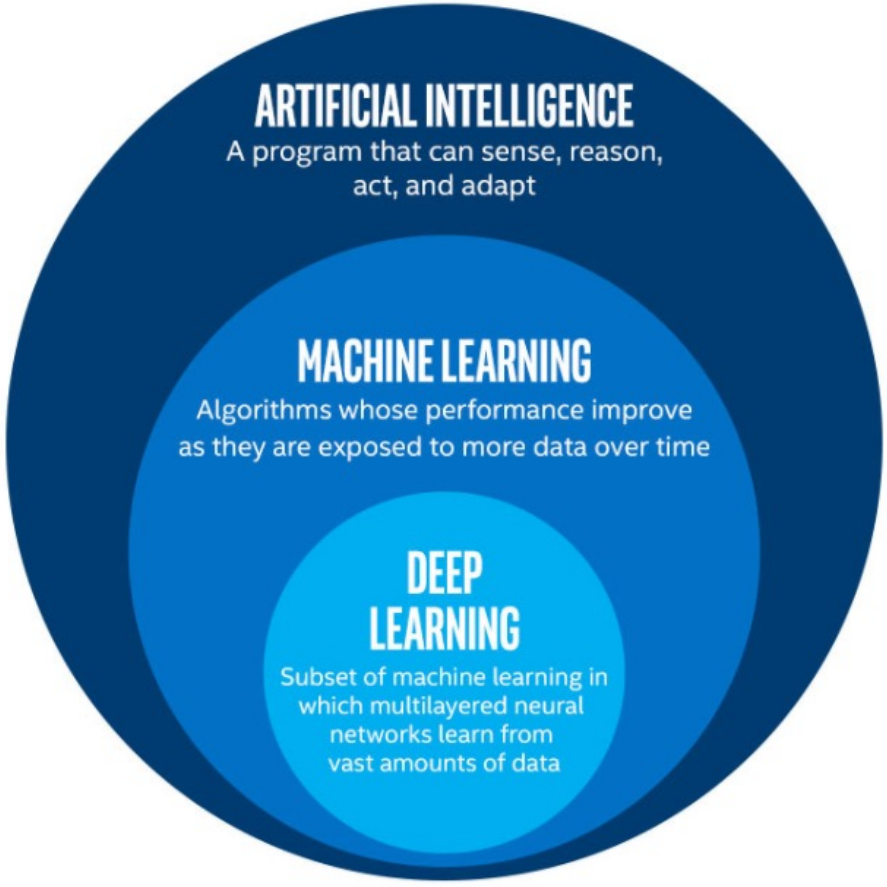
A gentle introduction to Deep Learning

Introduction



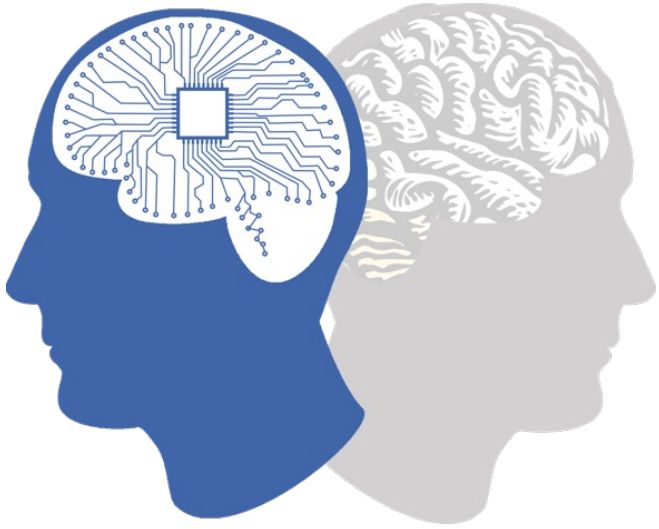
Introduction

Tabular data
(Traditional machine learning)

Photo, Video, Audio, Text data
(Deep learning)

AI applications

- Healthcare
- Automobile
- Banking, Finance
- Surveillance
- Social Media
- Entertainment



- Education
- Space Exploration
- Gaming
- Robotics
- Agriculture
- E-commerce

AI applications – Economics

GOOGL

Company	Symbol
Google	GOOGL
Apple	AAPL
Microsoft	MSFT
Amazon	AMZN
Tesla	TSLA

Beginning Stock Prediction ...

Loss after last Epoch (100) is: 0.0045591918751597404

Predicted Stock Price of GOOGL for date 16-02-2019 is: 1111.443$

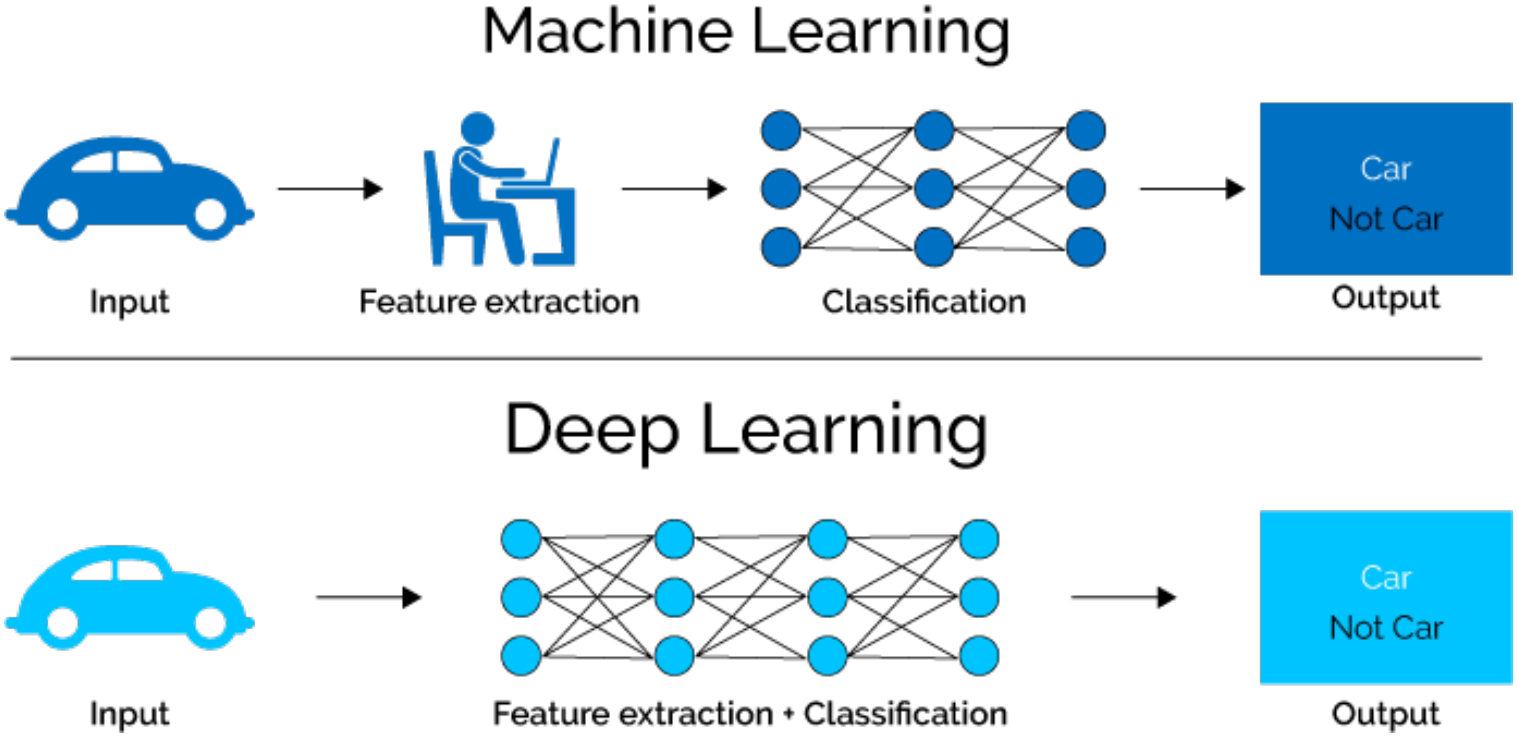
Stock / Cryptocurrency price prediction

What is Deep Learning (DL) ?

A machine learning subfield of learning **representations** of data. Exceptional effective at **learning patterns**.

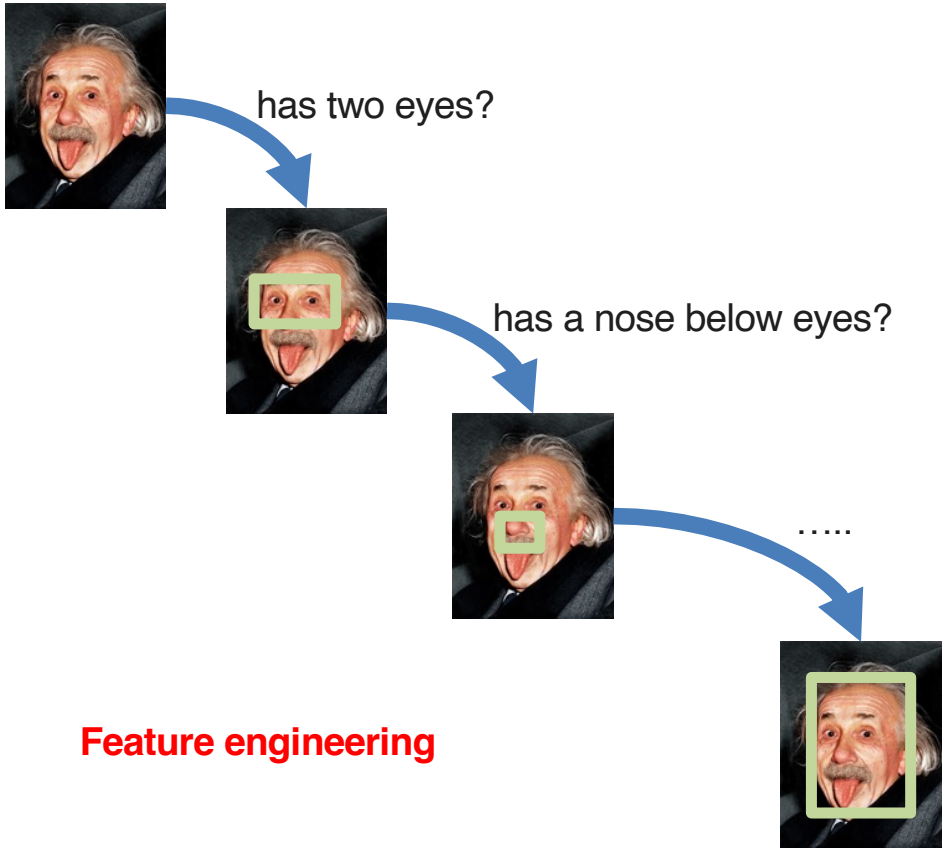
Deep learning algorithms attempt to learn (multiple levels of) representation by using a **hierarchy of multiple layers**

If you provide the system **tons of information**, it begins to understand it and respond in useful ways.



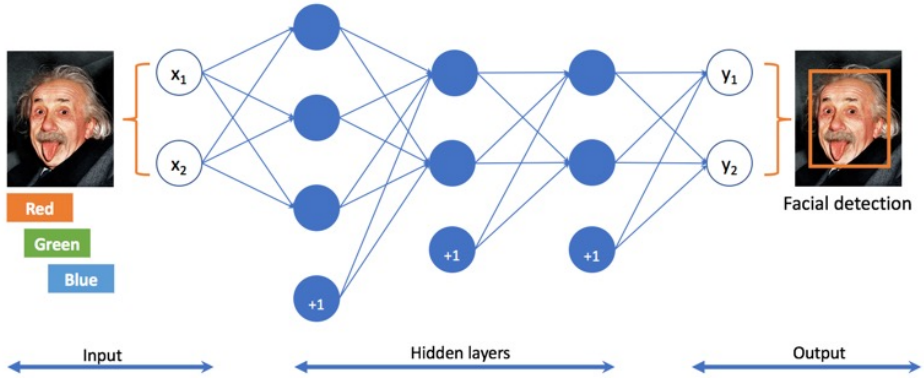
https://www.xenonstack.com/blog/static/public/uploads/media/machine-learning-vs-deep-learning.png

Traditional Machine Learning



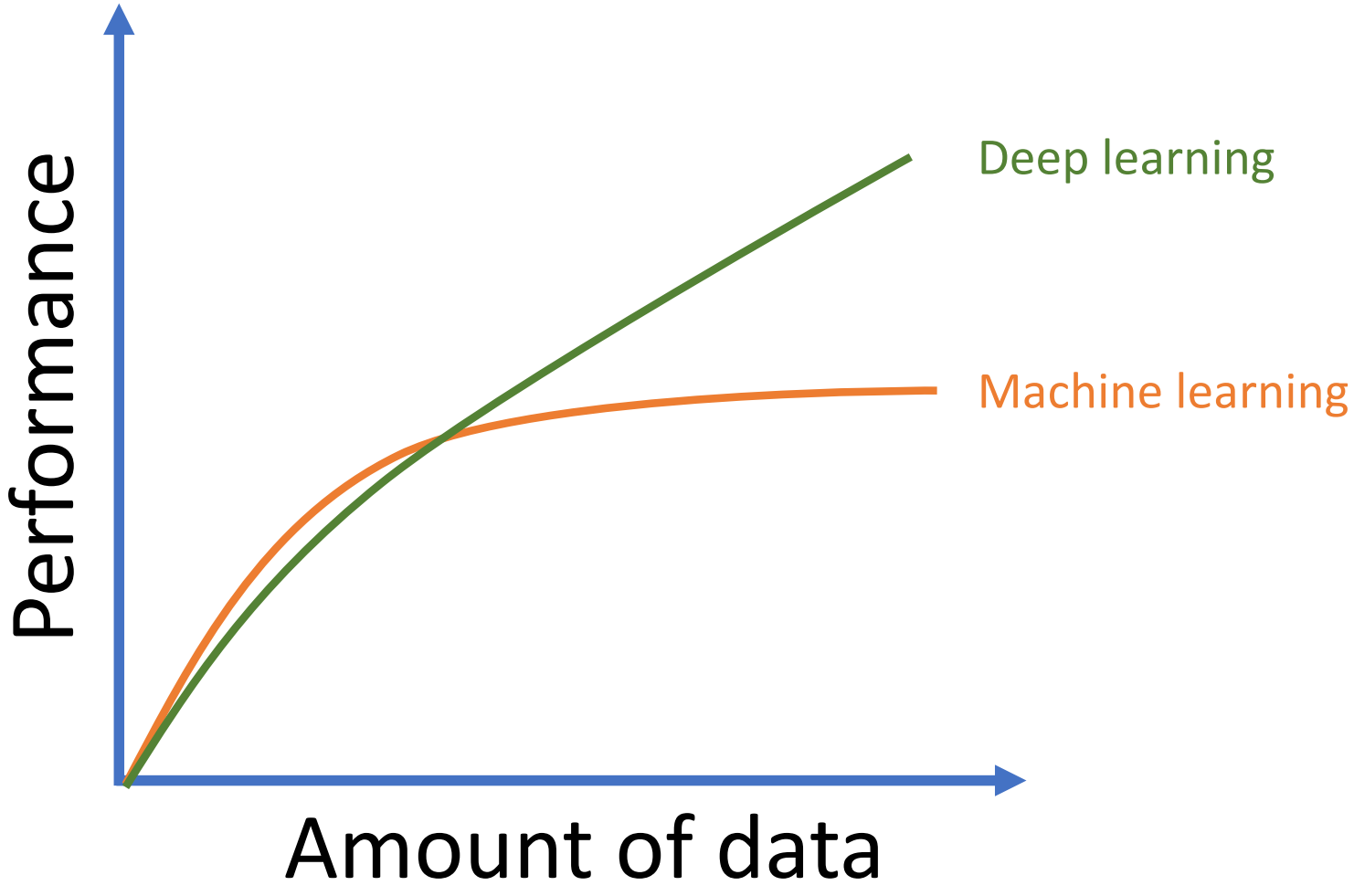
Feature engineering

Deep learning

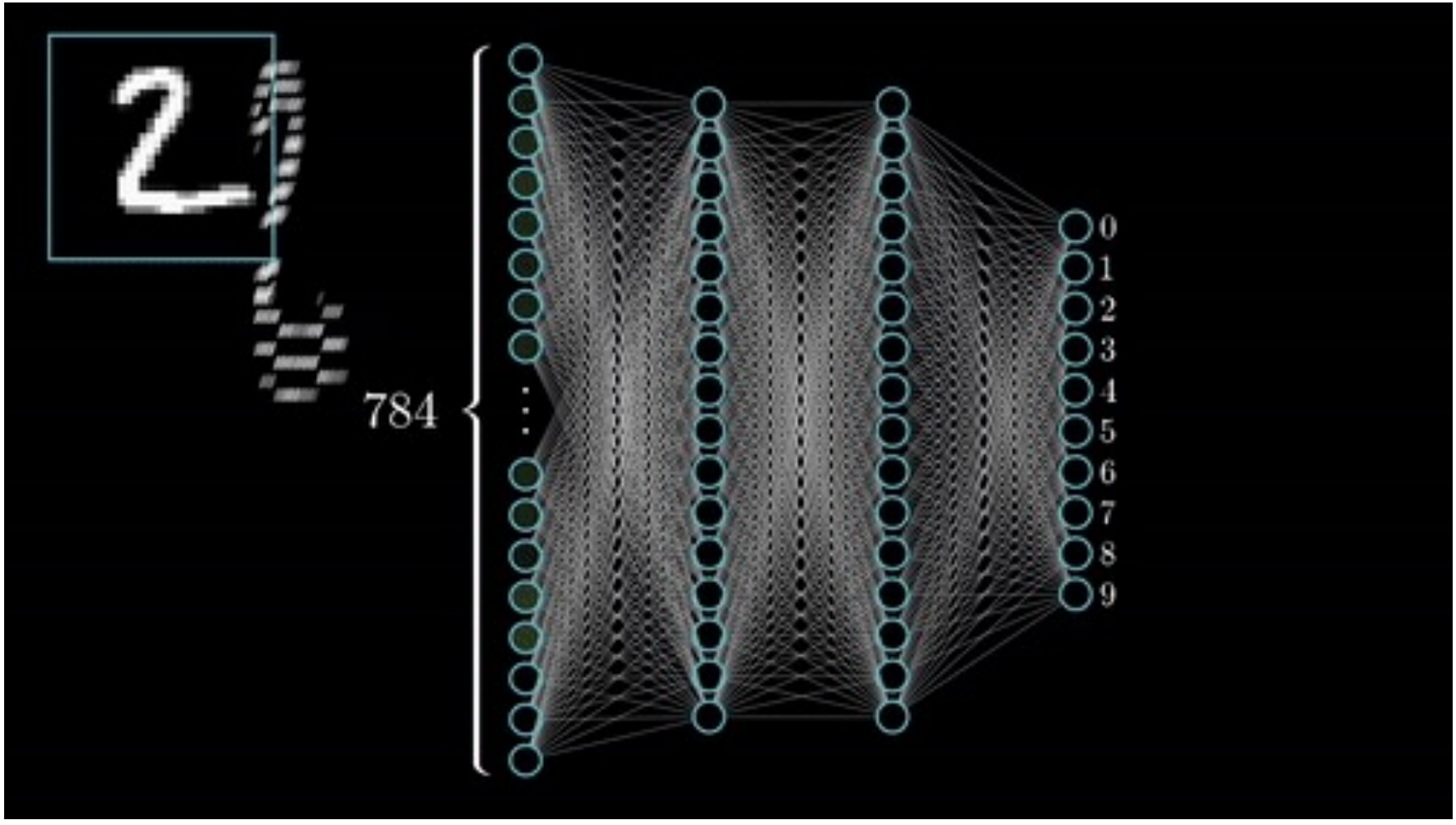


NO feature engineering

Why is DL useful?

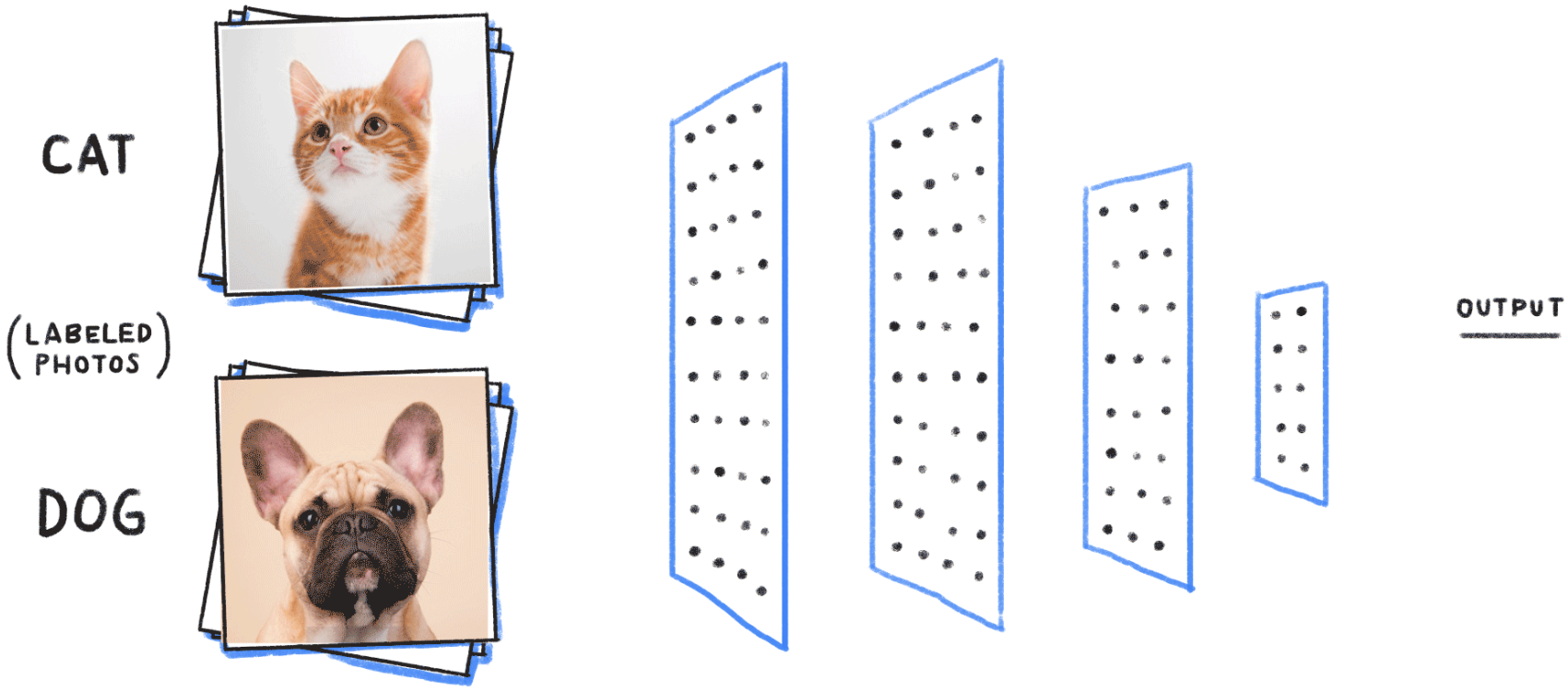


Deep Learning as a classifier



Neural Network is trying to predict the image (data sample) that given to it. it predicts that the number is 2 here.

Deep Learning as a classifier



Neural network is predicting cat/dog image

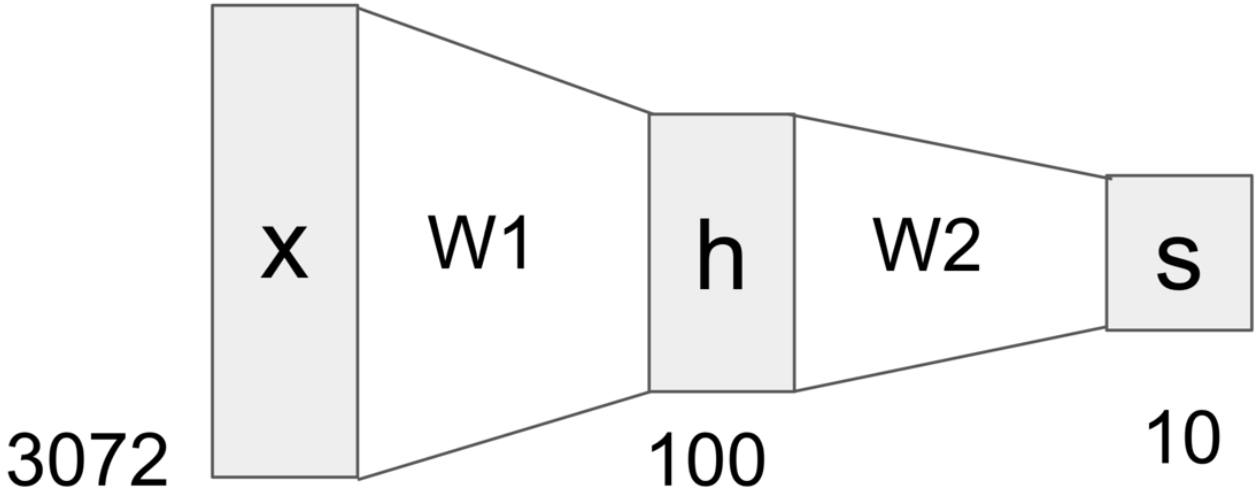
Neural Network

Linear score function:

$$f = Wx$$

2-layer Neural Network

$$f = W_2 \max(0, W_1 x)$$



Neural Network

Linear score function:

$$f = Wx$$

2-layer Neural Network

$$f = W_2 \max(0, W_1 x)$$

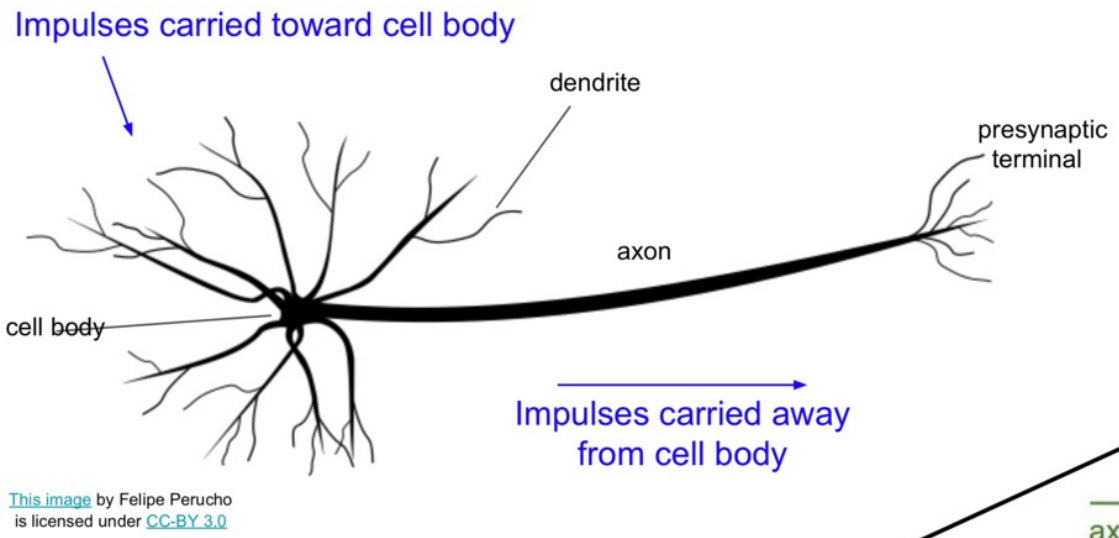
3-layer Neural Network

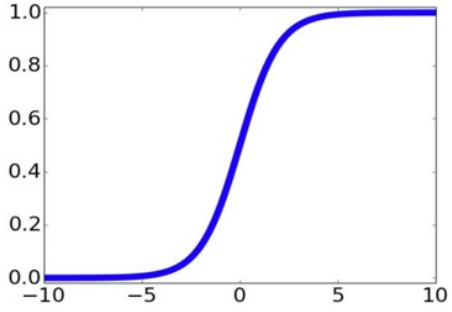
$$f = W_3 \max(0, W_2 \max(0, W_1 x))$$

... and so on.

Deep learning layers as Lego blocks

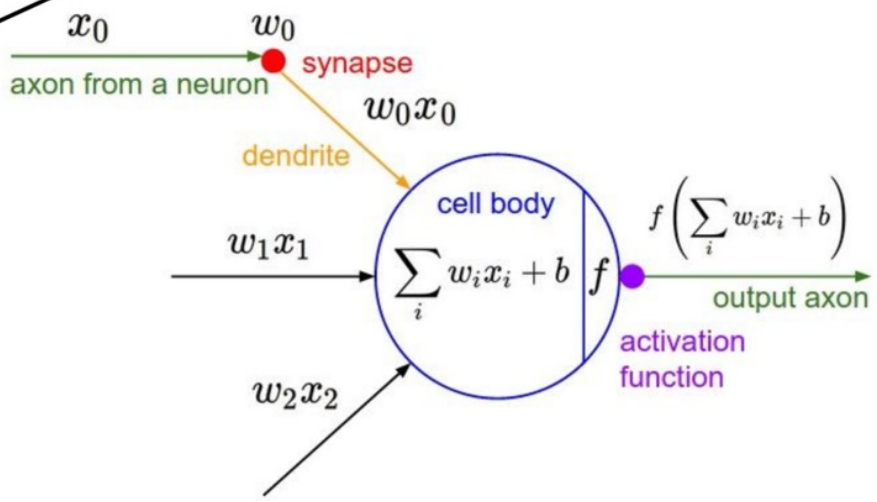
Neural Network





sigmoid activation function

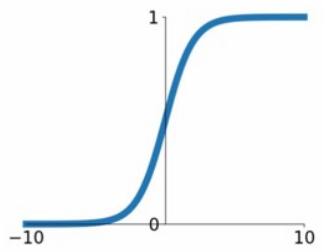
$$\frac{1}{1+e^{-x}}$$



Activation function

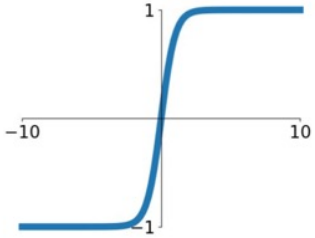
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



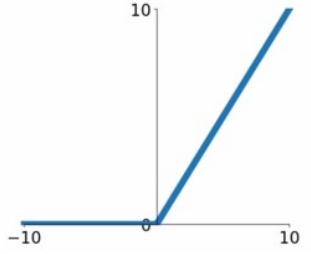
tanh

$$\tanh(x)$$



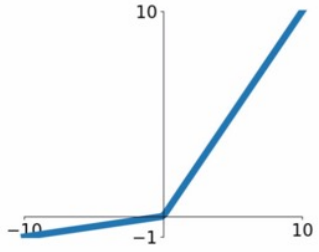
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

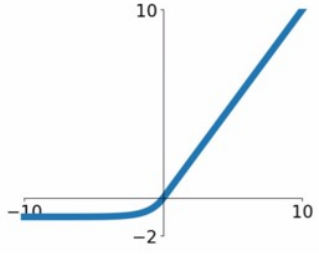


Maxout

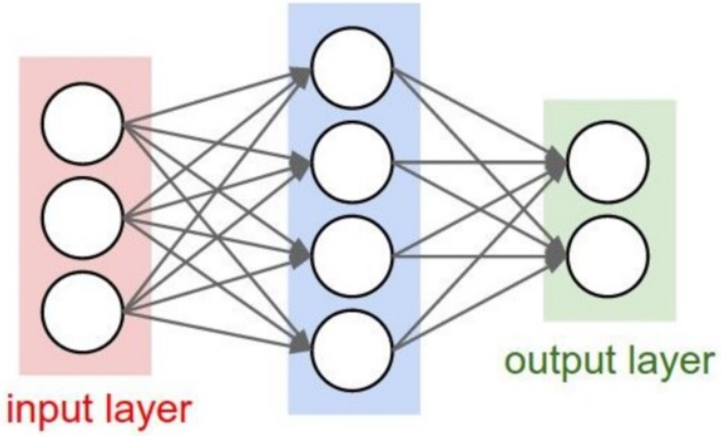
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

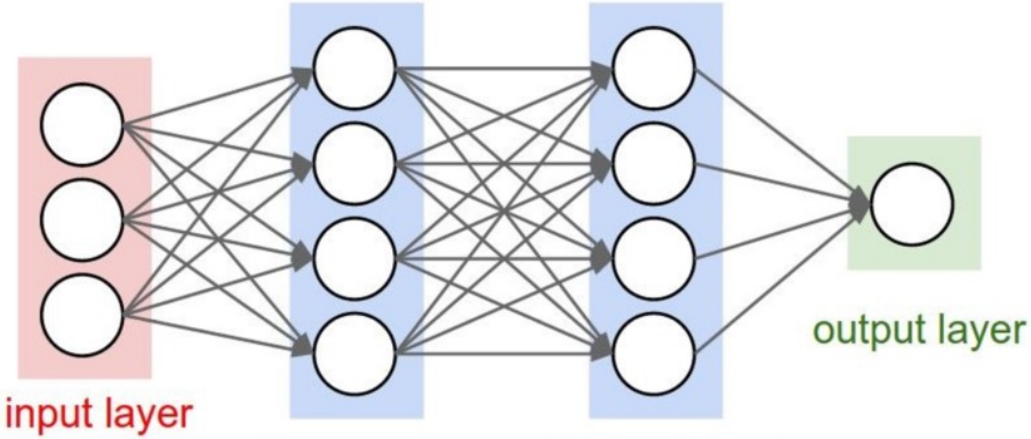
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Neural Network Architectures



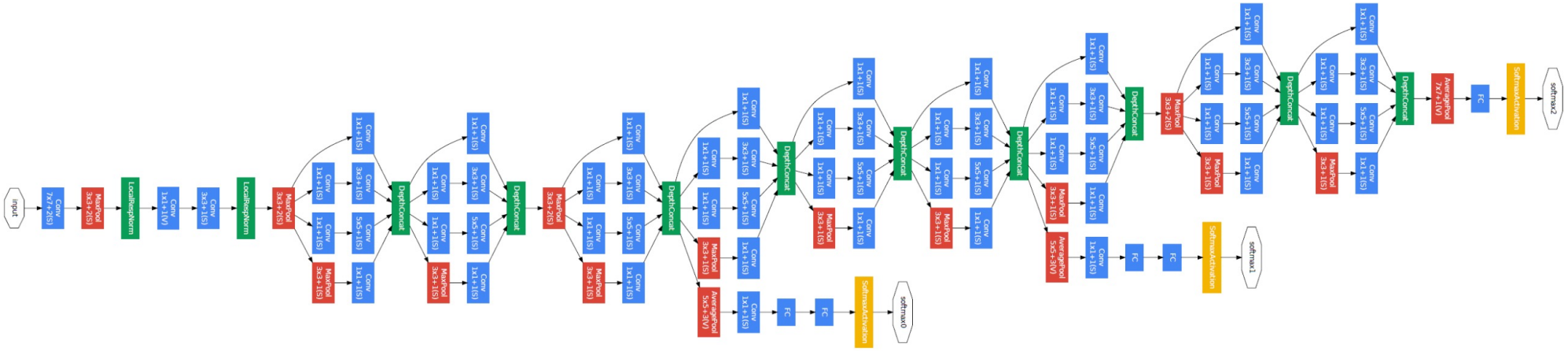


"2-layer Neural Net", or
"1-hidden-layer Neural Net"

"3-layer Neural Net", or
"2-hidden-layer Neural Net"

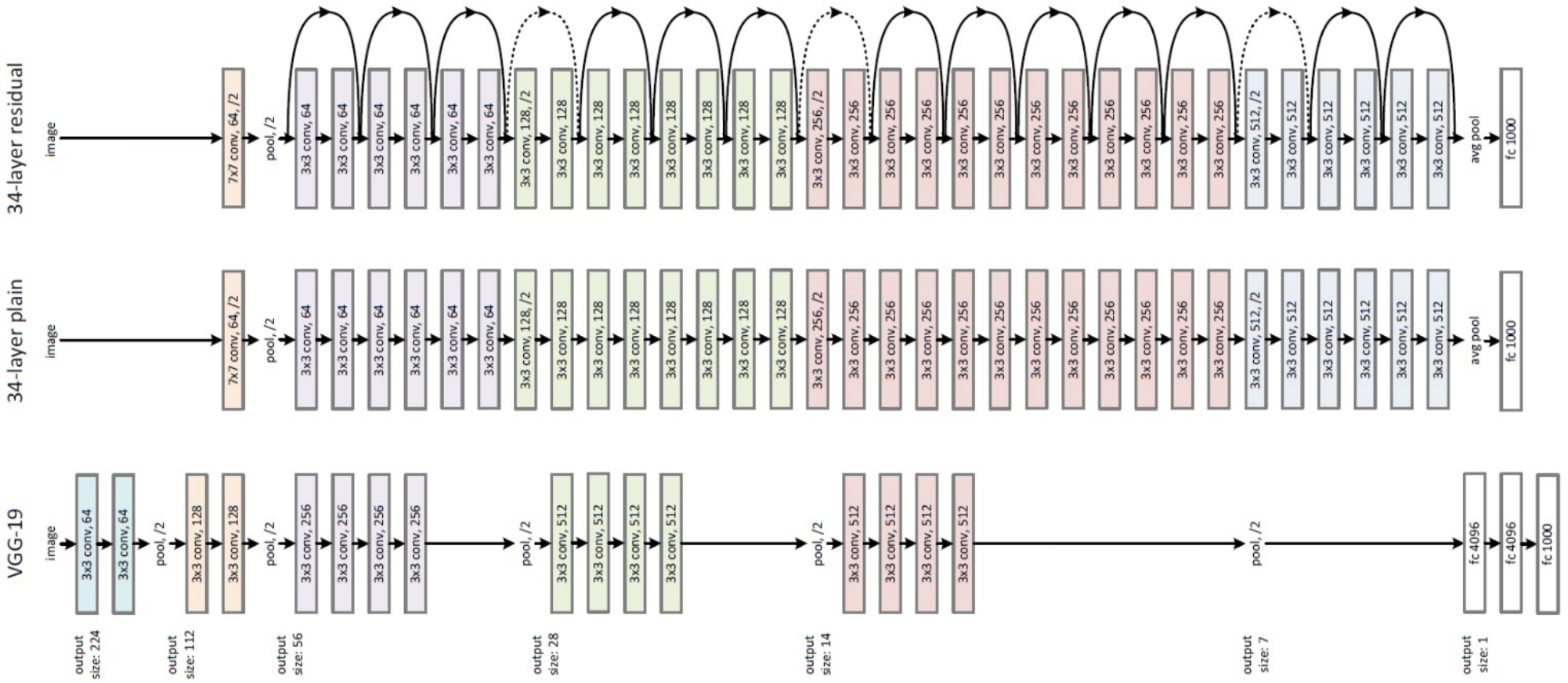
"Fully-connected" layers

Neural Network Architectures



GoogLe Net (Inception v1)

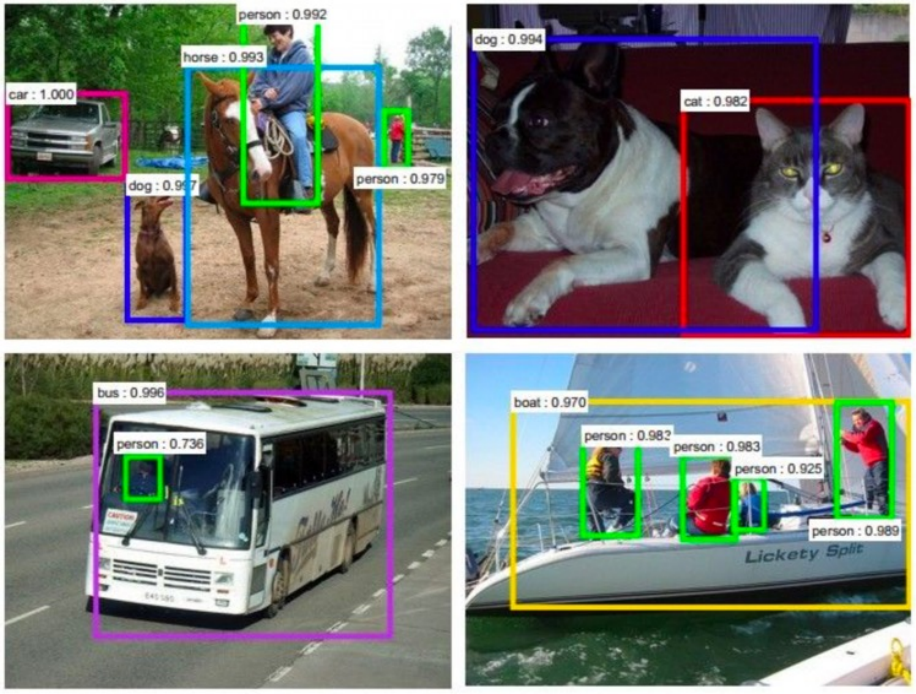
Neural Network Architectures



ResNet

Convolution Neural Network (ConvNet)

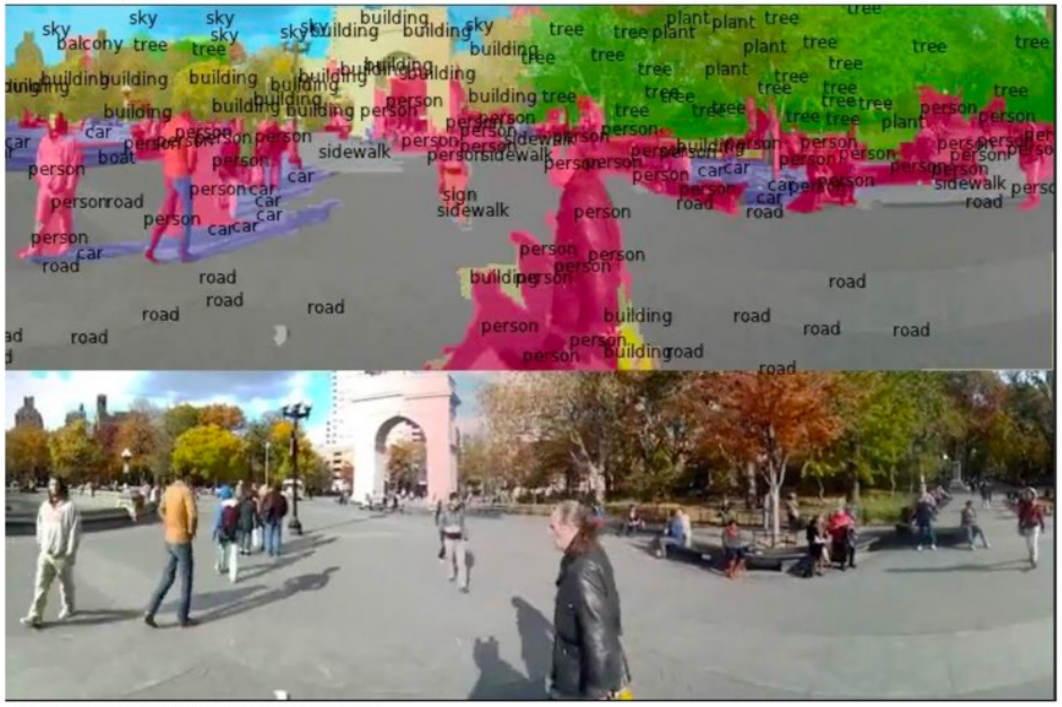
Detection



Figures copyright Shaoqing Ren, Kaiming He, Ross Girschick, Jian Sun, 2015. Reproduced with permission.

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

Segmentation

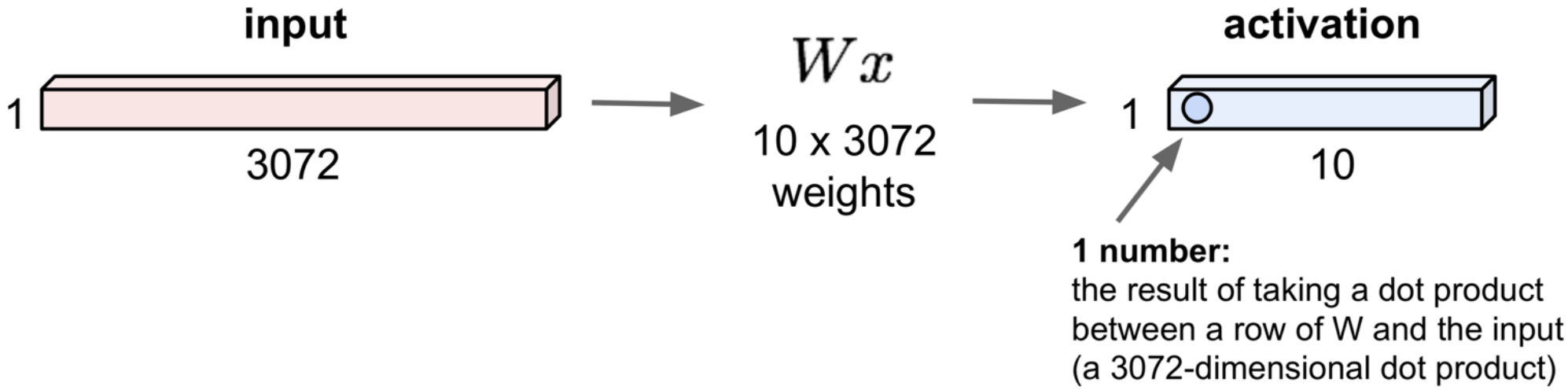


Figures copyright Clement Farabet, 2012. Reproduced with permission.

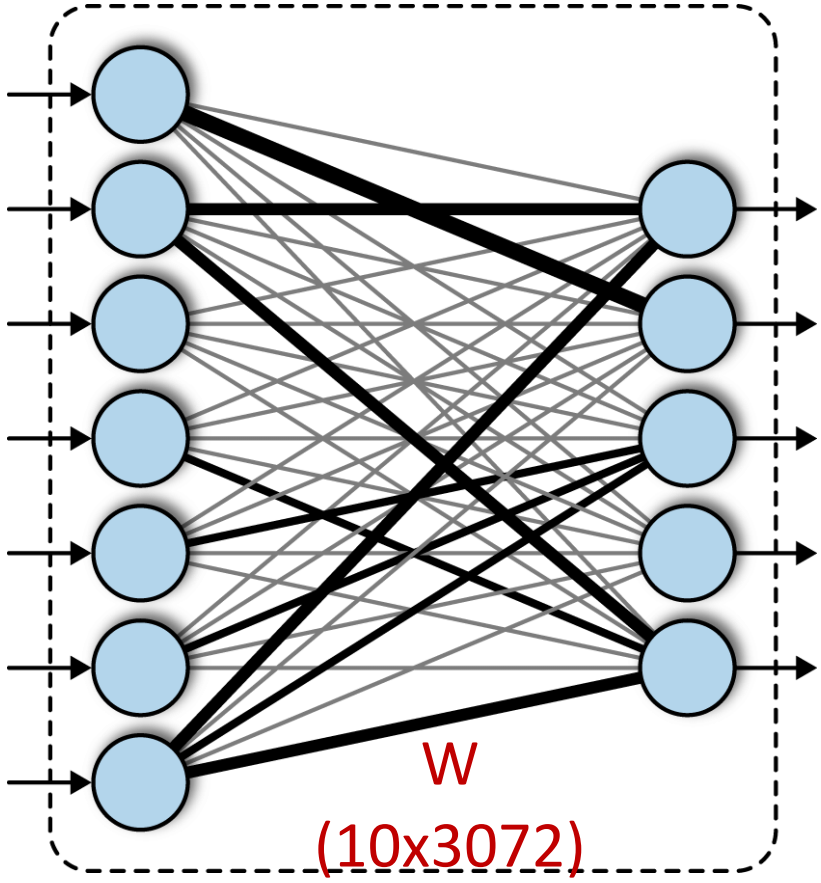
[Farabet et al., 2012]

Fully Connected Layer

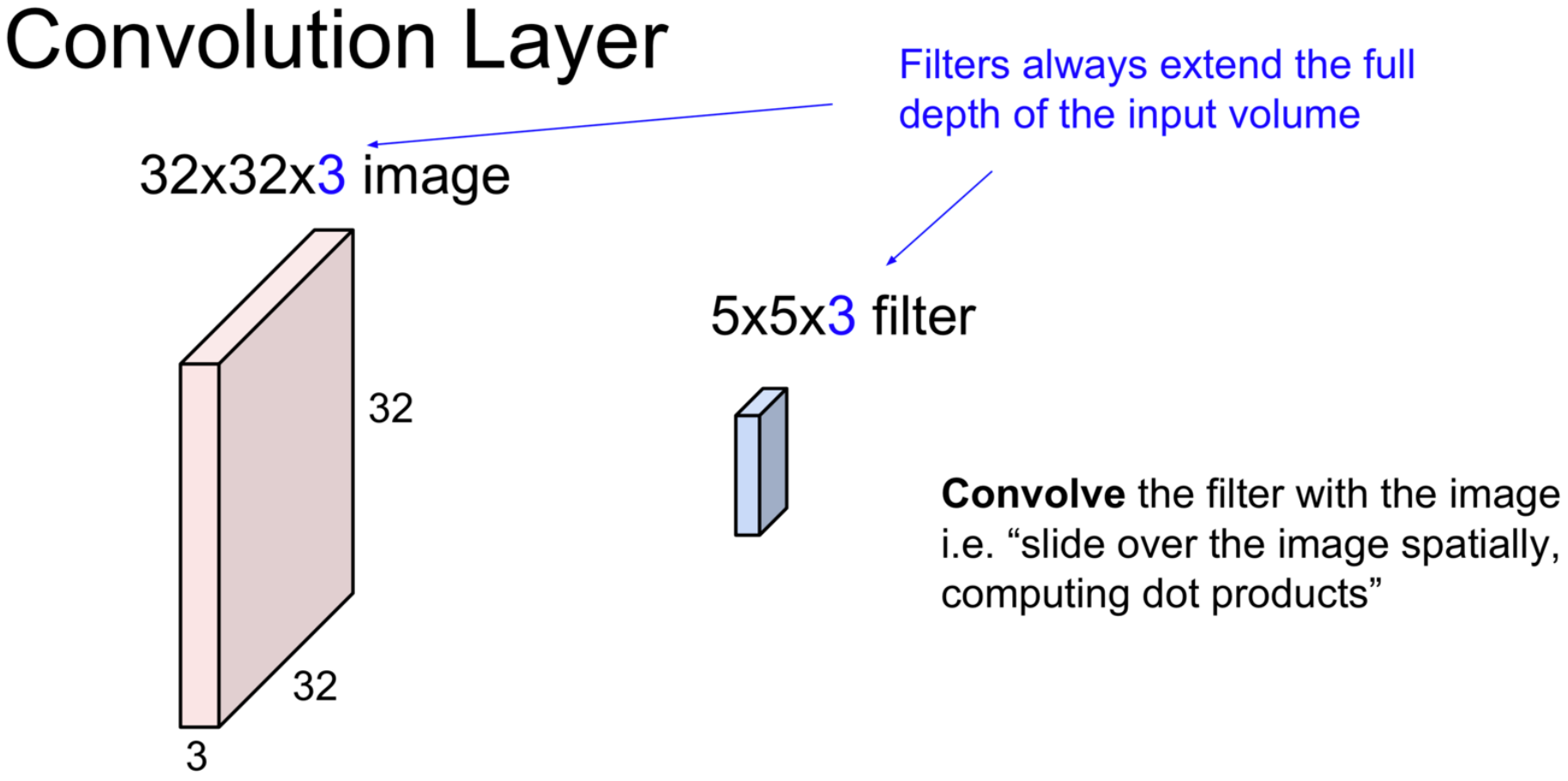
32x32x3 image -> stretch to 3072 x 1



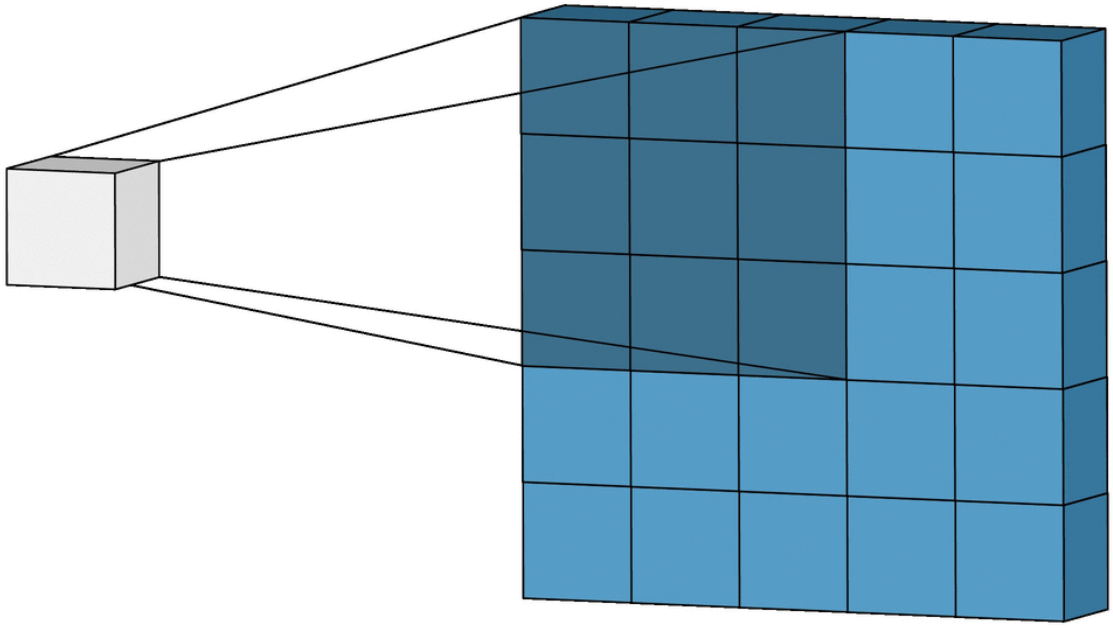
Fully Connected Layer



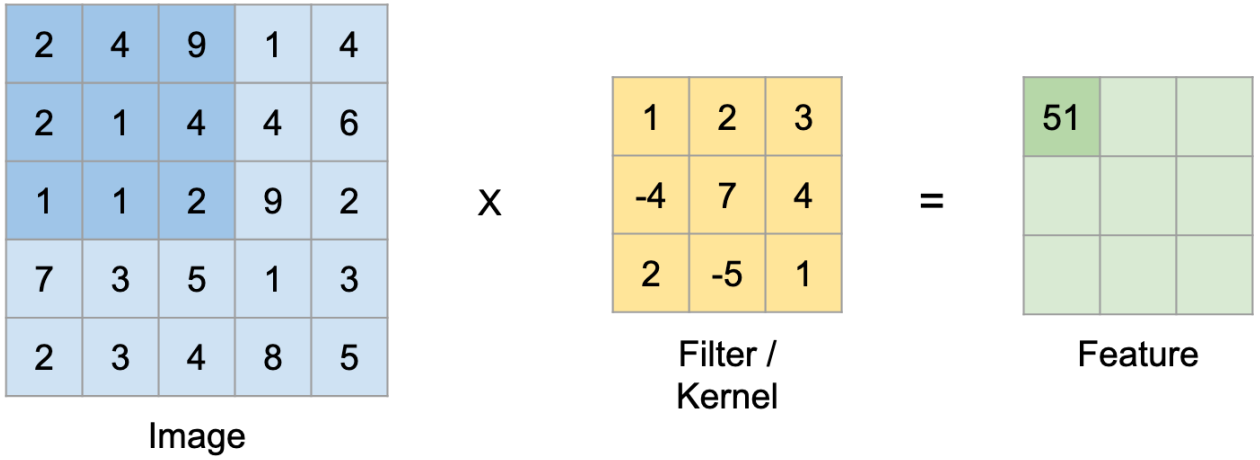
Convolutional layer

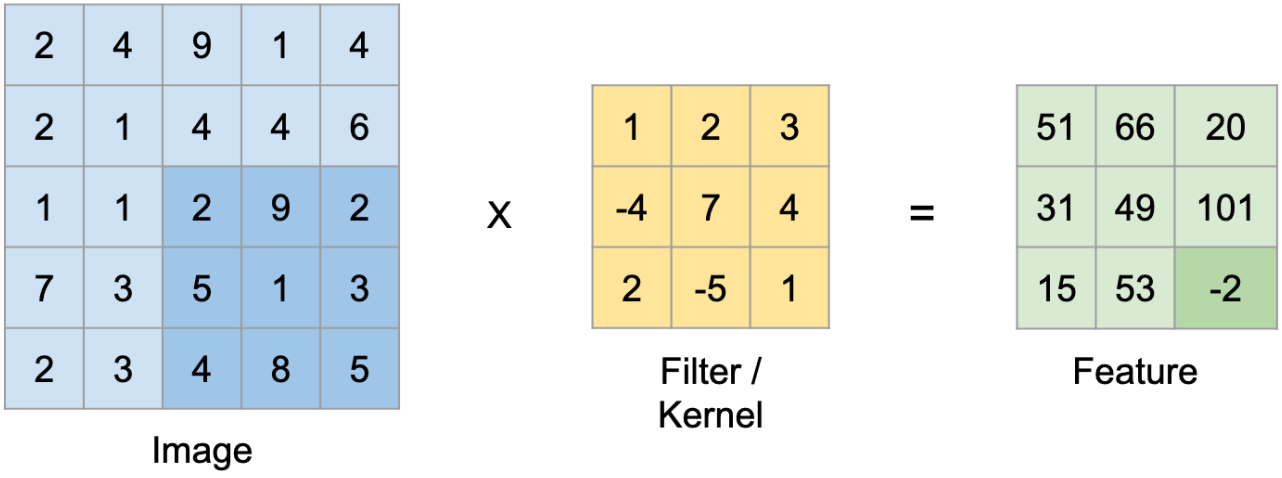


Convolutional layer

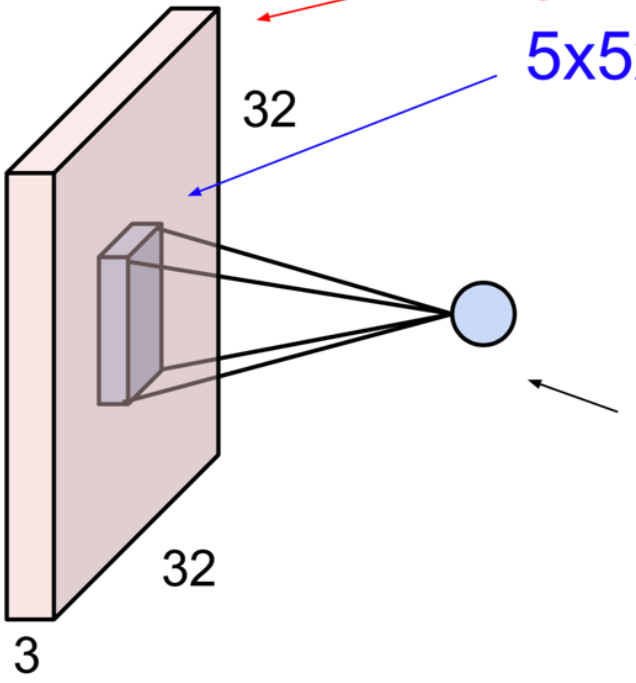


Convolutional layer





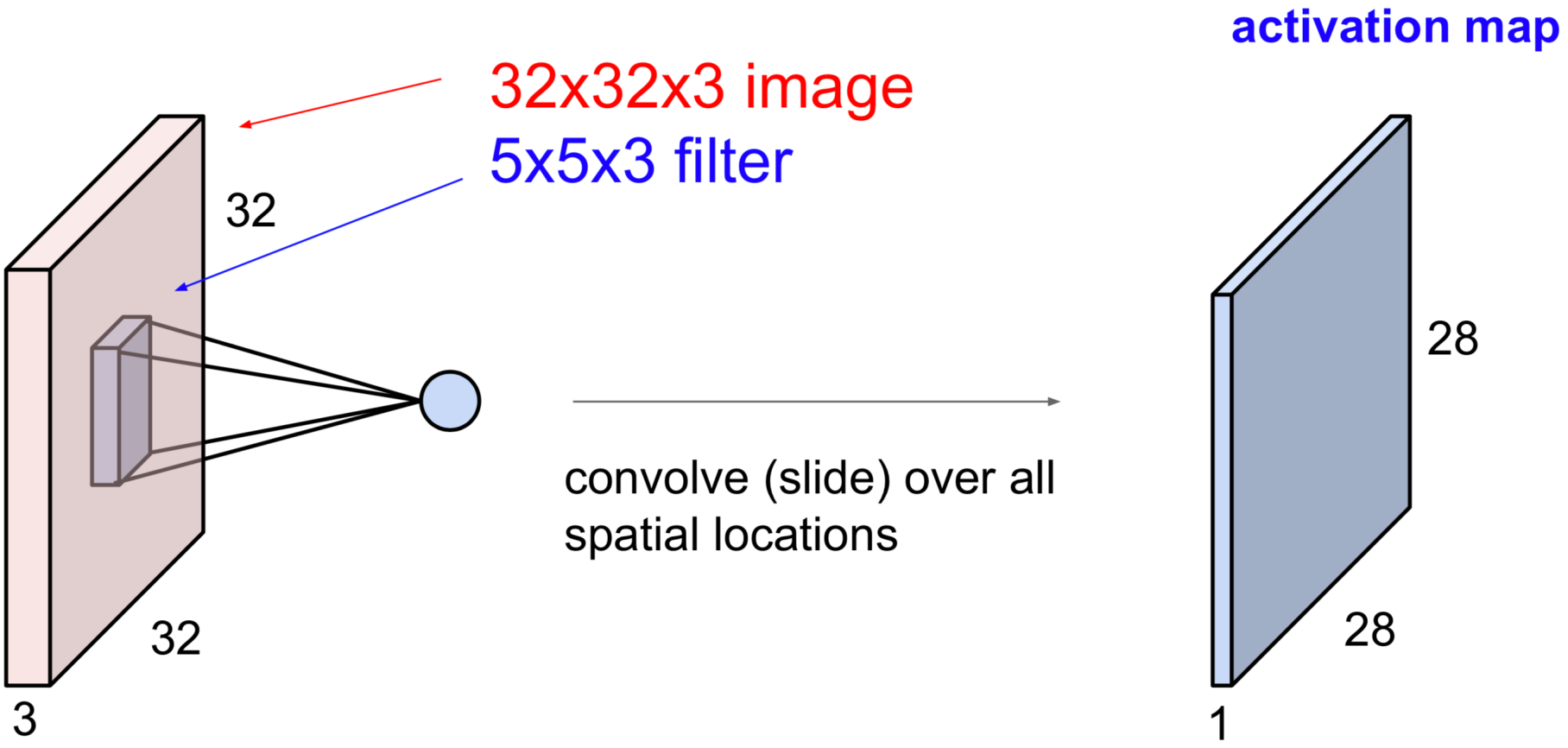
Convolutional layer



1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

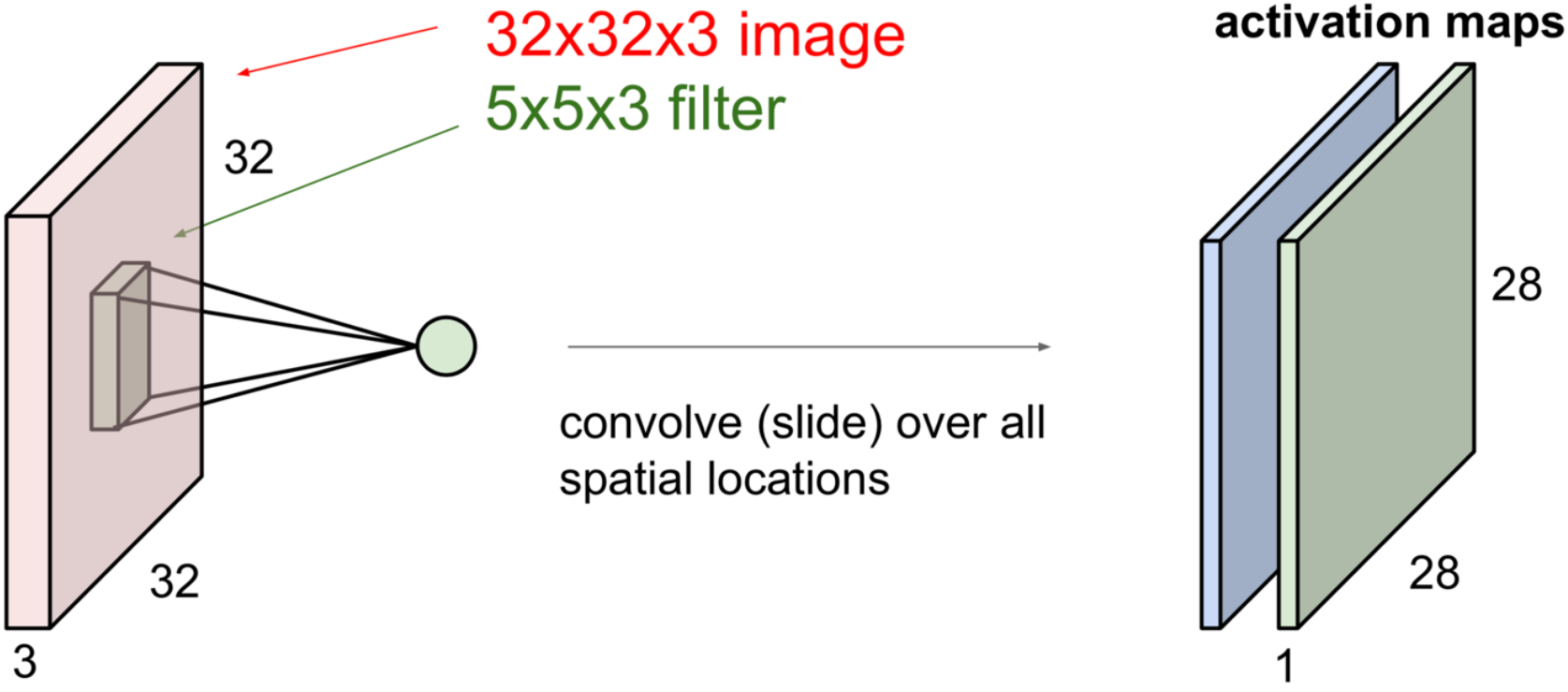
Convolutional layer



Convolutional layer

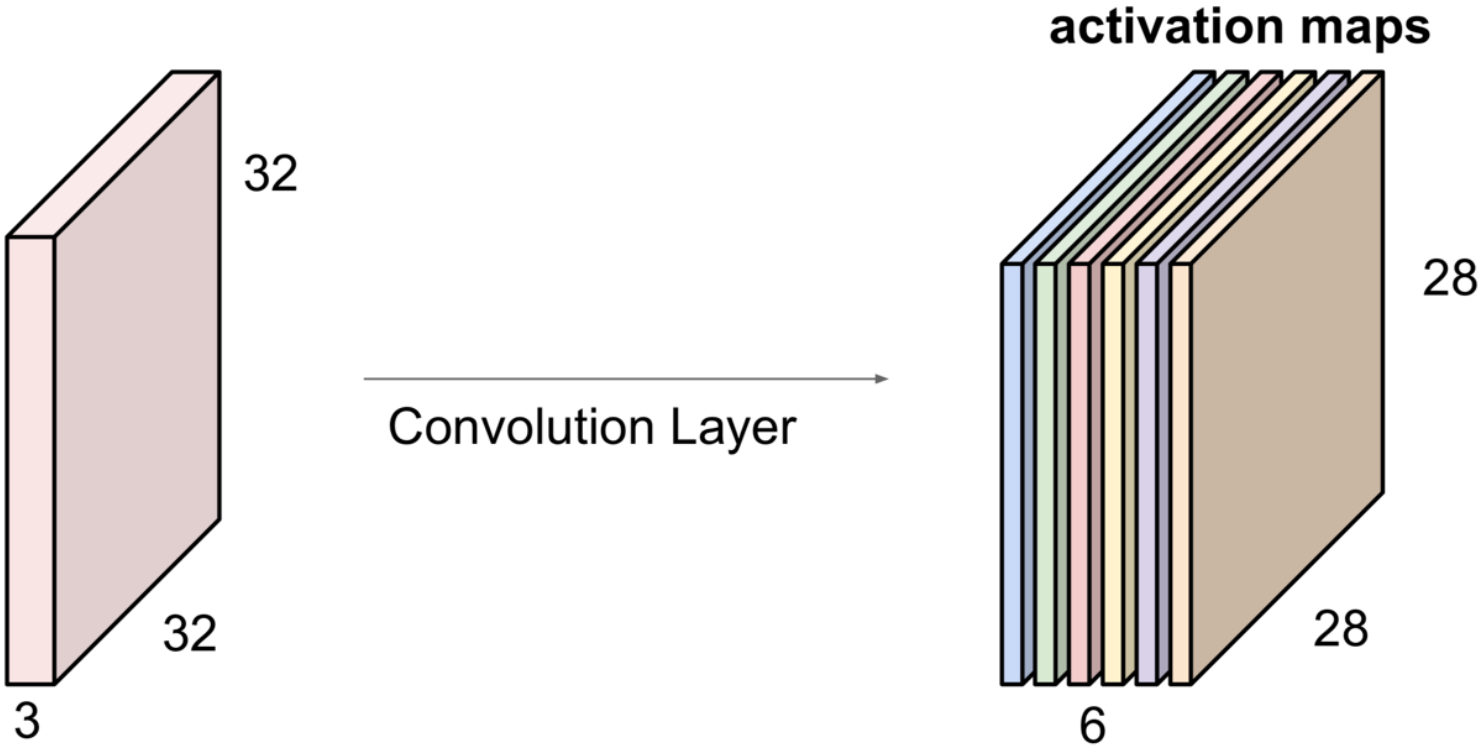
Convolution Layer

consider a second, green filter



Convolutional layer

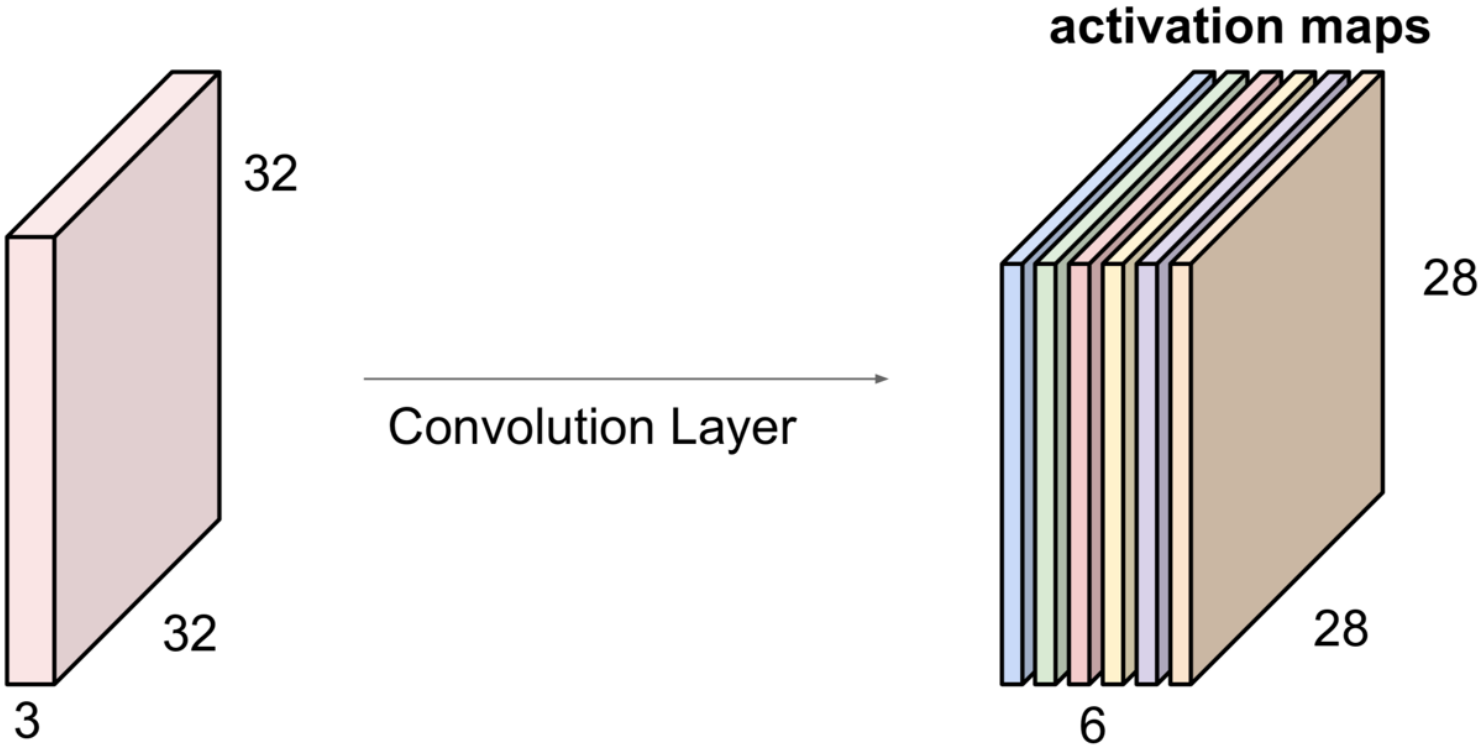
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a "new image" of size 28x28x6!

Convolutional layer

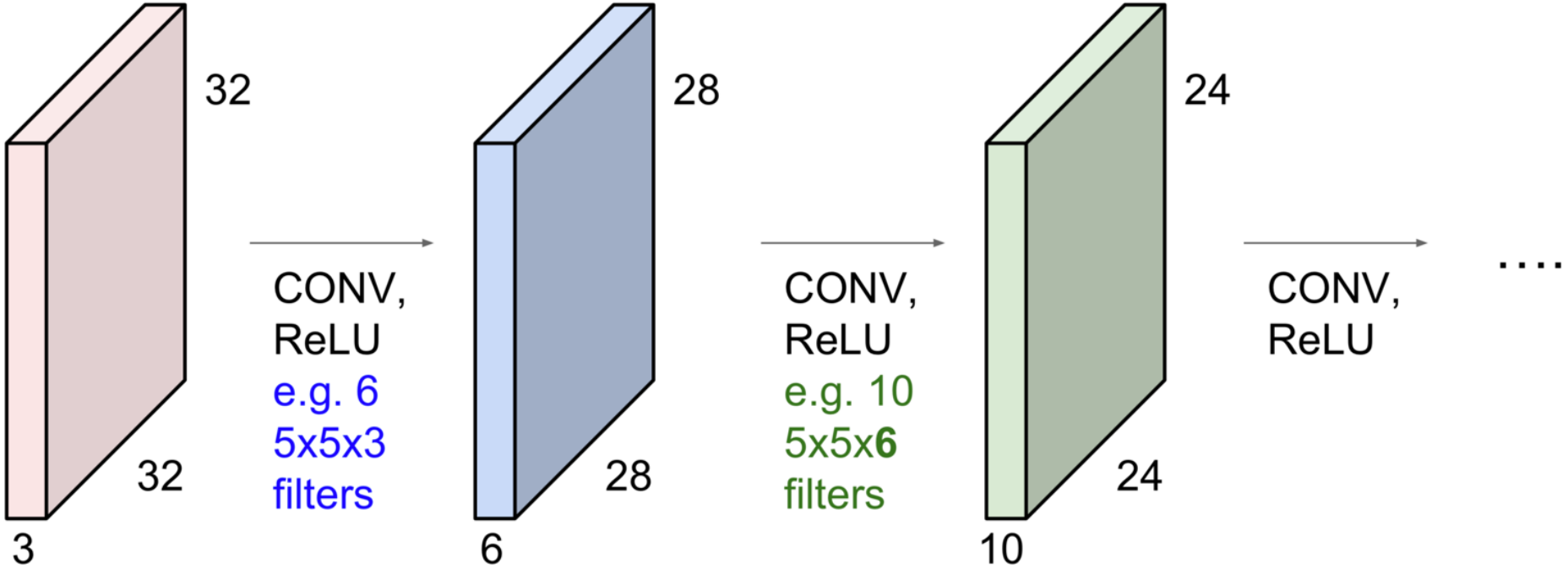
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



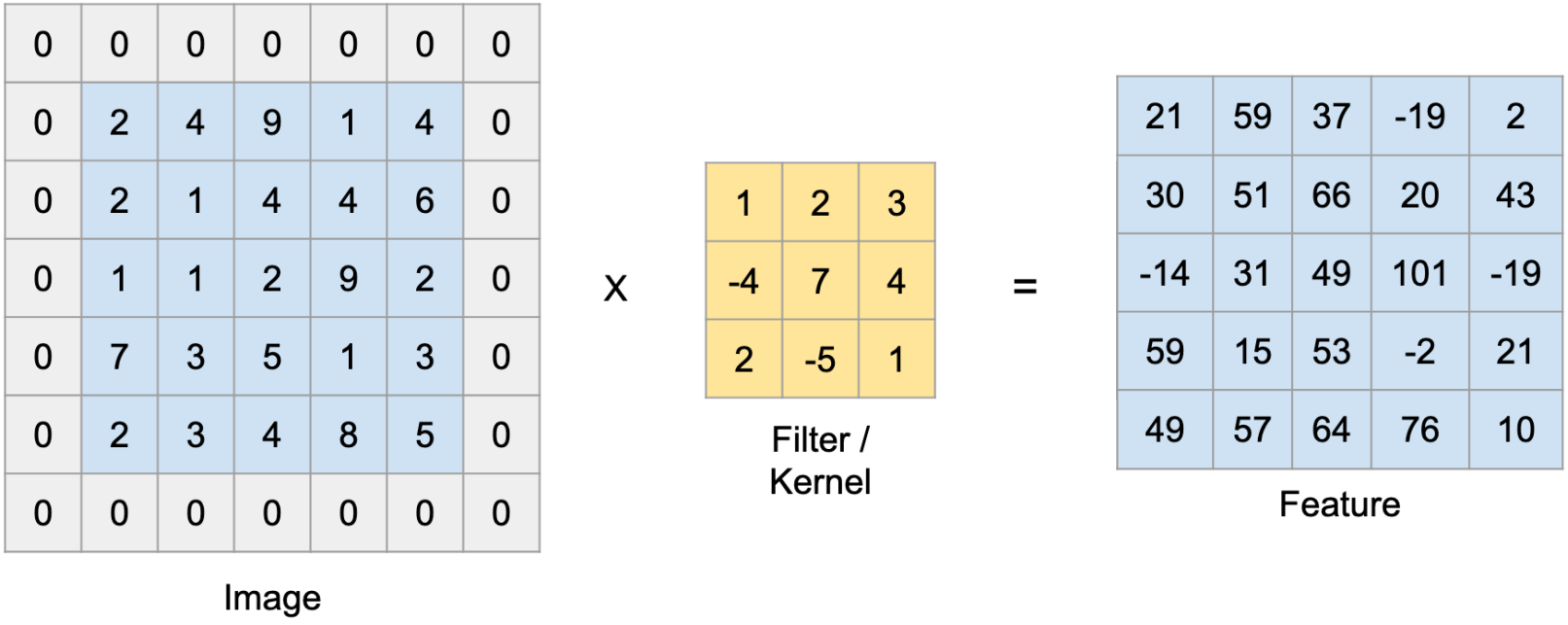
We stack these up to get a "new image" of size 28x28x6!

Convolutional layer

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



Convolutional layer



- Input 5x5 with padding 1.
- Filter with kernel size 3x3 and stride 1.

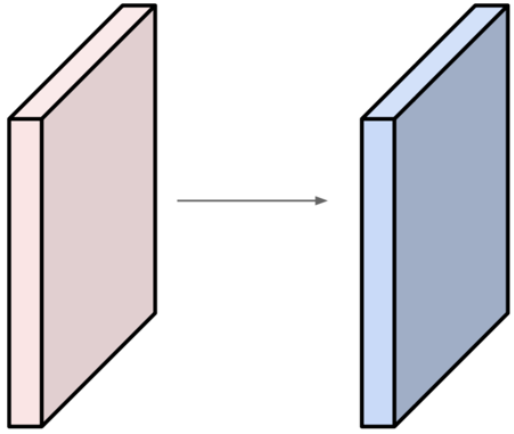
→ Output is 5x5.

Convolutional layer

Examples time:

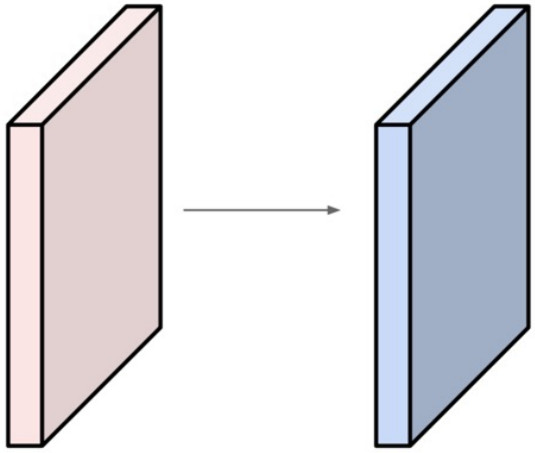
Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

Output volume size: ?



Convolutional layer

Examples time:

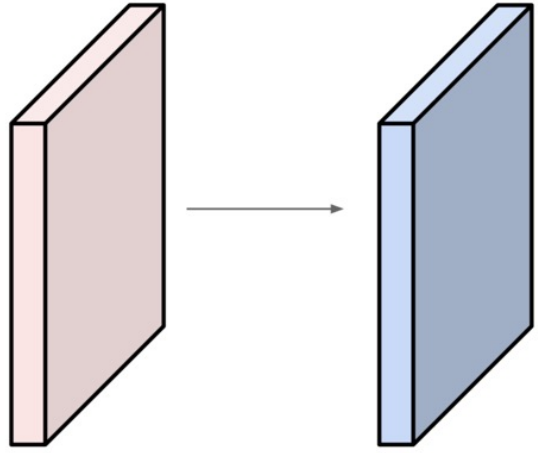


Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

Output volume size:
(32+2*2-5)/1+1 = 32 spatially, so
32x32x10

Convolutional layer

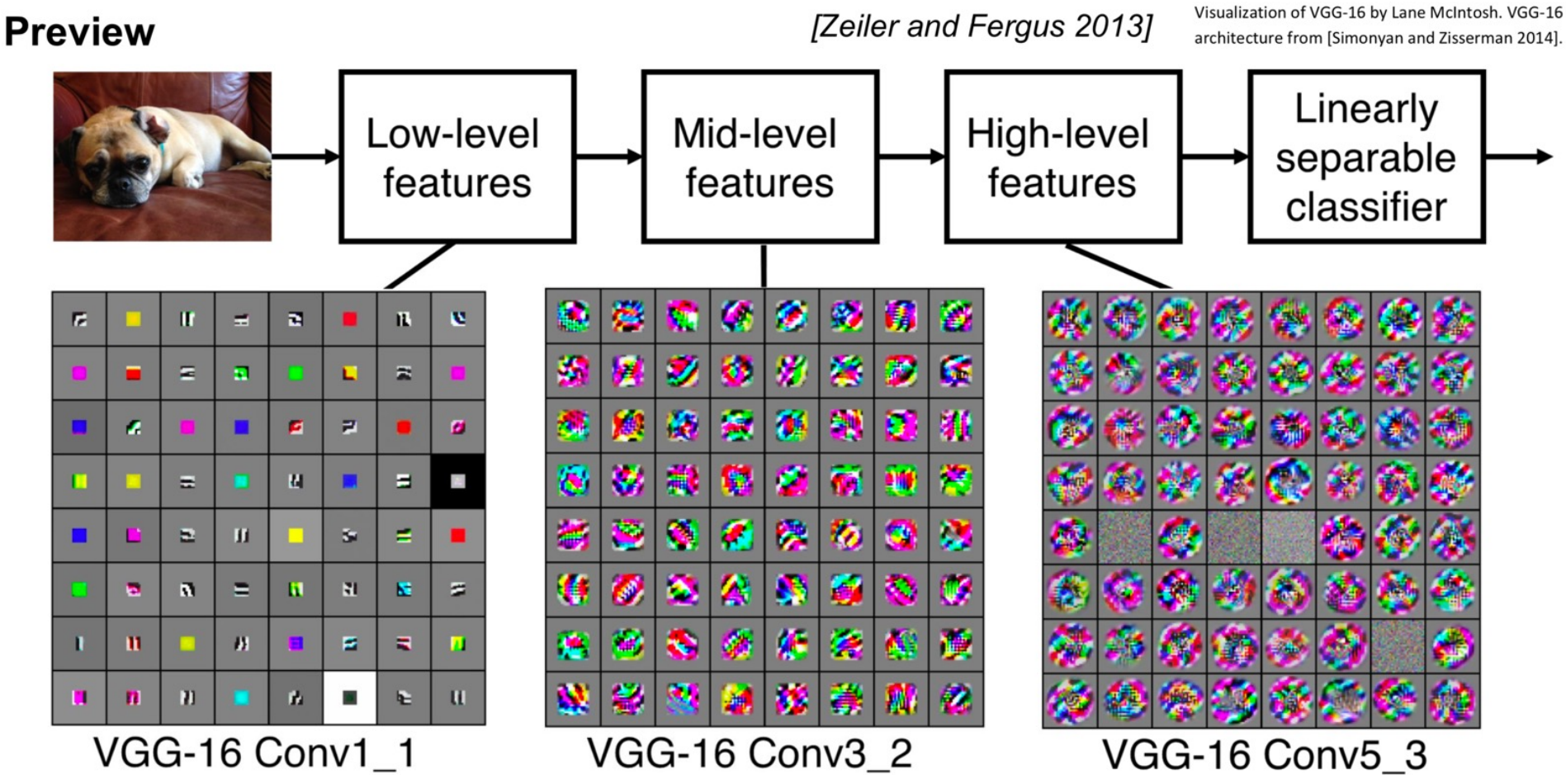
Examples time:



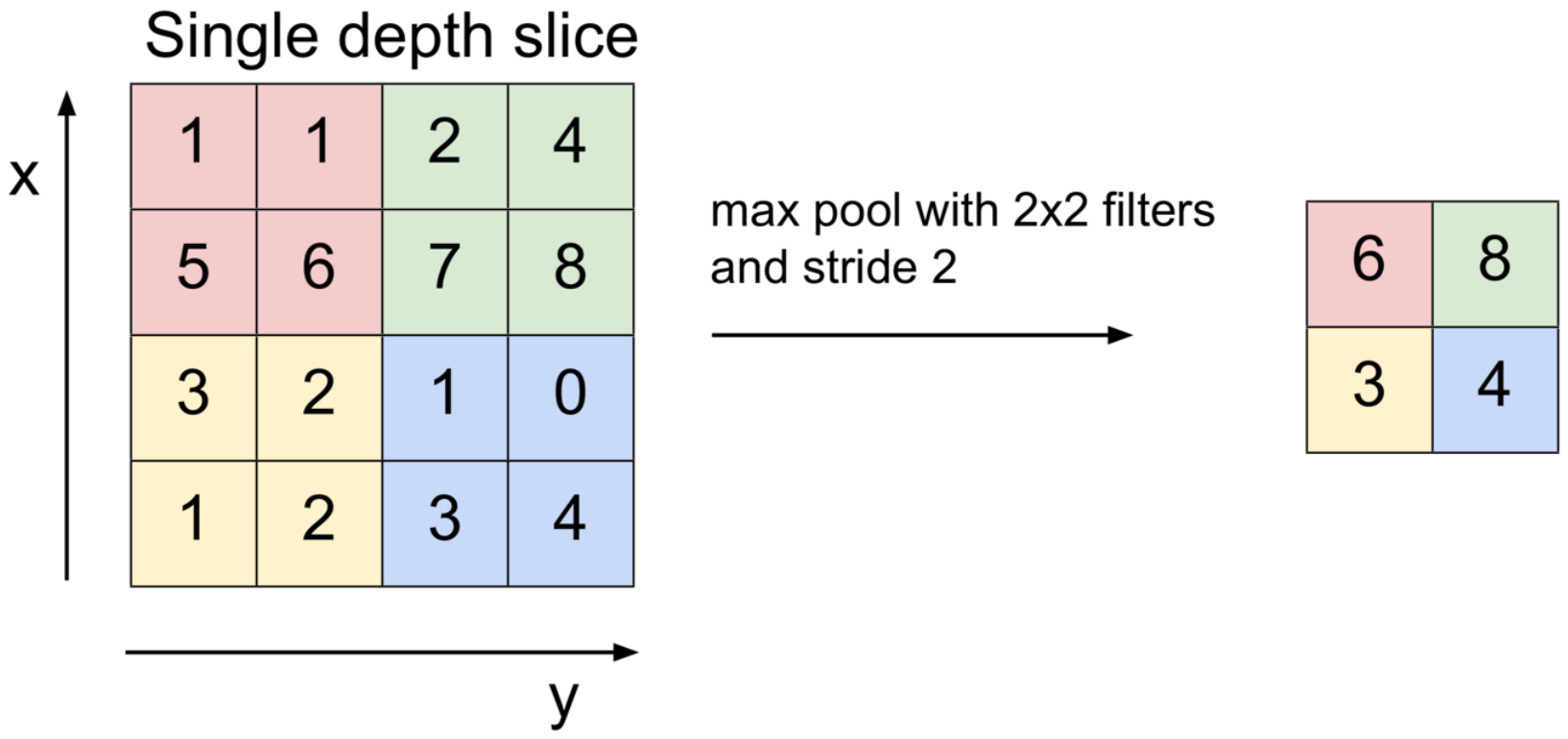
Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

Number of parameters in this layer?
each filter has 5*5*3 + 1 = 76 params (+1 for bias)
=> 76*10 = **760**

Convolutional layer

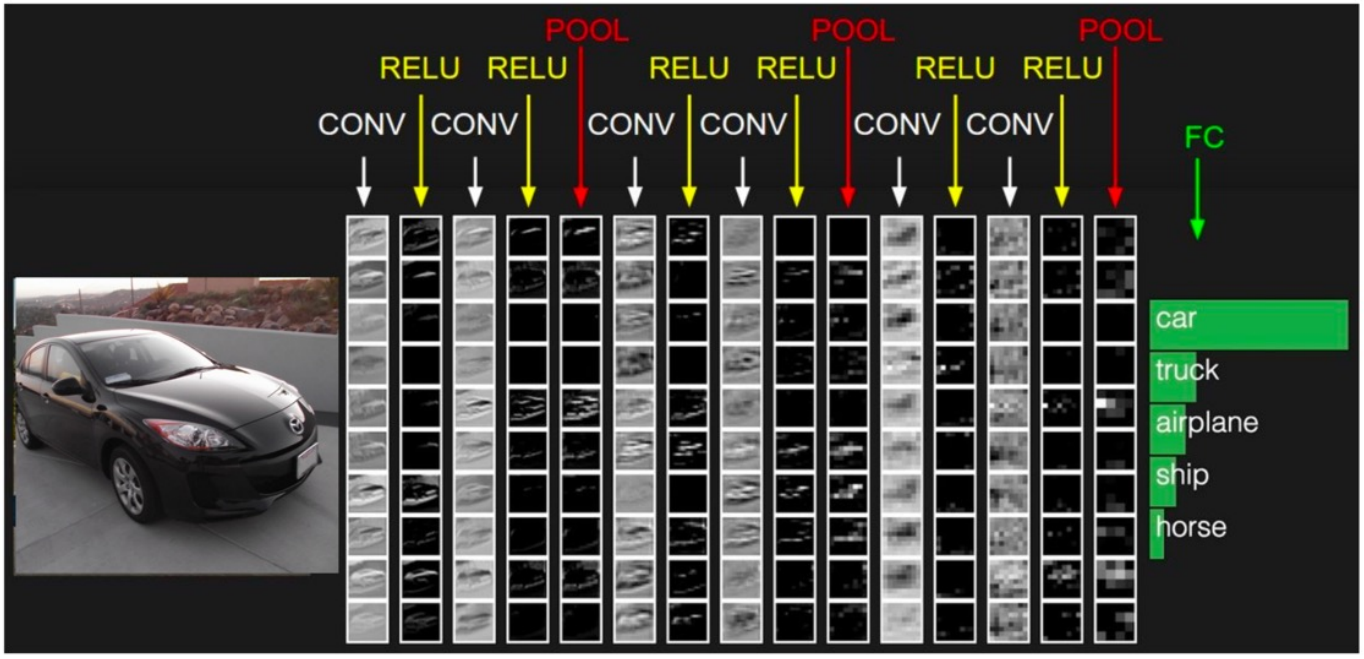


Pooling layer



Deep Neural network - An example structure

- Contains neurons that connect to the entire input volume, as in ordinary Neural Networks



Summary

- Introduction to deep learning and neural networks.
- Feature engineering in machine learning and deep learning.
- Fully connected layer.
- Convolutional layer.
- Activation function.
- Pooling layer.

Q&A

Thank you