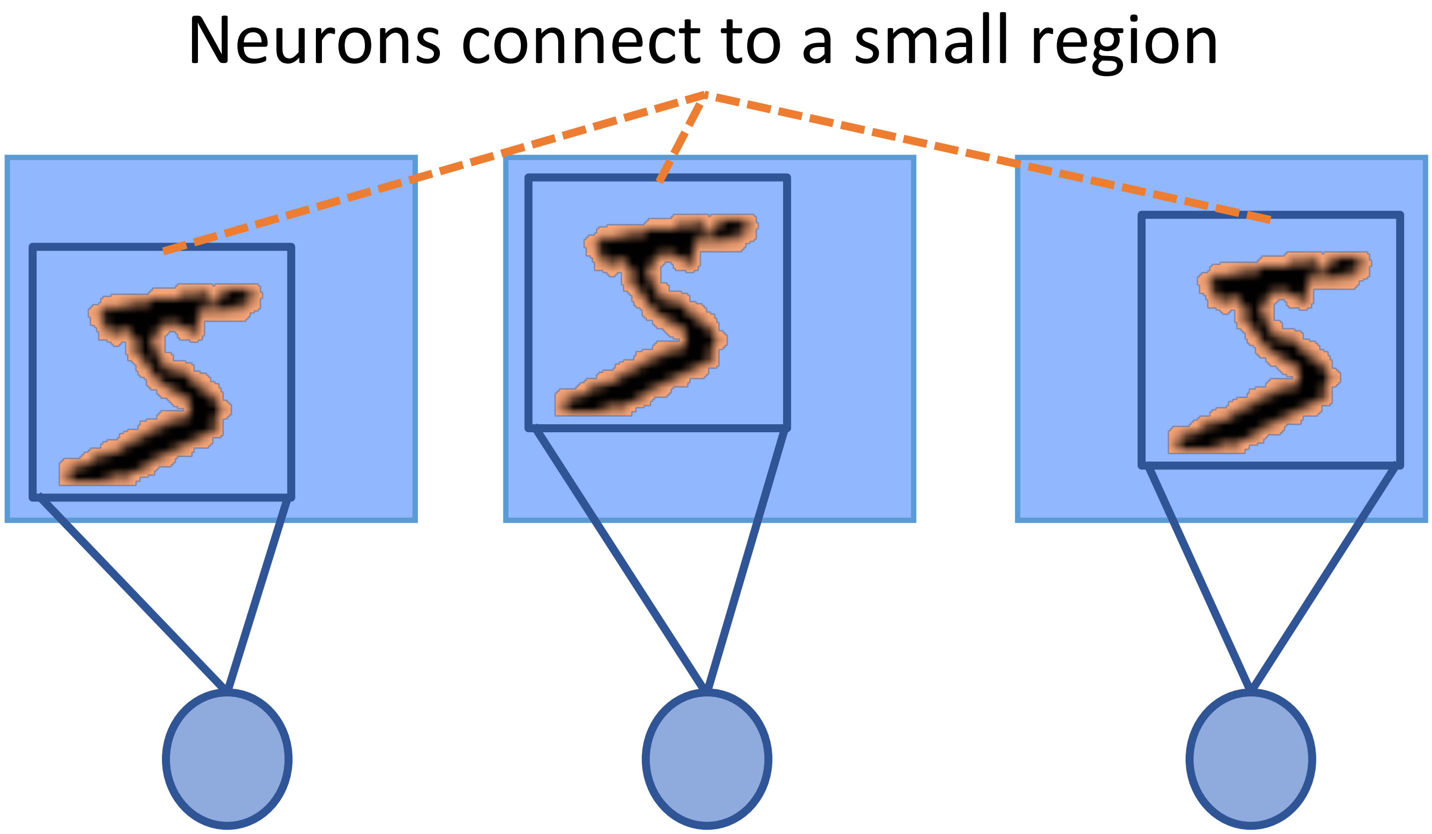


http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf

Image Recognition

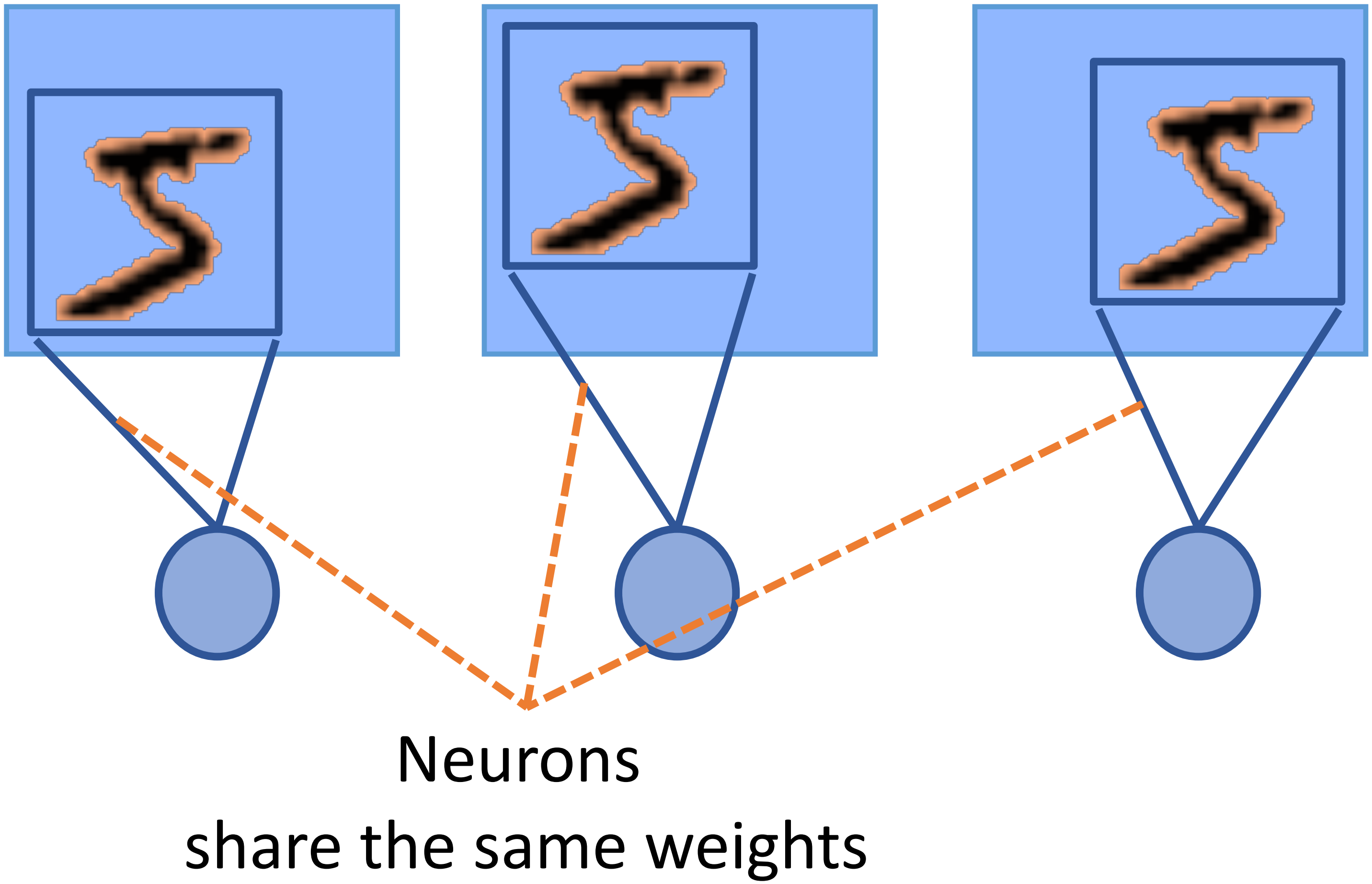


Local Connectivity



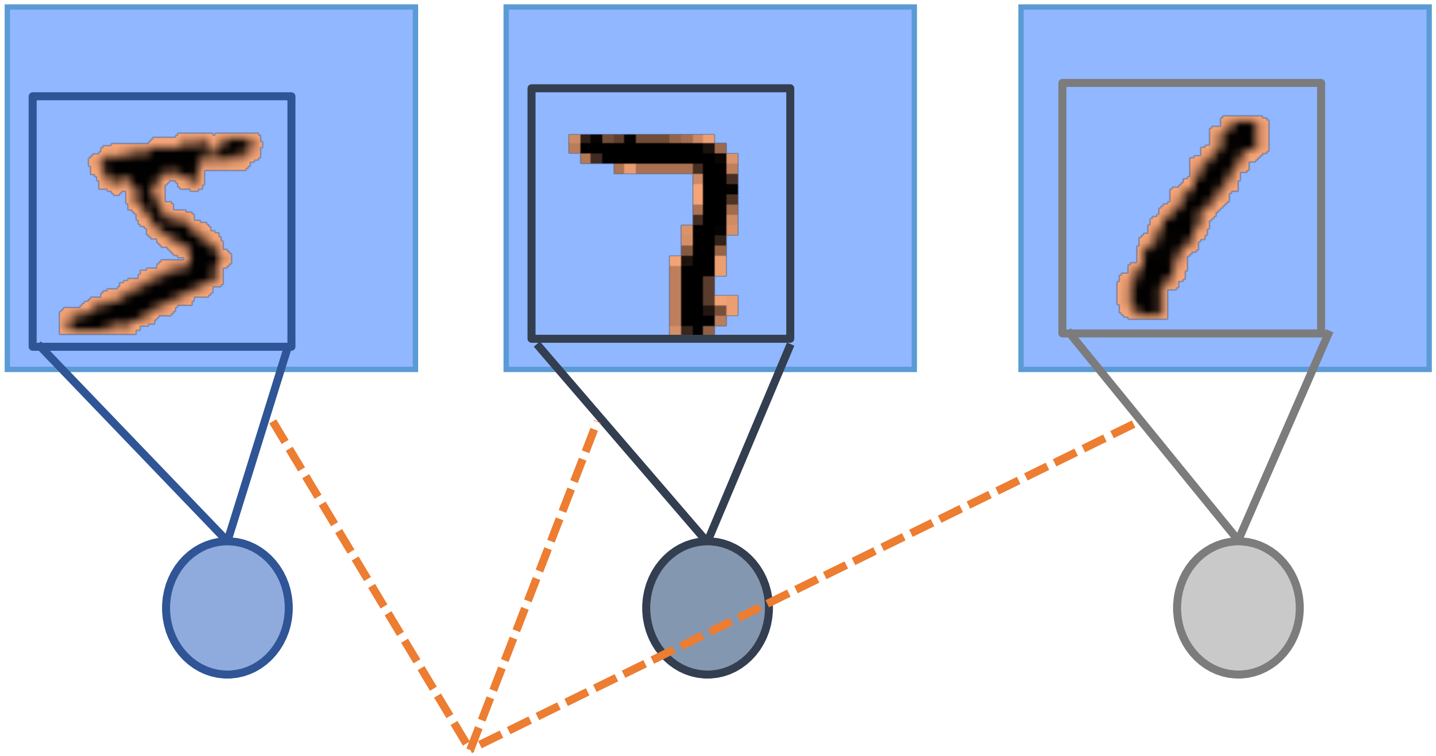
Parameter Sharing

- The same feature in different positions

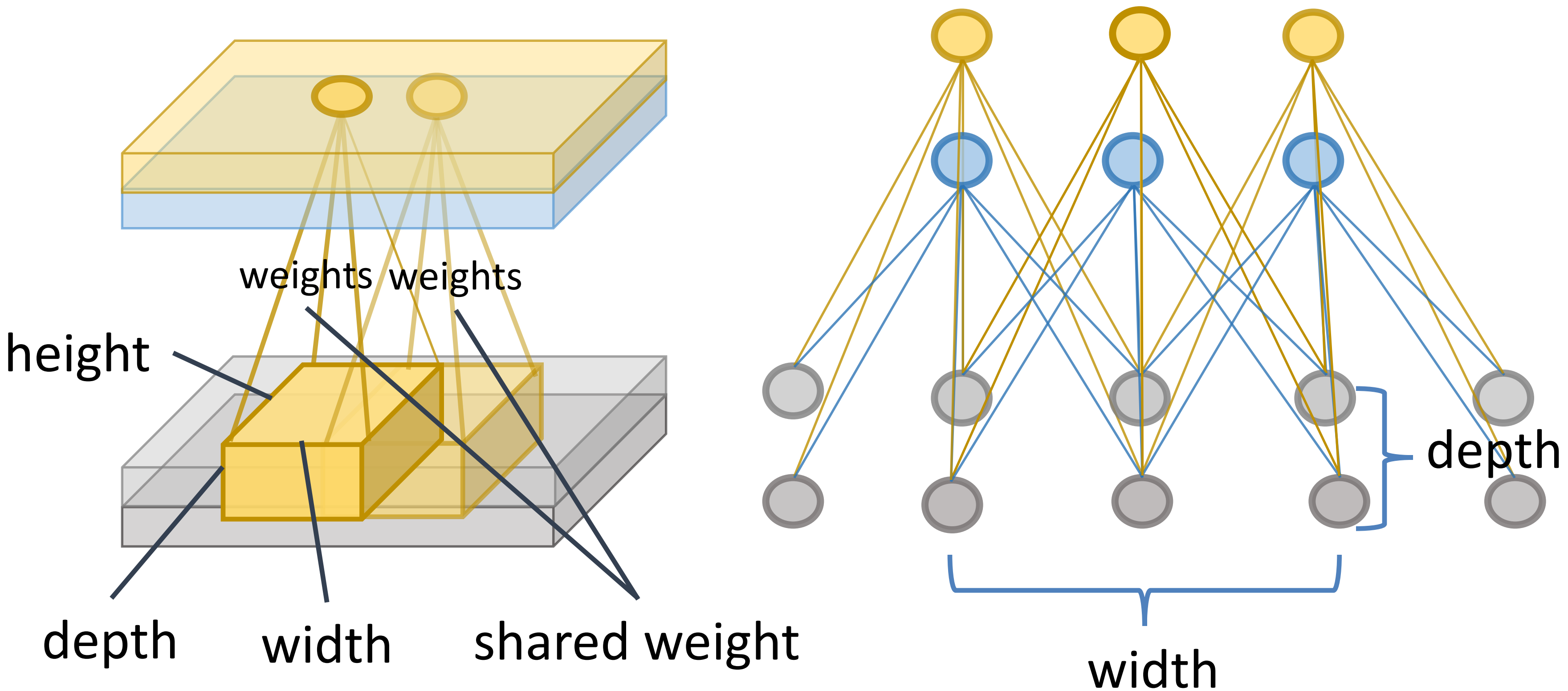


Parameter Sharing

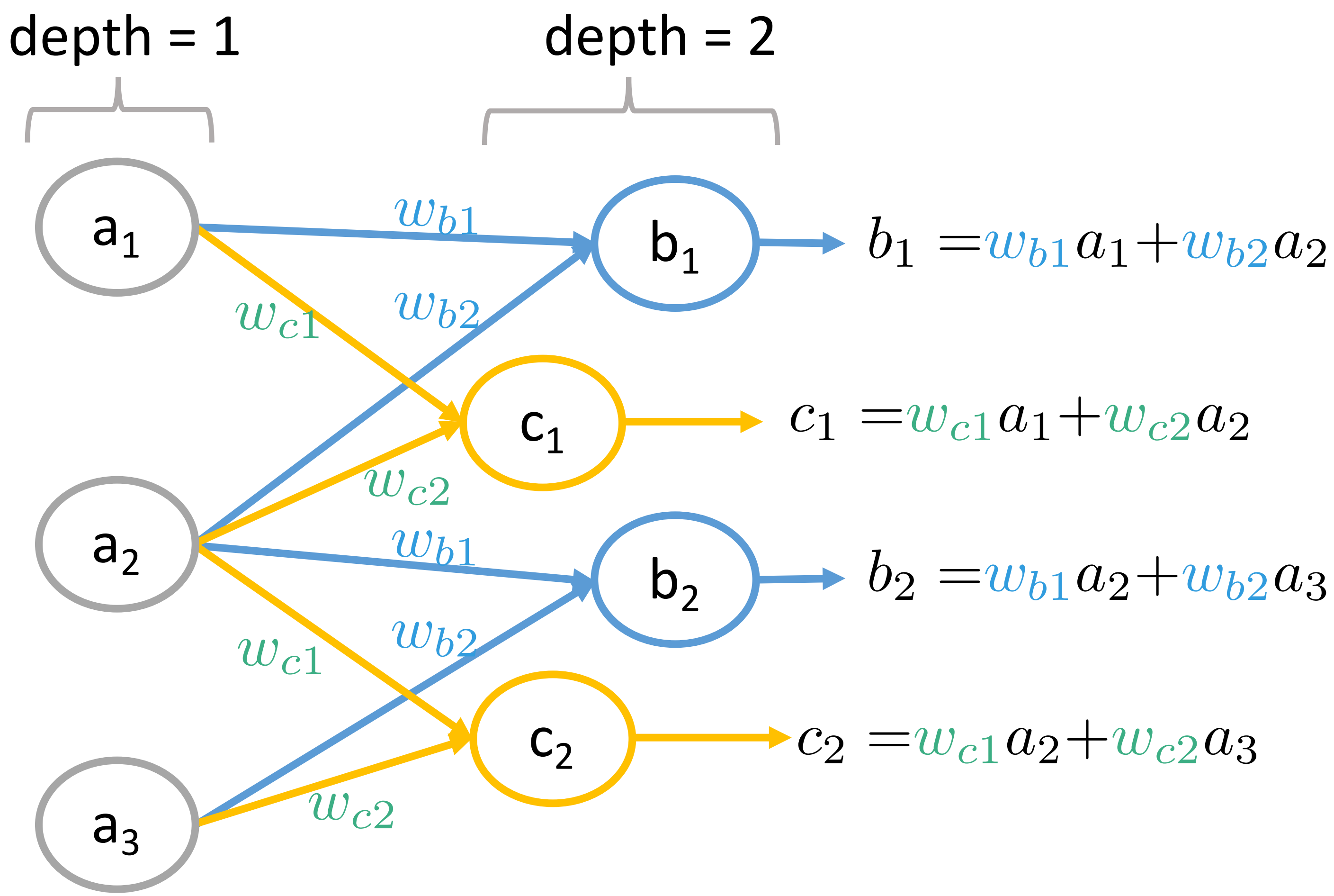
- Different features in the same position



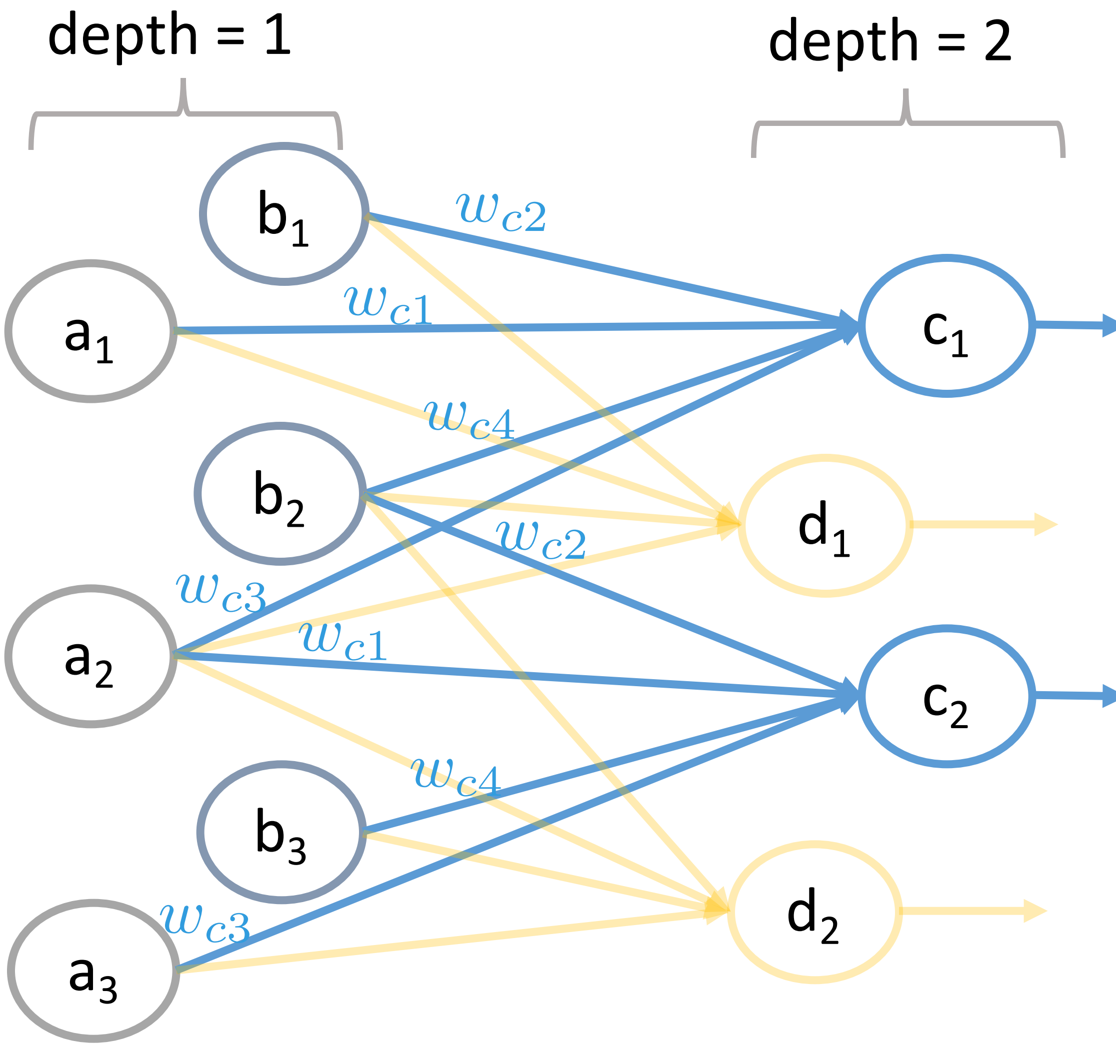
Convolutional Layers: model local contexts. Sliding window in an image / sentence.



Convolutional Layers: Window Size of 2.



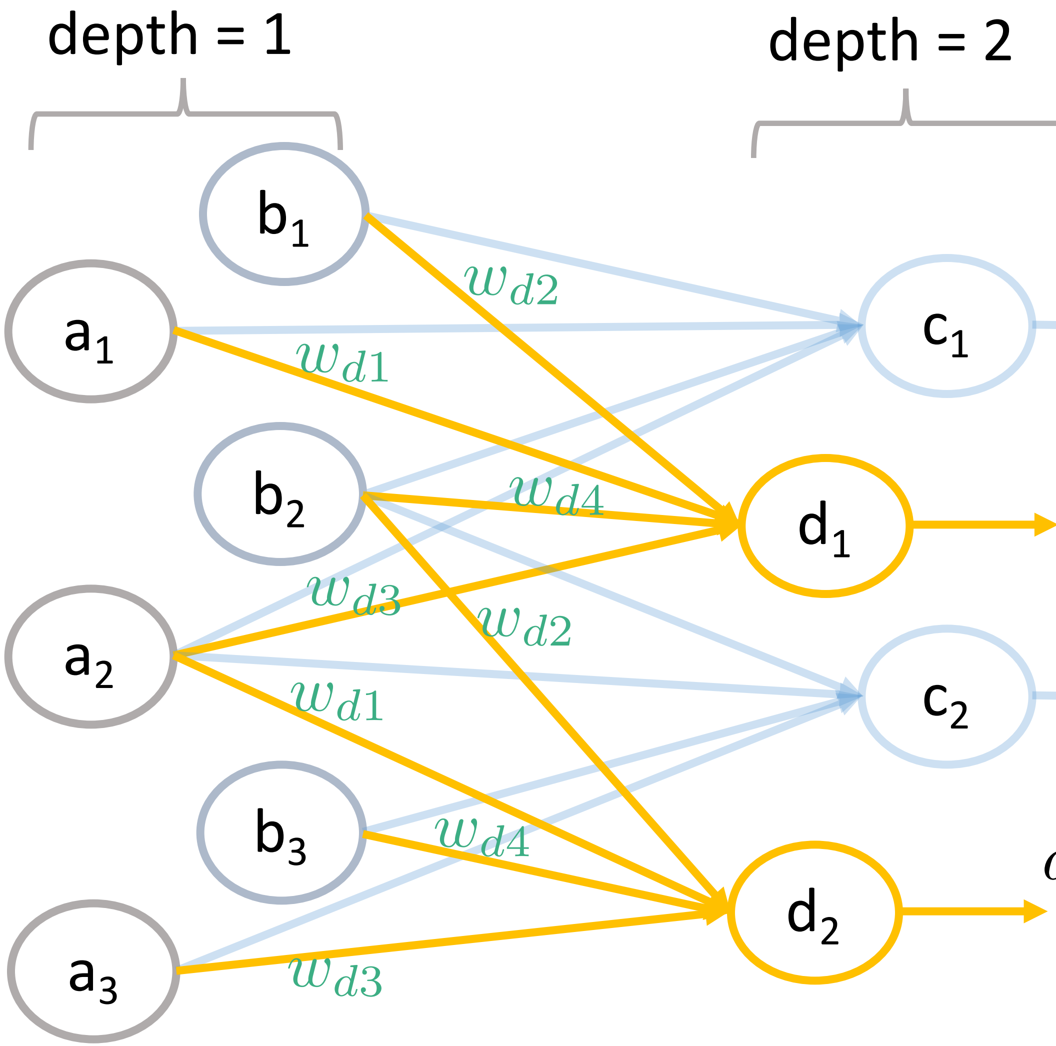
Convolutional Layers: Window Size of 4.



$$c_1 = a_1 w_{c1} + b_1 w_{c2} + a_2 w_{c3} + b_2 w_{c4}$$

$$c_2 = a_2 w_{c1} + b_2 w_{c2} + a_3 w_{c3} + b_3 w_{c4}$$

Convolutional Layers: Window Size of 4.



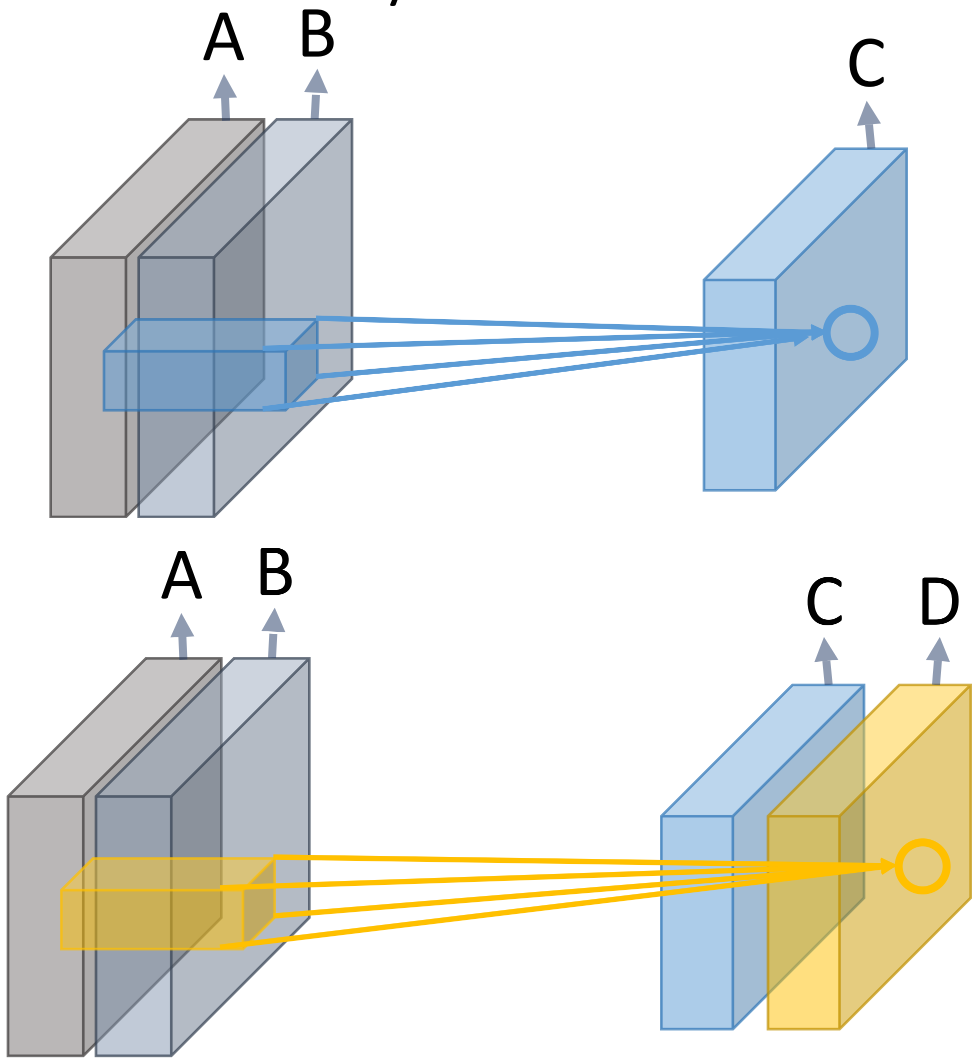
$$c_1 = a_1 w_{c1} + b_1 w_{c2} + a_2 w_{c3} + b_2 w_{c4}$$

$$d_1 = a_1 w_{d1} + b_1 w_{d2} + a_2 w_{d3} + b_2 w_{d4}$$

$$c_2 = a_2 w_{c1} + b_2 w_{c2} + a_3 w_{c3} + b_3 w_{c4}$$

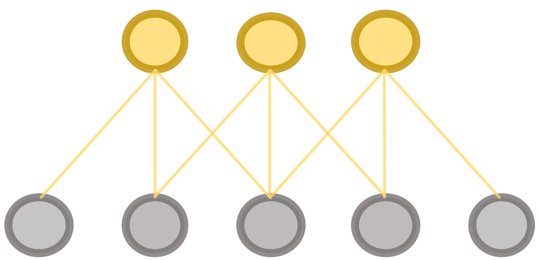
$$d_2 = a_2 w_{d1} + b_2 w_{d2} + a_3 w_{d3} + b_3 w_{d4}$$

Convolutional Layers

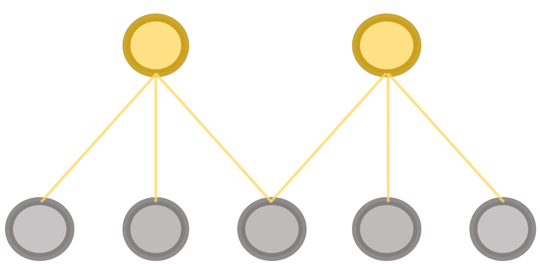


Hyper-parameters of CNN

- Stride

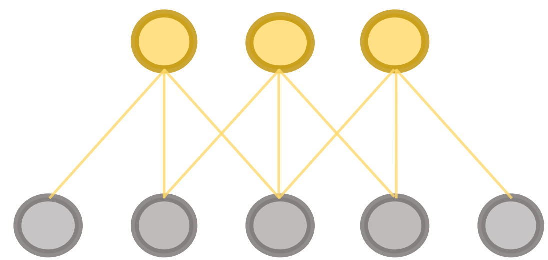


Stride = 1

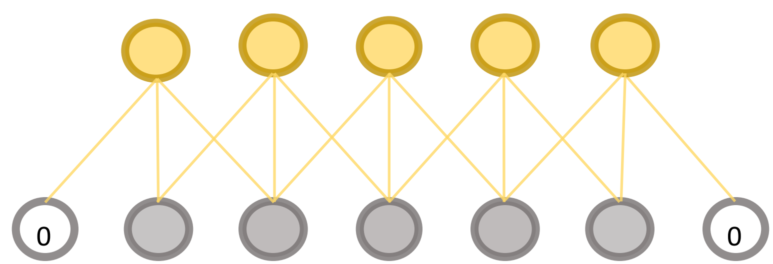


Stride = 2

- Padding



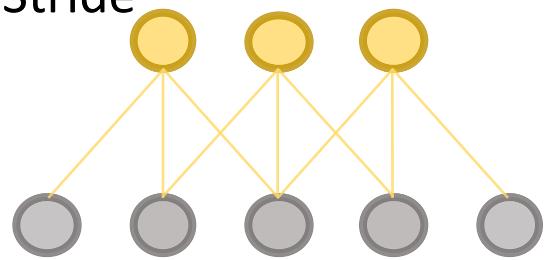
Padding = 0



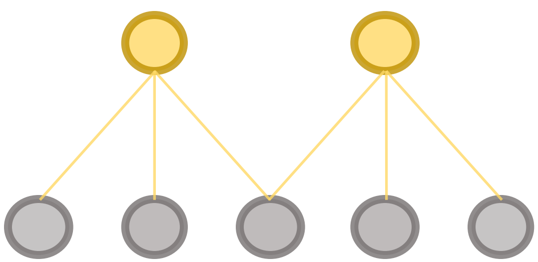
Padding = 1

Piazza Poll: Why do we need padding?

- Stride

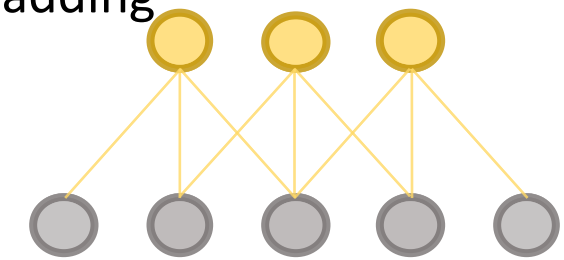


Stride = 1

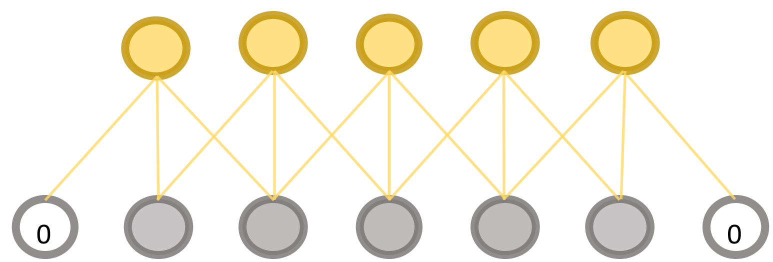


Stride = 2

- Padding

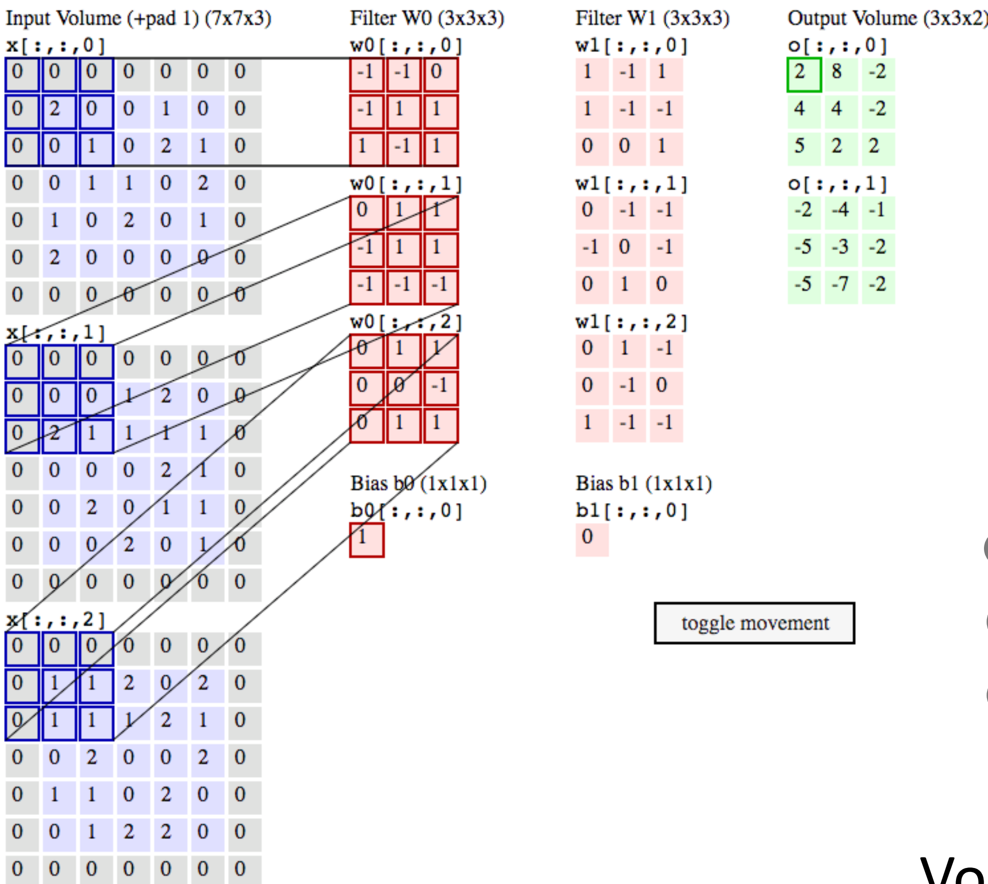


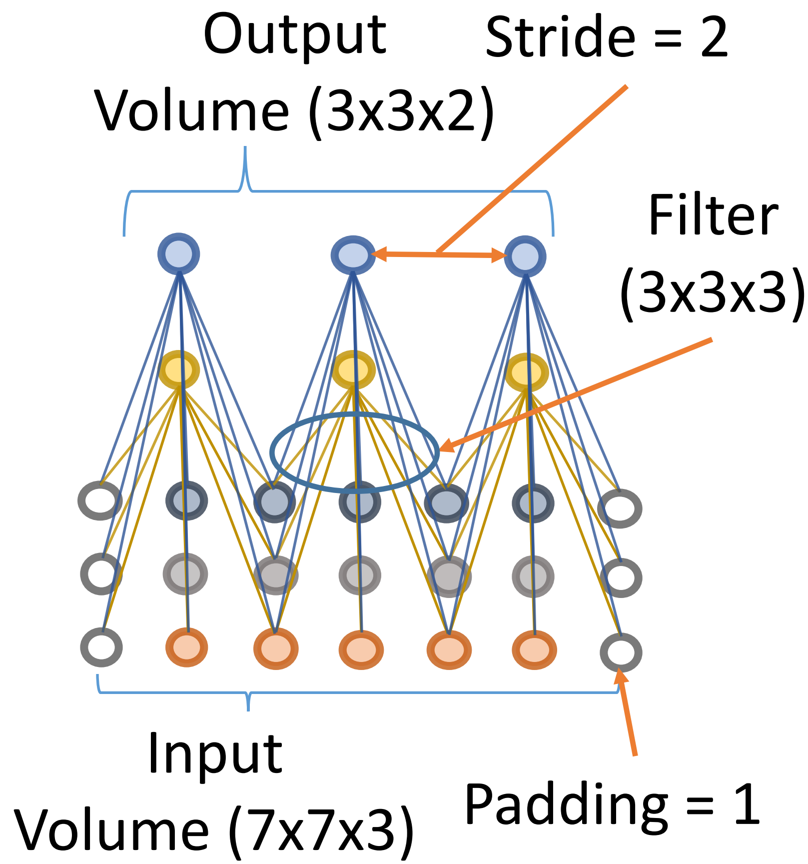
Padding = 0



Padding = 1

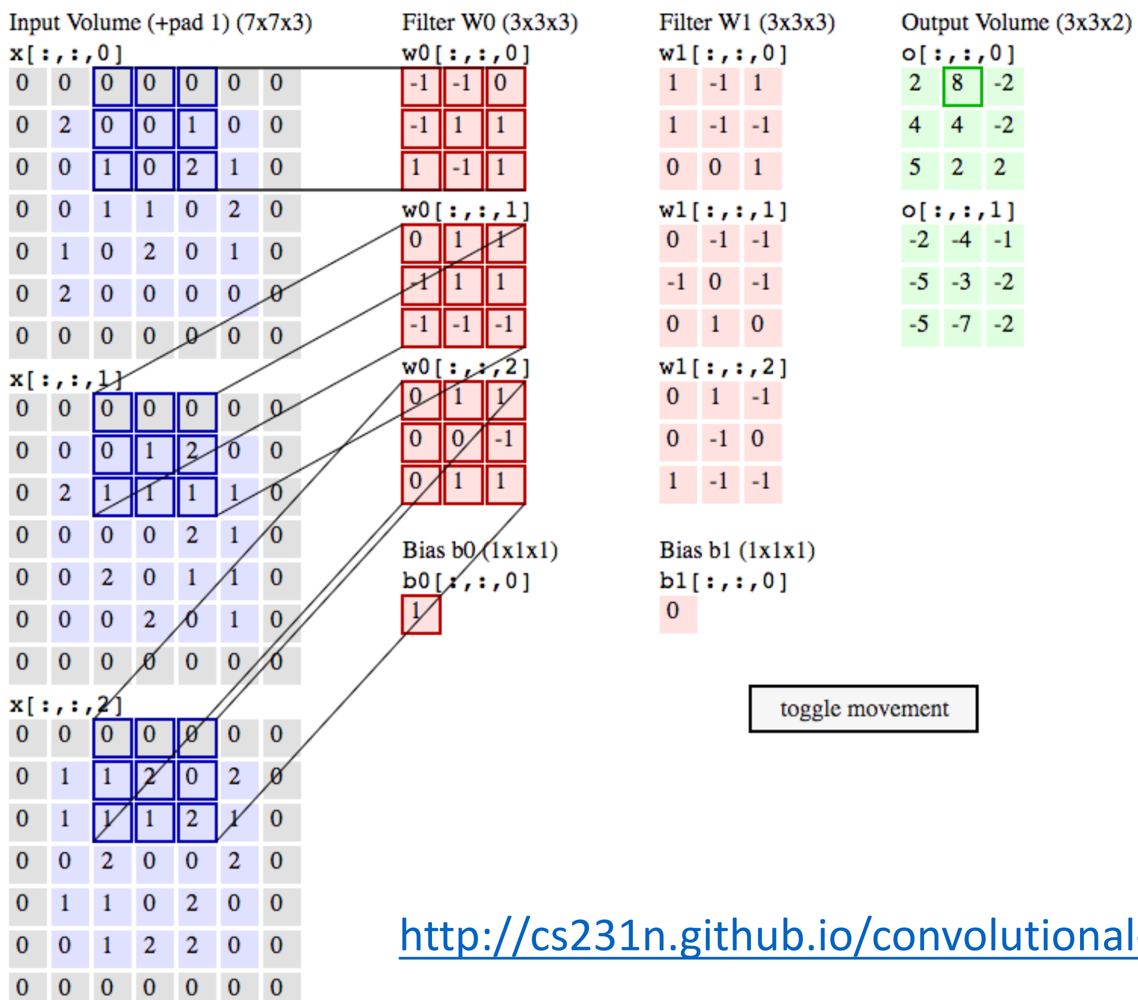
CNN Example





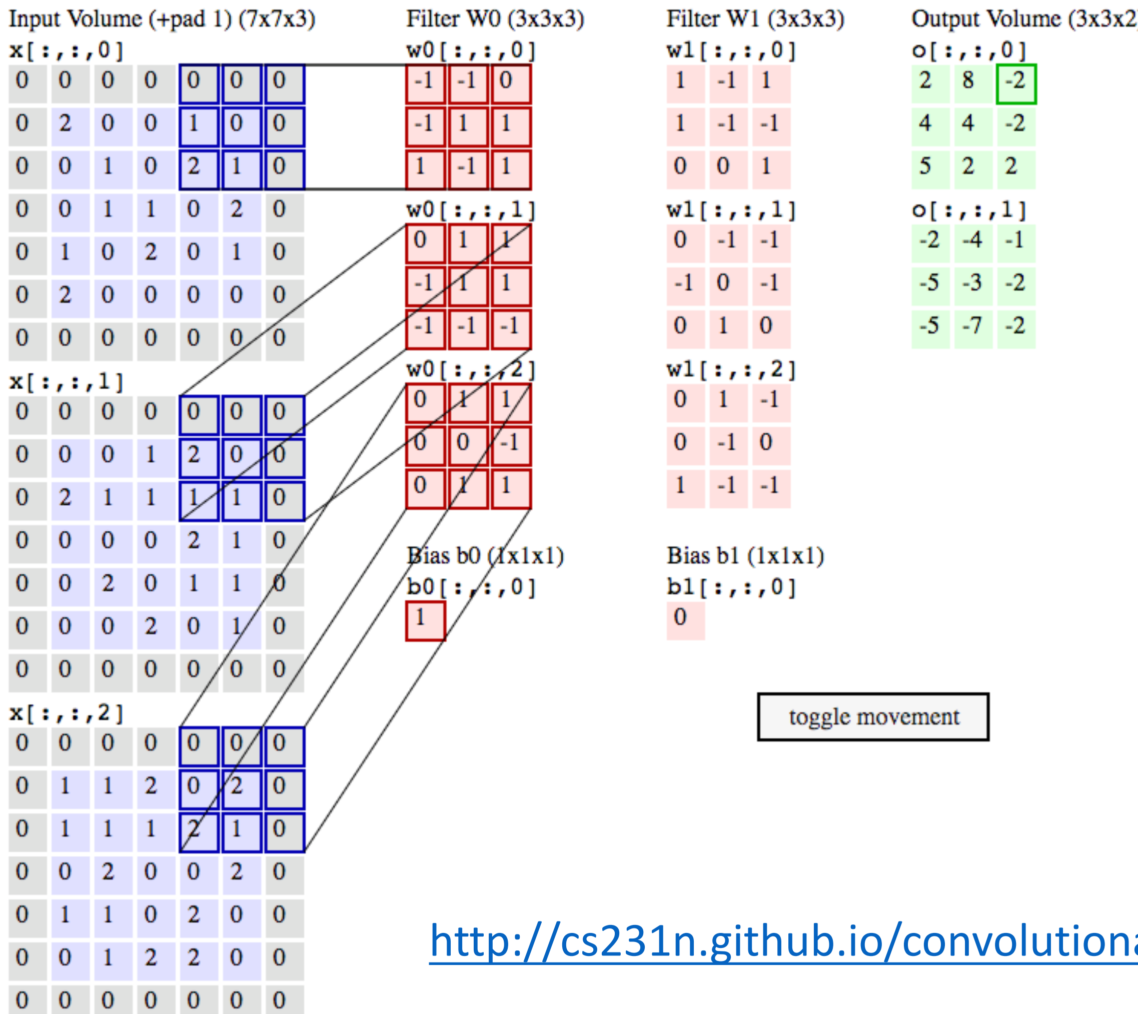
http://cs231n.github.io/convolutional-networks/

Convolutional Layers



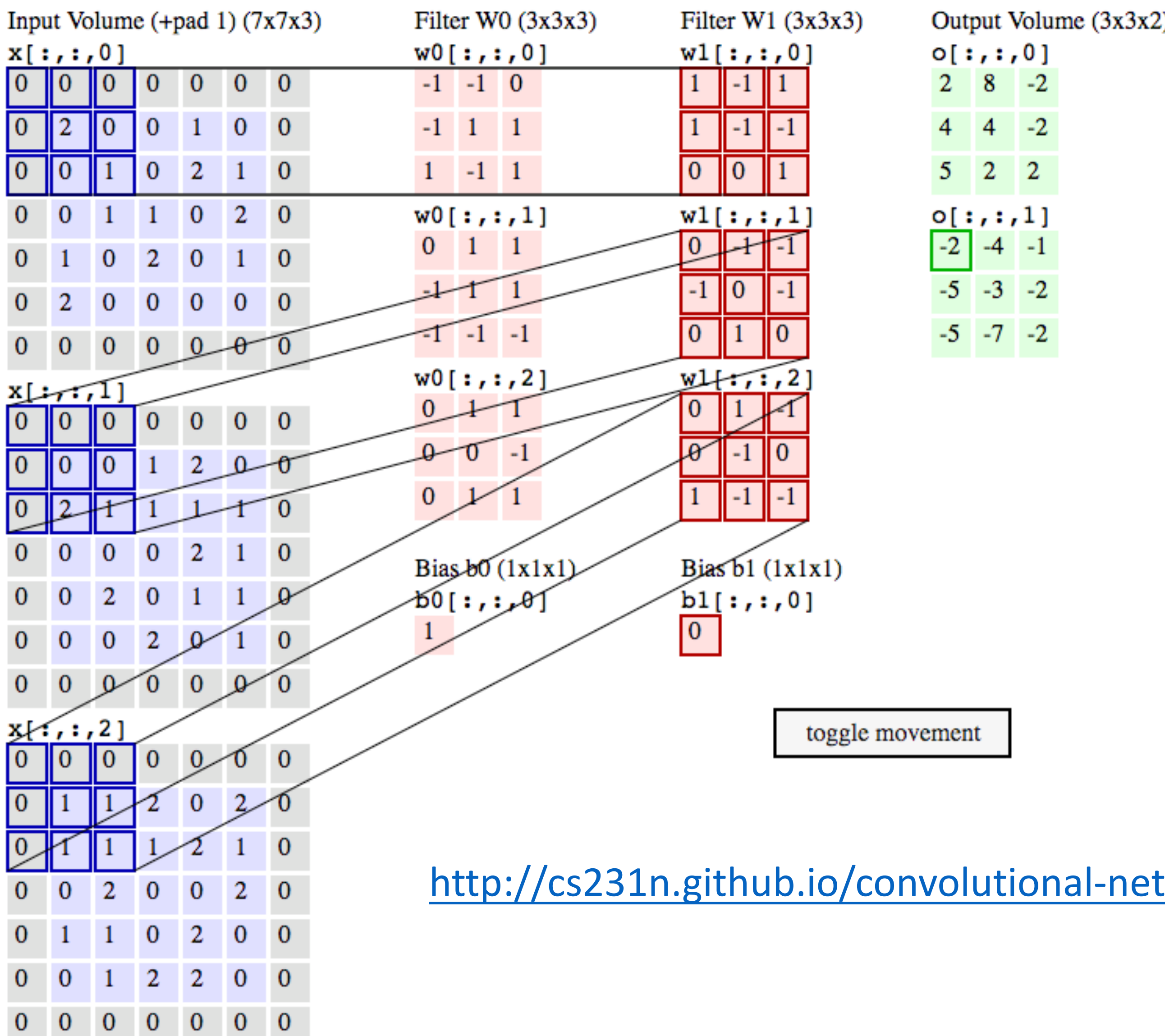
http://cs231n.github.io/convolutional-networks/

Convolutional Layers



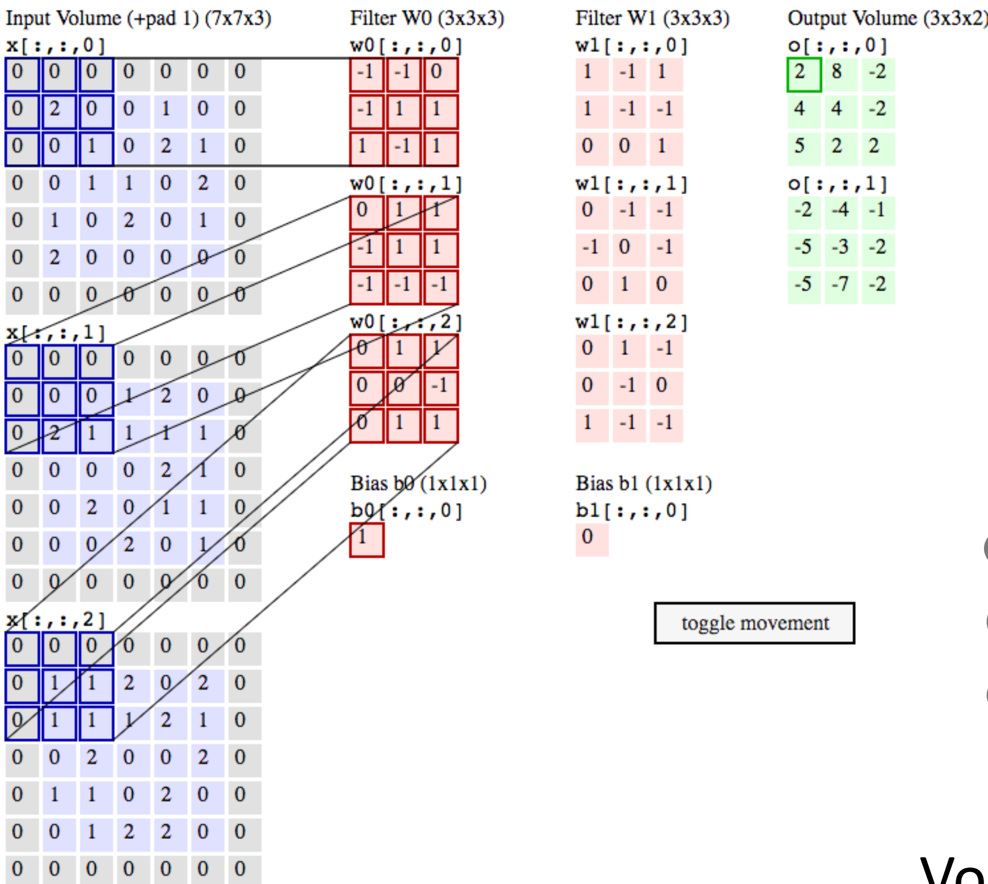
http://cs231n.github.io/convolutional-networks/

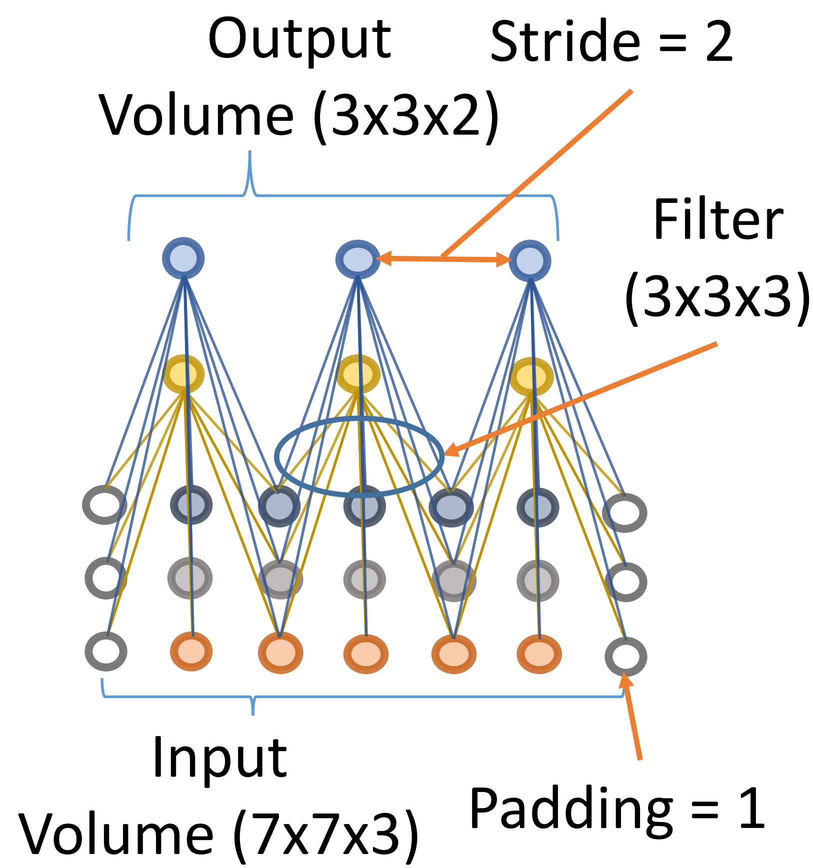
Convolutional Layers



http://cs231n.github.io/convolutional-networks/

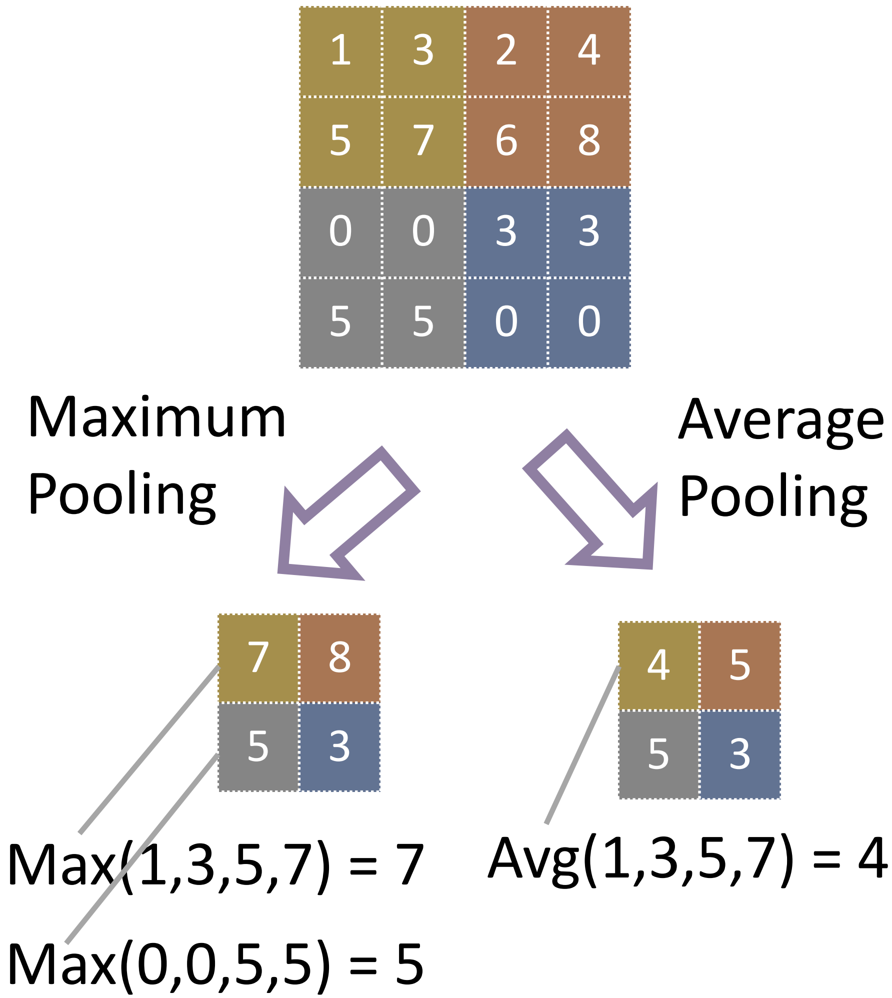
Quick question: how to change one parameter and generate an output of 5x5x2?

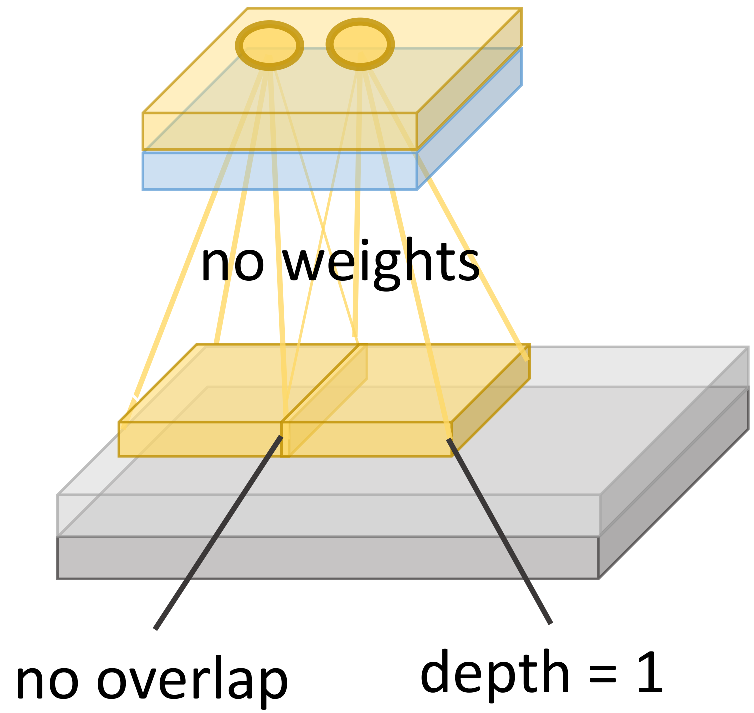




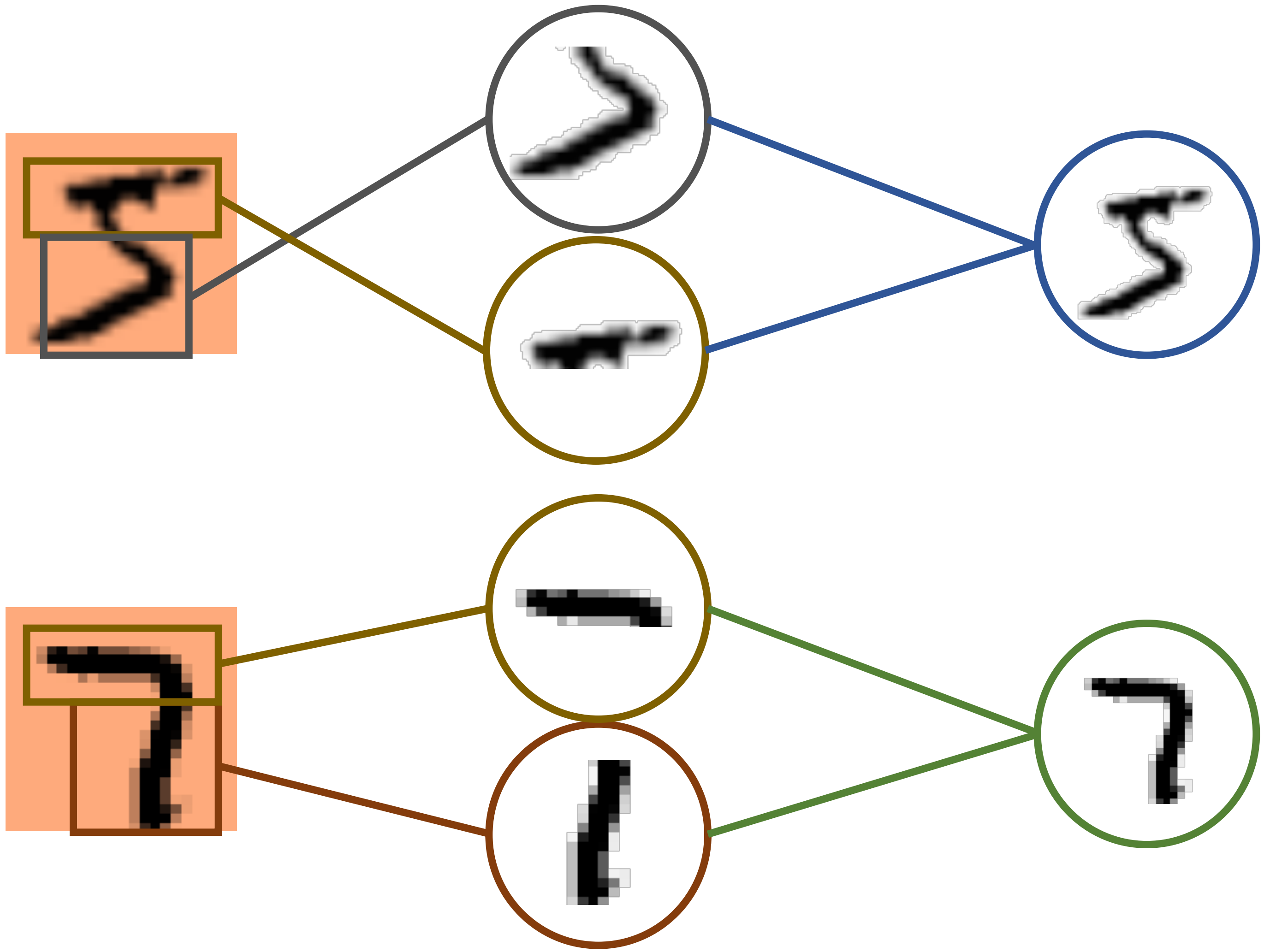
http://cs231n.github.io/convolutional-networks/

Pooling Layer



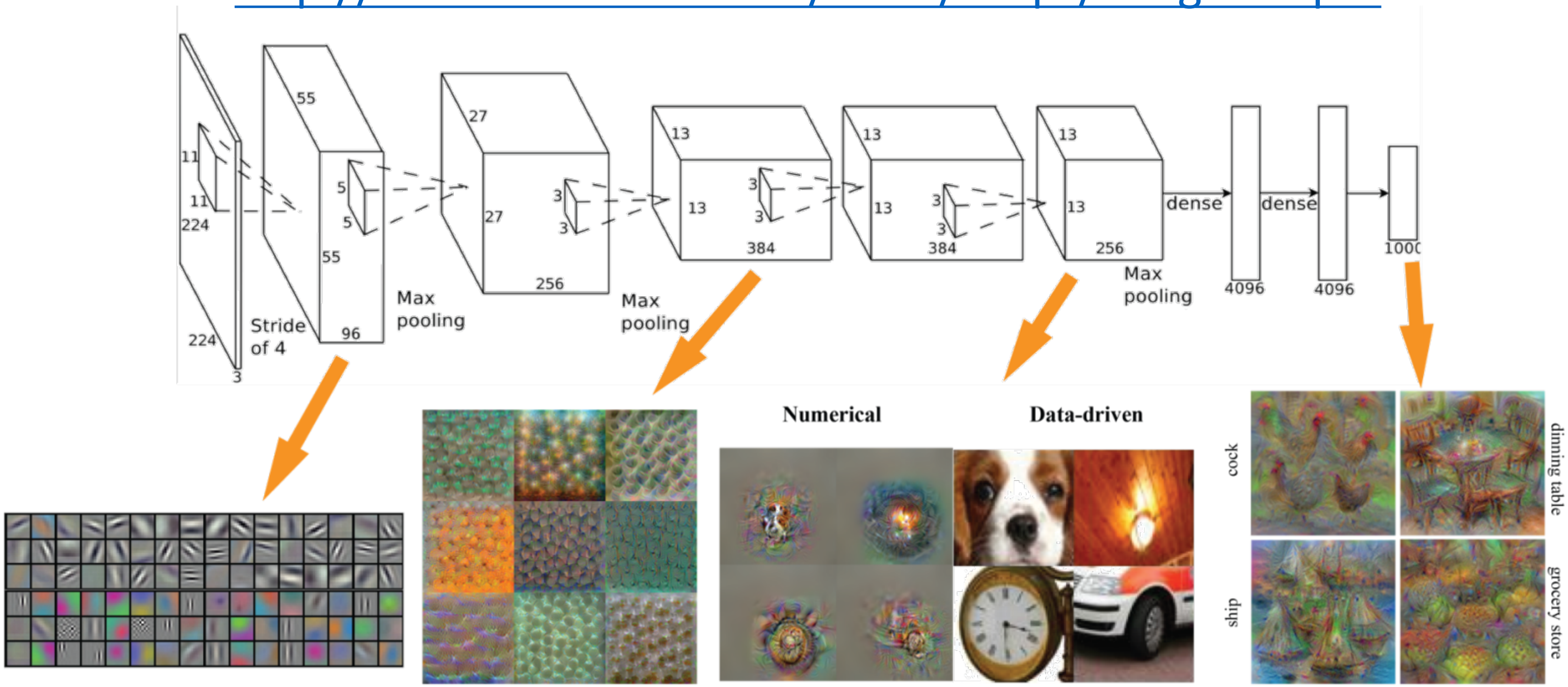


Why “Deep” Learning?



AlexNet

- Alexnet http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf



Conv 1: Edge+Blob

Conv 3: Texture

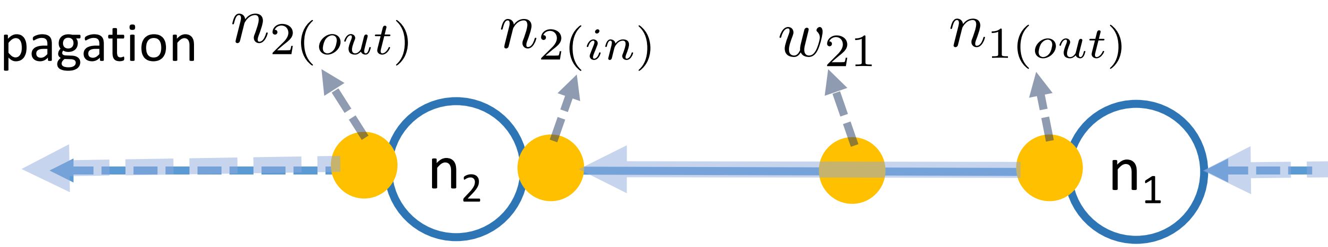
Conv 5: Object Parts

Fc8: Object Classes

http://vision03.csail.mit.edu/cnn_art/data/single_layer.png

Training

- Forward Propagation

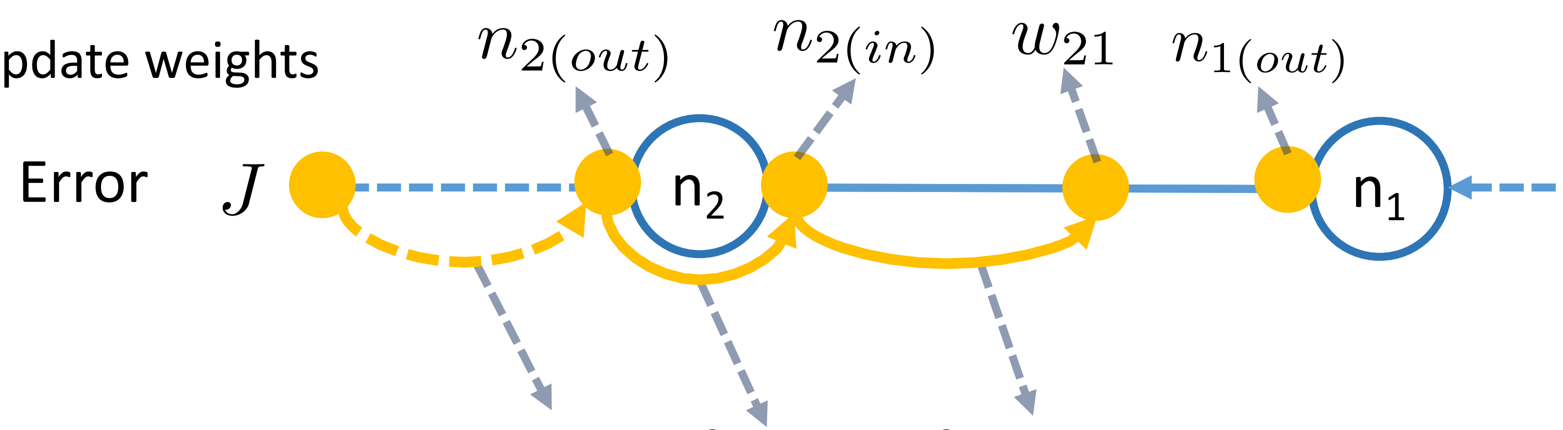


$$n_{2(in)} = w_{21} n_{1(out)}$$

$$n_{2(out)} = g(n_{2(in)}), \quad g \text{ is activation function}$$

Training

- Update weights



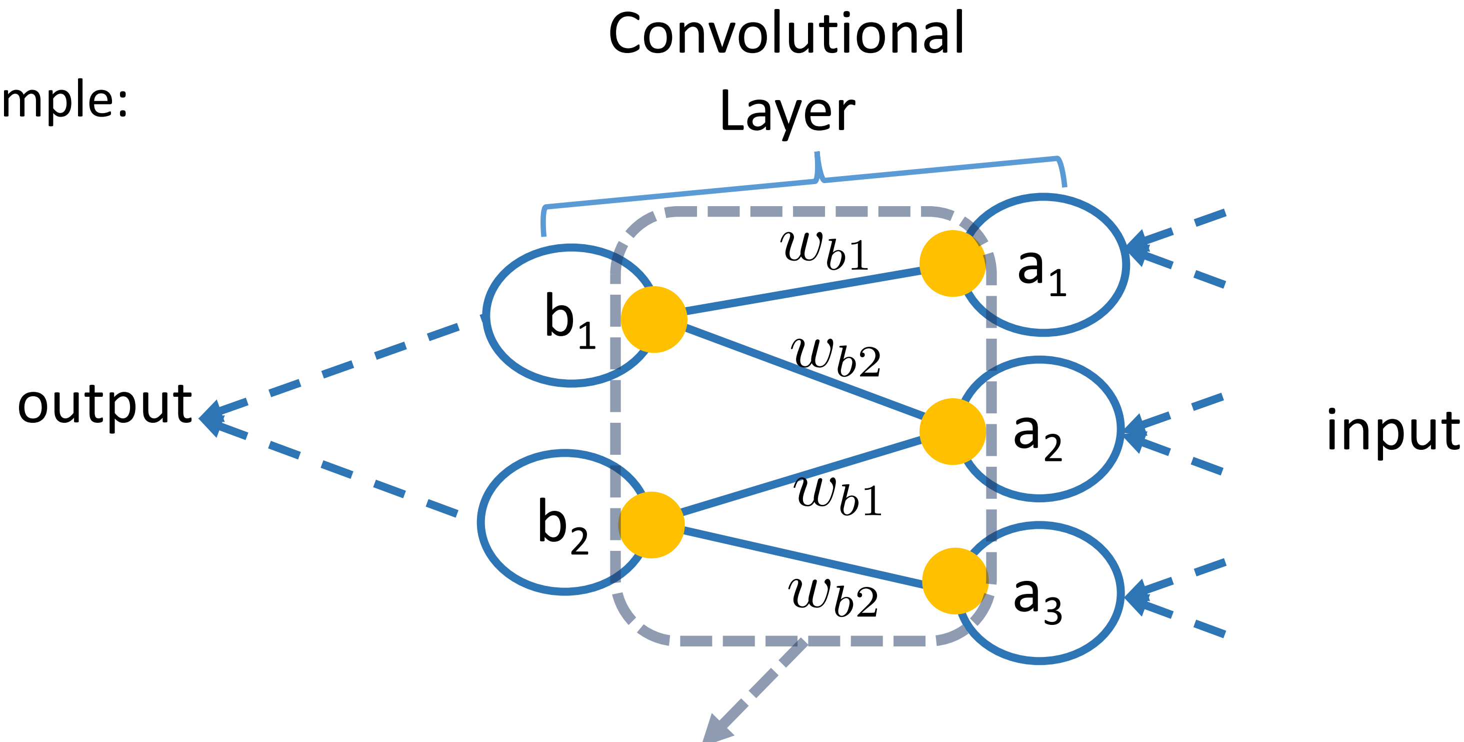
$$\frac{\partial J}{\partial w_{21}} = \frac{\partial J}{\partial n_{2(out)}} \frac{\partial n_{2(out)}}{\partial n_{2(in)}} \frac{\partial n_{2(in)}}{\partial w_{21}}$$

$$w_{21} \leftarrow w_{21} - \eta \frac{\partial J}{\partial w_{21}}$$

$$\Rightarrow w_{21} \leftarrow w_{21} - \eta \frac{\partial J}{\partial n_{2(out)}} \frac{\partial n_{2(out)}}{\partial n_{2(in)}} \frac{\partial n_{2(in)}}{\partial w_{21}}$$

Training Convolutional Layers

- example:



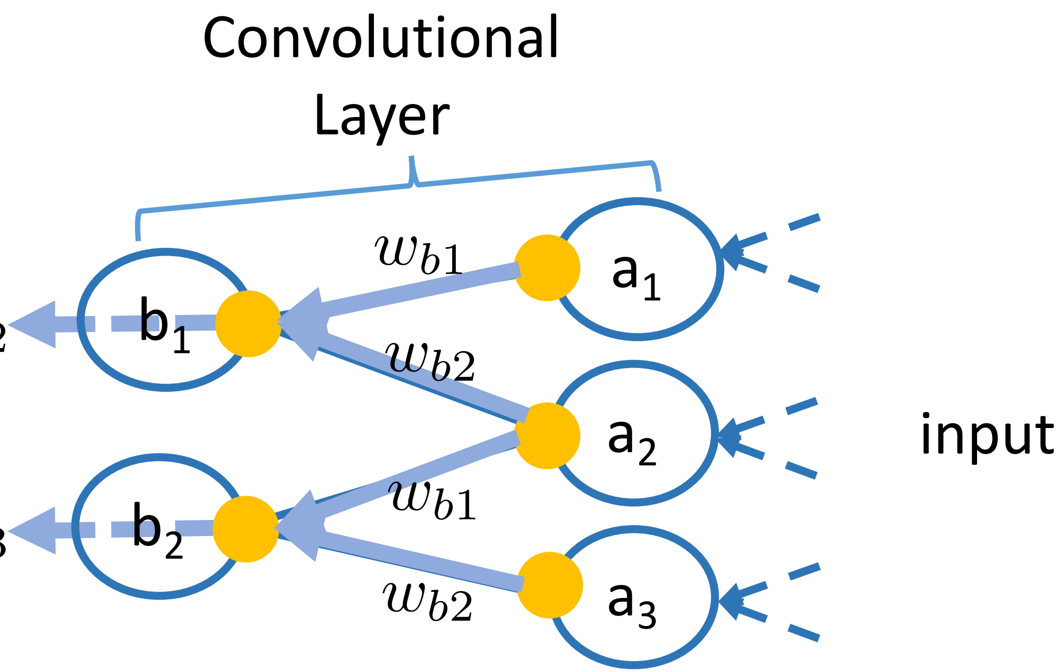
To simplify the notations, in the following slides, we make: b_1 means $b_{1(in)}$, a_1 means $a_{1(out)}$, and so on.

Training Convolutional Layers

- Forward propagation

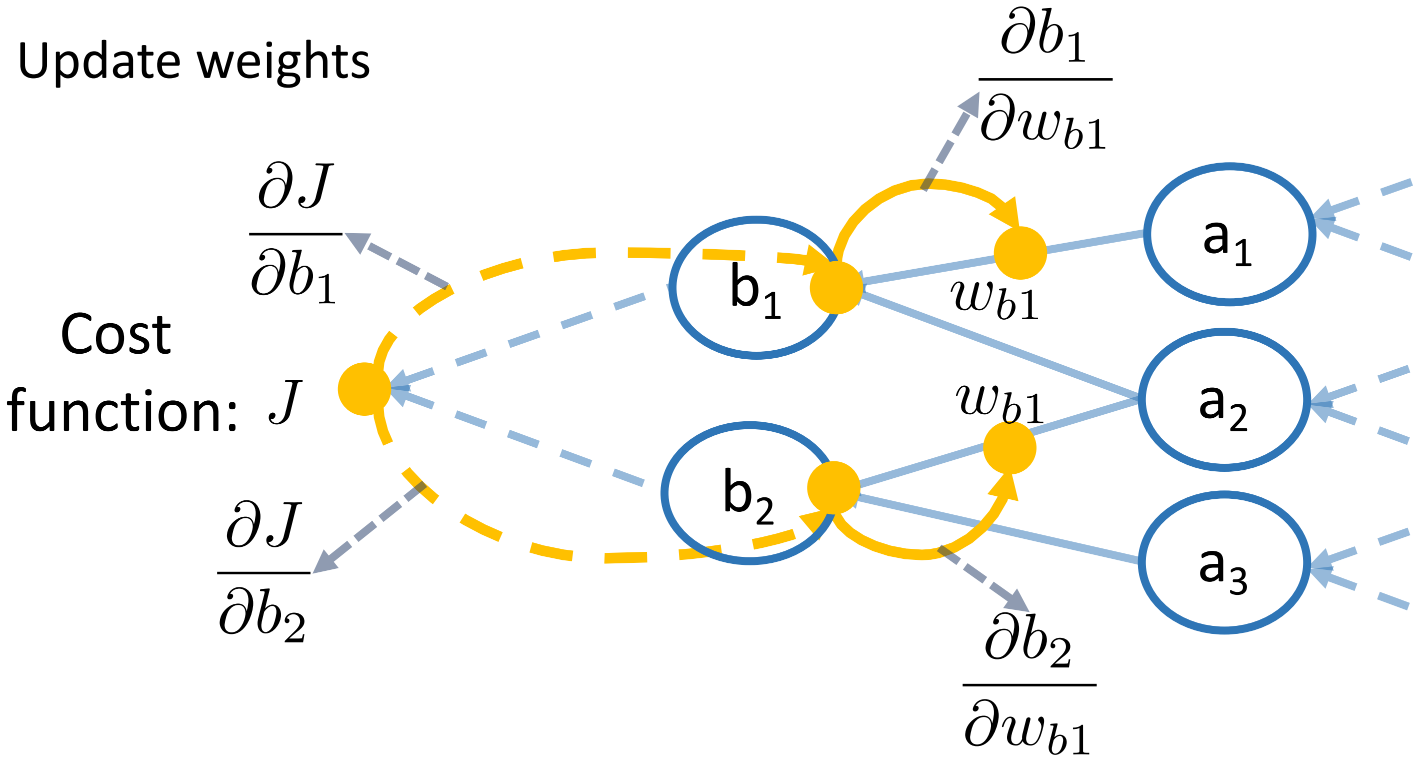
$$b_1 = w_{b1}a_1 + w_{b2}a_2$$

$$b_2 = w_{b1}a_2 + w_{b2}a_3$$



Training Convolutional Layers

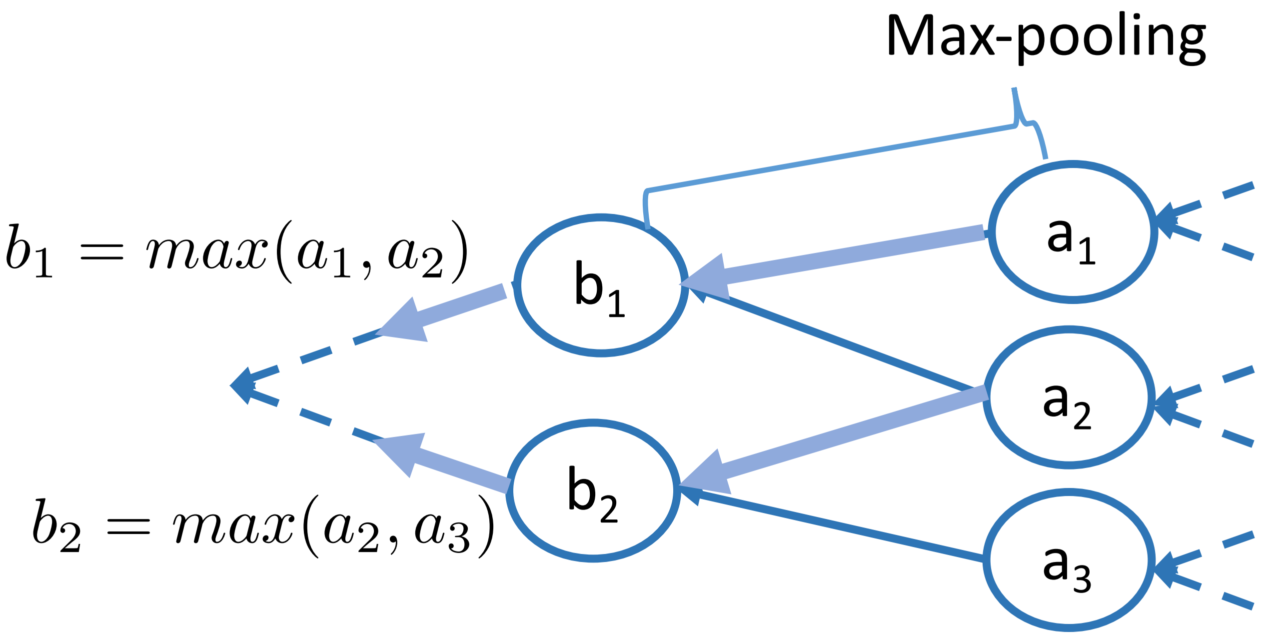
- Update weights



$$w_{b1} \leftarrow w_{b1} - \eta\left(\frac{\partial J}{\partial b_1}\frac{\partial b_1}{\partial w_{b1}} + \frac{\partial J}{\partial b_2}\frac{\partial b_2}{\partial w_{b1}}\right)$$

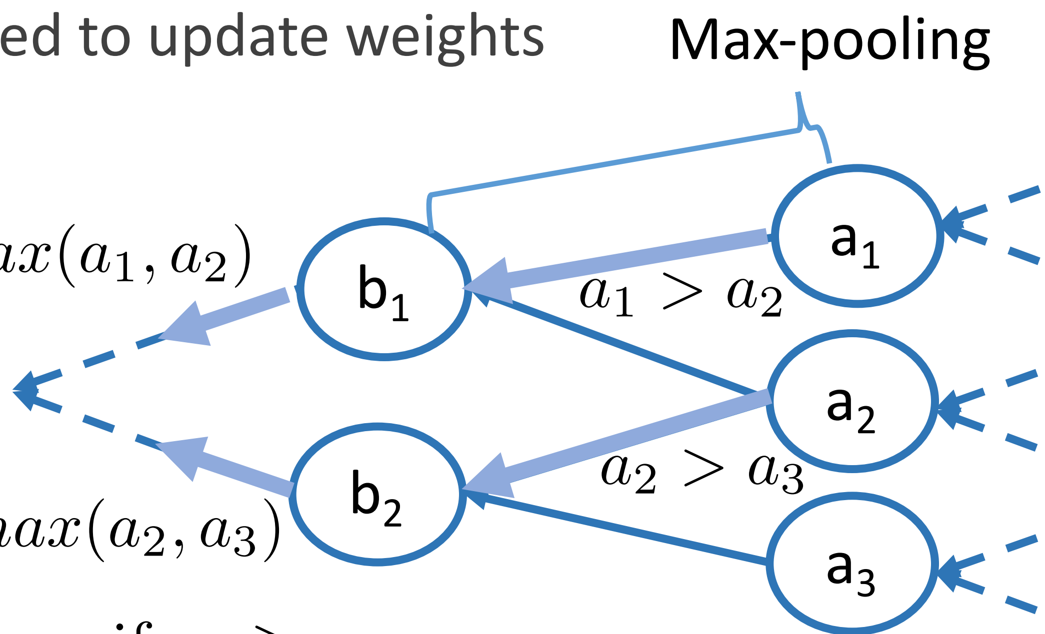
Piazza Poll:
Max-Pooling Layers during Training

- How to update weights for max-pooling layers?



Max-Pooling Layers during Training

- Pooling layers have no weights
- No need to update weights



$$b_1 = max(a_1, a_2)$$

$$b_2 = max(a_2, a_3)$$

$$= \begin{cases} a_2 \text{ if } a_2 \geq a_3 \\ a_3 \text{ otherwise} \end{cases}$$

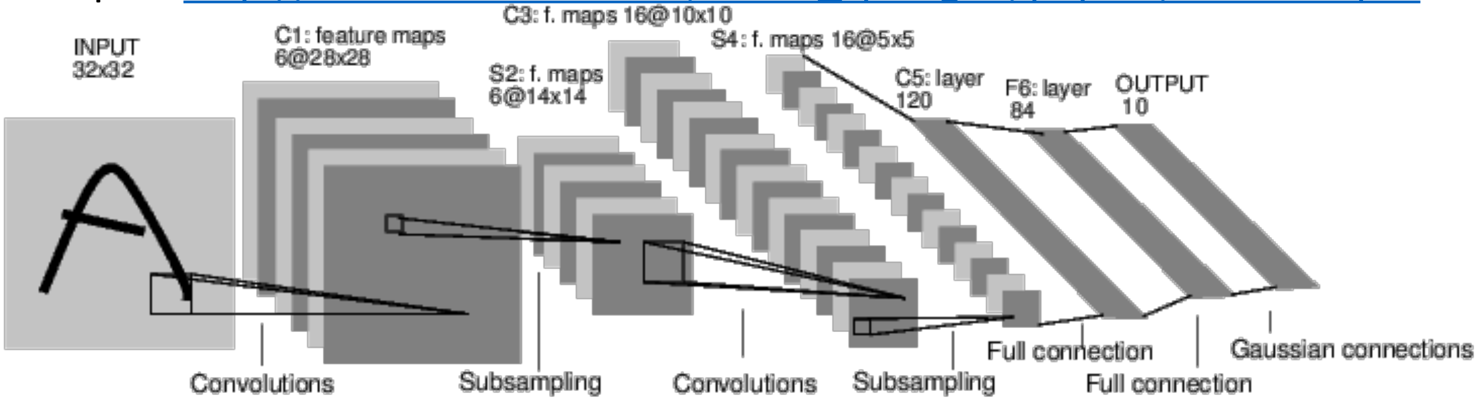
$$\frac{\partial b_2}{\partial a_2} = \begin{cases} 1 \text{ if } a_2 \geq a_3 \\ 0 \text{ otherwise} \end{cases}$$

Outline

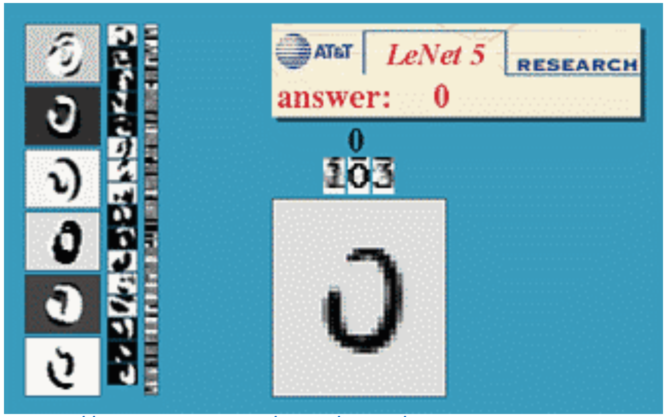
- CNN(Convolutional Neural Networks) Introduction
- Evolution of CNN
- Visualizing the Features
- CNN as Artist
- Sentiment Analysis by CNN

LeNet

- Paper: http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf



Yann LeCun



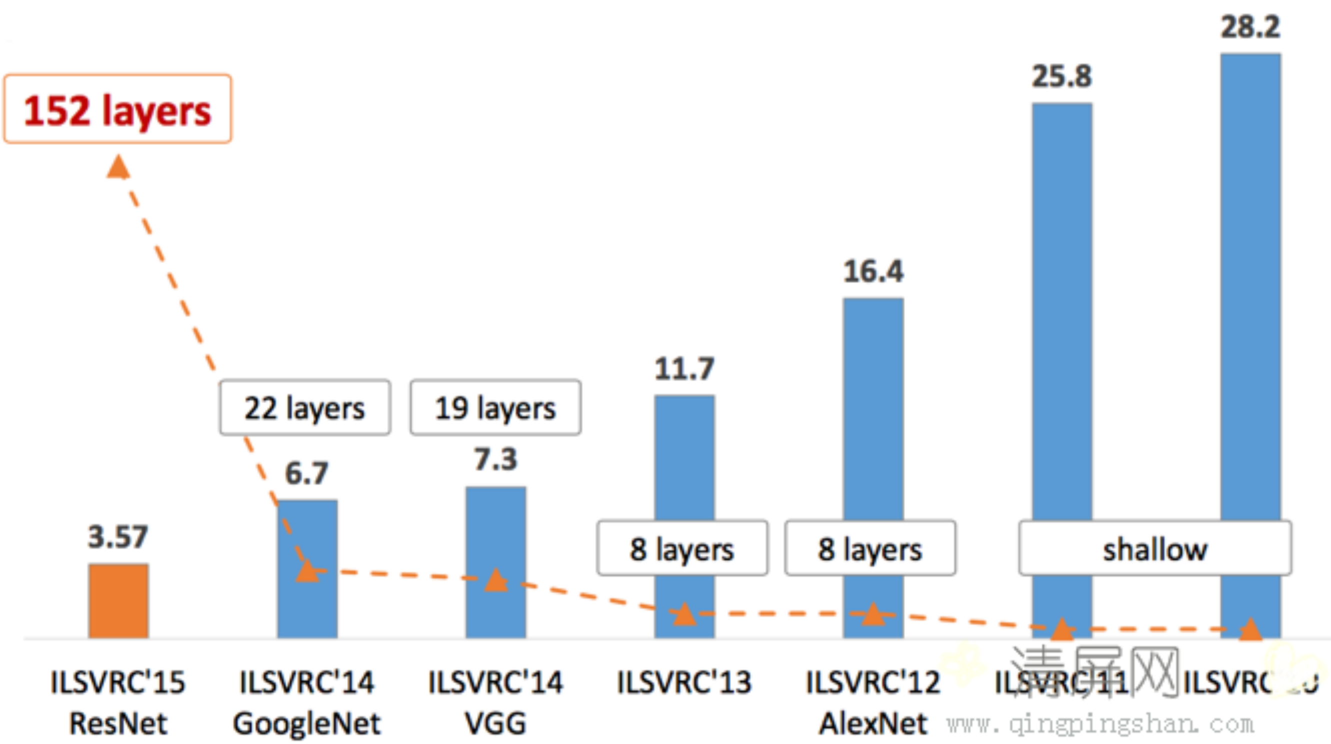
http://yann.lecun.com/exdb/lenet/

ImageNet Challenge

- ImageNet Large Scale Visual Recognition Challenge
  - http://image-net.org/challenges/LSVRC/
- Dataset :
  - 1000 categories
  - Training: 1,200,000
  - Validation: 50,000
  - Testing: 100,000

http://vision.stanford.edu/Datasets/collage_s.png

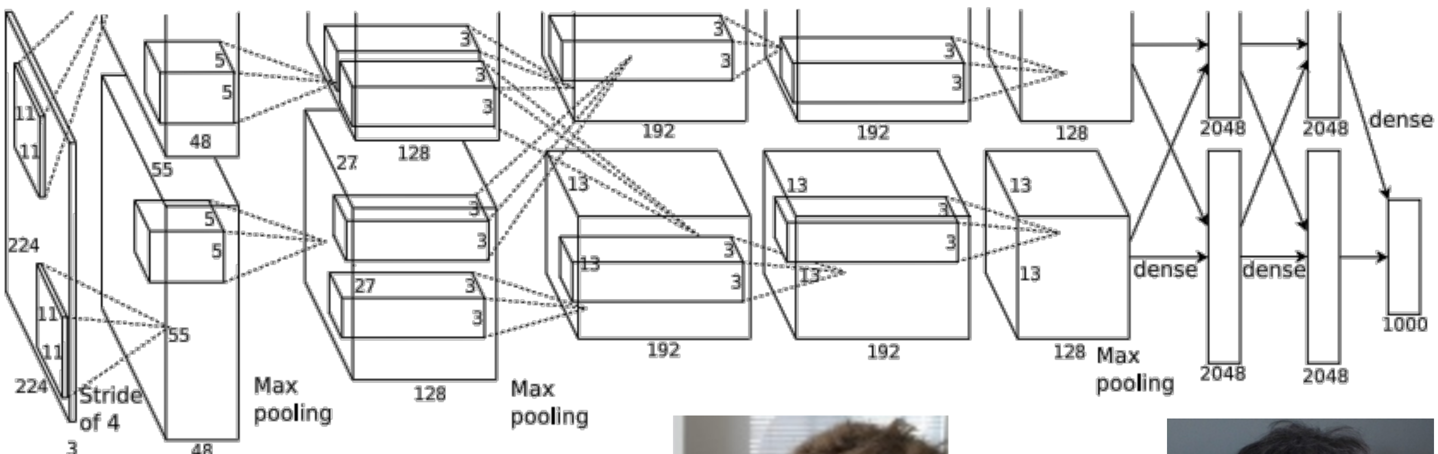
ImageNet Challenge

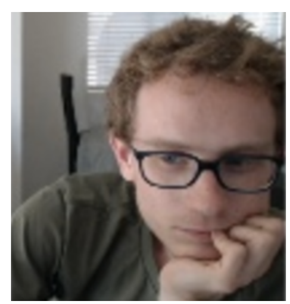


http://www.qingpingshan.com/uploads/allimg/160818/1J22QI5-0.png

AlexNet (2012)

- Paper: http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf
- The resurgence of Deep Learning




Alex Krizhevsky


Geoffrey Hinton

VGGNet (2014)

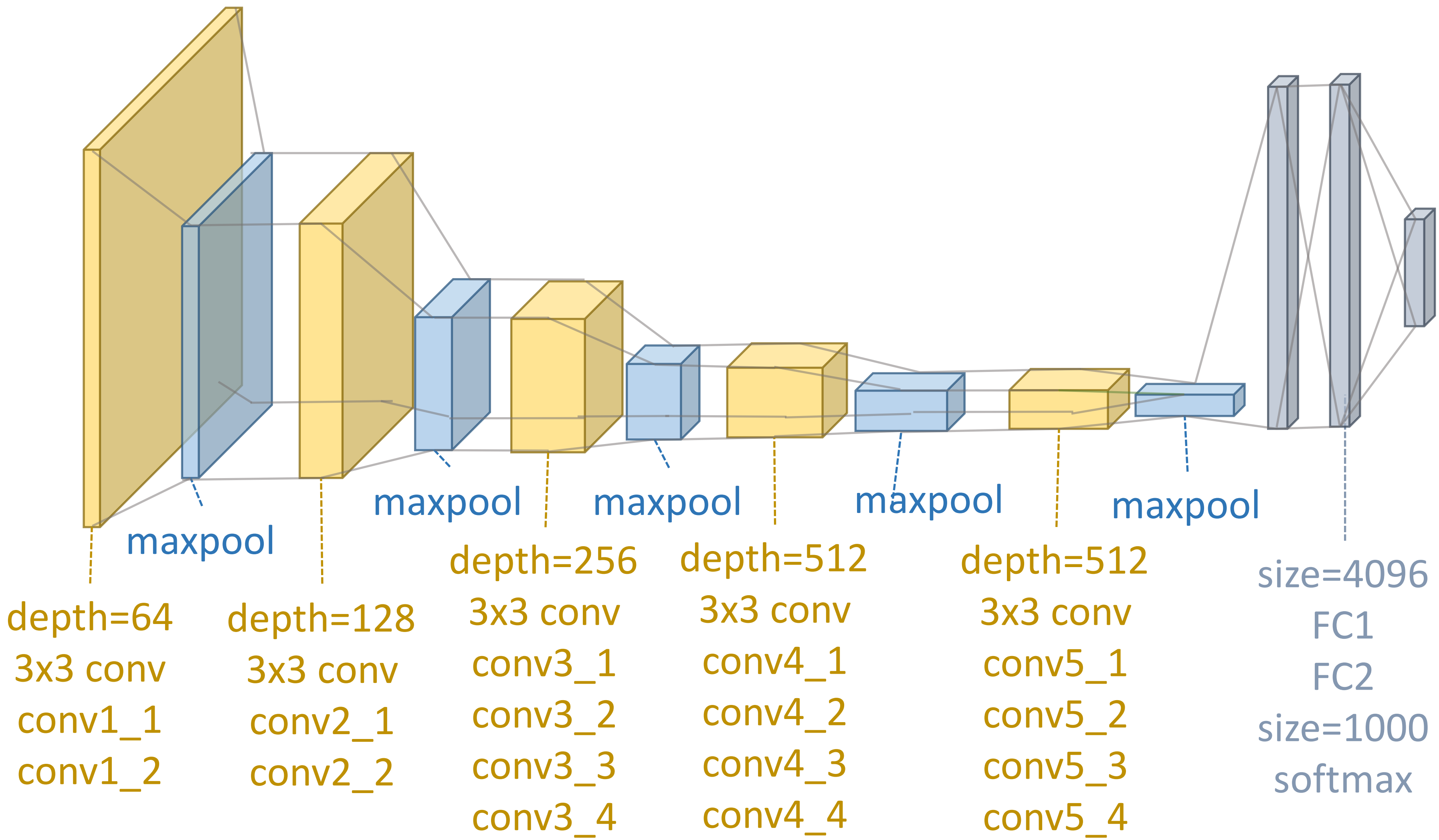
More layers & smaller filters (3x3) is better
More non-linearity, fewer parameters

- Paper: https://arxiv.org/abs/1409.1556

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					

D: VGG16
E: VGG19
All filters are 3x3

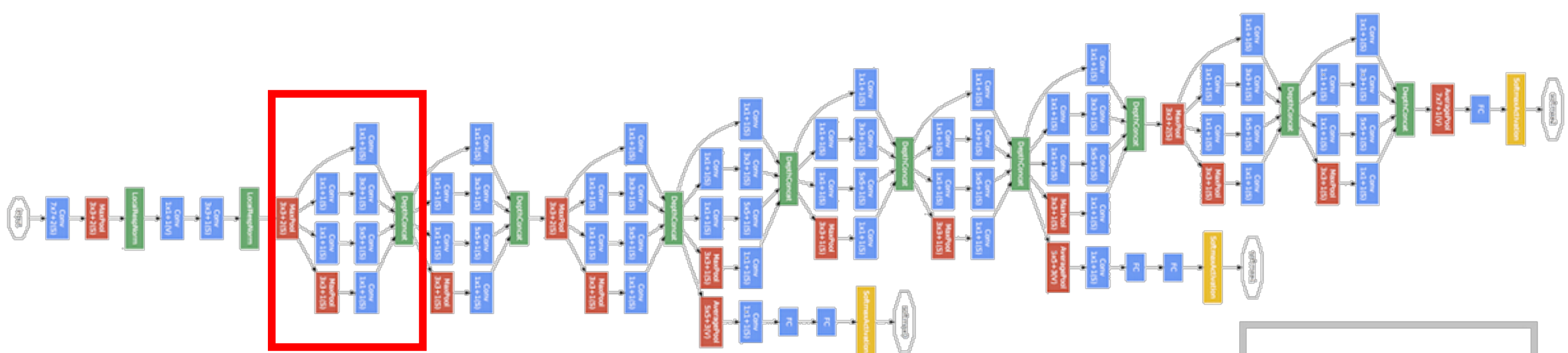
VGG 19


maxpool
maxpool
maxpool
maxpool
maxpool
depth=64
3x3 conv
conv1_1
conv1_2
depth=128
3x3 conv
conv2_1
conv2_2
depth=256
3x3 conv
conv3_1
conv3_2
conv3_3
conv3_4
depth=512
3x3 conv
conv4_1
conv4_2
conv4_3
conv4_4
depth=512
3x3 conv
conv5_1
conv5_2
conv5_3
conv5_4
size=4096
FC1
FC2
size=1000
softmax

GoogLeNet (2014)

- Paper: http://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf

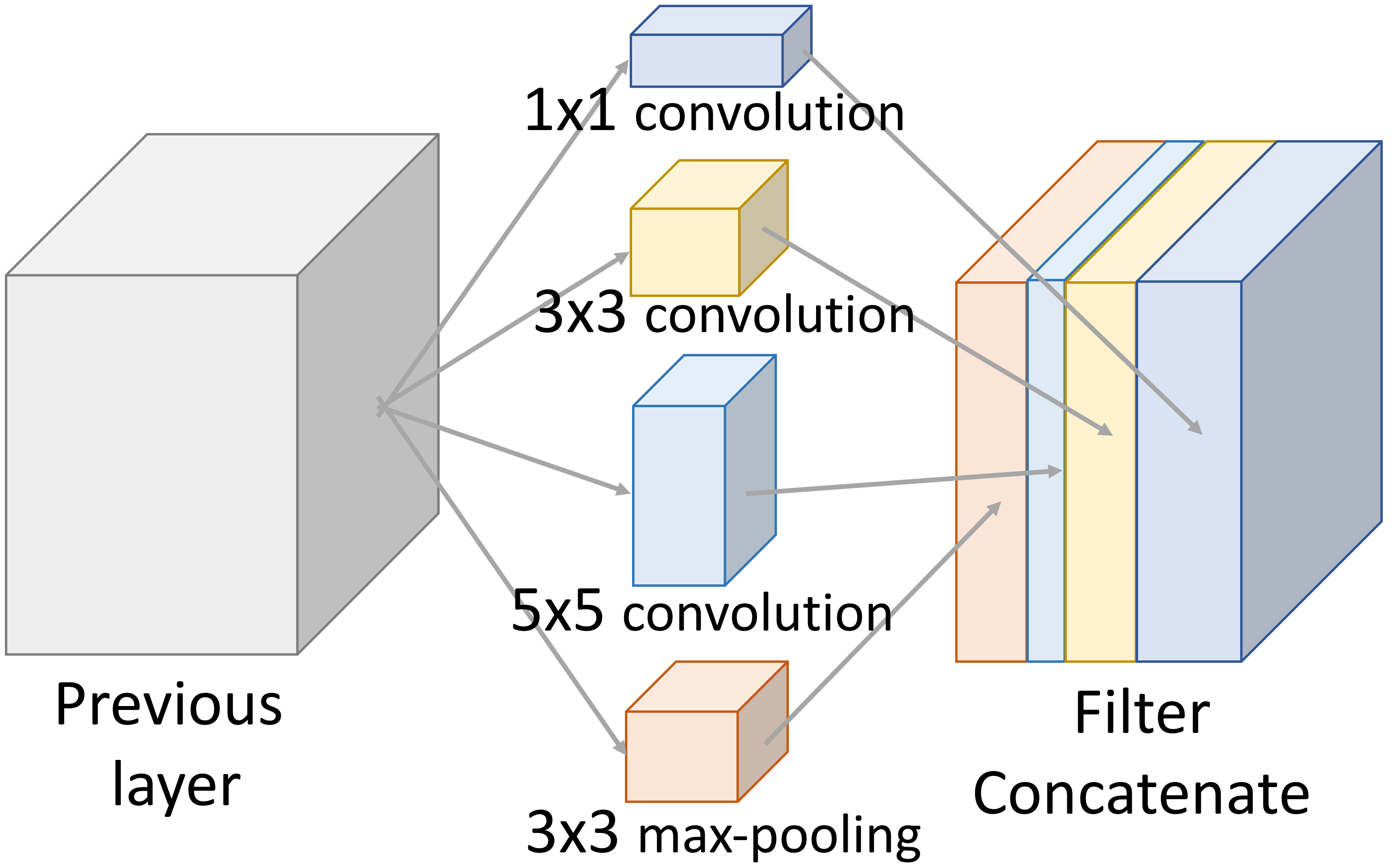
22 layers deep network



Inception Module

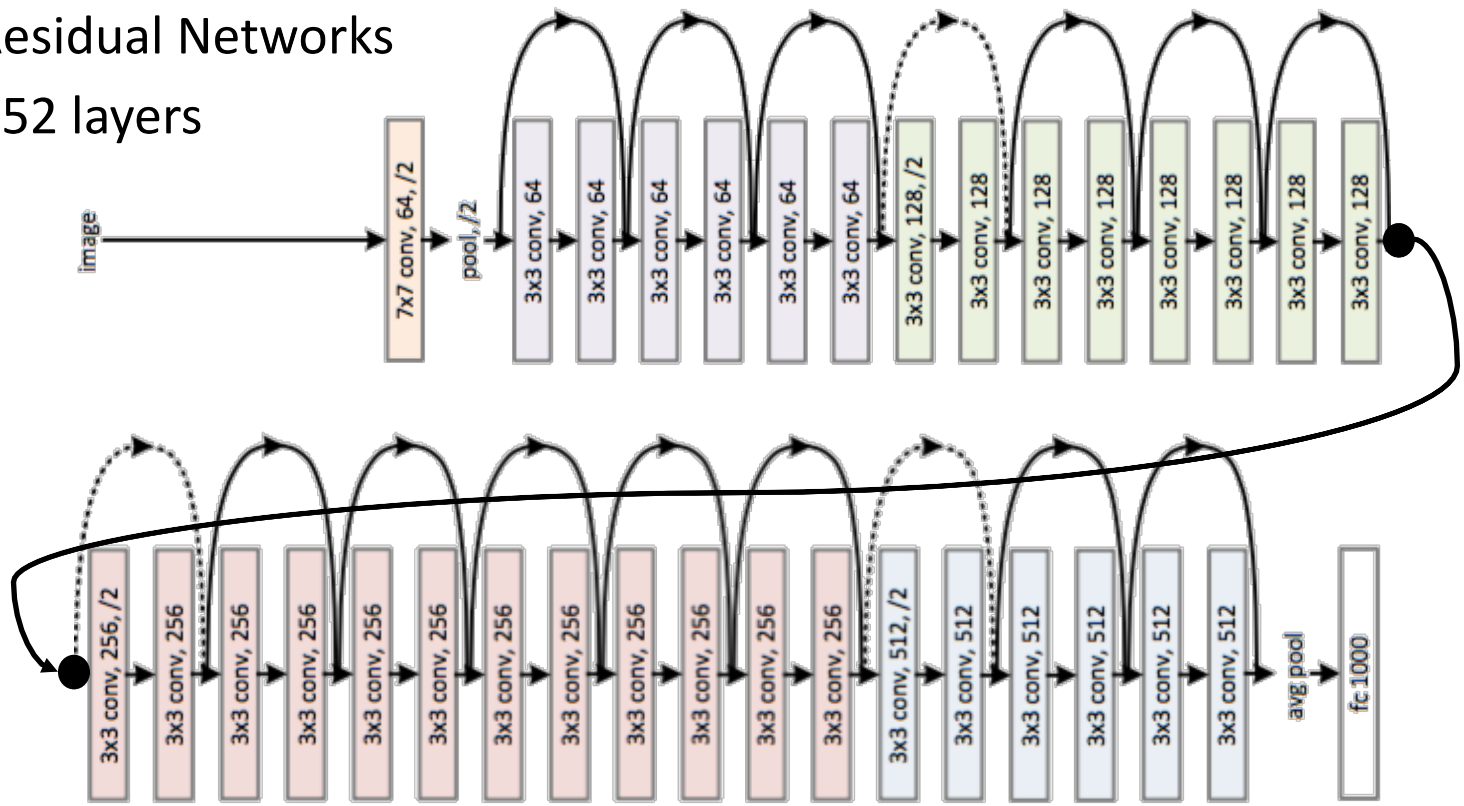
Convolution
Pooling
Softmax
Other

Inception Module



ResNet (2015)

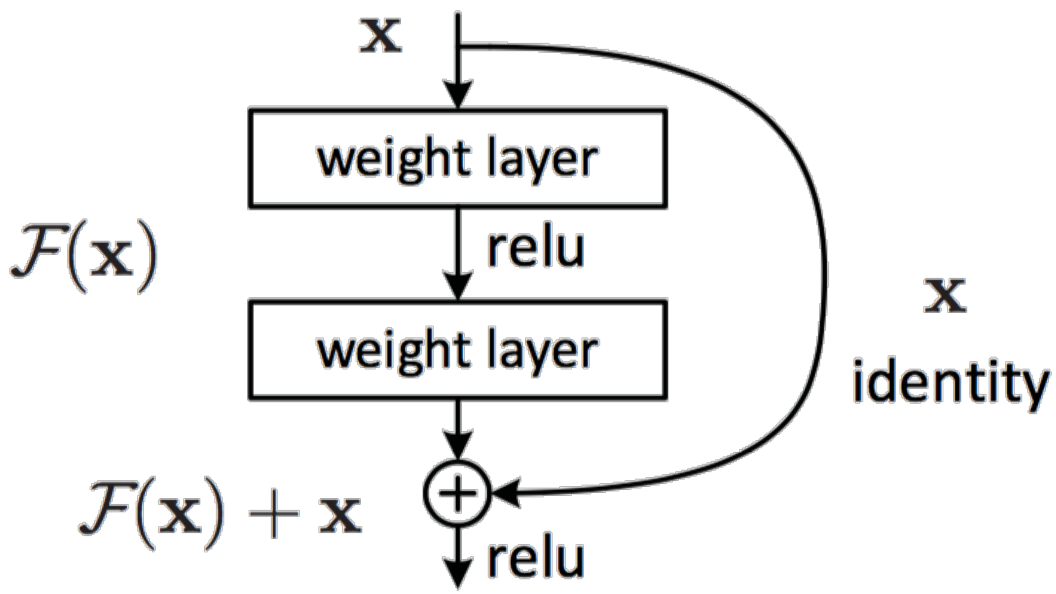
- Paper: https://arxiv.org/abs/1512.03385
- Residual Networks
- 152 layers



ResNet

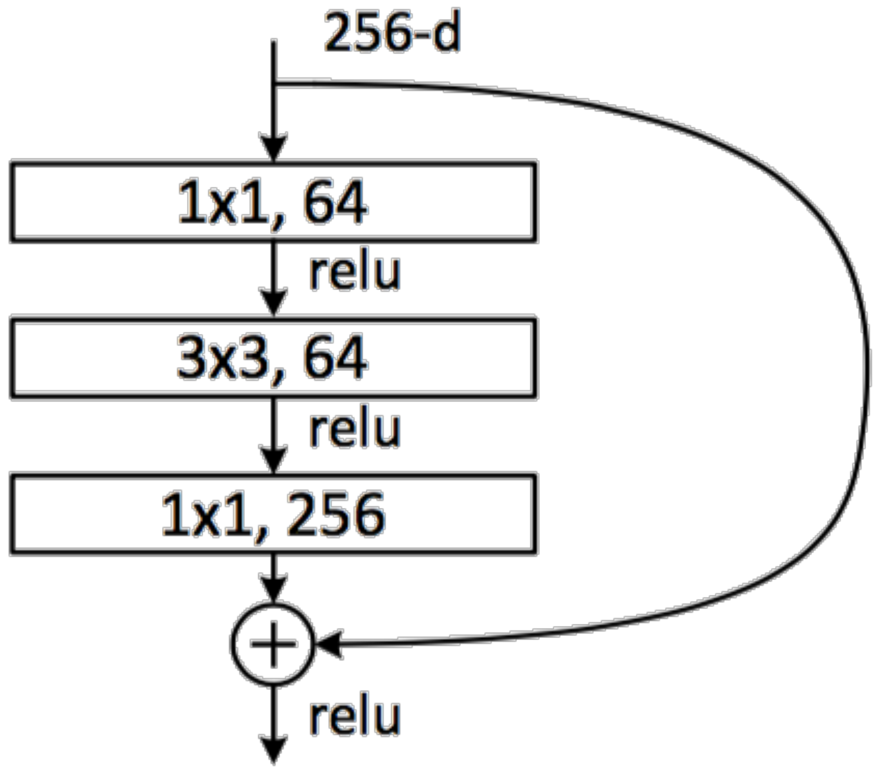
- Residual learning: a building block

Residual function



Residual Learning with Dimension Reduction

- using 1x1 filters

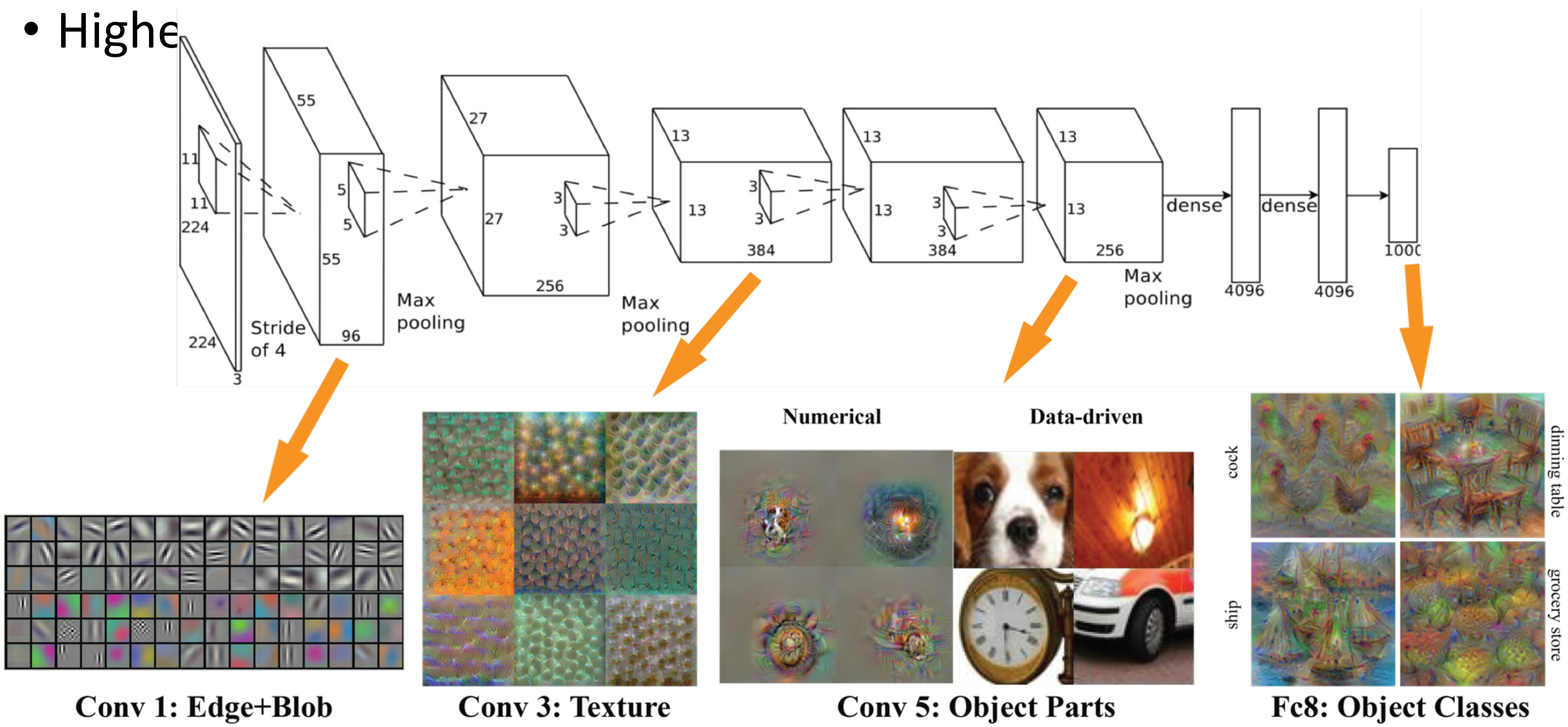


Pretrained Model Download

- http://www.vlfeat.org/matconvnet/pretrained/
  - Alexnet:
    - http://www.vlfeat.org/matconvnet/models/imagenet-matconvnet-alex.mat
  - VGG19:
    - http://www.vlfeat.org/matconvnet/models/imagenet-vgg-verydeep-19.mat
  - GoogLeNet:
    - http://www.vlfeat.org/matconvnet/models/imagenet-googlenet-dag.mat
  - ResNet
    - http://www.vlfeat.org/matconvnet/models/imagenet-resnet-152-dag.mat

Using Pretrained Model

- Lower layers：edge, blob, texture (more general)
- Highe

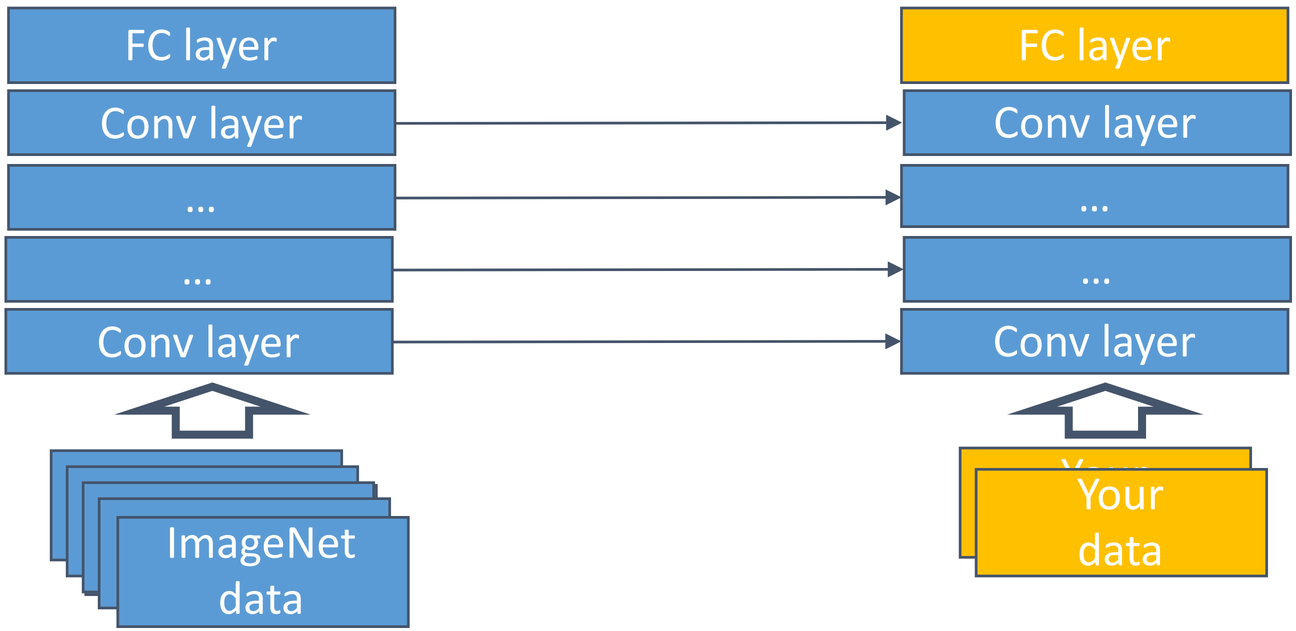


http://vision03.csail.mit.edu/cnn_art/data/single_layer.png

Transfer Learning

- The Pretrained Model is trained on ImageNet dataset

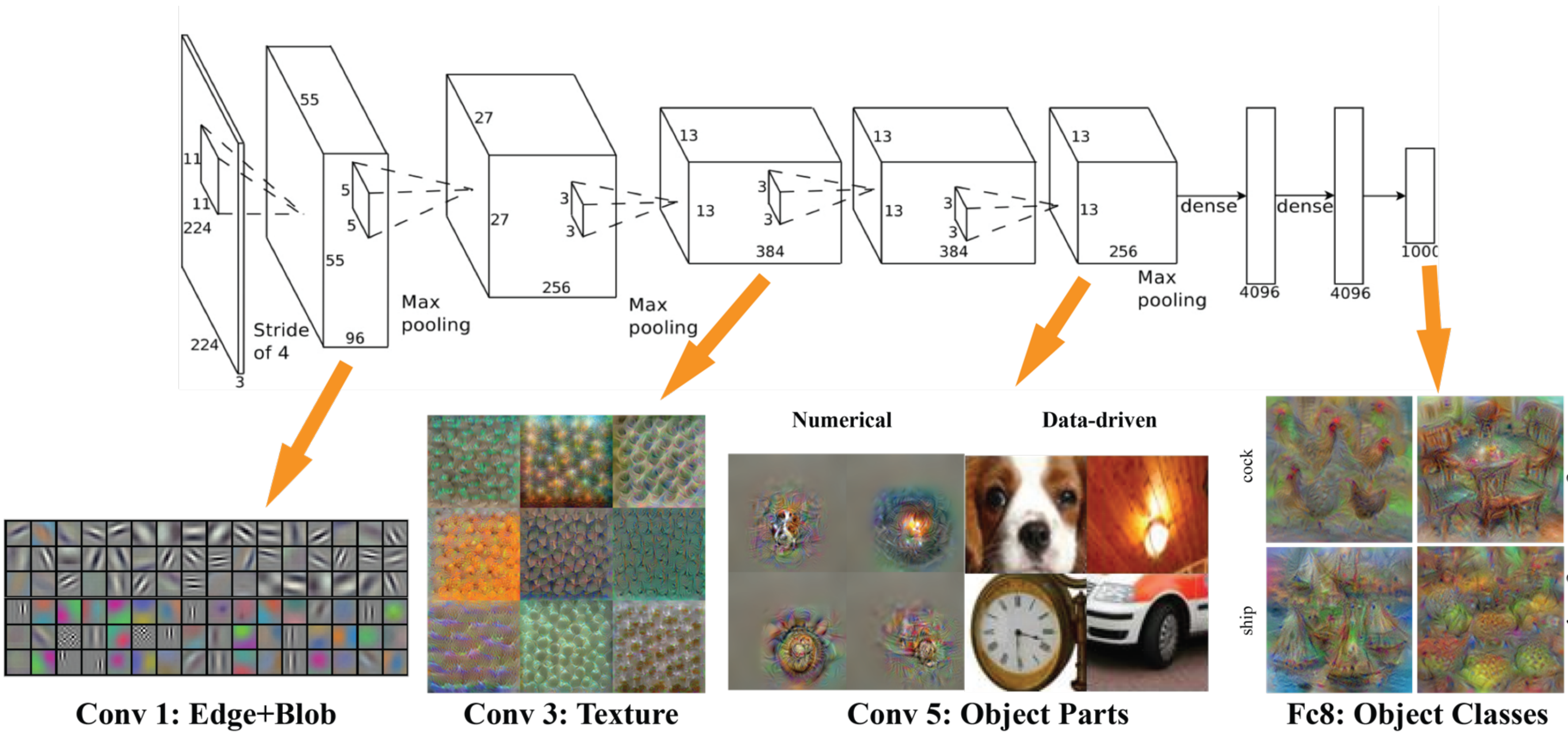
- If your data is similar to the ImageNet data
  - Fix all CNN Layers
  - Train FC layer



Outline

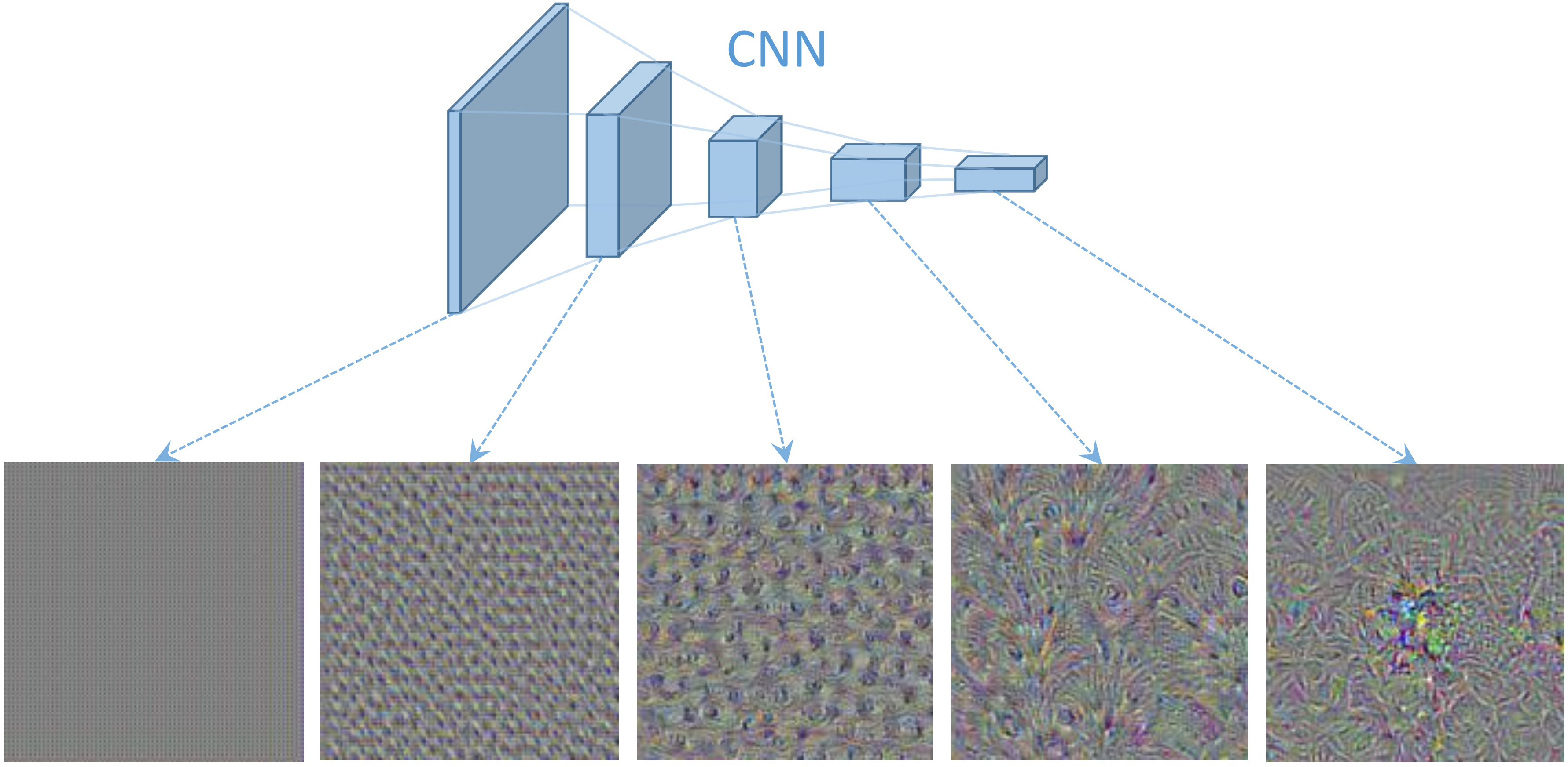
- CNN(Convolutional Neural Networks) Introduction
- Evolution of CNN
- Visualizing the Features
- Sentiment Analysis by CNN

Visualizing CNN



http://vision03.csail.mit.edu/cnn_art/data/single_layer.png

Different Layers of Visualization



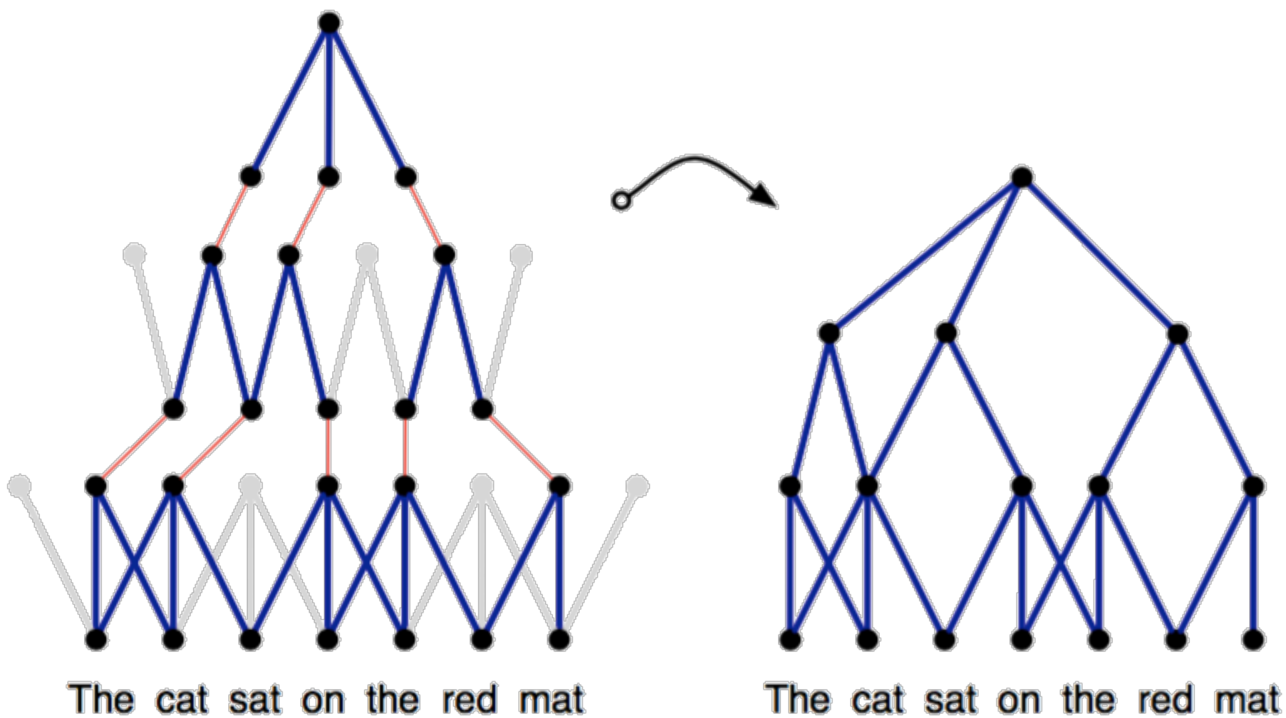
Multiscale Image Generation

Outline

- CNN(Convolutional Neural Networks) Introduction
- Evolution of CNN
- Visualizing the Features
- Sentiment Analysis by CNN

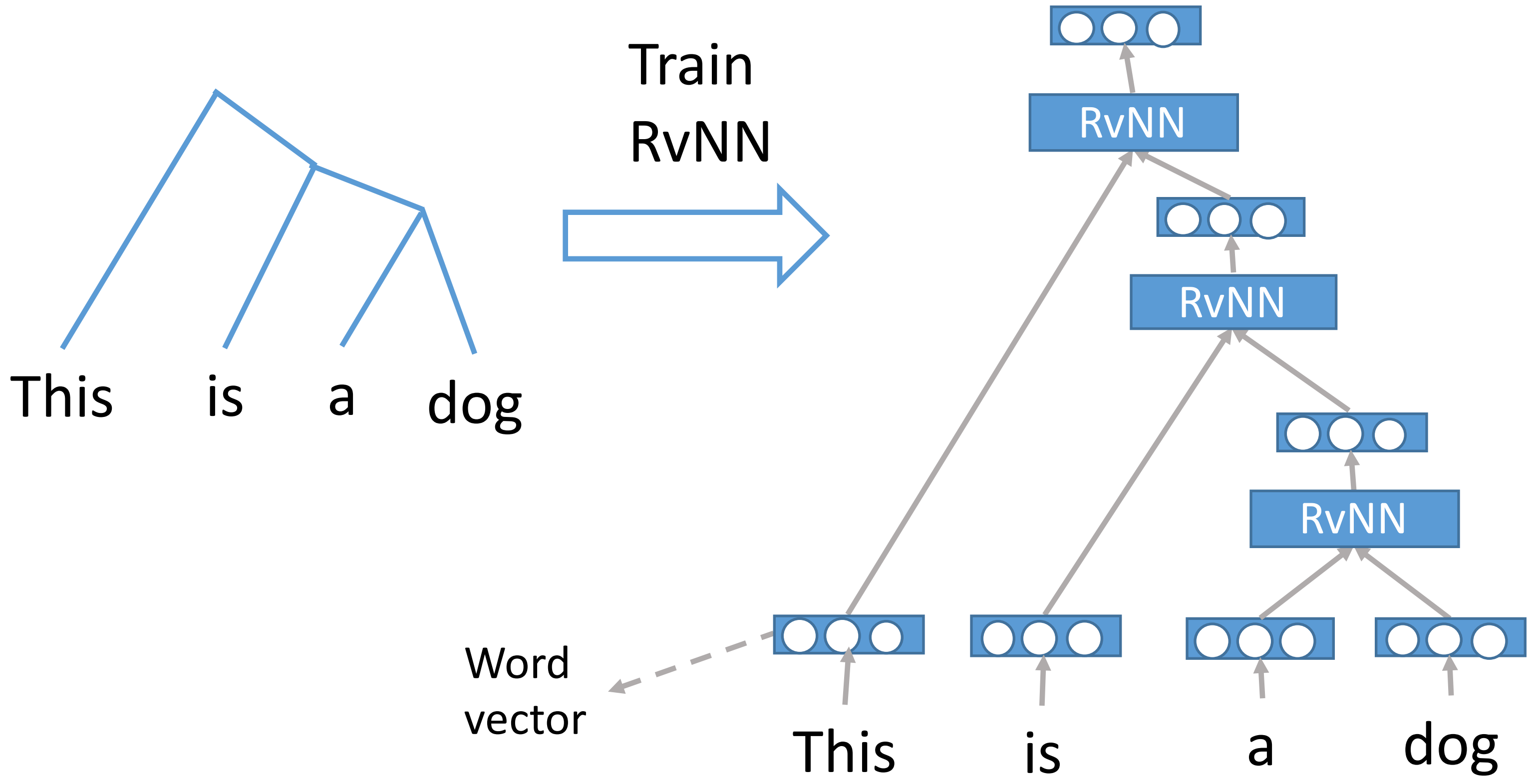
A Convolutional Neural Network for Modelling Sentences

- Paper: https://arxiv.org/abs/1404.2188
- Source code: https://github.com/FredericGodin/DynamicCNN



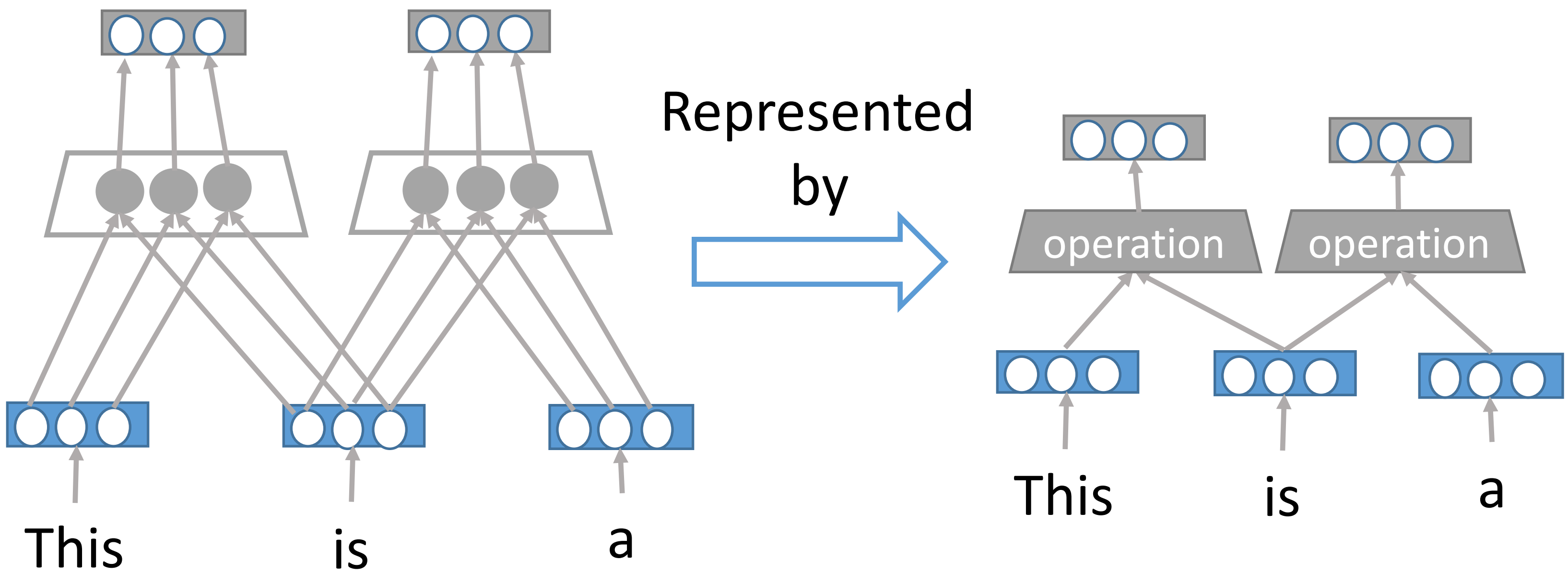
Drawbacks of Recursive Neural Networks(RvNN)

- Need human-labeled syntax tree during training



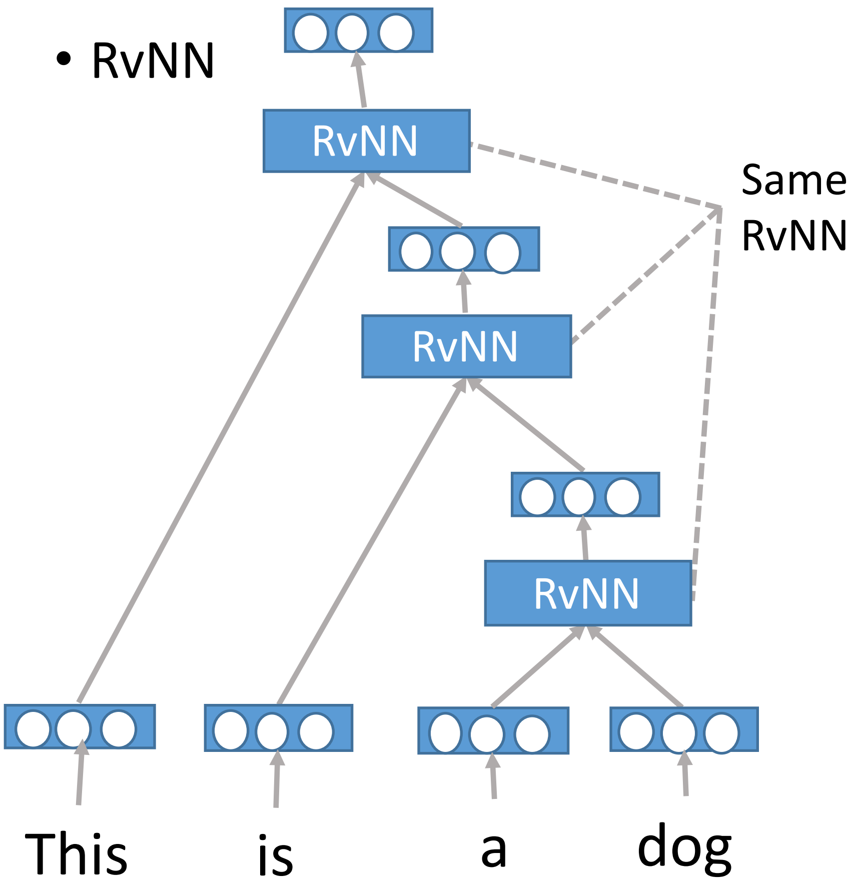
Element-wise 1D operations on word vectors

- 1D Convolution or 1D Pooling

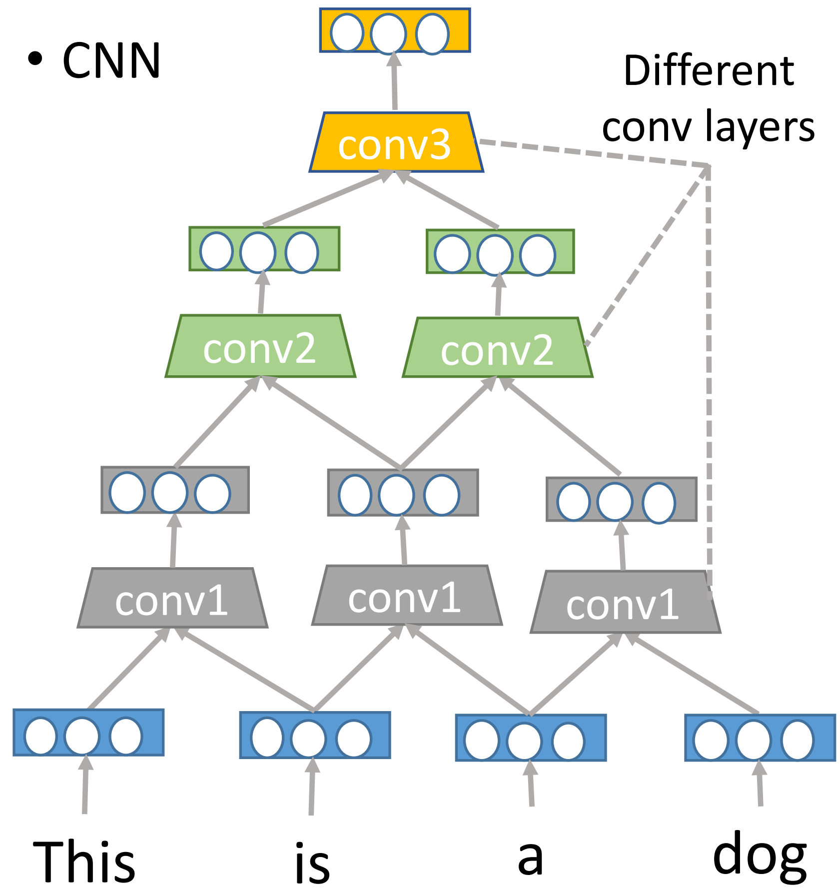


From RvNN to CNN

- RvNN

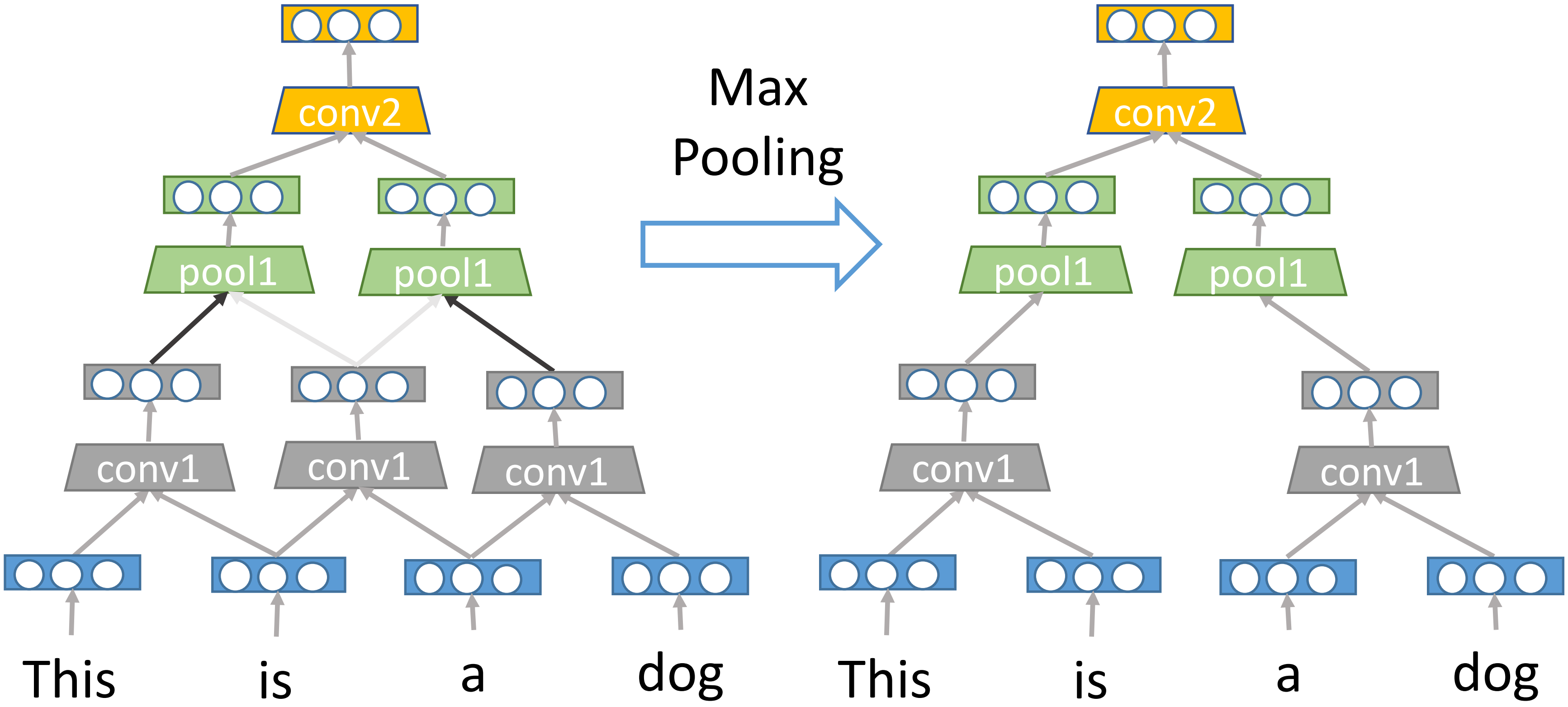


- CNN



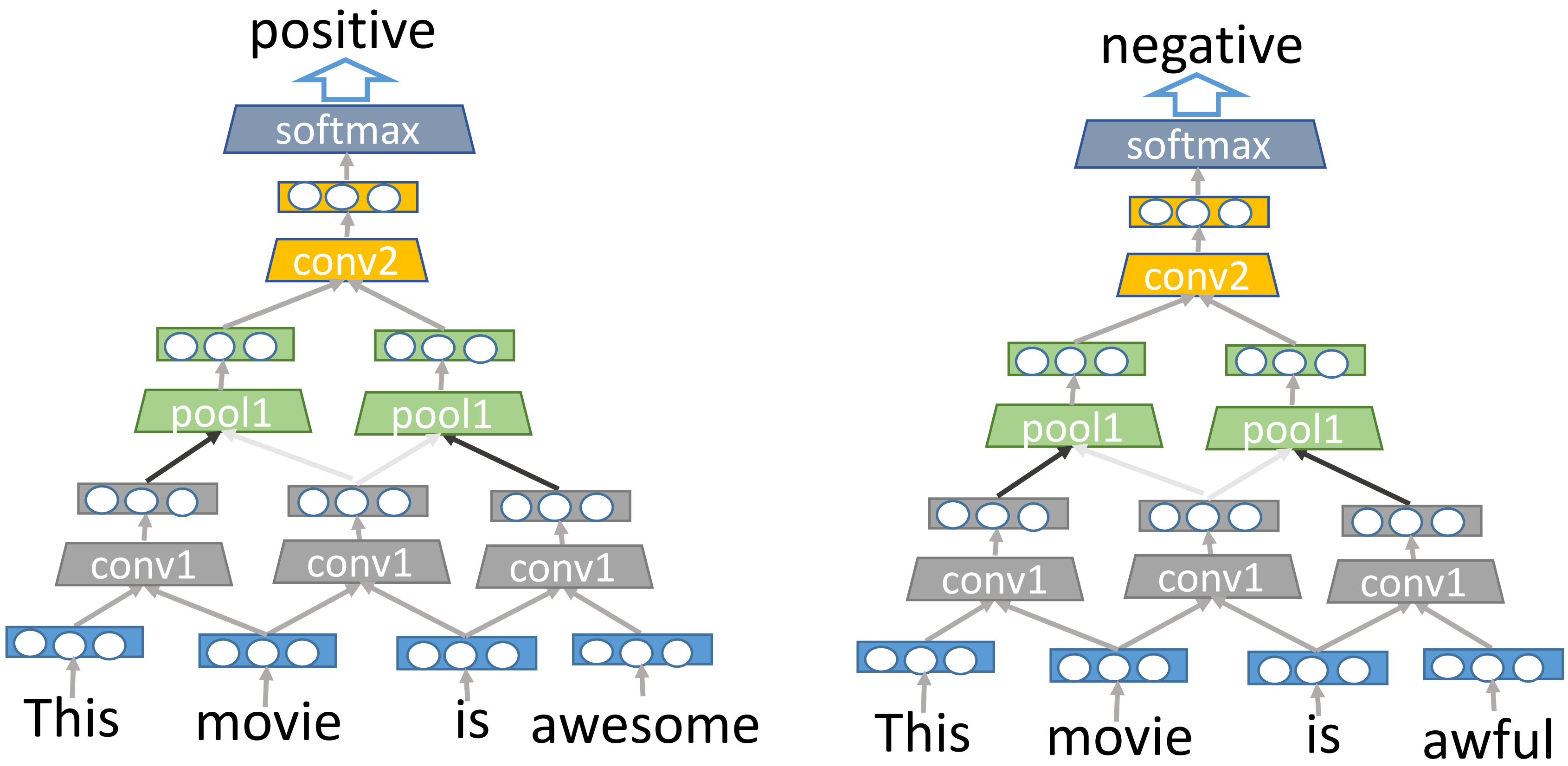
CNN with Max-Pooling Layers

- Similar to syntax tree
- But human-labeled syntax tree is not needed



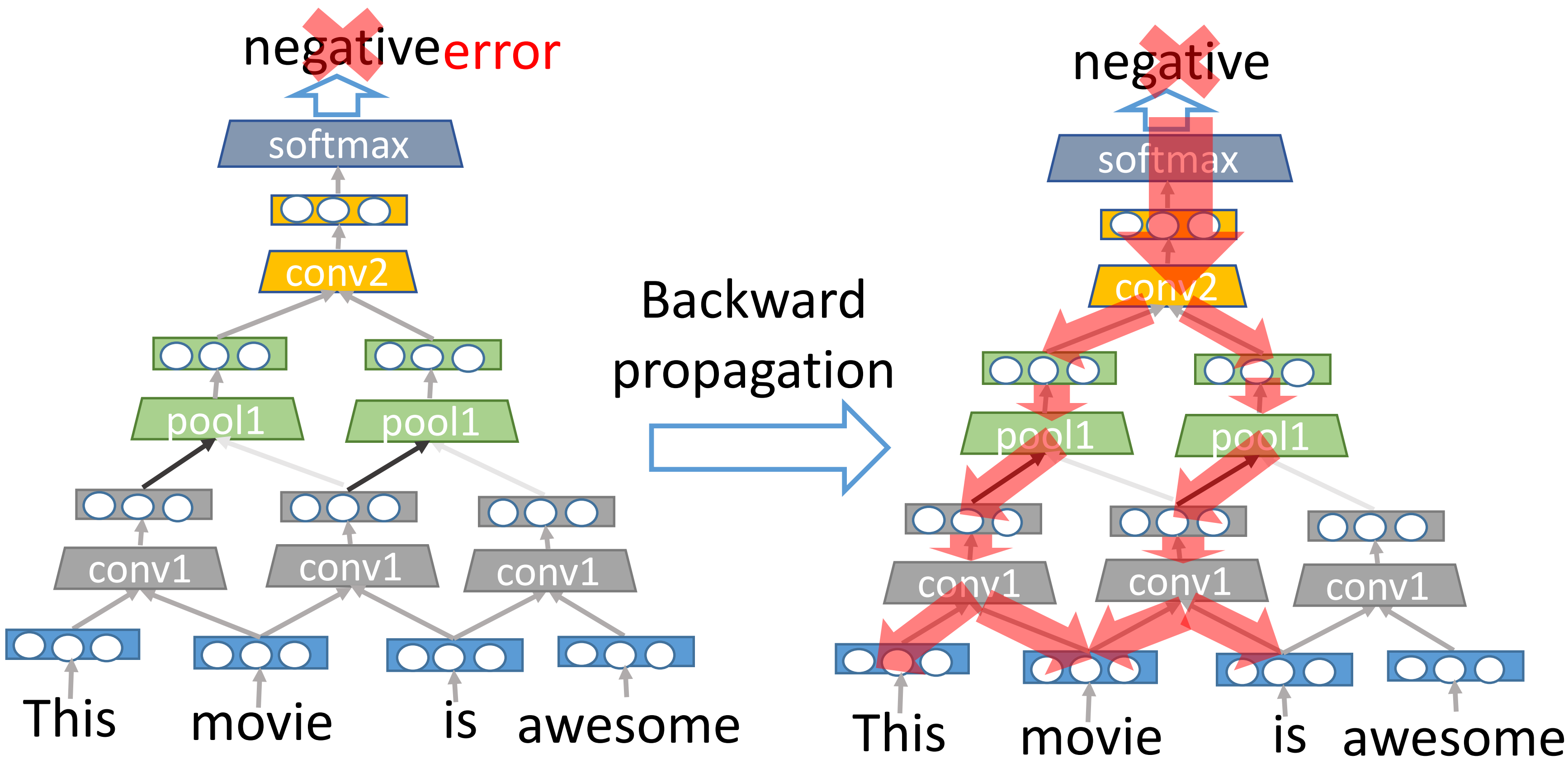
Sentiment Analysis by CNN

- Use softmax layer to classify the sentiments



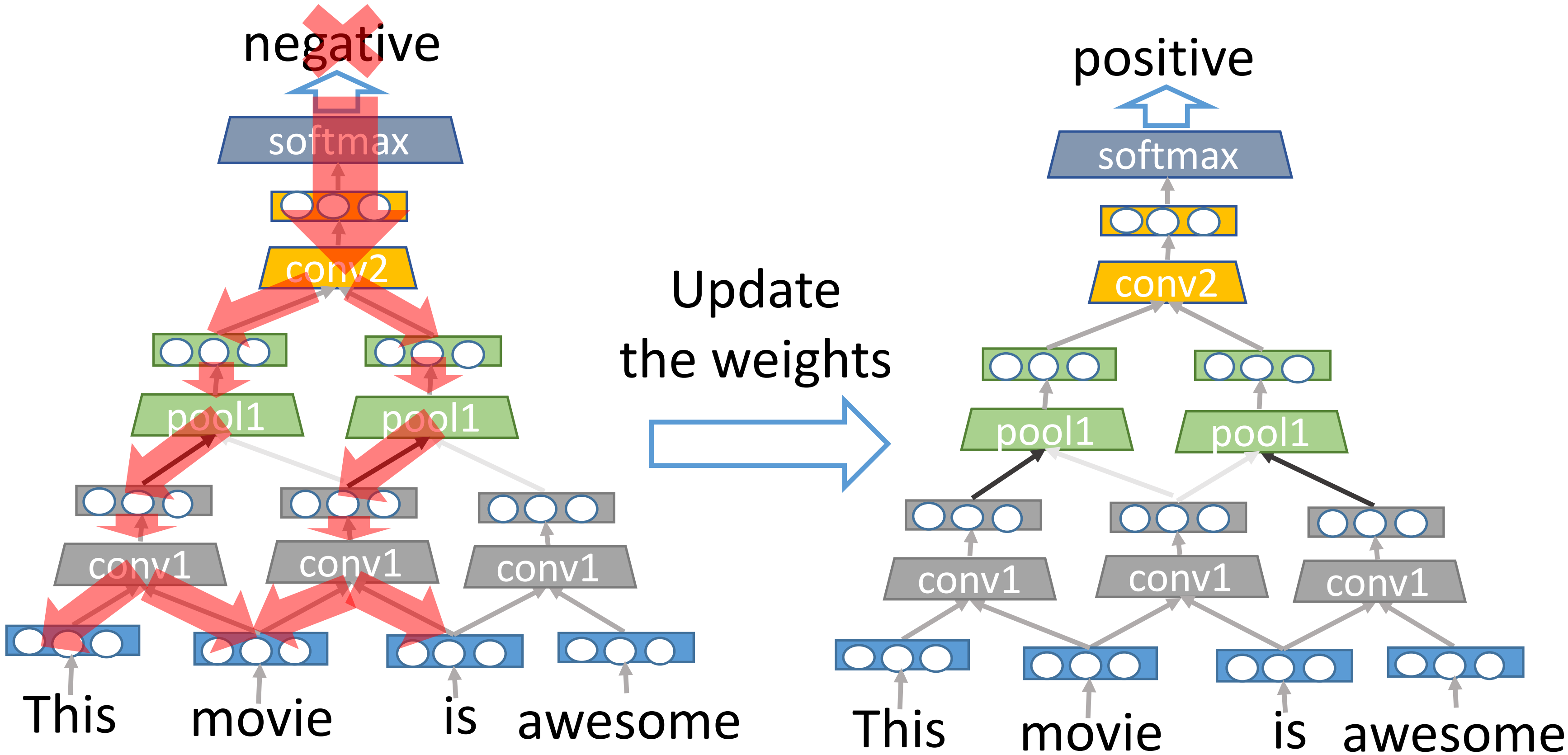
Sentiment Analysis by CNN

- Build the “correct syntax tree” by training



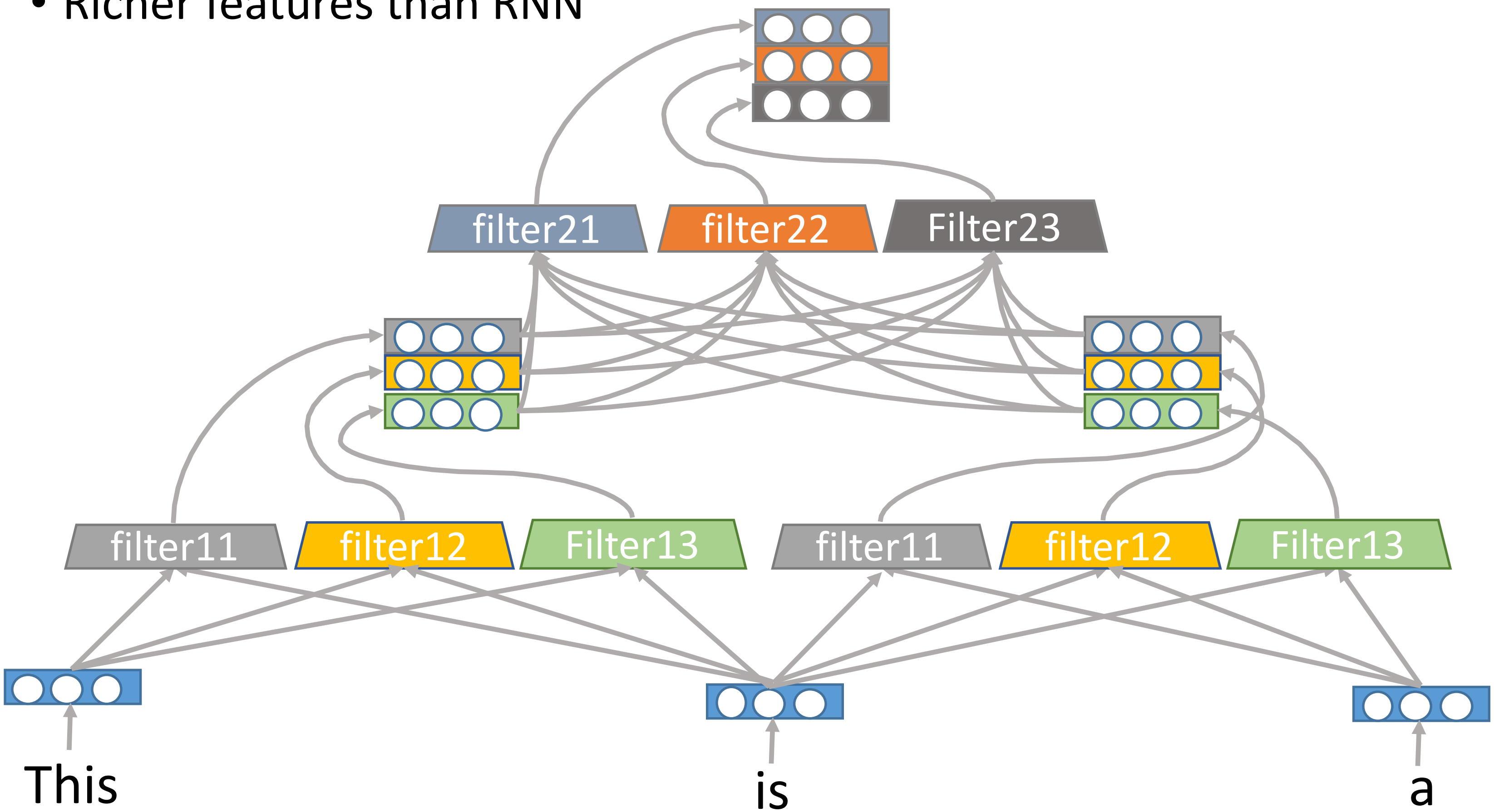
Sentiment Analysis by CNN

- Build the “correct syntax tree” by training



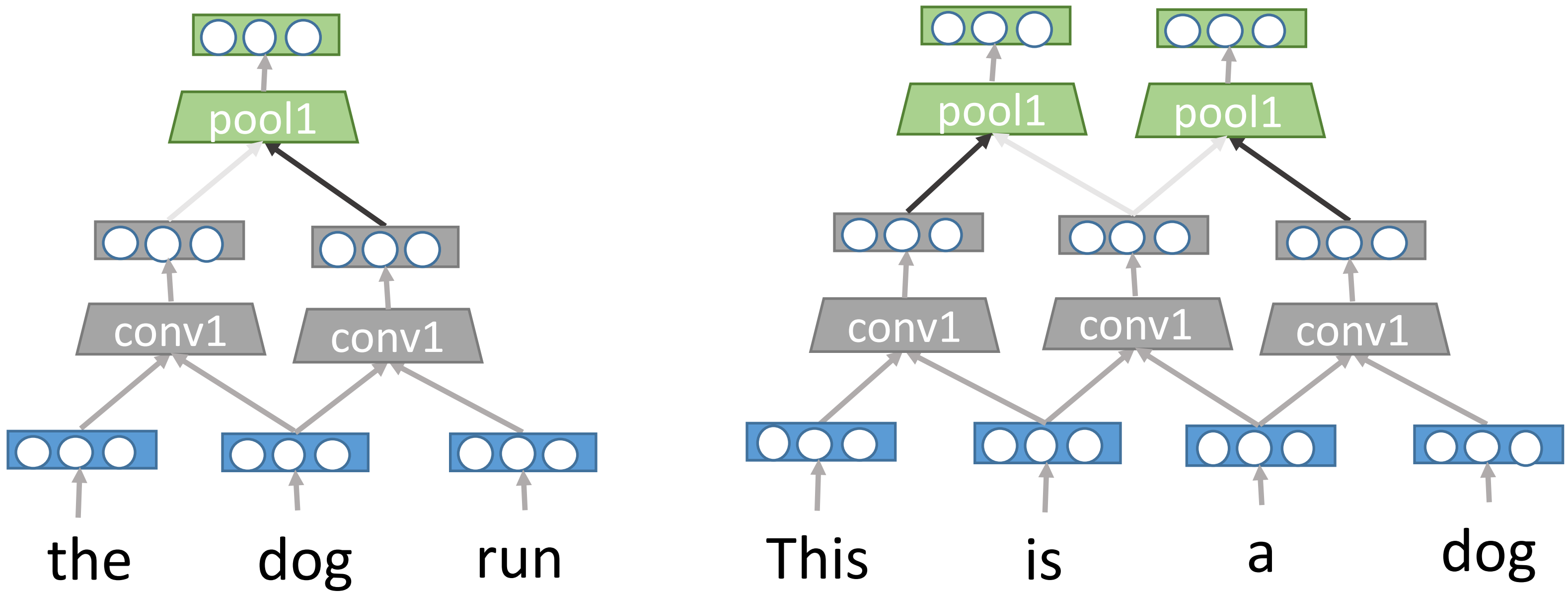
Multiple Filters

- Richer features than RNN



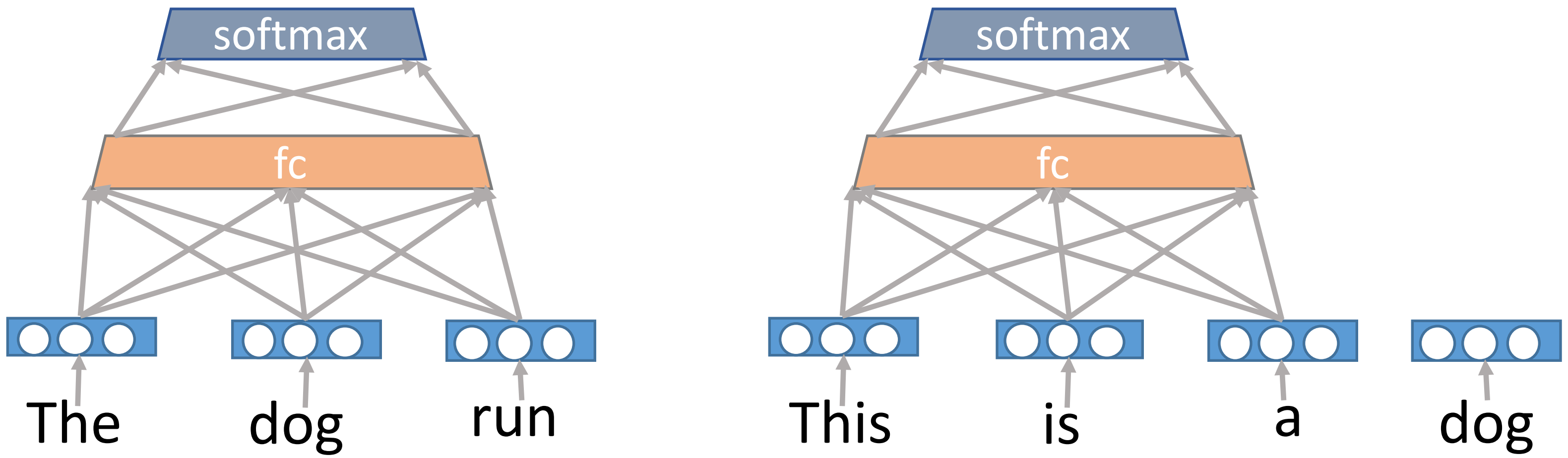
Various Input Size

- Convolutional layers and pooling layers
  - can handle input with various size



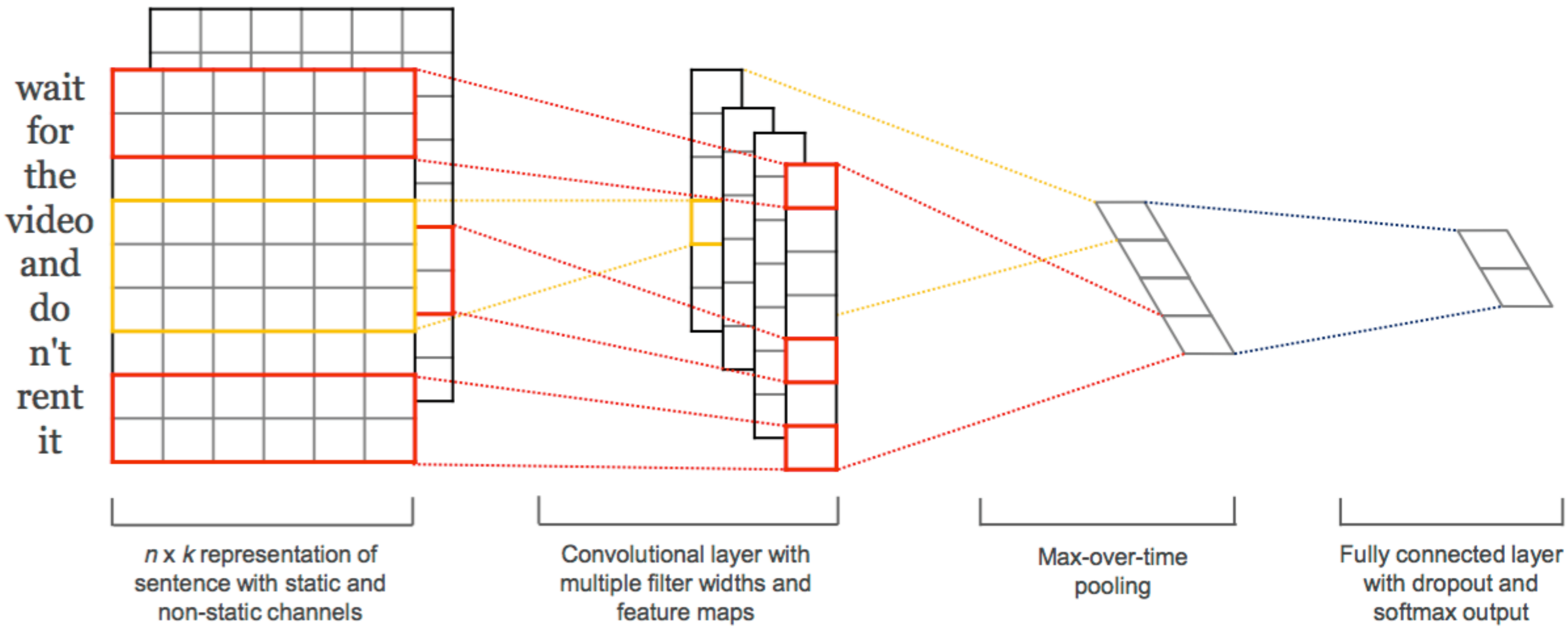
Various Input Size

- Fully-connected layer and softmax layer
  - need fixed-size input



Convolutional Neural Networks for Sentence Classification

- Paper: http://www.aclweb.org/anthology/D14-1181
- Sourcee code: https://github.com/yoonkim/CNN_sentence



Static & Non-Static Channel

- Pretrained by word2vec
- Static: fix the values during training
- Non-Static: update the values during training