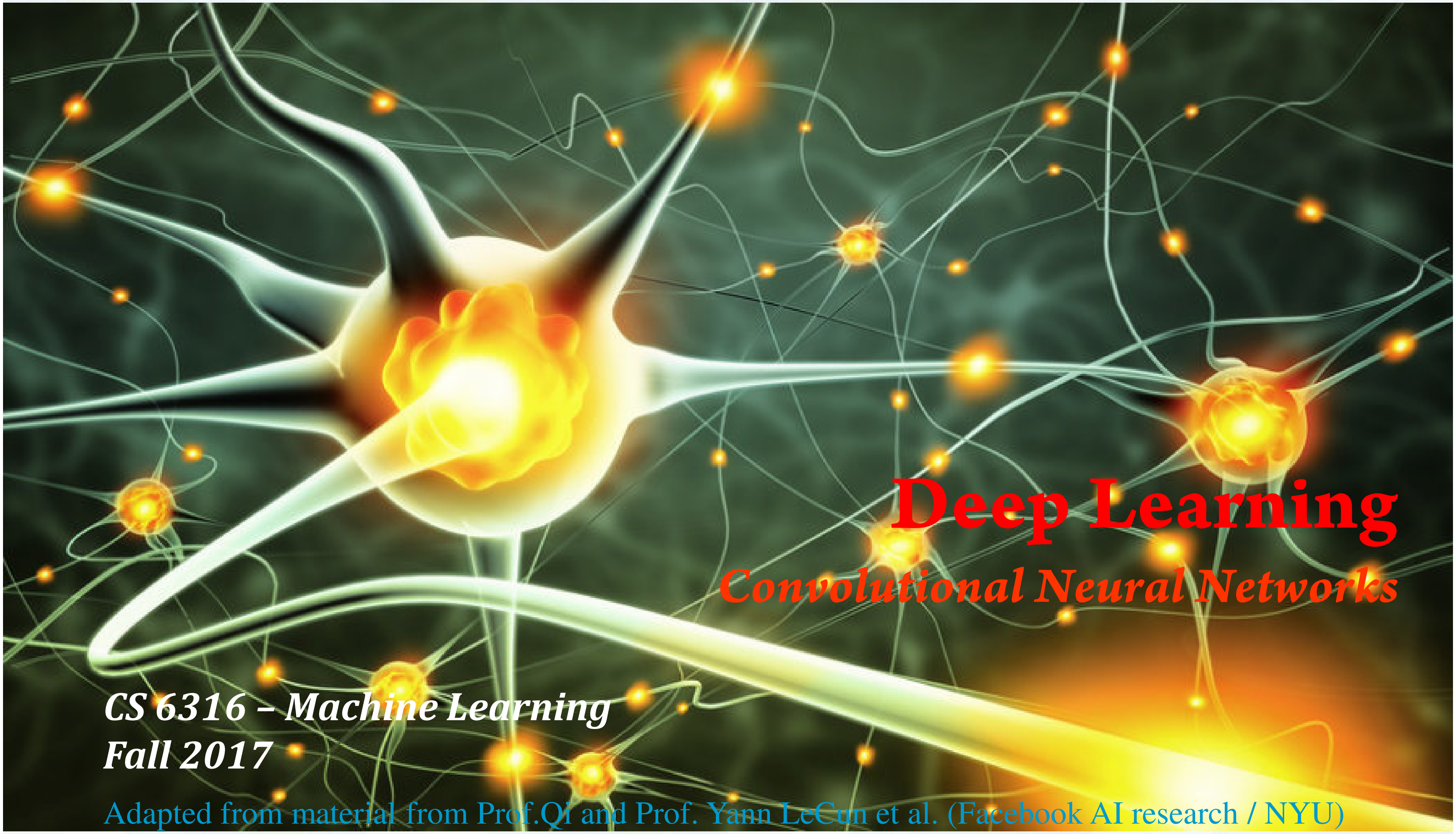


Deep Learning

Convolutional Neural Networks

CS 6316 – Machine Learning
Fall 2017

Adapted from material from Prof.Qi and Prof. Yann LeCun et al. (Facebook AI research / NYU)

Deep Learning

(Quote from Yann LeCun et al. – Nature, Vol. 521, 2015)

Machine Learning technology powers many aspects of modern society: from web searches to content filtering on social networks to recommendations on e-commerce websites, and it is increasingly present in consumer products such as cameras and smartphones.

Deep Learning

(Quote from Yann LeCun et al. – Nature, Vol. 521, 2015)

Machine Learning systems are used to identify objects in images, transcribe speech into text, match news items, posts or products with users' interests, and select relevant results of search.

Increasingly, these applications make use of a class of techniques called ***deep learning.***

Convolutional Neural Networks and Deep Learning

- Convolutional neural networks (CNN) are a type of artificial neural network (ANN) that includes both fully-connected layers and locally-connected layers known as convolutional layers

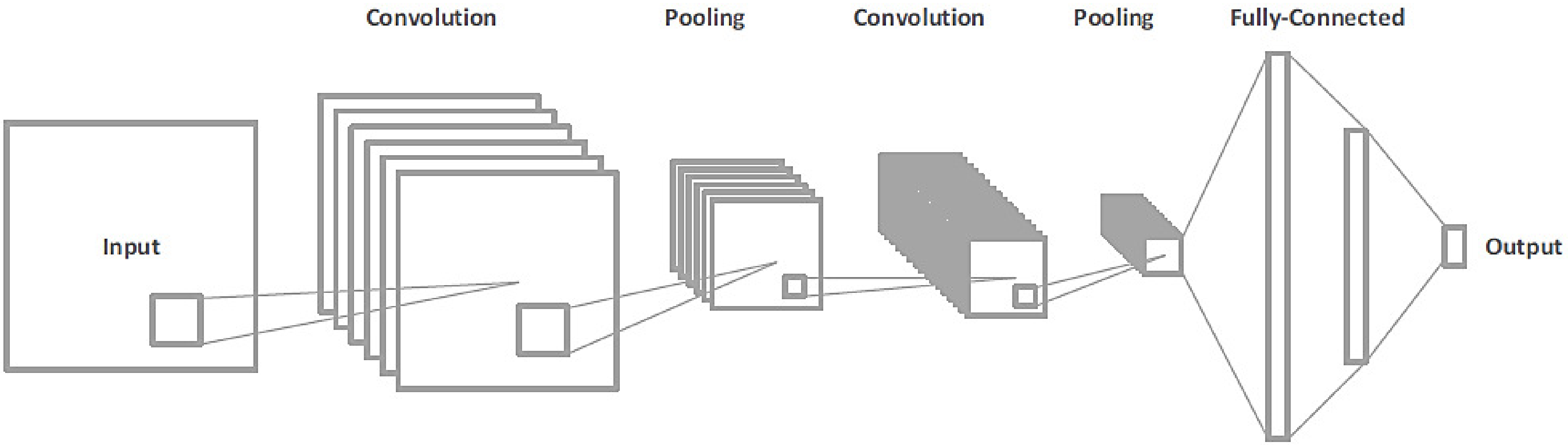


Figure 1: Convolutional neural network composed of convolution, pooling, and fully-connected layers

Convolutional Neural Networks and Deep Learning

- In large (“*deep*”) convolutional networks, it is common to see other types of layers such as pooling layers, activation layers, and batch normalization layers

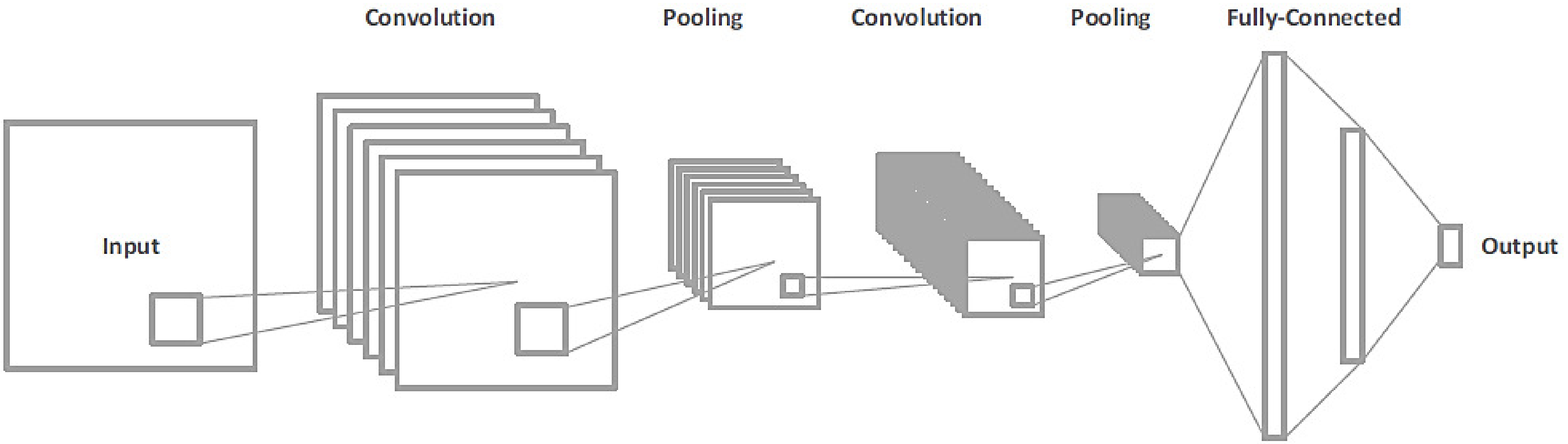


Figure 1: Convolutional neural network composed of convolution, pooling, and fully-connected layers

Convolutional Neural Networks and Deep Learning

- Here we see the architecture of a simple convolutional neural network consisting of two convolutional layers, two pooling layers, and three fully connected layers:

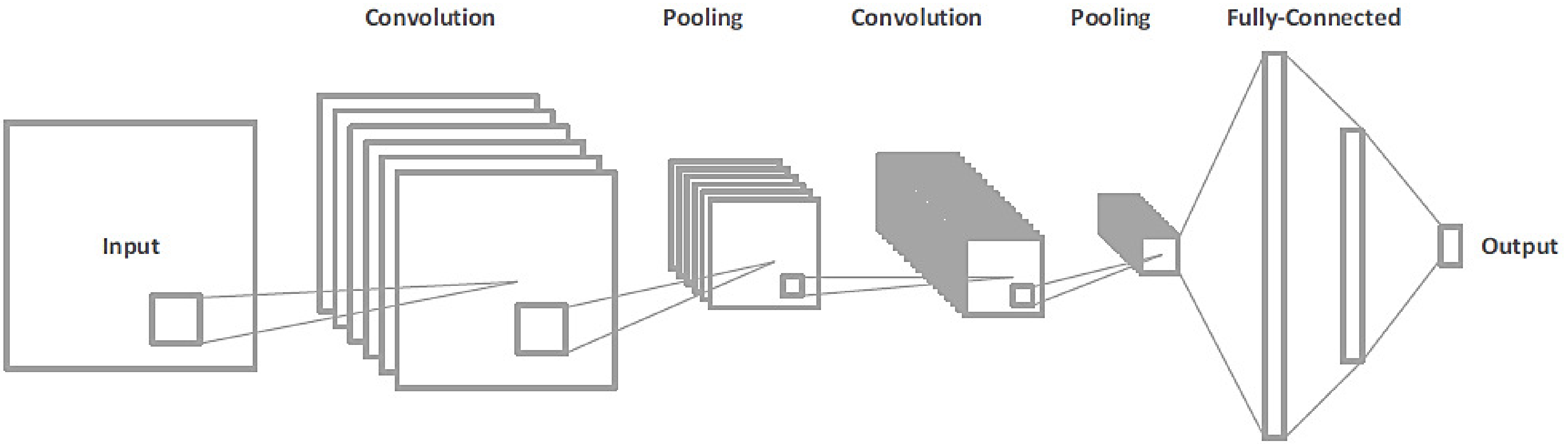


Figure 1: Convolutional neural network composed of convolution, pooling, and fully-connected layers

Data: Multiple Arrays

- Convolutional Neural Networks are designed to process data that come in the form of multiple arrays, for example a color image composed of three 2D arrays containing pixel intensities in the three color channels

Data: Multiple Arrays

- Many data modalities are in the form of **multiple arrays**:
  - 1D for signals and sequences, including language
  - 2D for images or audio spectrograms
  - 3D for video of volumetric images
- There are four key ideas behind CNNs that take advantage of the properties of natural signals:
  - Local connections
  - Shared weights
  - Pooling
  - **Use of many layers**

Convolutional Neural Networks

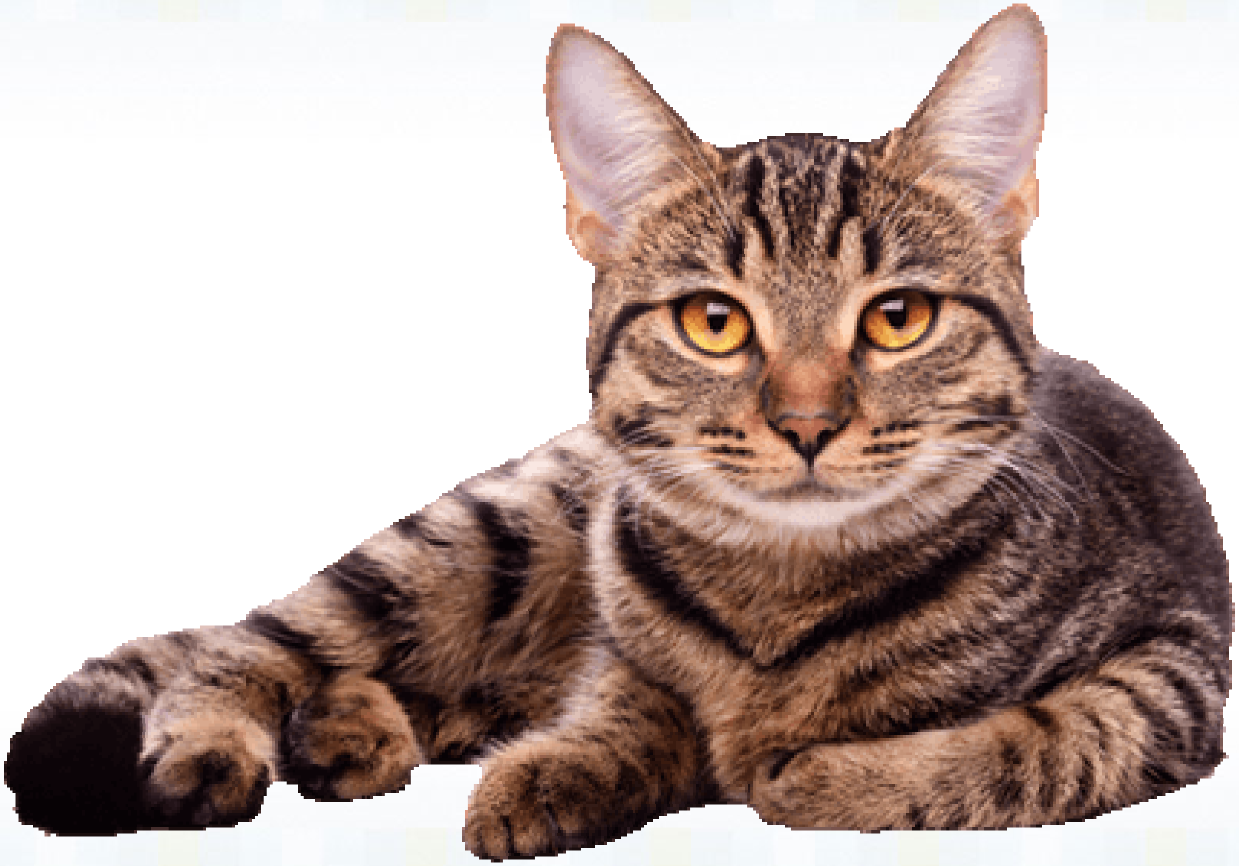
- Training deep convolutional neural networks from scratch is difficult since training can require
  - extensive computational resources and
  - large amounts of training data
- If such resources are not available, a pre-trained CNN can be used to perform a new classification task by either
  - [A] fine-tuning the pre-trained network's weights using a new target dataset, or by
  - [B] using the pre-trained network’s activations as automatic feature extractors.

Convolutional Neural Networks

- Pre-trained networks for many popular CNN architectures such as AlexNet (A.Krizhevsky et al.), VGGNet (K.Simonyan and A Zisserman), and GoogLeNet (C.Szegedy et al.) can be downloaded and adapted for different problems

- Many applications, but popular application is visual (image) recognition

What we see …

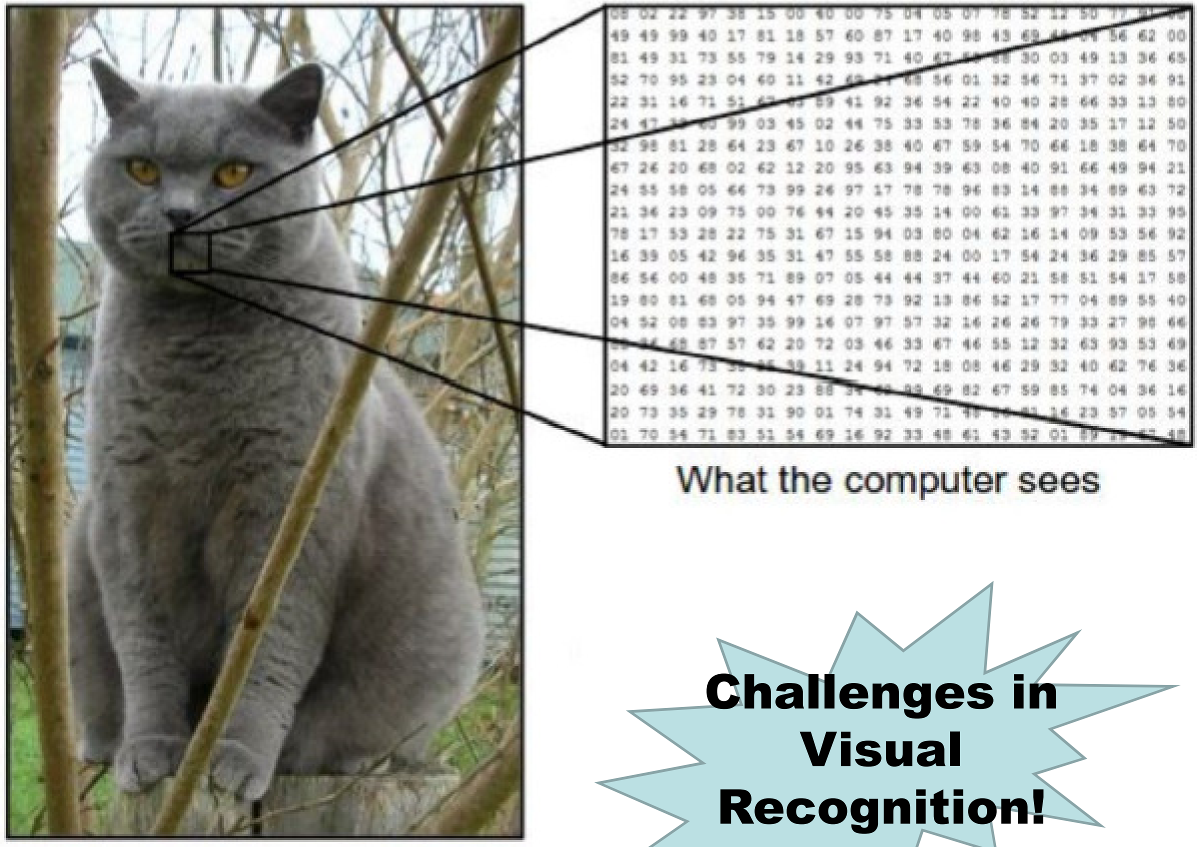


What we see…

What we see…

Convolutional Neural Networks (aka CNNs and ConvNets)

- The problem:
  semantic gap
- Images are represented as 3D arrays of numbers, with integers between [0, 255]
- E.g. 300 x 300 x 3 (3 for e color channels RGB)



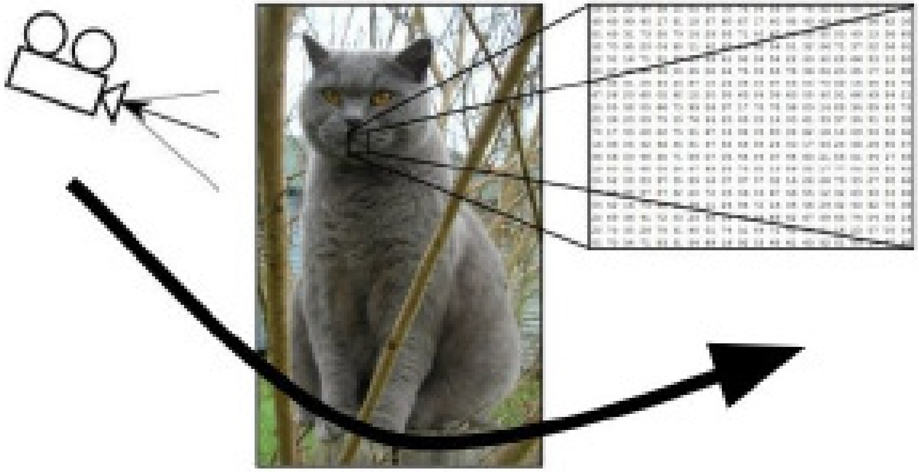
08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	91	08
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	24	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	63	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	25	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	86	81	16	23	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	48

What the computer sees

Challenges in Visual Recognition!

Challenges in Visual Recognition

- Camera pose
- Illumination



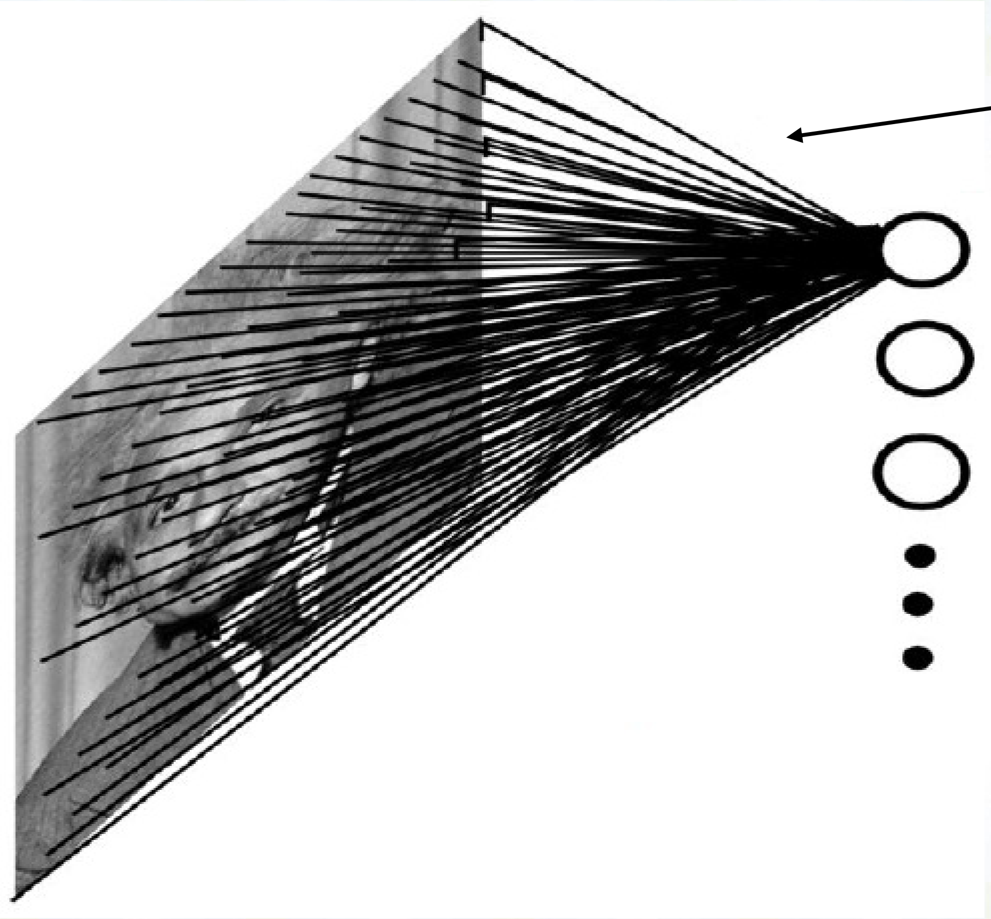
Challenges in Visual Recognition

- Deformation
- Occlusion

Challenges in Visual Recognition

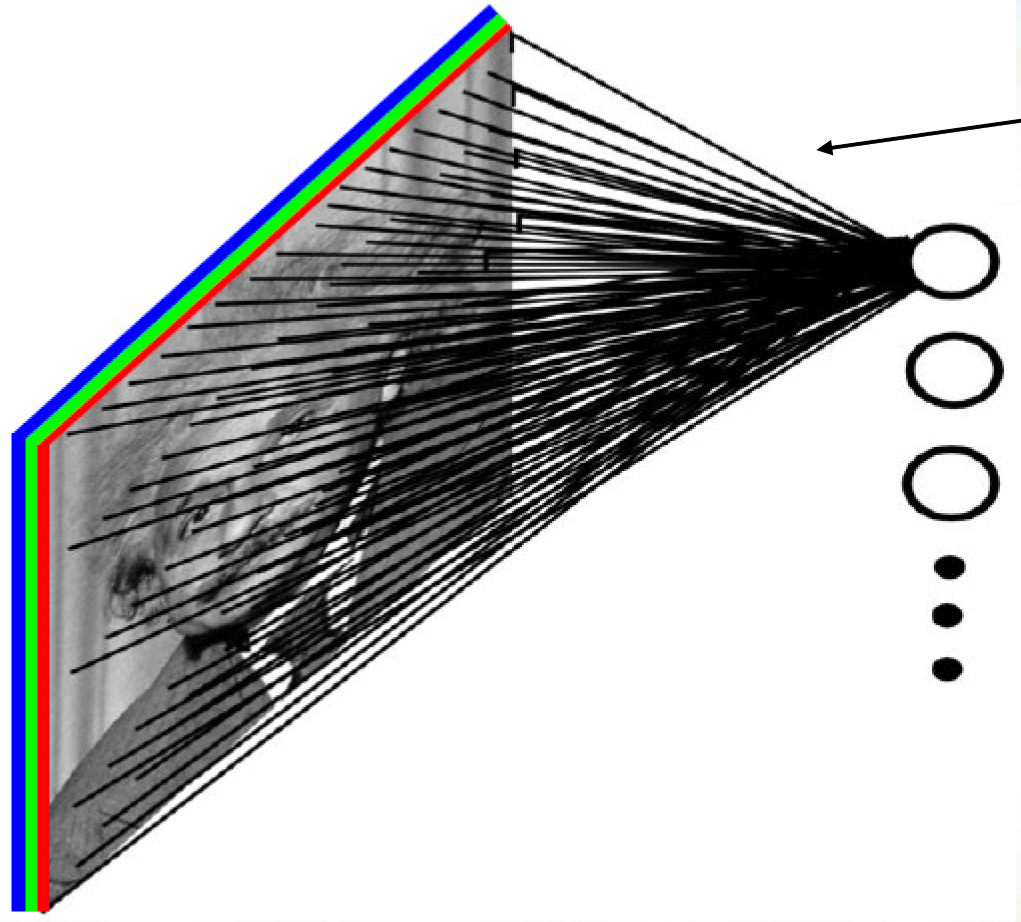
- Background clutter
- Intra-class variation

Problems with “Fully Connected Networks” on Images



Each neuron corresponds to a ***specific*** pixel

Problems with “Fully Connected Networks” on Images

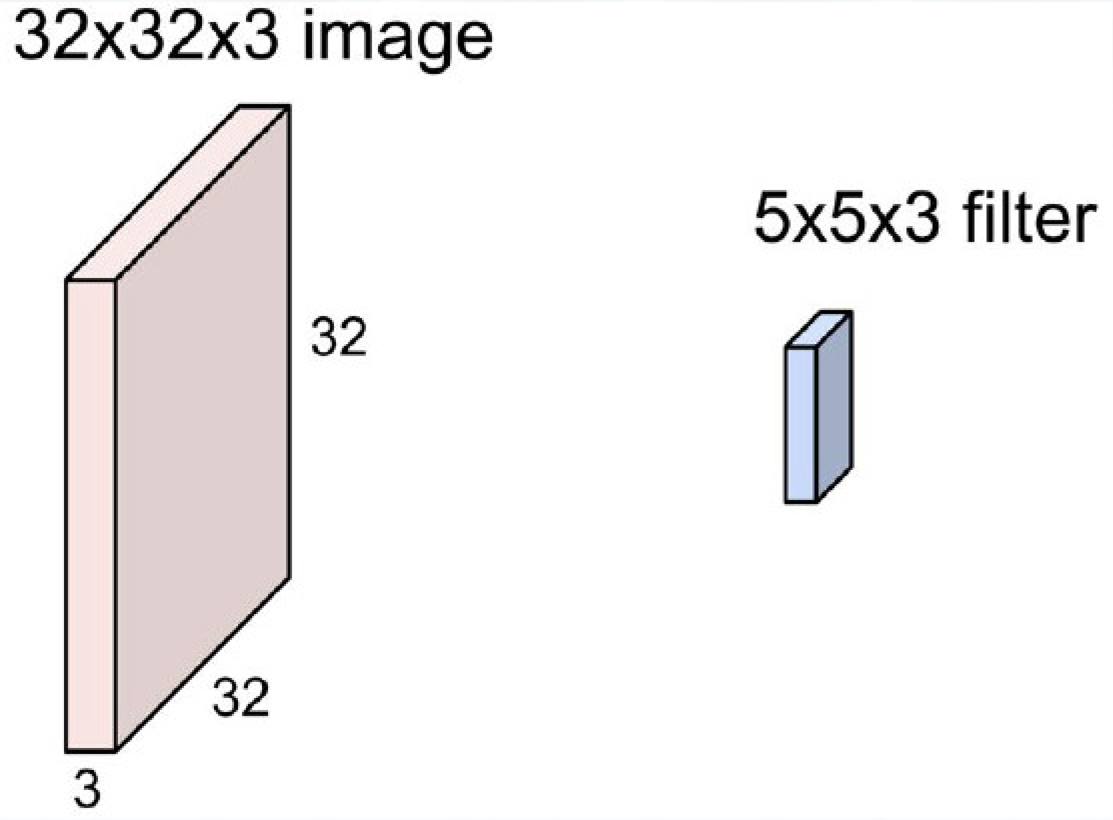


Each neuron corresponds to a ***specific*** pixel

How do we deal with multiple dimensions in the input?

Length, height, channel (RGB)

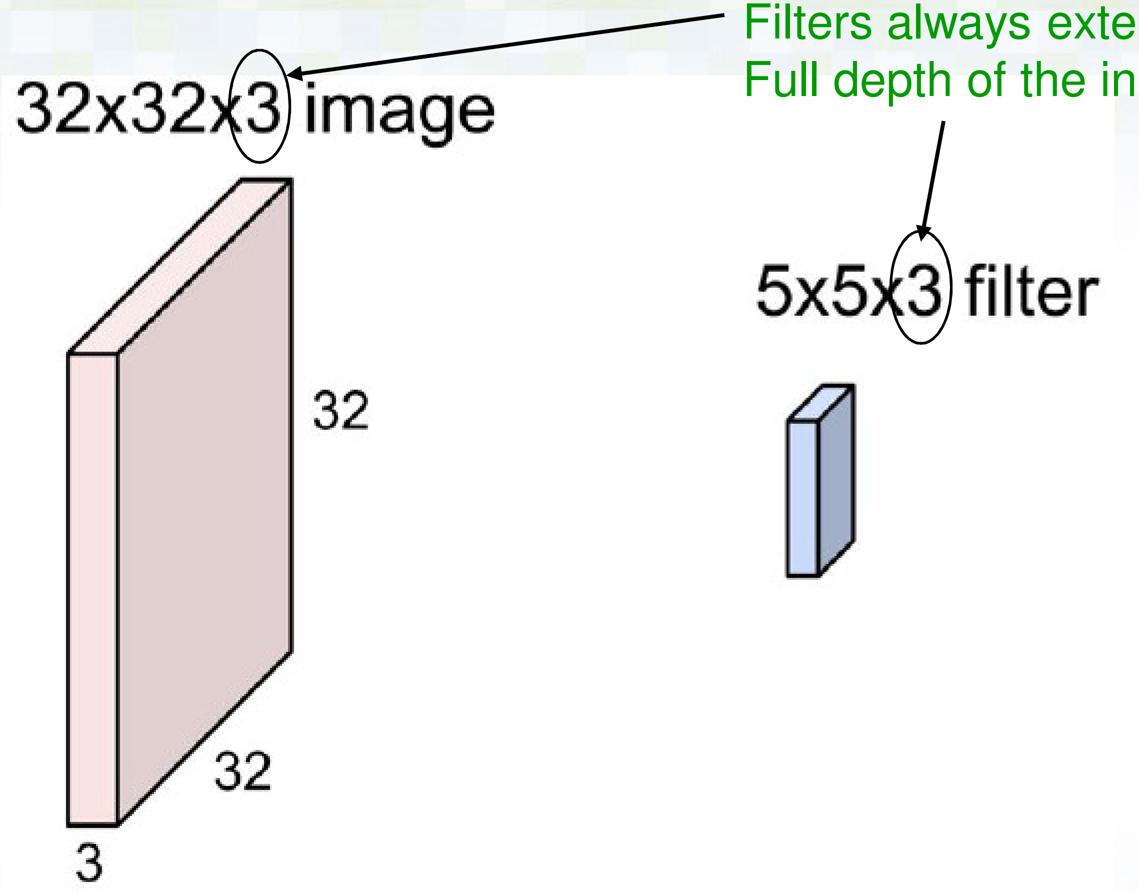
Convolutional Layer



Convolve the filter with the image

i.e. "*slide over the image spatially, computing dot products*"

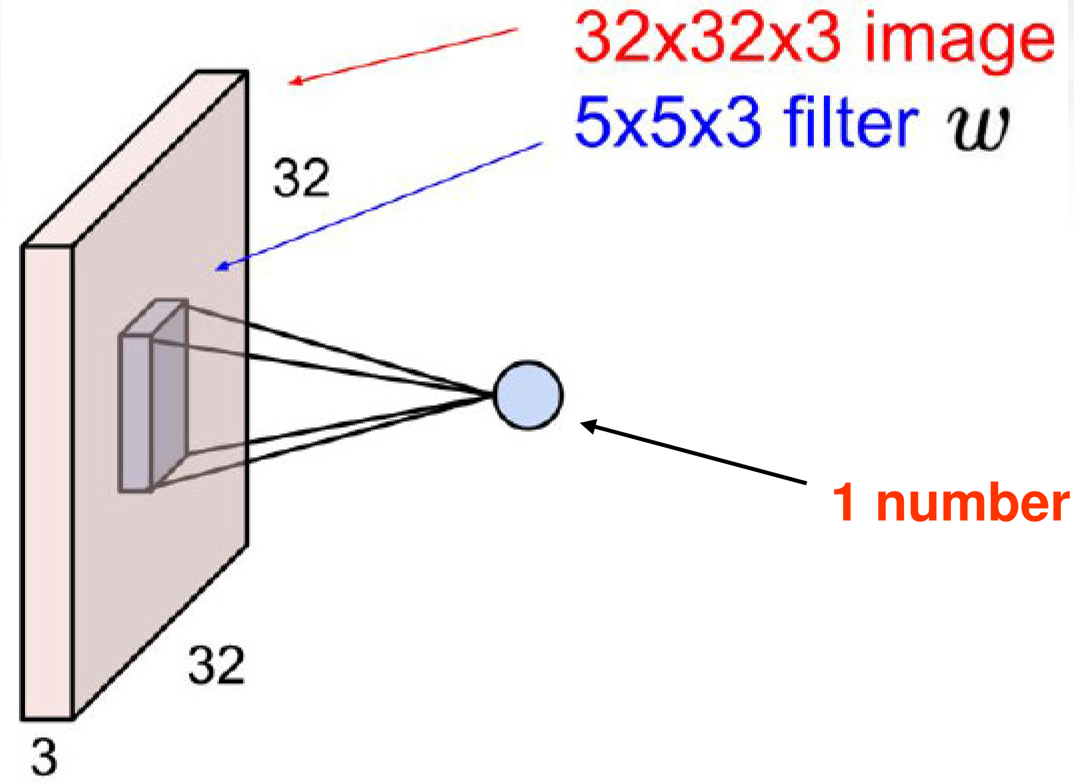
Convolutional Layer



Filters always extend the
Full depth of the input volume

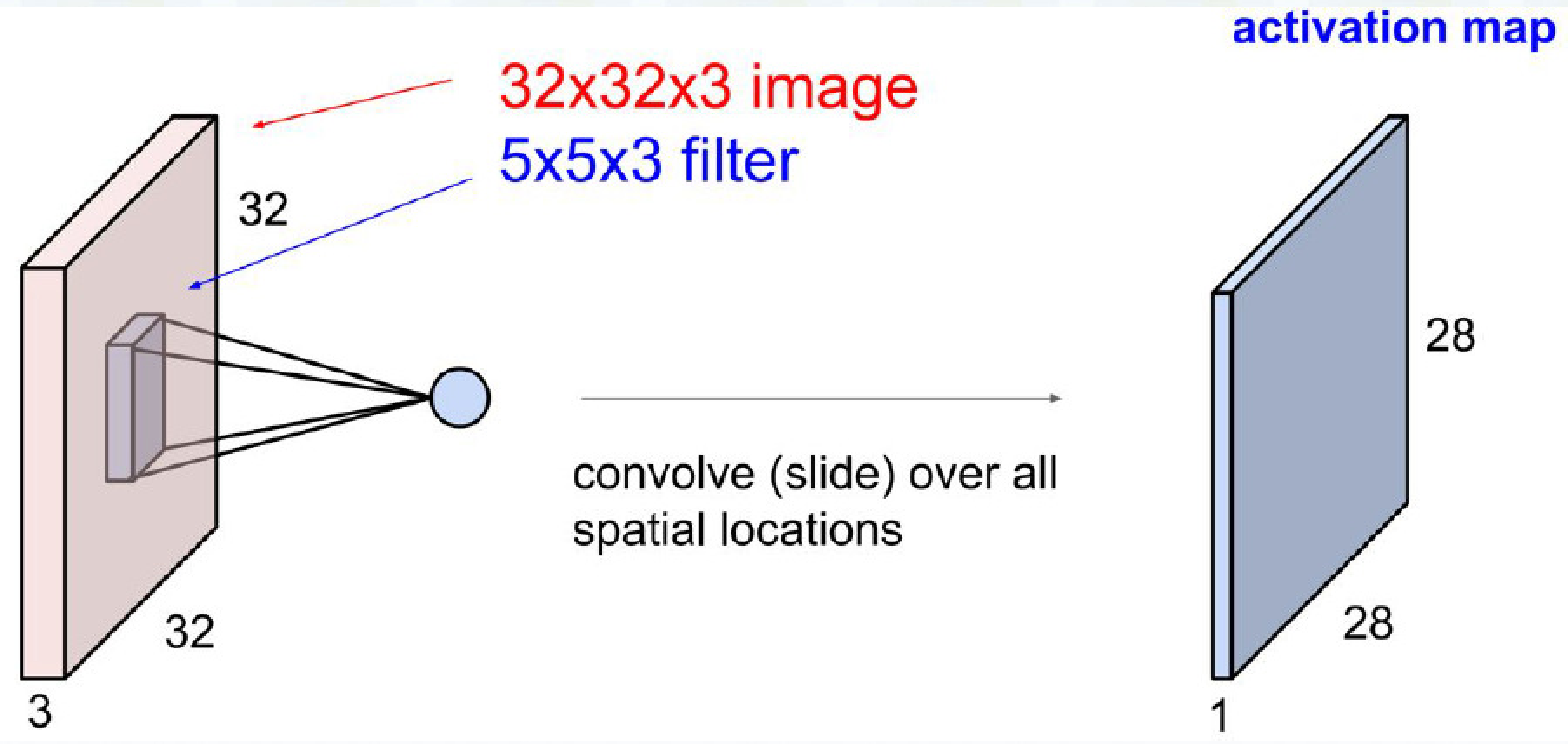
Convolve the filter with the image
i.e. “*slide over the image spatially, computing dot products*”

Convolutional Layer



- The result of taking a dot product between the filter and a small 5 x 5 x 3 chunk of the image (i.e. 5*5*3 = 75-dimensional dot product+ bias) $w^T x + b$

Convolutional Layer



activation map

32x32x3 image

5x5x3 filter

convolve (slide) over all spatial locations

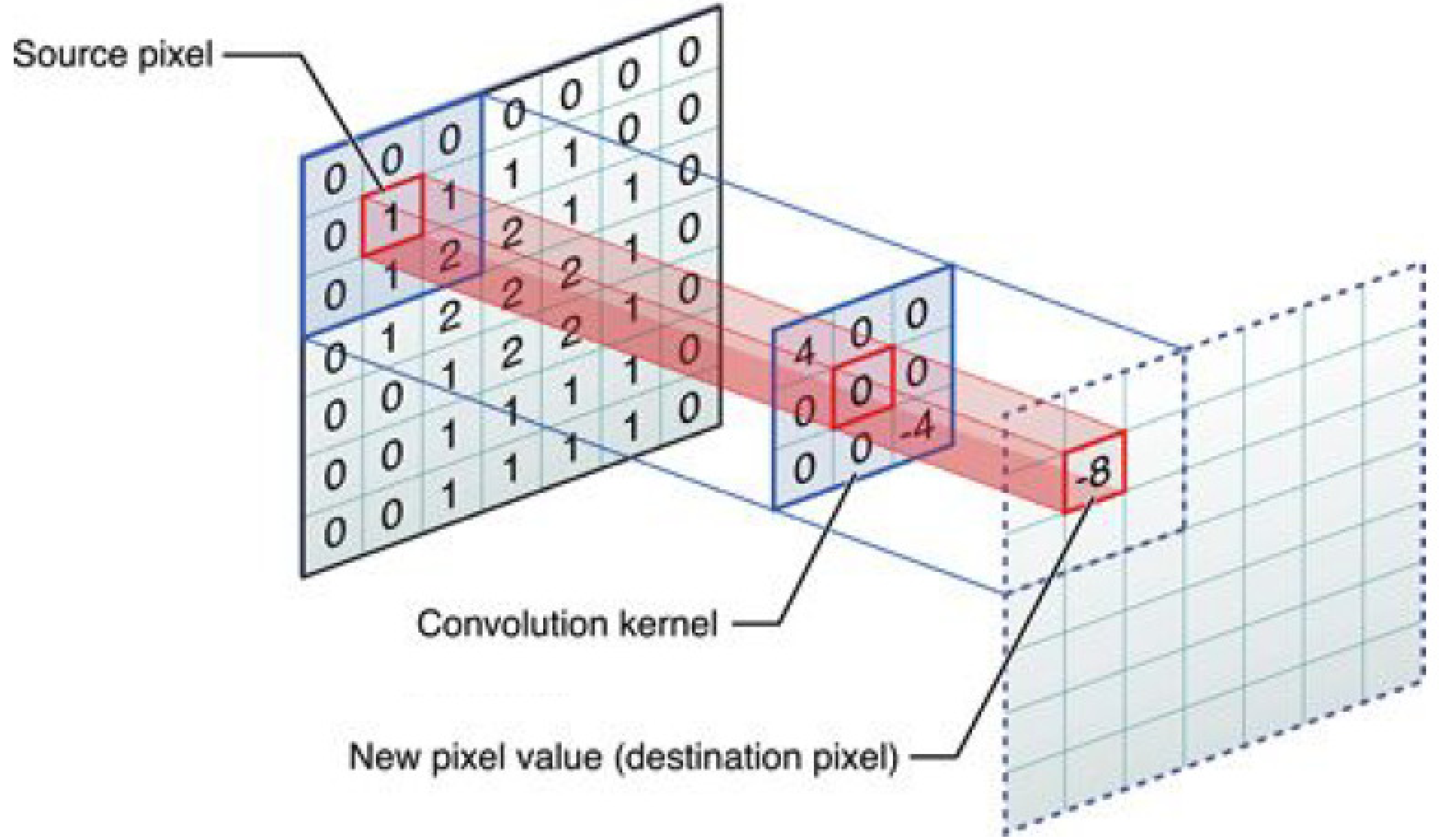
Convolution

- We call the layer convolutional because it is related to convolution of two signals:

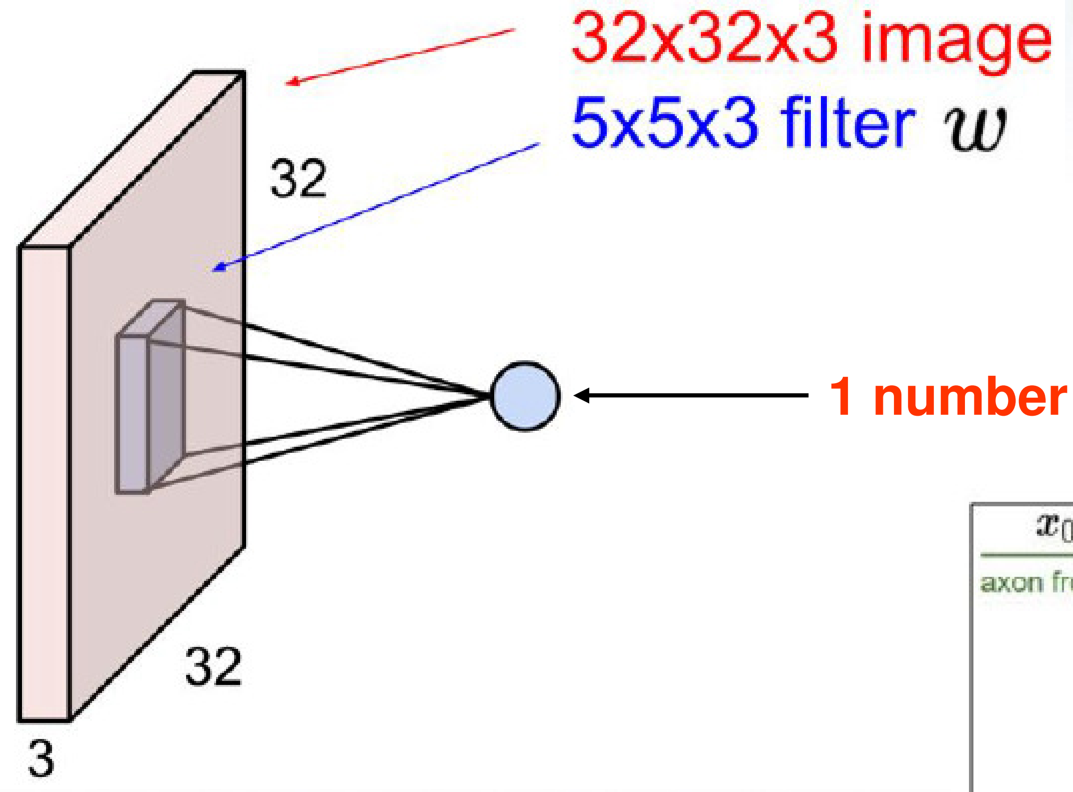
$$f[x,y] * g[x,y] \;=\; \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x-n_1, y-n_2]$$

elementwise multiplication and sum of
a filter and the signal (image)

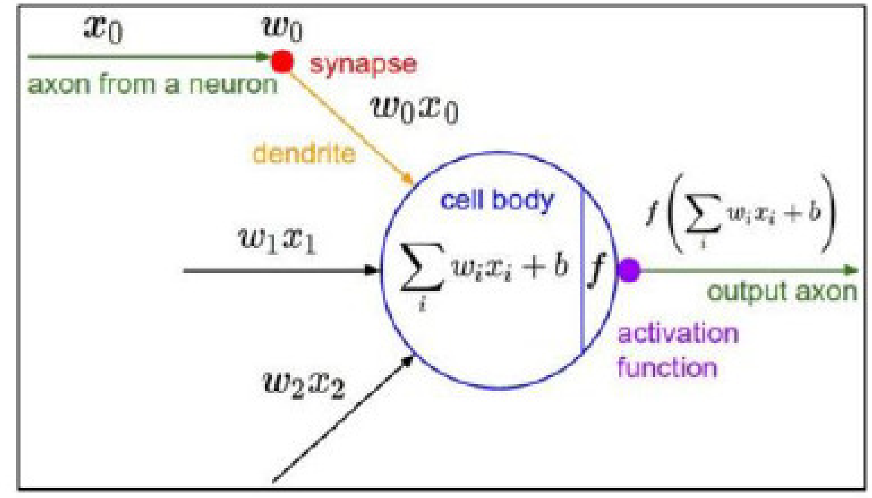
Convolution



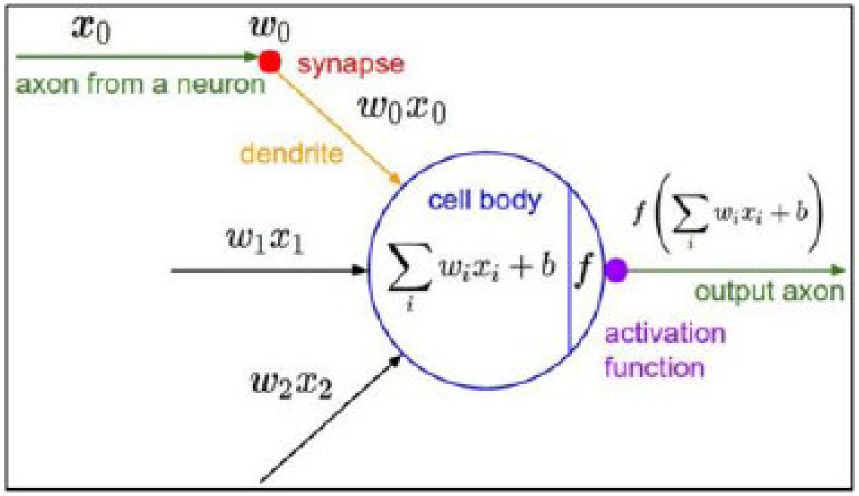
Neuron View of Convolutional Layer

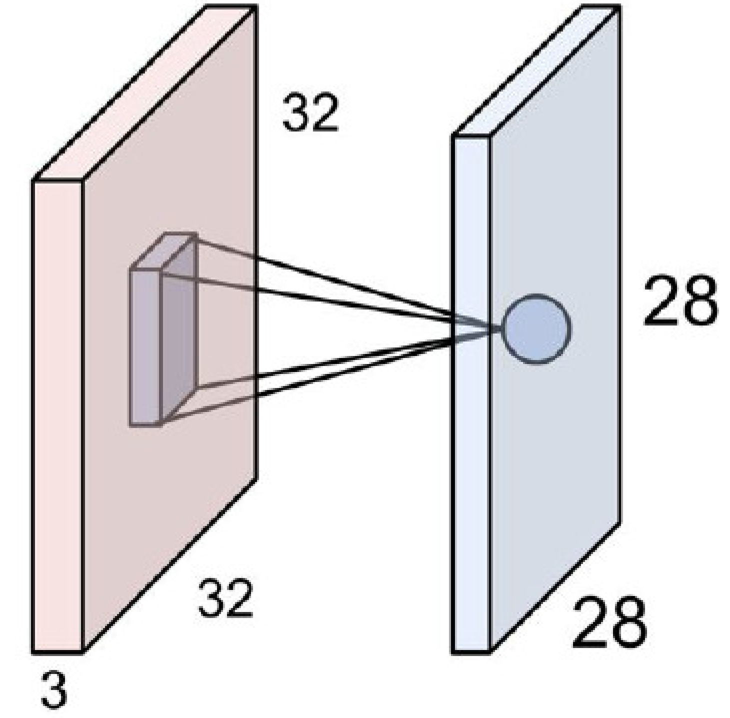


It is just a neuron with
Local connectivity…



Neuron View of Convolutional Layer

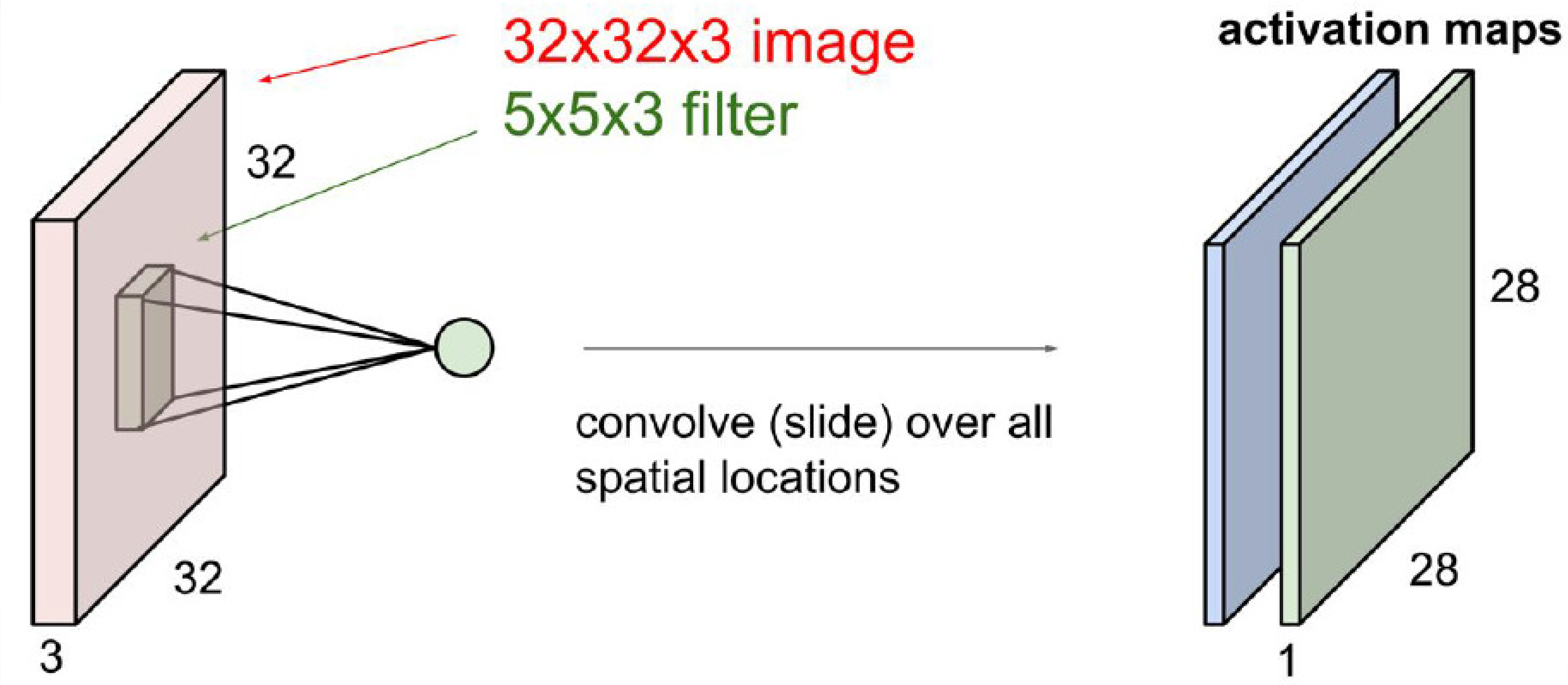




- An activation map is a 28 x 28 sheet of neuron outputs:
  1. Each is connected to a small region in the input
  2. All of them share parameters

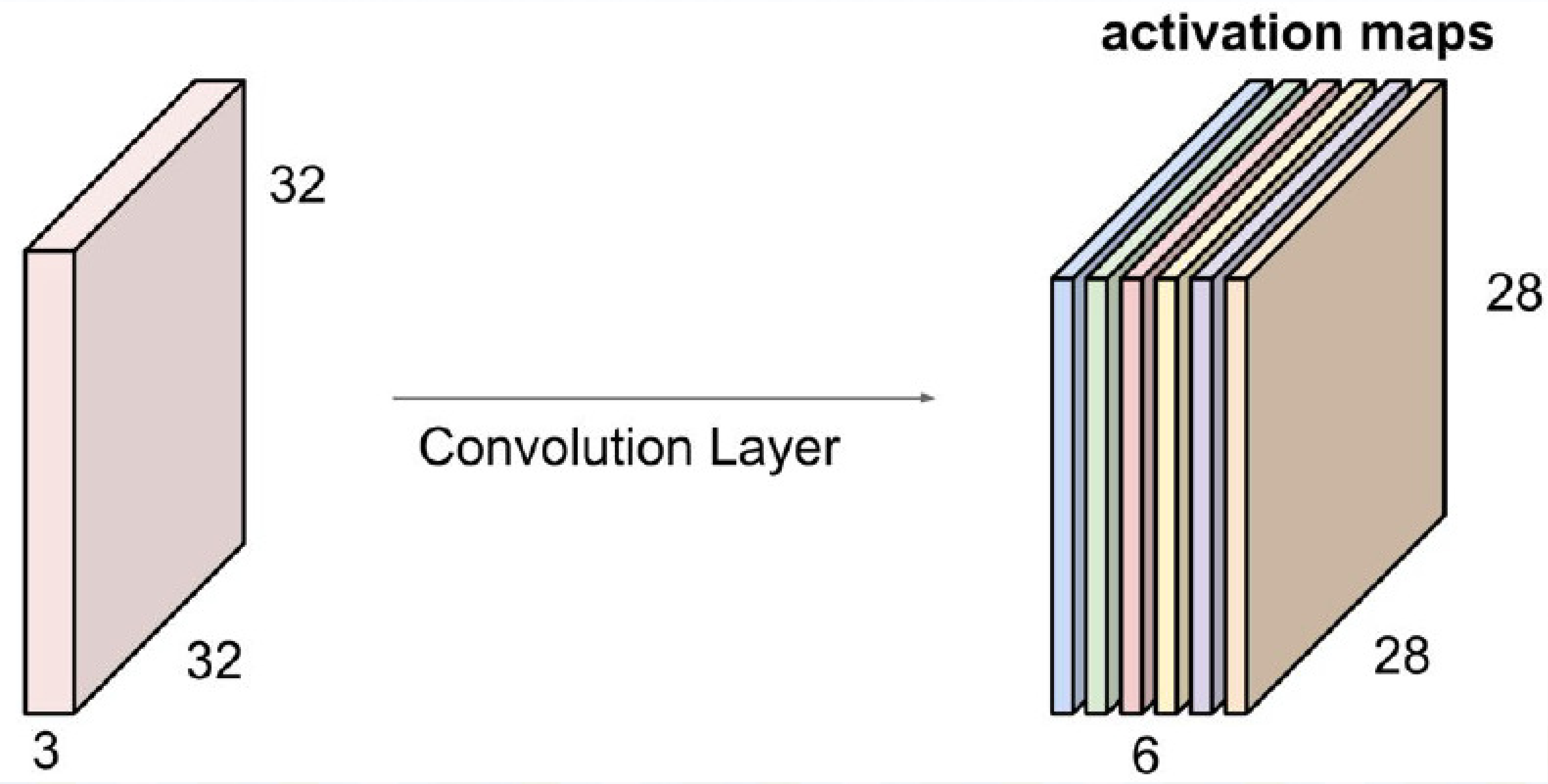
"5x5 filter" → "*5x5 receptive field for each neuron*"

Convolutional Layer (consider a second, GREEN filter)



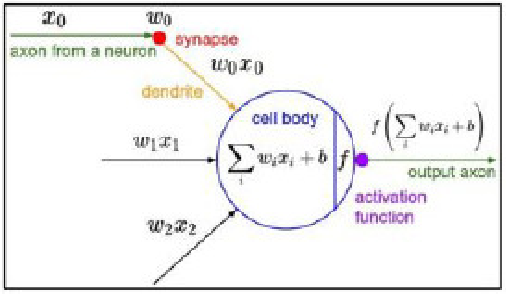
Convolutional Layer

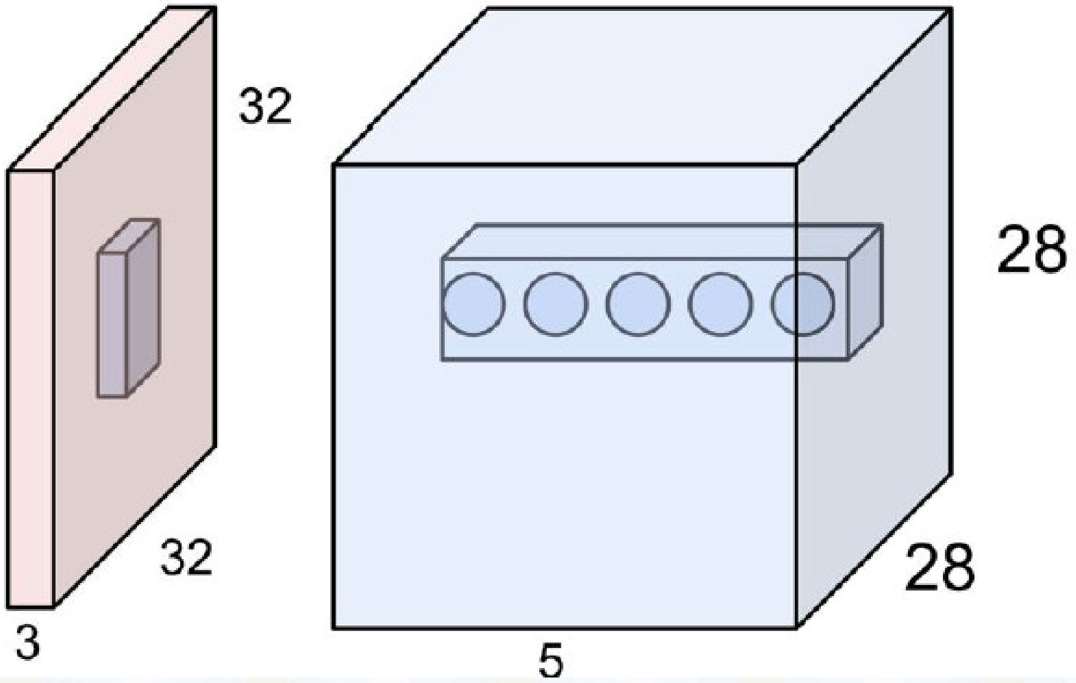
- For example, if we had **six (6) 5x5 filters**, we'll get 6 separate activation maps:



- We stack these up to get a "new image" of size 28 x 28 x 6!

Neuron View of Convolutional Layer

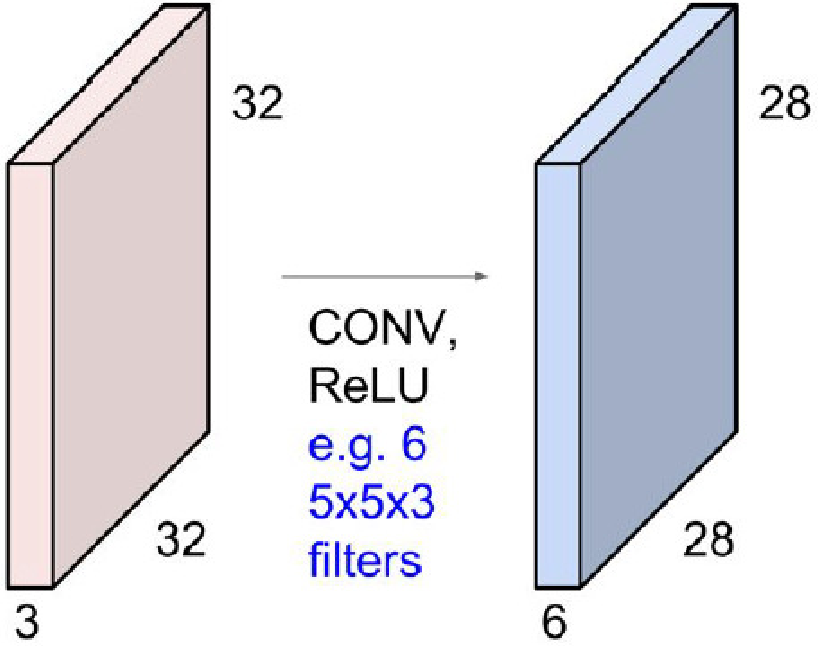




E.g. with 5 filters,
CONV layer consists of neurons arranged in a 3D grid (28x28x5)

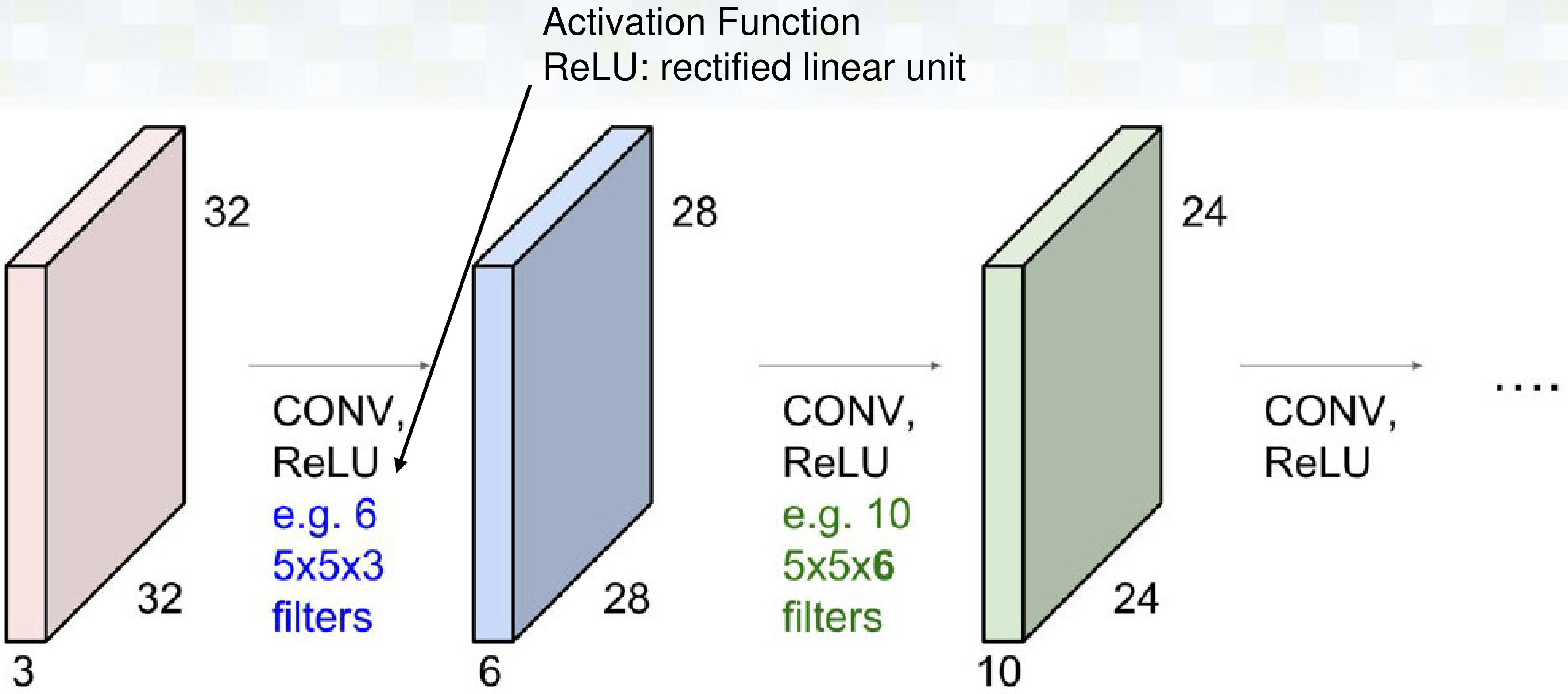
There will be 5 different neurons all looking at the same region in the input volume

Convolutional Neural Networks



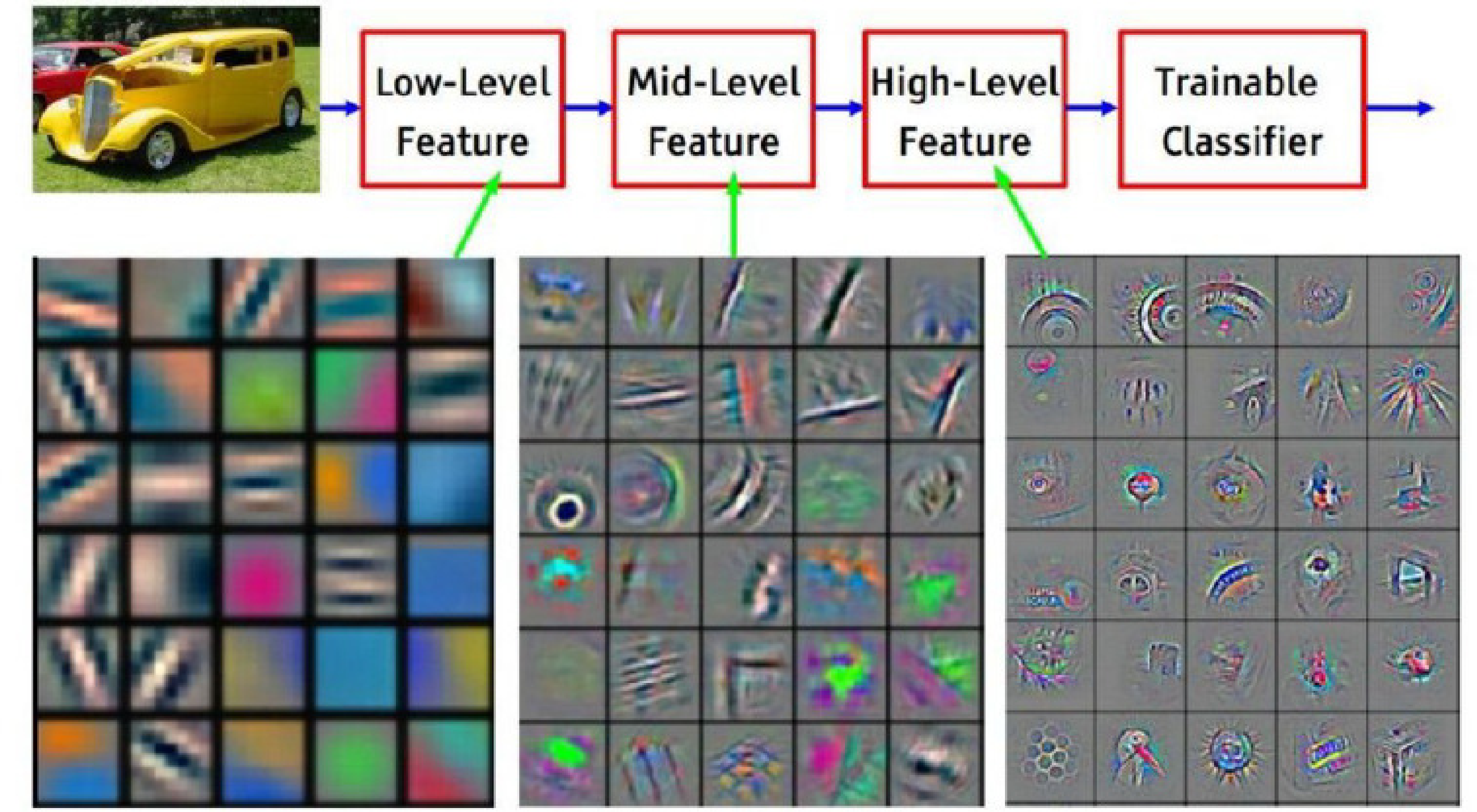
- Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions

Convolutional Neural Networks



- Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions

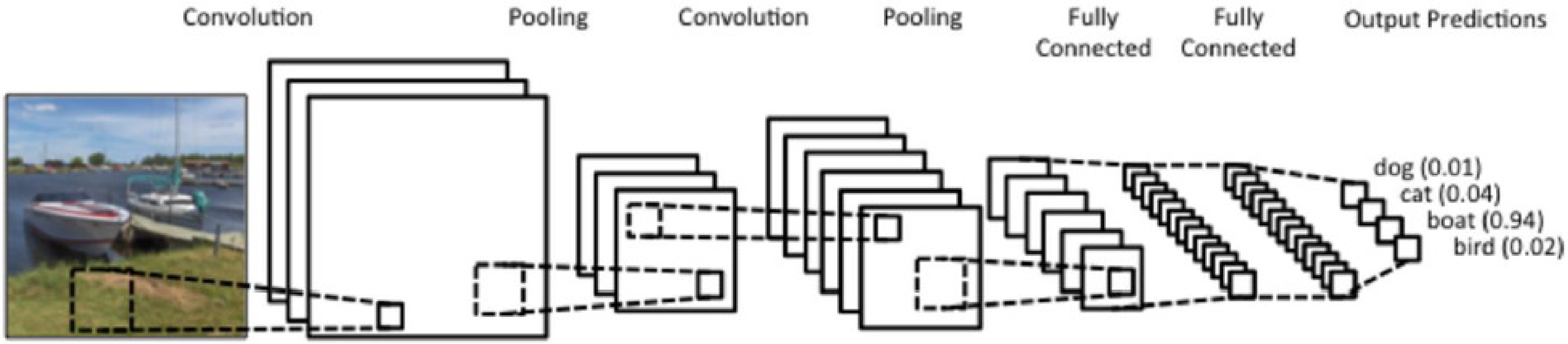
Convolutional Neural Networks



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

[From recent Yann LeCun slides]

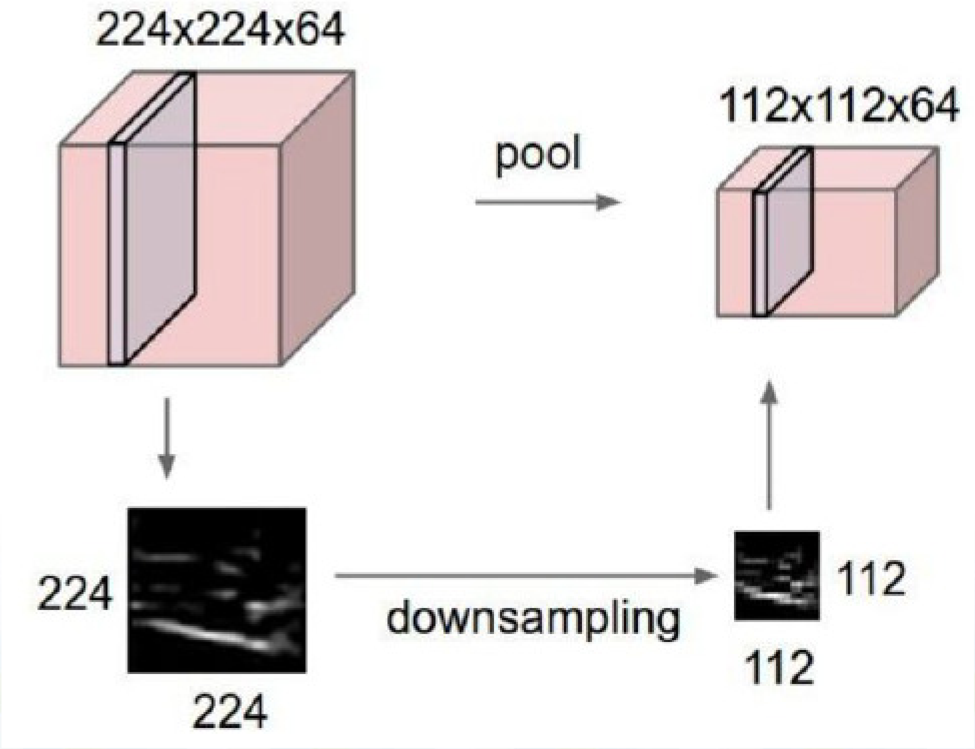
Convolutional Neural Networks



http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

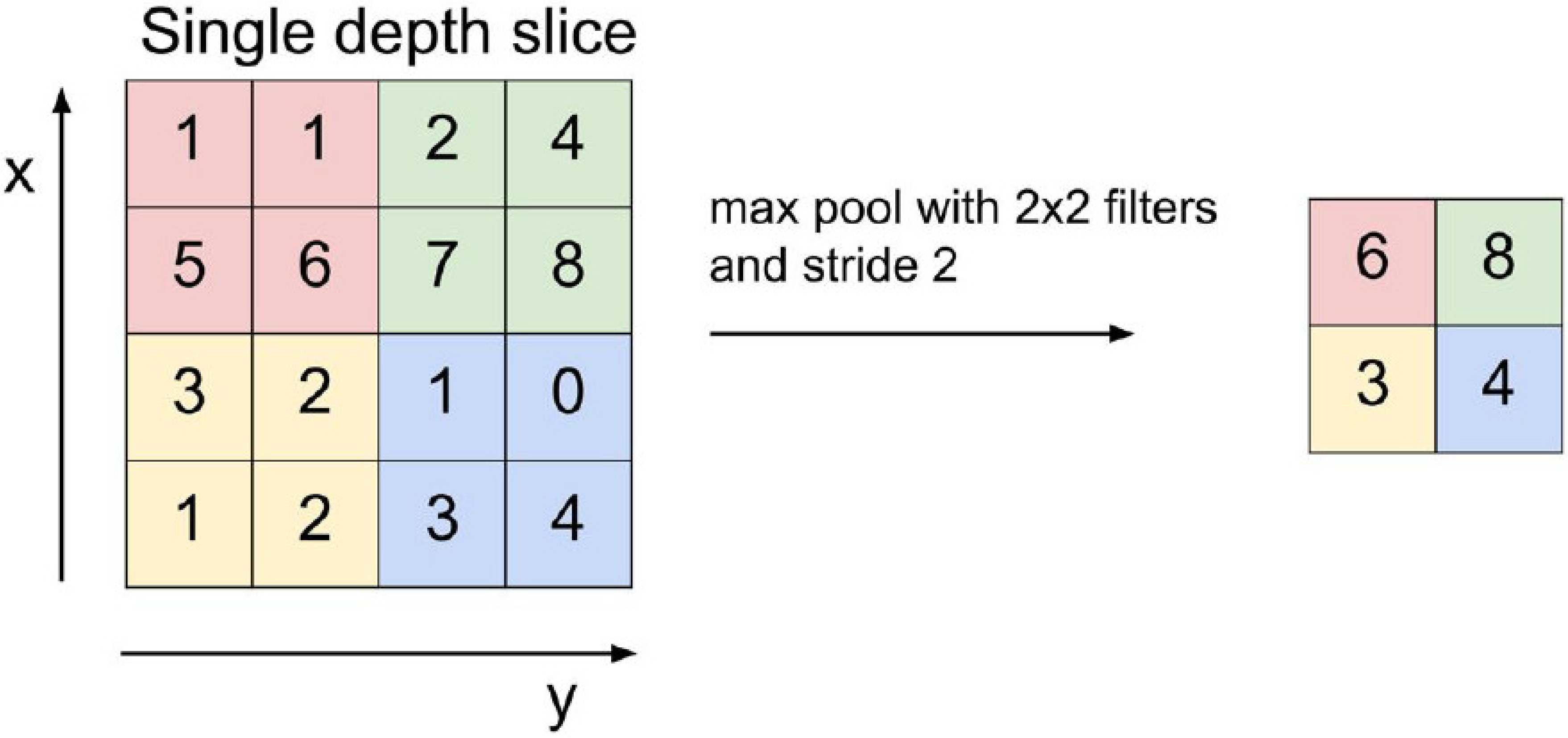
Pooling Layer

- Makes the representations smaller and more manageable
- Operates over each activation map independently



Pooling Layer (**Max Pooling** example)

- A typical pooling unit computes the maximum of a local patch of unites in one feature map (or in a few feature maps)



Single depth slice

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

max pool with 2x2 filters and stride 2

6	8
3	4

Pooling Layer

- Although the role of the convolutional layer is to detect local conjunctions of features from the previous layer
- The role of the **pooling layer** is to merge semantically similar features into one
- Because the relative positions of the features forming a motif can vary somewhat, reliably detecting the motif can be done by coarse-graining the position of each feature

Pooling Layer

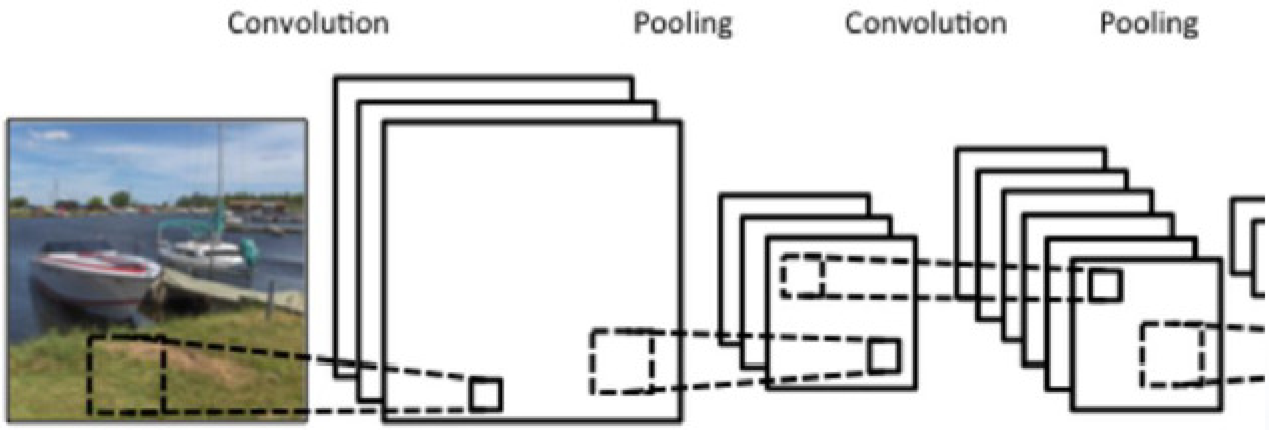
The convolutional and pooling layers in ConvNets are directly inspired by the classic notions of simple cells and complex cells in visual neuroscience

Hierarchies

- Two or three stages of convolution and pooling are stacked, followed by more convolutional and fully-connected layers
- Deep neural networks exploit the property that many natural signals are **compositional hierarchies**, in which higher-level features are obtained by composing lower-level ones
- In images, local combinations of edges form motifs, motifs assemble into parts, and parts form objects

Hierarchies and Pooling

- Neighboring pooling units take input from patches that are shifted by more than one row or column, thereby reducing the dimension of the representation and creating an invariance to small shifts and distortions

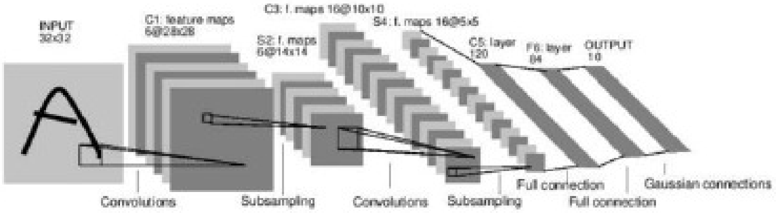


http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

History of CNNs

1998

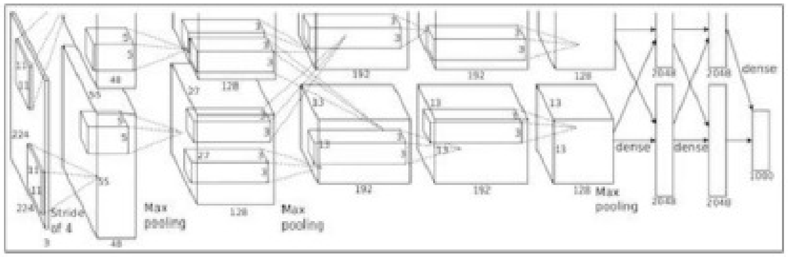
Gradient-based learning applied to document recognition [LeCun, Bottou, Bengio, Haffner]



LeNet-5

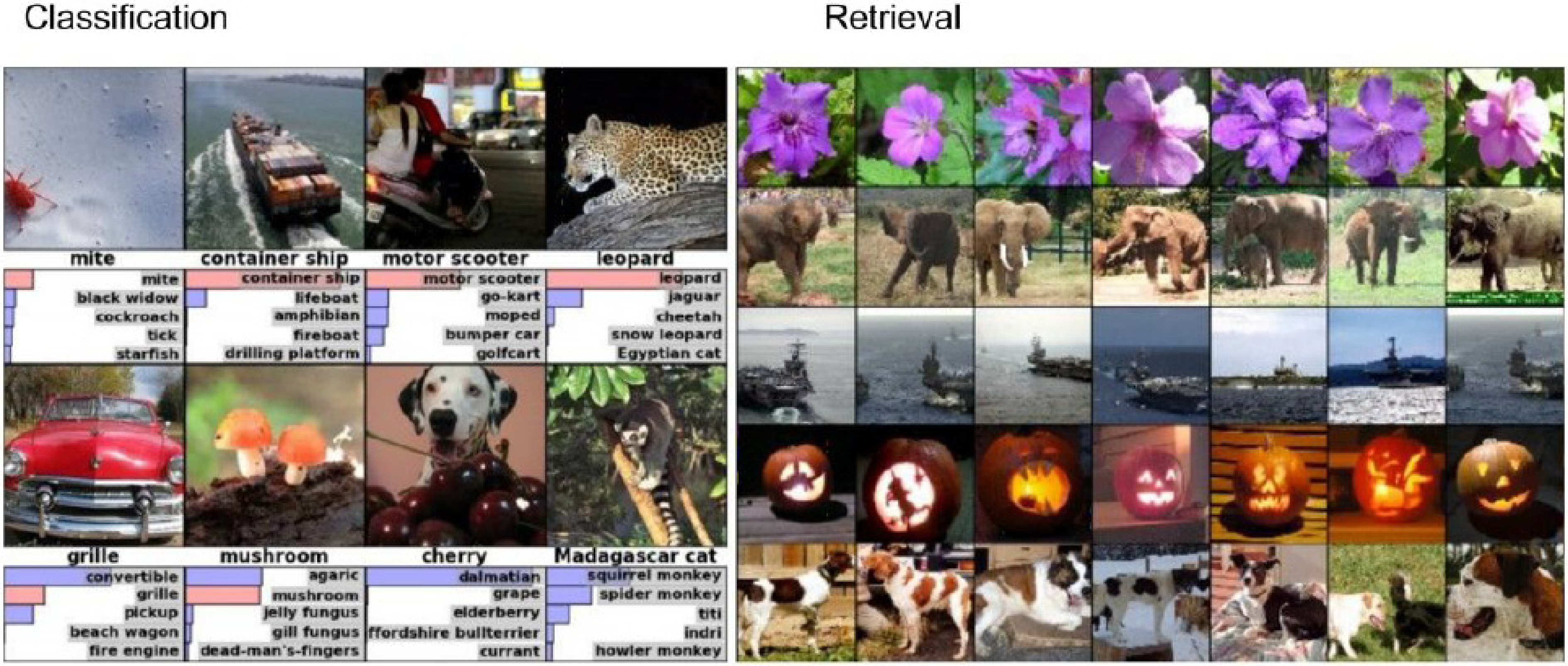
2012

ImageNet Classification with Deep Convolutional Neural Networks [Krizhevsky, Sutskever, Hinton, 2012]



"AlexNet"

Fast-forward to today: CNNs are everywhere



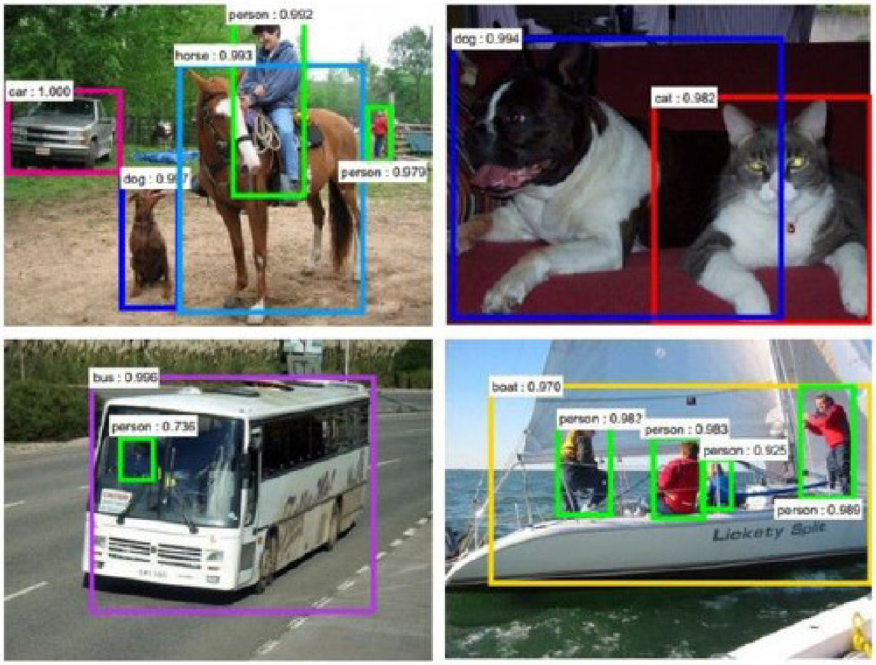
Classification

Retrieval

[Krizhevsky 2012]

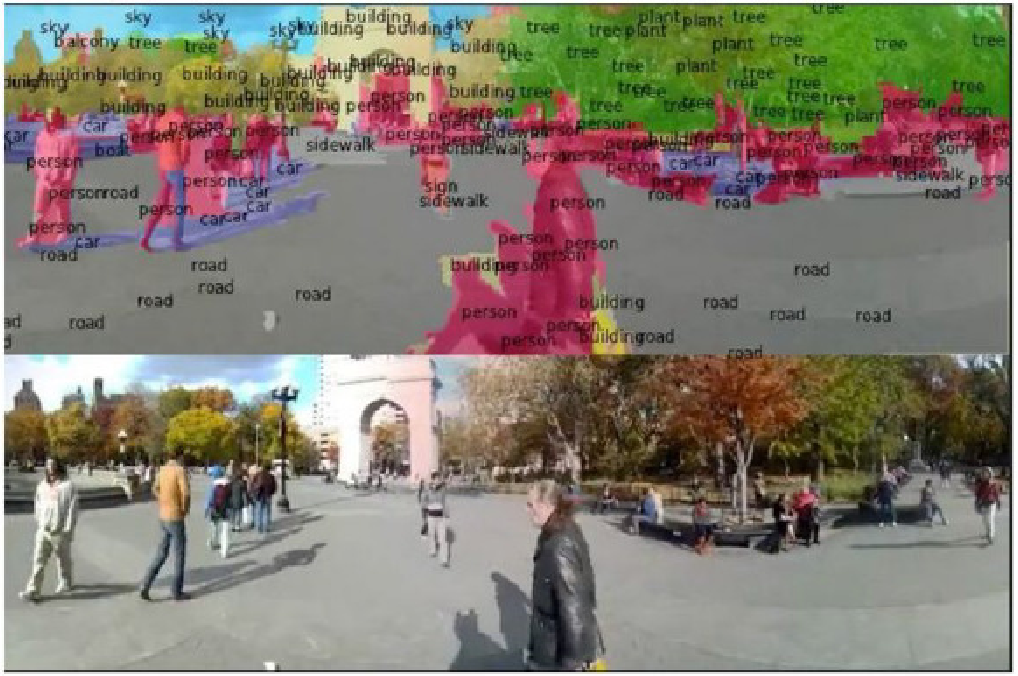
Fast-forward to today: CNNs are everywhere

Detection



[Faster R-CNN: Ren, He, Girshick, Sun 2015]

Segmentation



[Farabet et al., 2012]

… everywhere

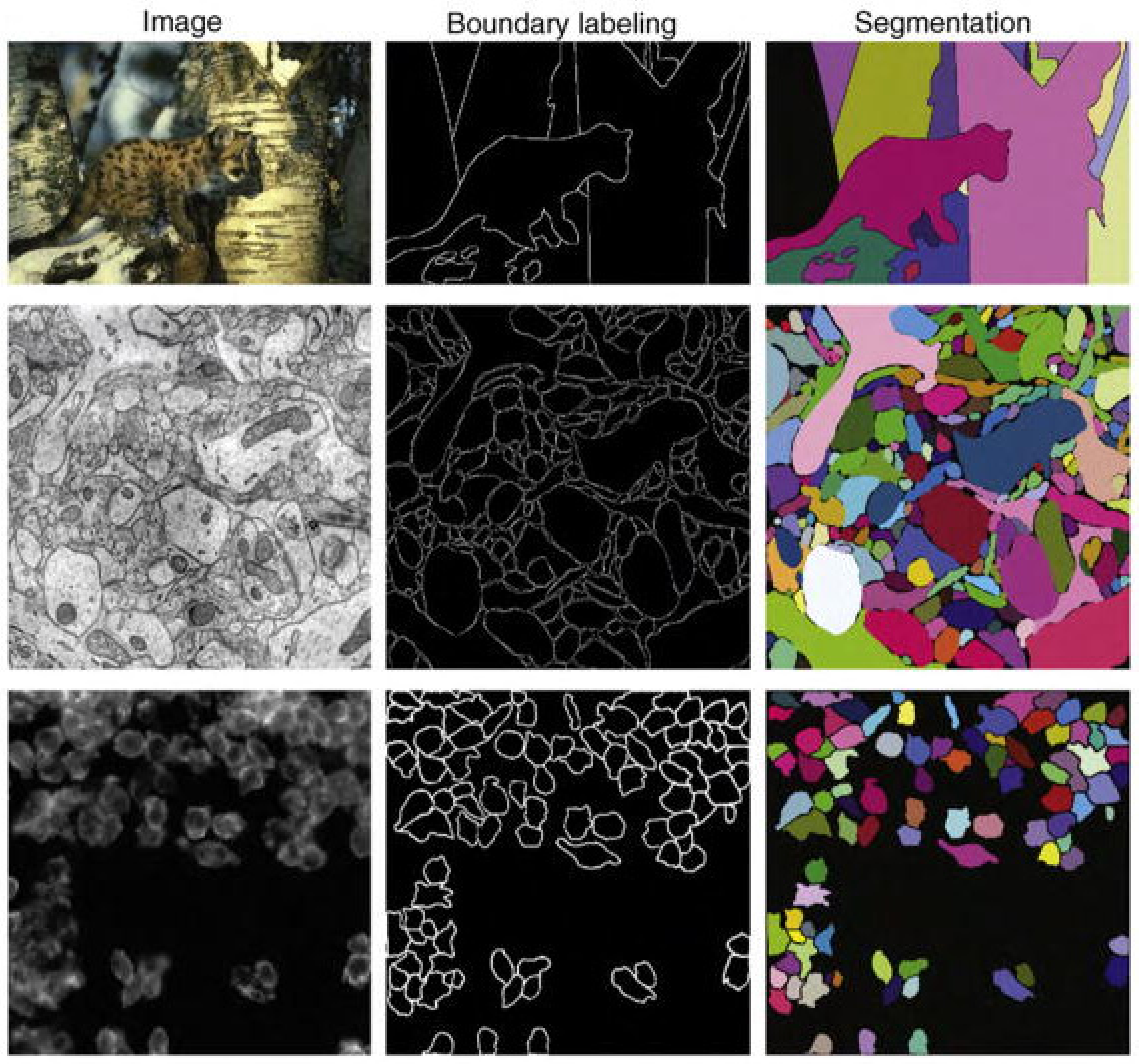
self-driving cars

NVIDIA Tegra X1

…everywhere

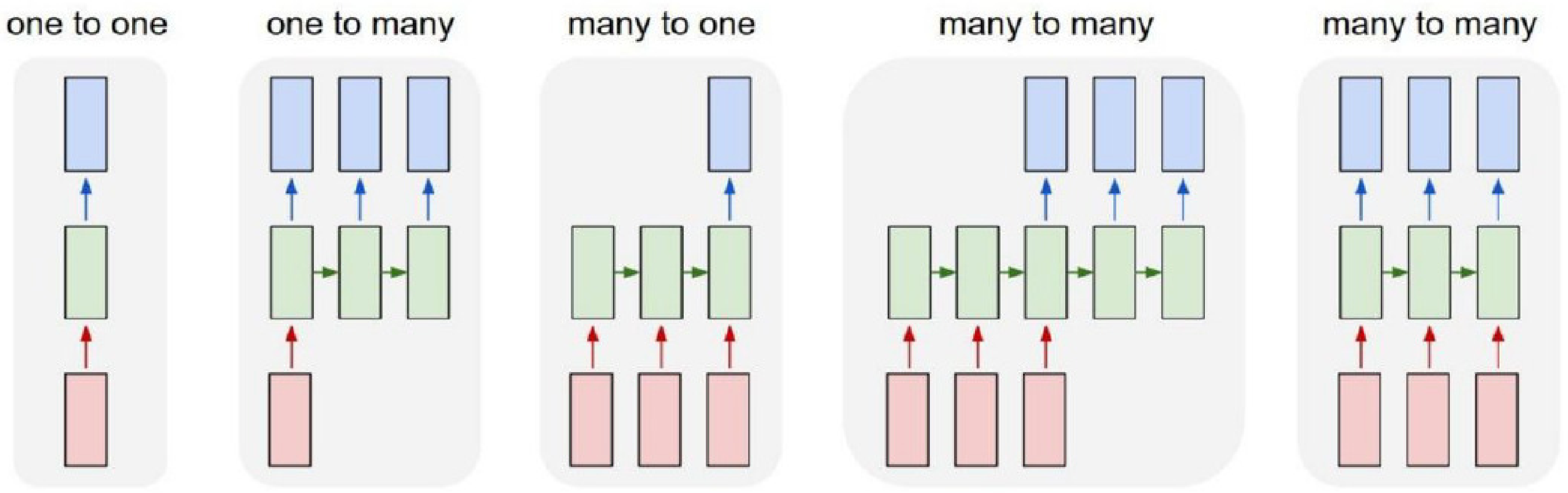
Segmentation

[Turaga et al., 2010]

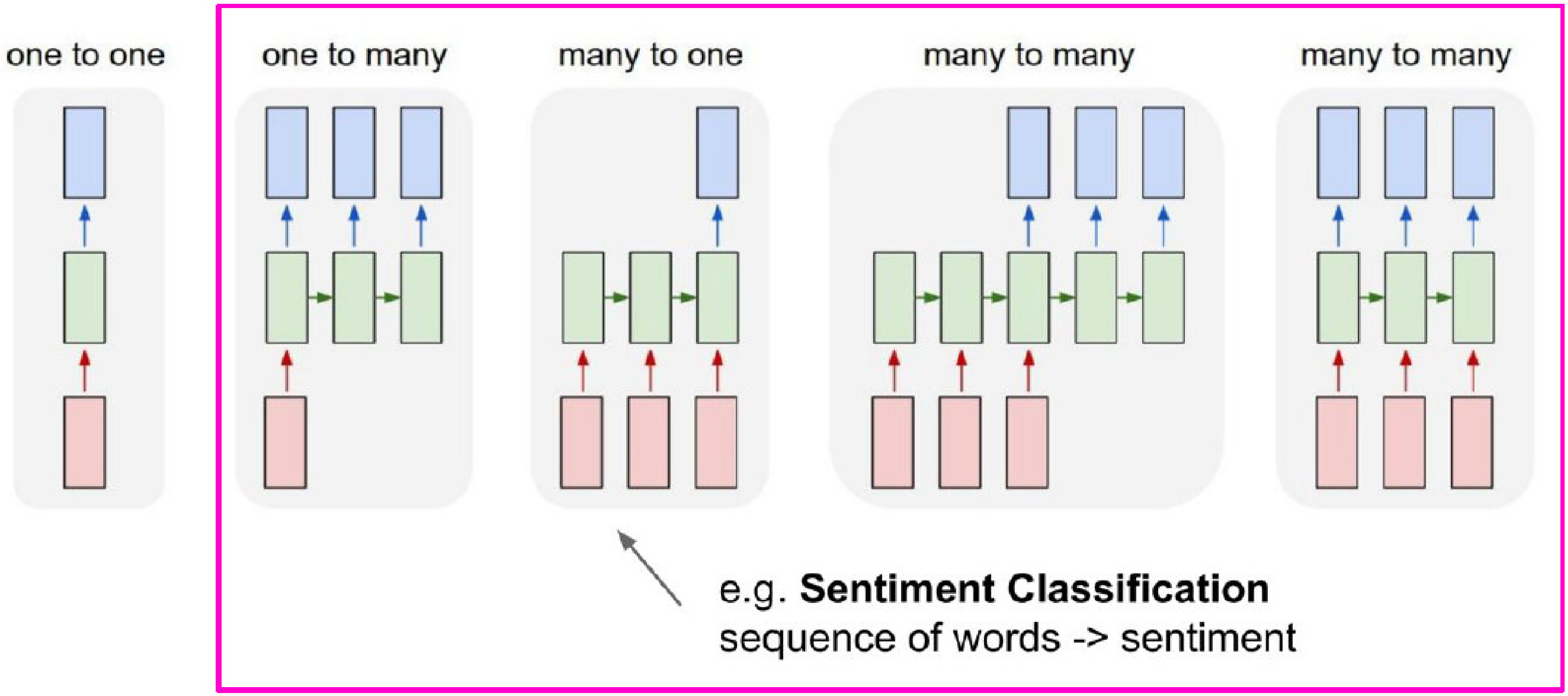


Recurrent Neural Networks (***Brief***)

Standard "Feed-Forward" Neural Network



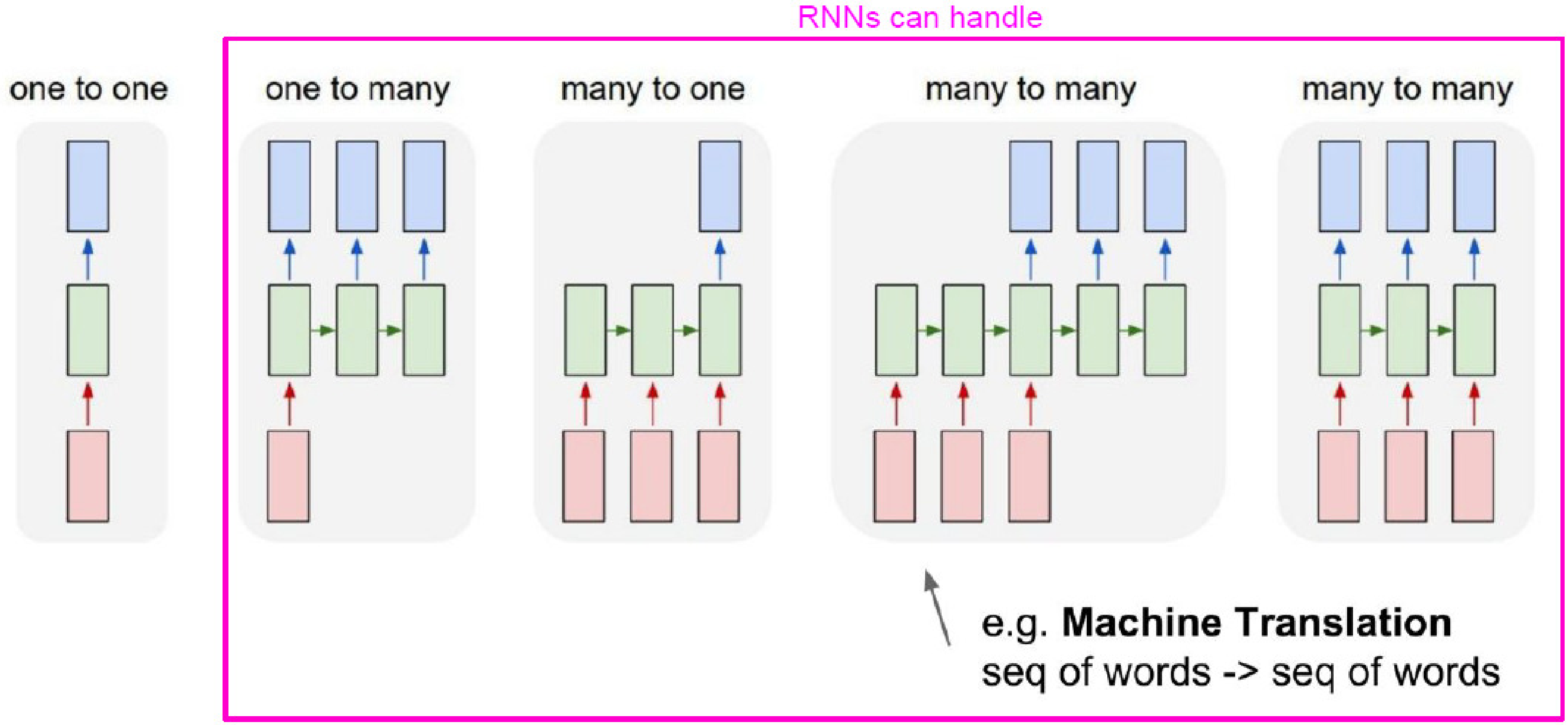
Recurrent Neural Networks (RNNs)



RNNs can handle

e.g. **Sentiment Classification**
sequence of words -> sentiment

Recurrent Neural Networks (RNNs)



RNNs can handle

one to one

one to many

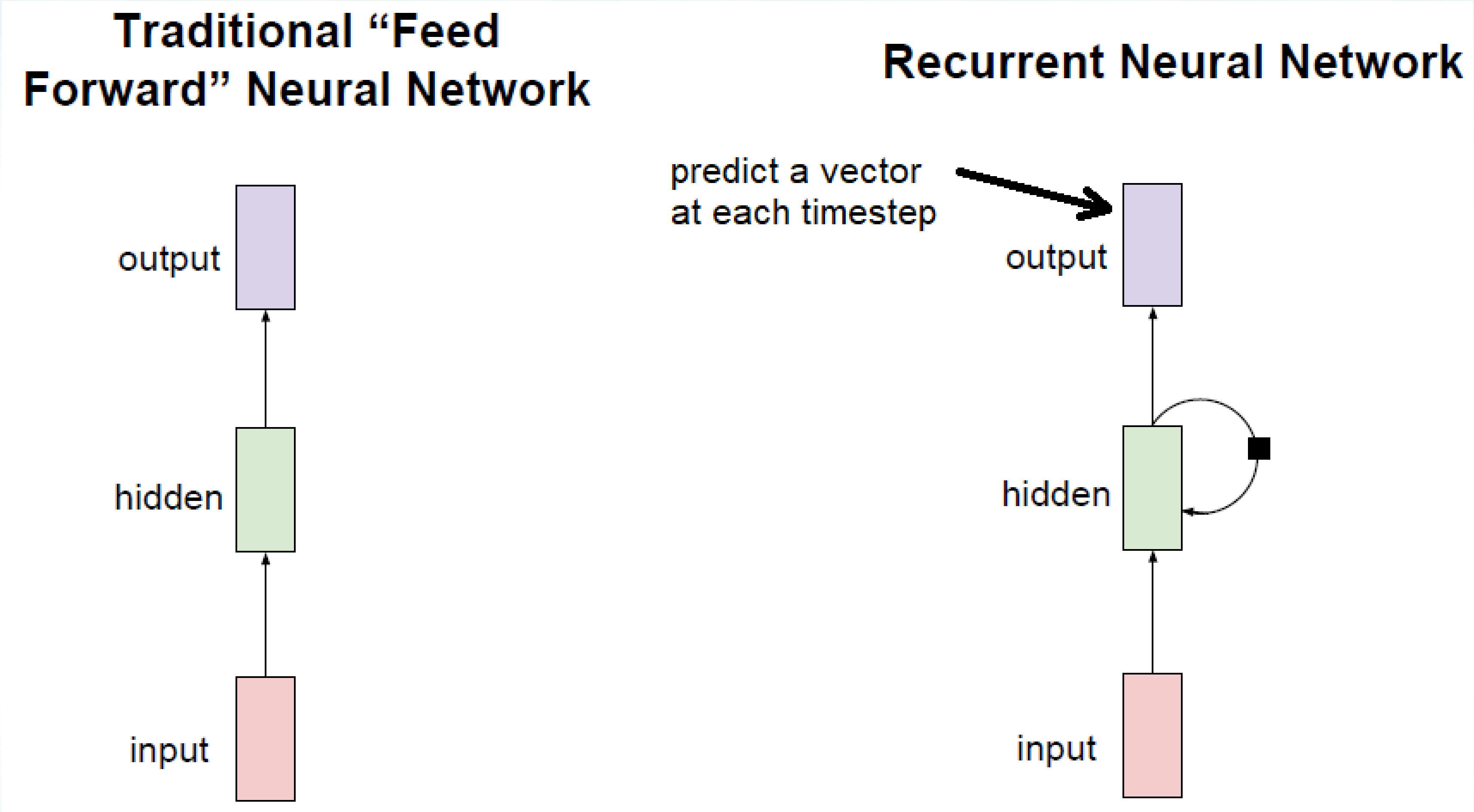
many to one

many to many

many to many

e.g. **Machine Translation**
seq of words -> seq of words

Recurrent Neural Networks (RNNs)



Traditional "Feed Forward" Neural Network

Recurrent Neural Network

predict a vector at each timestep

Recurrent Neural Networks

We can process a sequence of vectors **x** by applying a recurrence formula at every time step:

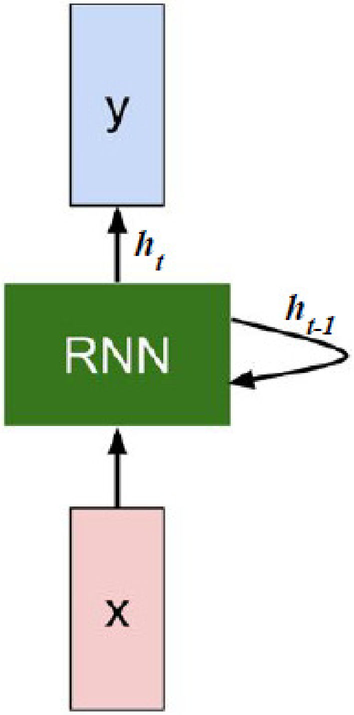
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

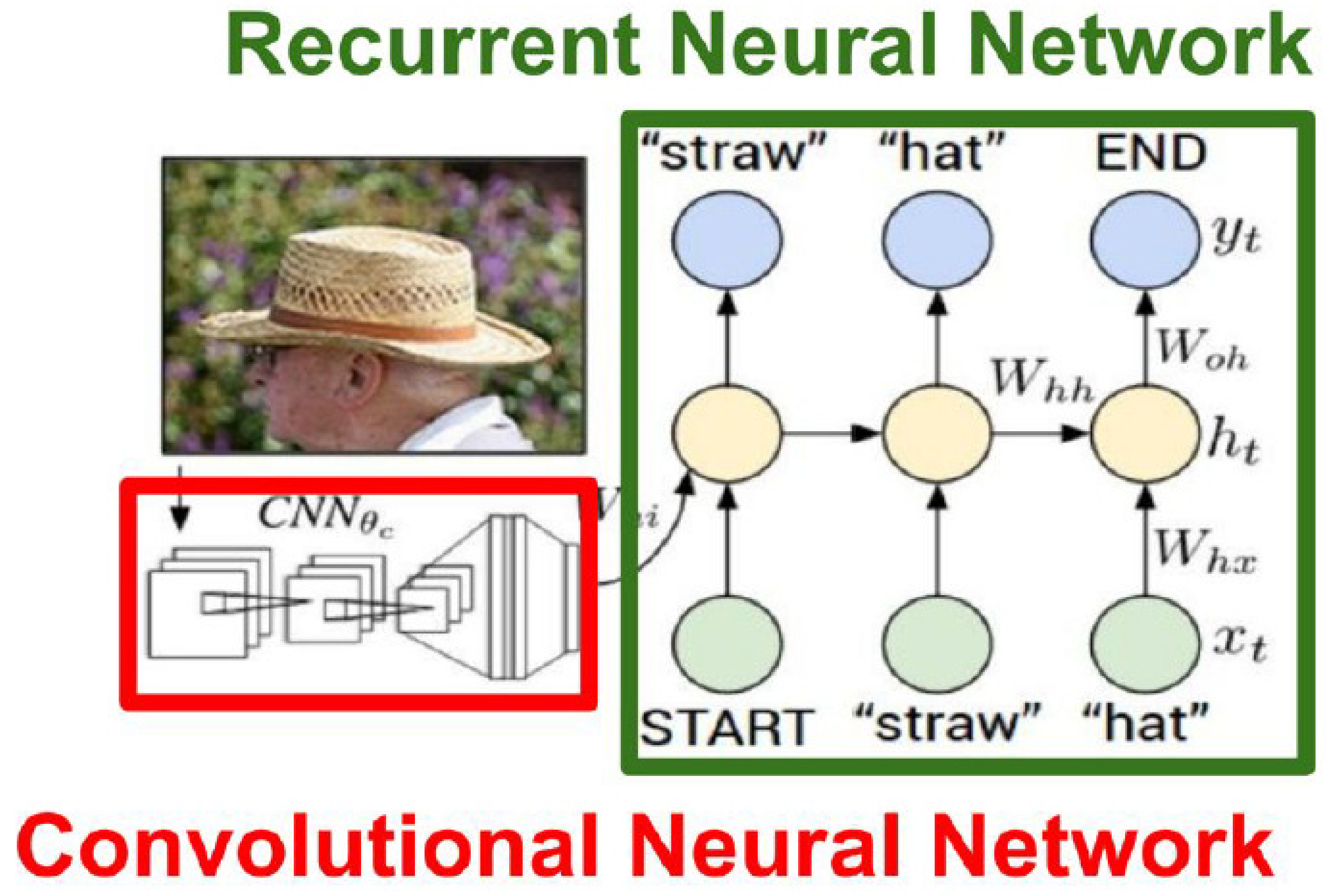
old state

input vector at some time step

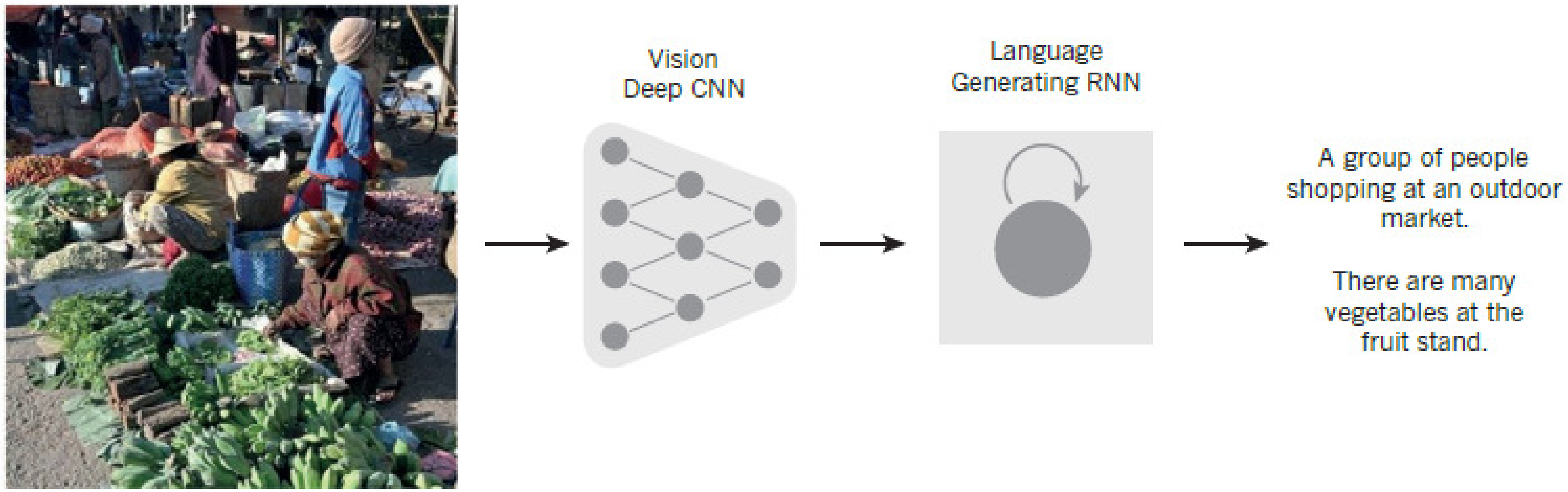
some function with parameters W



RNN and CNN Together (*Caption Generation*)

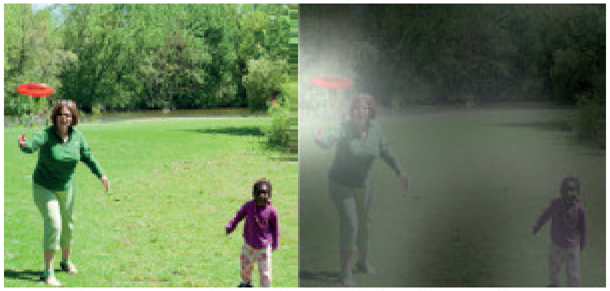


RNN and CNN Together (*Caption Generation*)

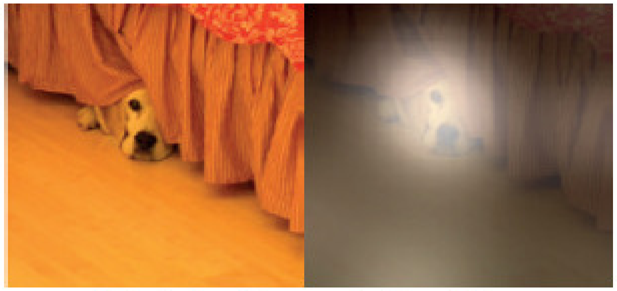

Vision
Deep CNN
Language
Generating RNN
A group of people shopping at an outdoor market.
There are many vegetables at the fruit stand.

Yann LeCun et al. Nature, Vol. 521, 2015

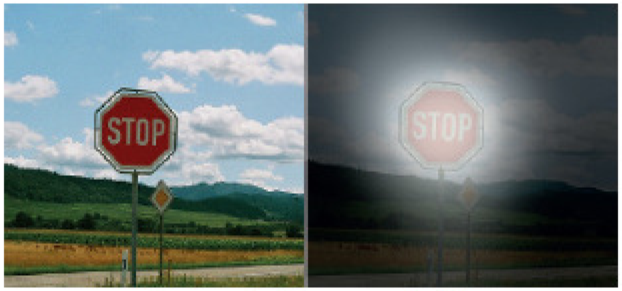
RNN and CNN Together (*Caption Generation*)



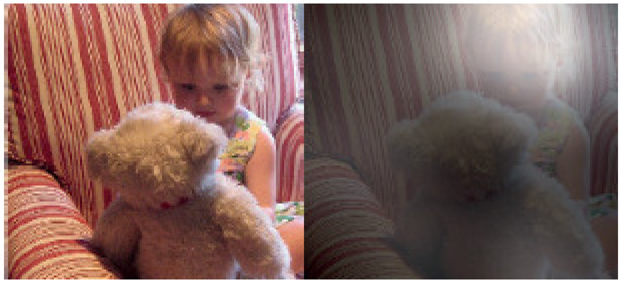
A woman is throwing a **frisbee** in a park.



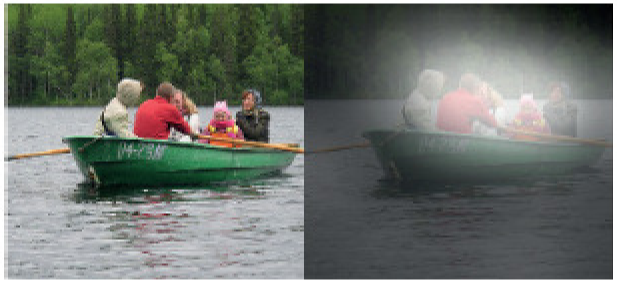
A **dog** is standing on a hardwood floor.



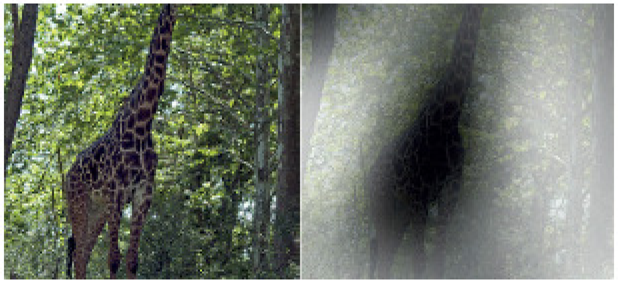
A **stop** sign is on a road with a mountain in the background



A little **girl** sitting on a bed with a teddy bear.



A group of **people** sitting on a boat in the water.



A giraffe standing in a forest with **trees** in the background.

Vinyals, O., Toshev, A., Bengio, S. & Erhan, D. Show and tell: a neural image caption generator. In *Proc. International Conference on Machine Learning* http:// arxiv.org/abs/1502.03044 (2014).

Yann LeCun et al. Nature, Vol. 521, 2015

CNNs Are a Success!

- The success of CNNs has brought about a revolution in computer vision
- CNNs are now the dominant approach for almost all recognition and detection tasks
- CNNs approach human performance on some tasks!
- Stunning demonstration of CNN + RNN for the generation of image captions (*seen in previous two slides*)

Some Software

- Many options exist for deep learning
- Main programming languages:
  - Python (https://www.python.org)
  - Lua (https://www.lua.org)
  - Matlab (https://www.mathworks.com)
- Main frameworks:
  - Theano
  - TensorFlow
  - Keras
  - …more!

Python Software

- Anaconda Python (Python with some pre-loaded libraries) [https://www.continuum.io/downloads]
- Natural Language Toolkit / NLTK (Python library for natural language processing) [installed with Anaconda]
- Theano (Symbolic functions, needed for deep neural networks)
- Keras (Layer for arbitrary neural networks, including RNNs/CNNs/etc)
- Gensim (Natural Language Processing, word2vec)

References:

~NN and Deep Learning ~ Dr.Qi
~Deep Learning ~ Yann LeCun, Yoshua Bengio & Geoffrey Hinton ~Nature, Vol 521, May 2015
~Imagenet classification with deep convolutional neural networks ~ A.Krizhevsky et al. ~ NIPS 2012
~Very deep convolutional networks for large-scale image recognition ~ K.Simonyan and A Zisserman ~ ICLR 2015
~Going deeper with convolutions ~ C.Szegedy et al. ~ CVPR 2015
~Filters and Convolution ~ https://en.Wikipedia.org/wiki/Convolution
~Convolutional Neural Networks & Recurrent Neural Networks: cs231n.standford.edu/