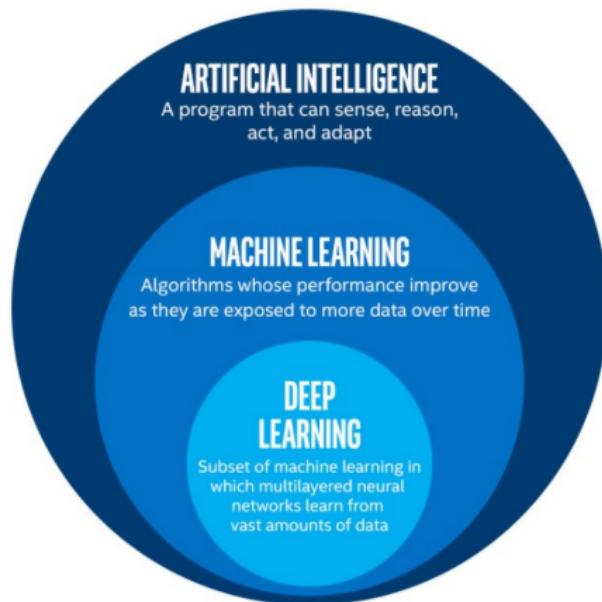


A gentle introduction to Deep Learning

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

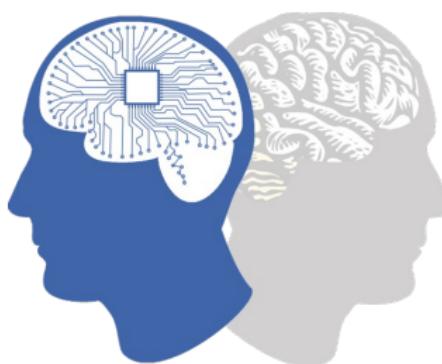
Space
Exploration

Gaming

Robotics

Agriculture

E-commerce



AI applications – Economics

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

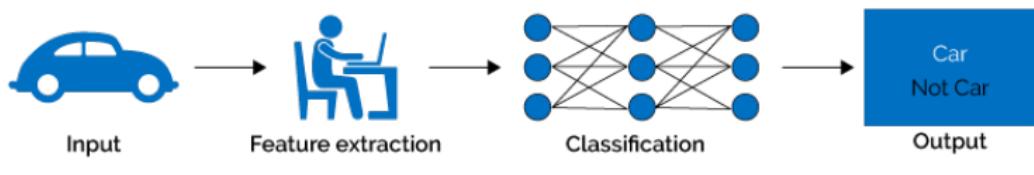
What is Deep Learning (DL) ?

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

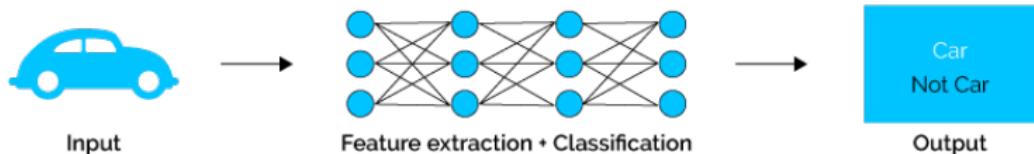
Deep learning algorithms attempt to learn (multiple levels of) representation by using a **hierarchy of multiple layers**

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

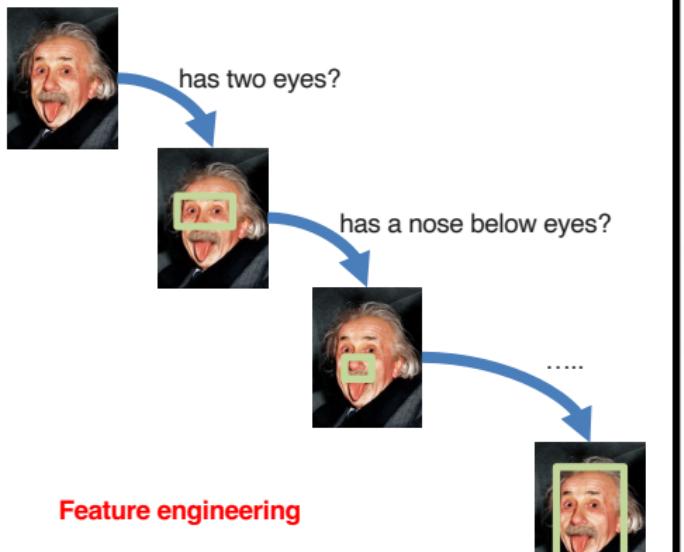
Machine Learning



Deep Learning

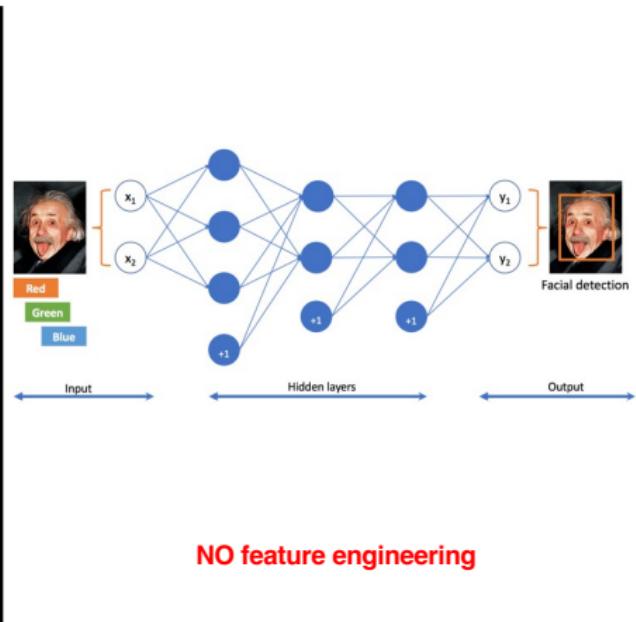


Traditional Machine Learning



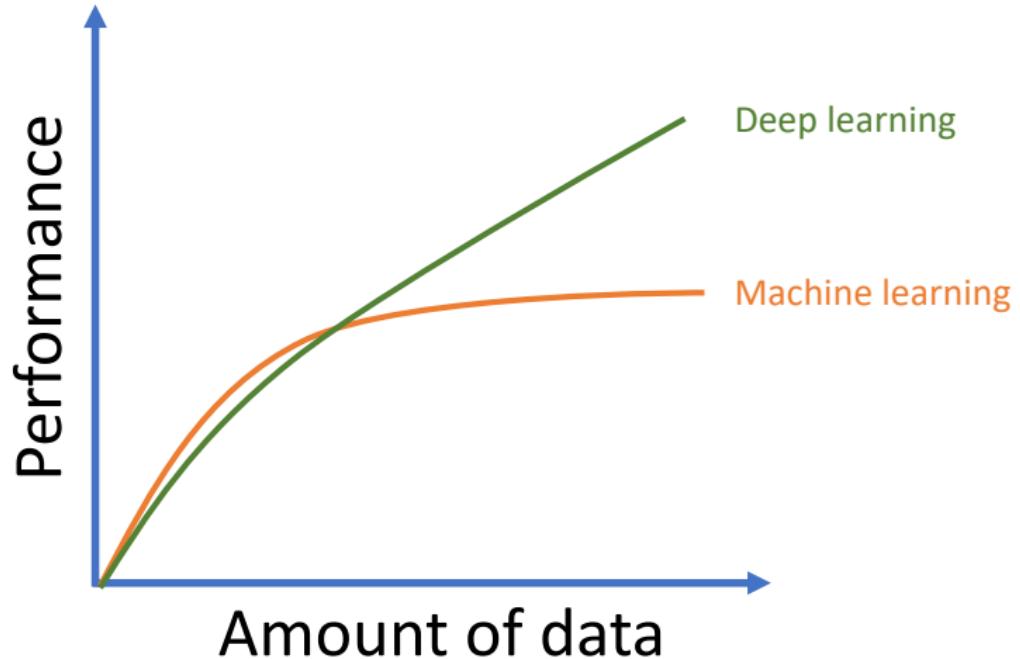
Feature engineering

Deep learning

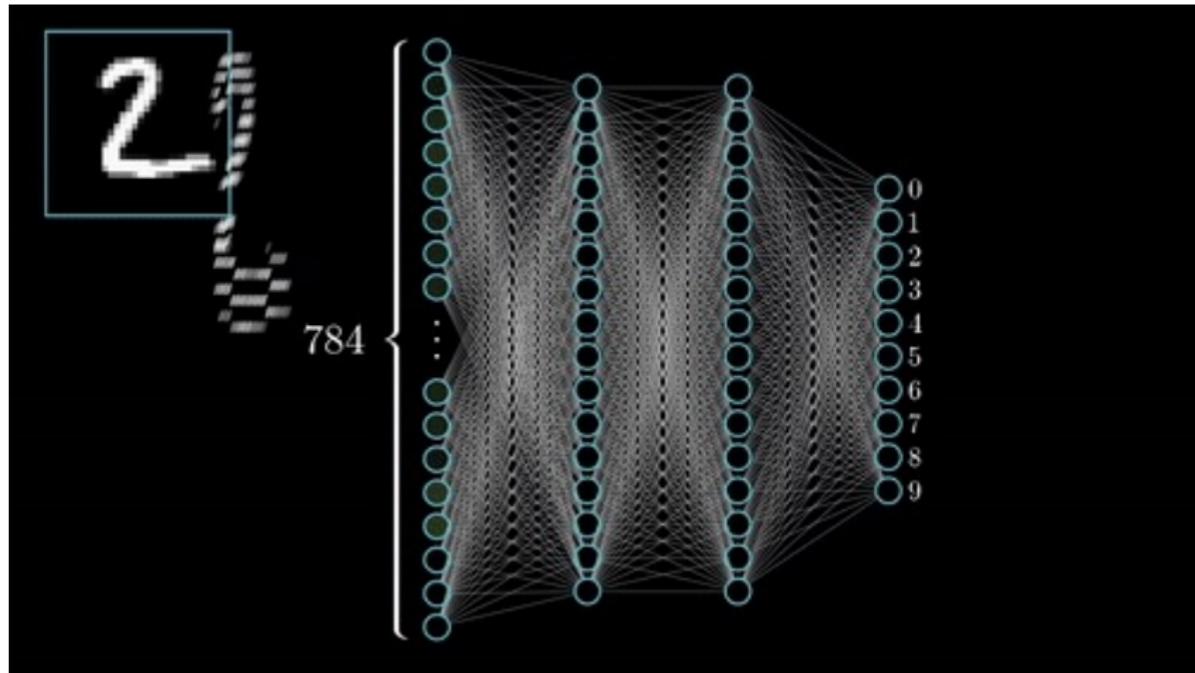


NO feature engineering

Why is DL useful?

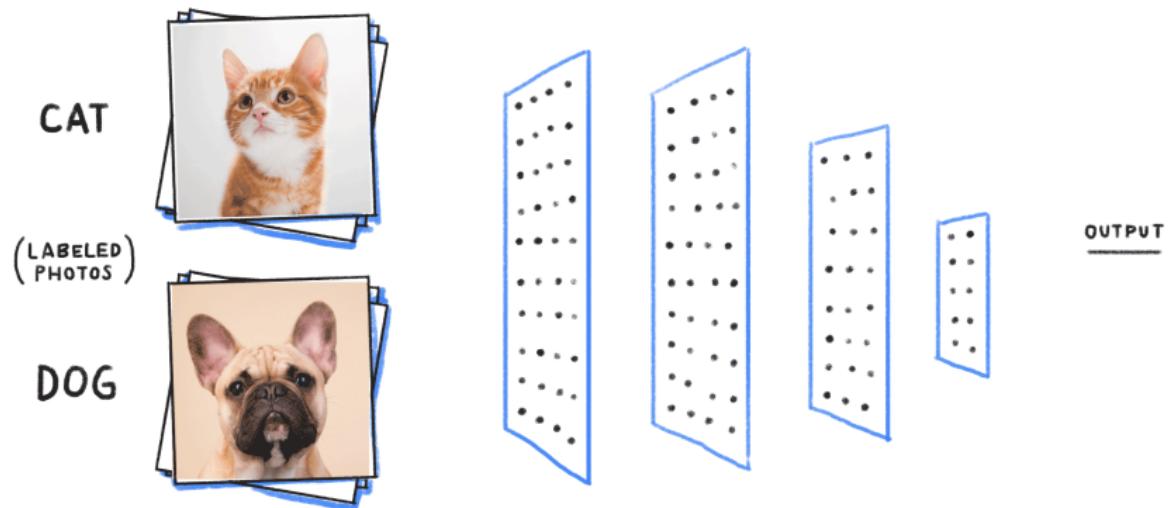


Deep Learning as a classifier



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

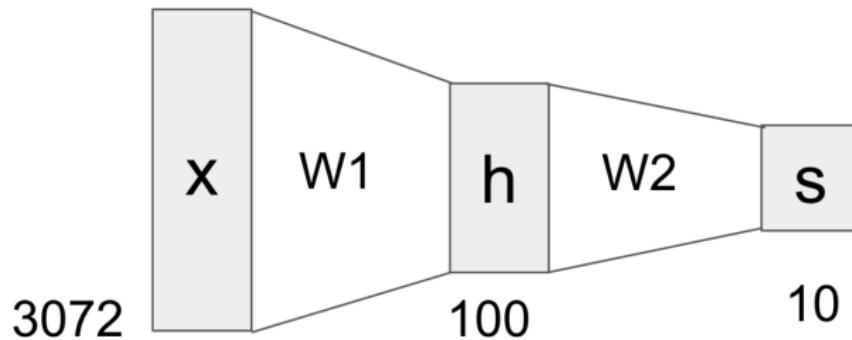
Neural Network

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

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

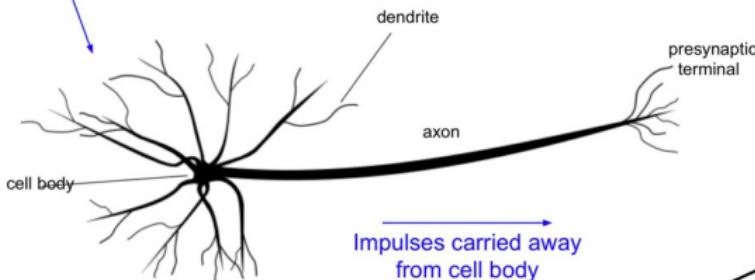
... and so on.

Deep learning layers as Lego blocks

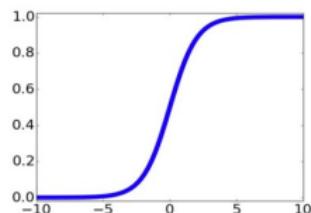


Neural Network

Impulses carried toward cell body



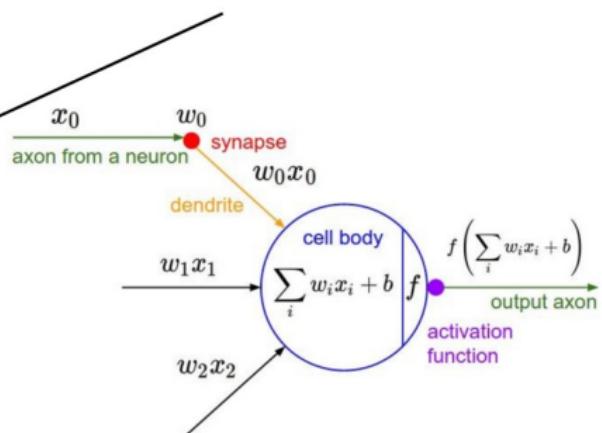
This image by Felipe Perucho
is licensed under CC-BY 3.0



sigmoid activation function

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

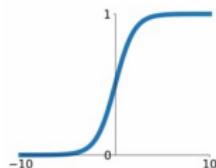
Impulses carried away from cell body



Activation function

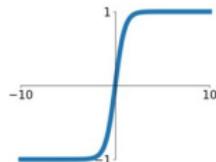
Sigmoid

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



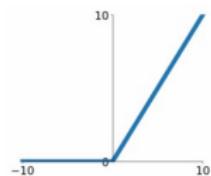
tanh

$$\tanh(x)$$



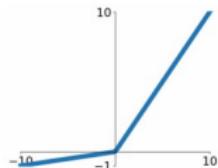
ReLU

$$\max(0, x)$$



Leaky ReLU

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

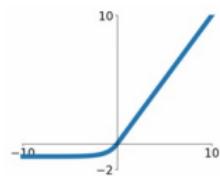


Maxout

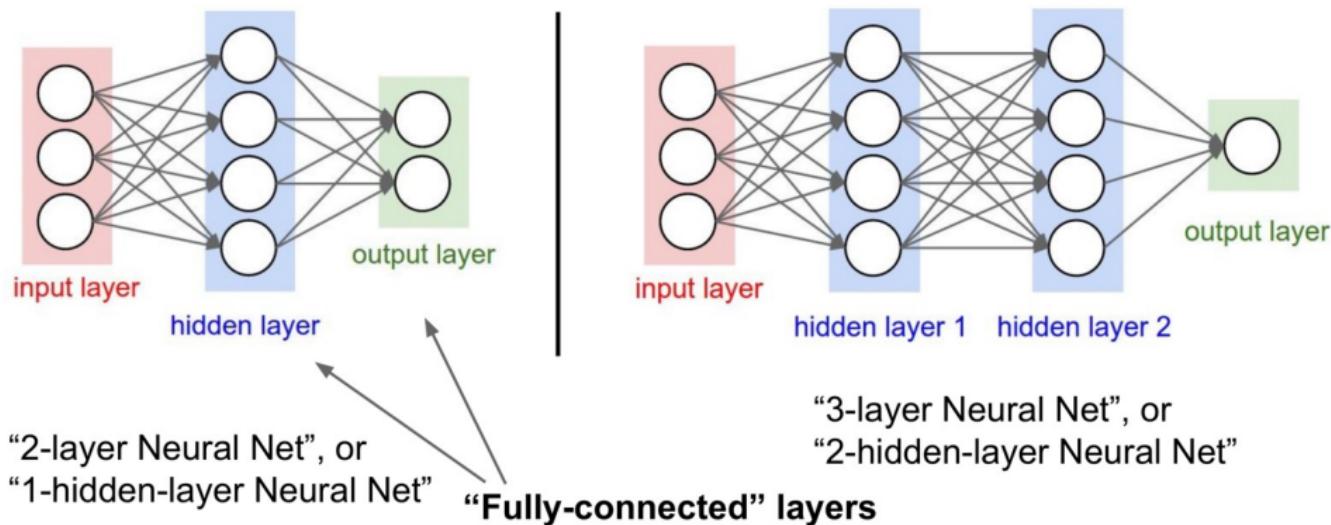
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

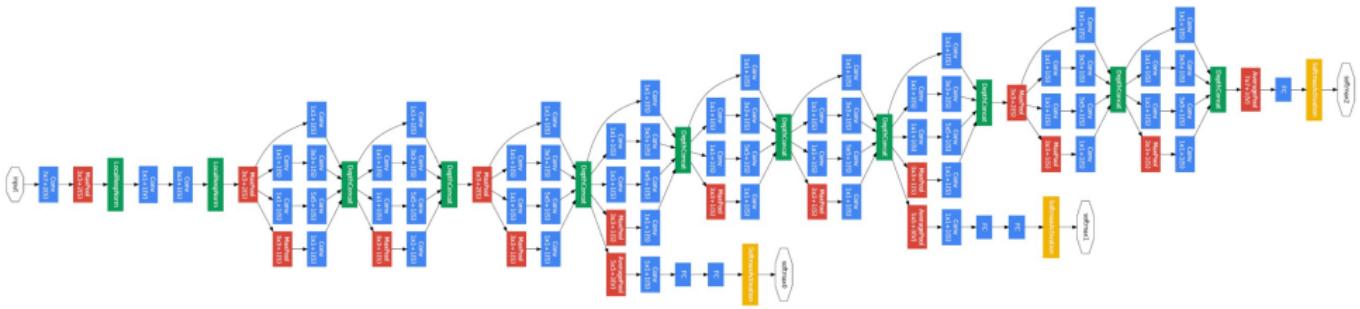
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Neural Network Architectures

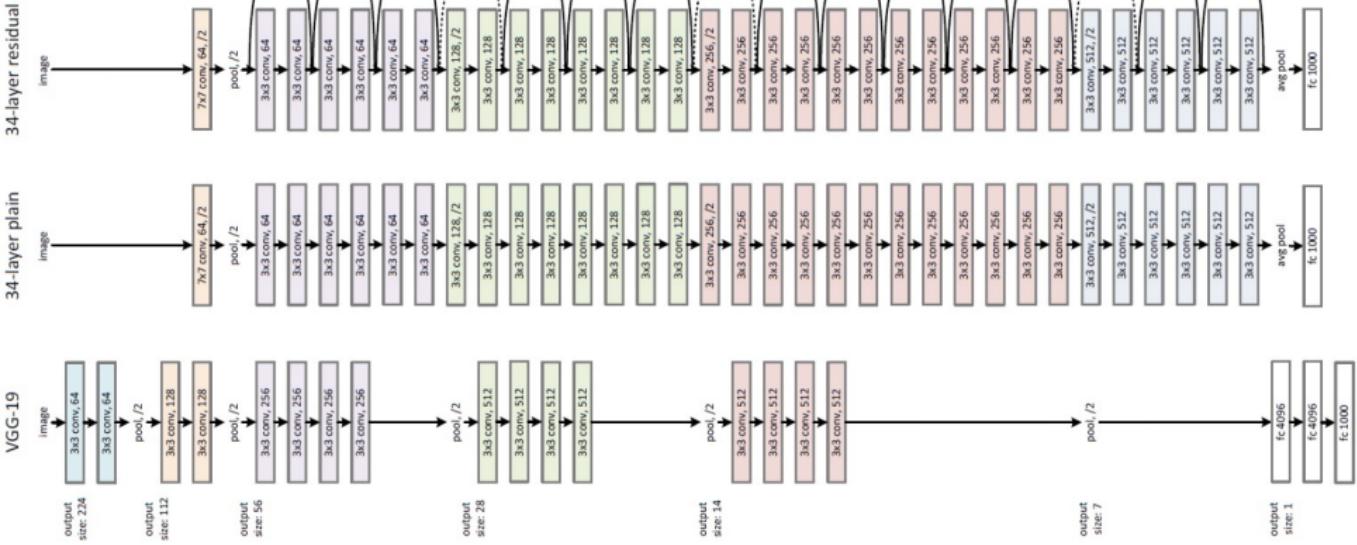


Neural Network Architectures



GoogLe Net (Inception v1)

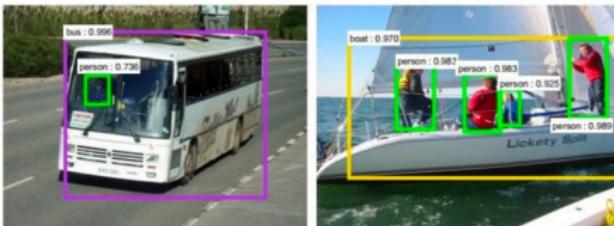
Neural Network Architectures



ResNet

Convolution Neural Network (ConvNet)

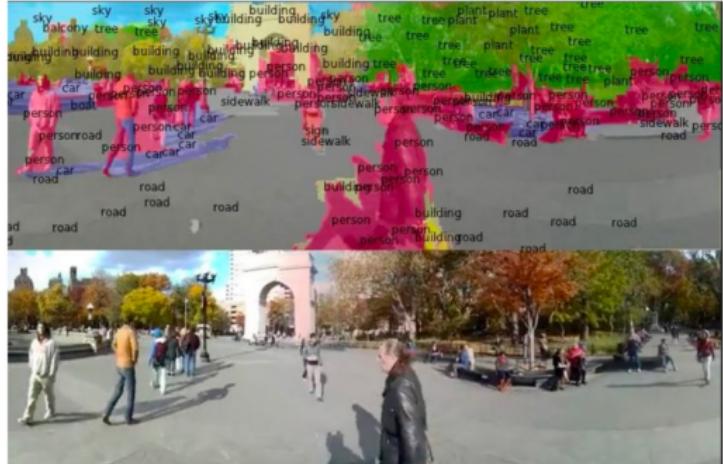
Detection



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

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

Segmentation

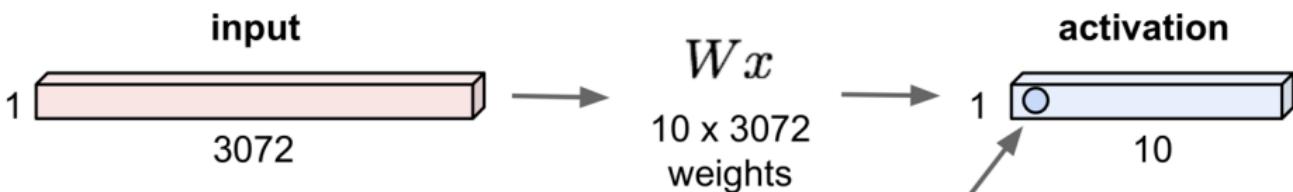


Figures copyright Clement Farabet, 2012.
Reproduced with permission.

[Farabet et al., 2012]

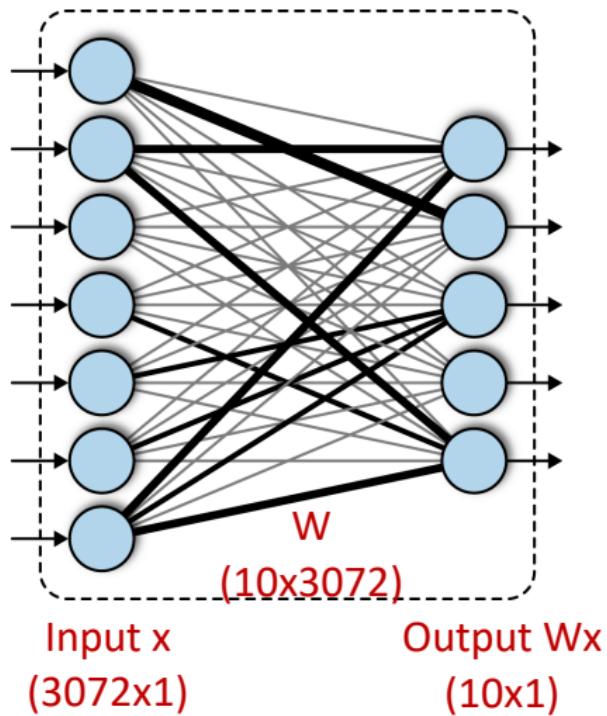
Fully Connected Layer

32x32x3 image -> stretch to 3072 x 1



1 number:
the result of taking a dot product
between a row of W and the input
(a 3072-dimensional dot product)

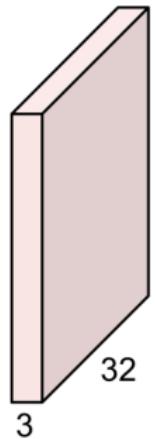
Fully Connected Layer



Convolutional layer

Convolution Layer

32x32x3 image



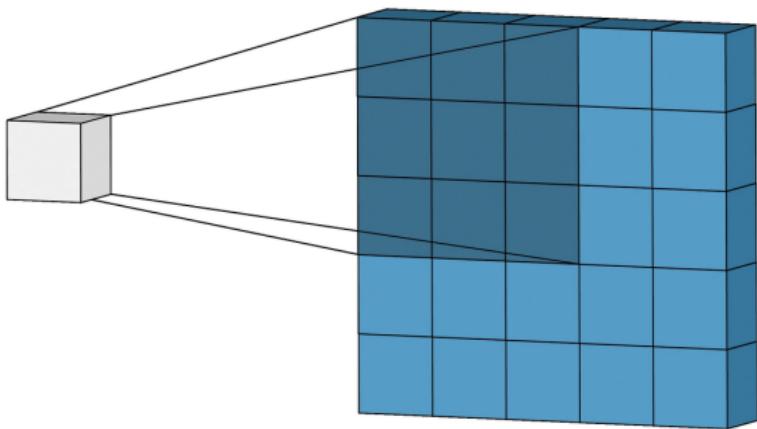
5x5x3 filter



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



Convolutional layer

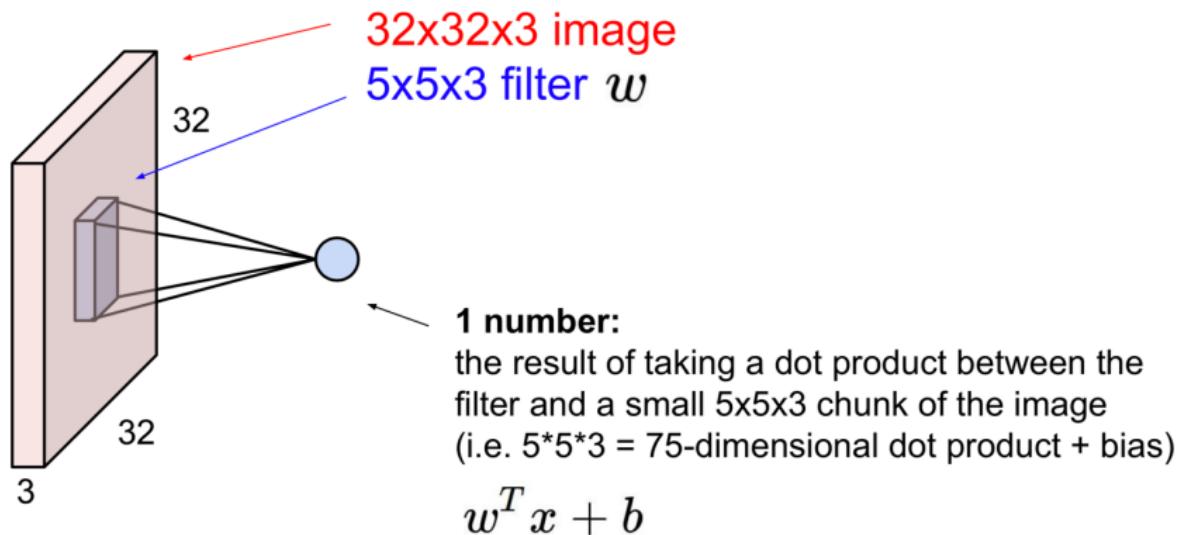
$$\begin{matrix} 2 & 4 & 9 & 1 & 4 \\ 2 & 1 & 4 & 4 & 6 \\ 1 & 1 & 2 & 9 & 2 \\ 7 & 3 & 5 & 1 & 3 \\ 2 & 3 & 4 & 8 & 5 \end{matrix} \times \begin{matrix} 1 & 2 & 3 \\ -4 & 7 & 4 \\ 2 & -5 & 1 \end{matrix} = \begin{matrix} 51 & & \\ & & \\ & & \end{matrix}$$

Image Filter / Kernel Feature

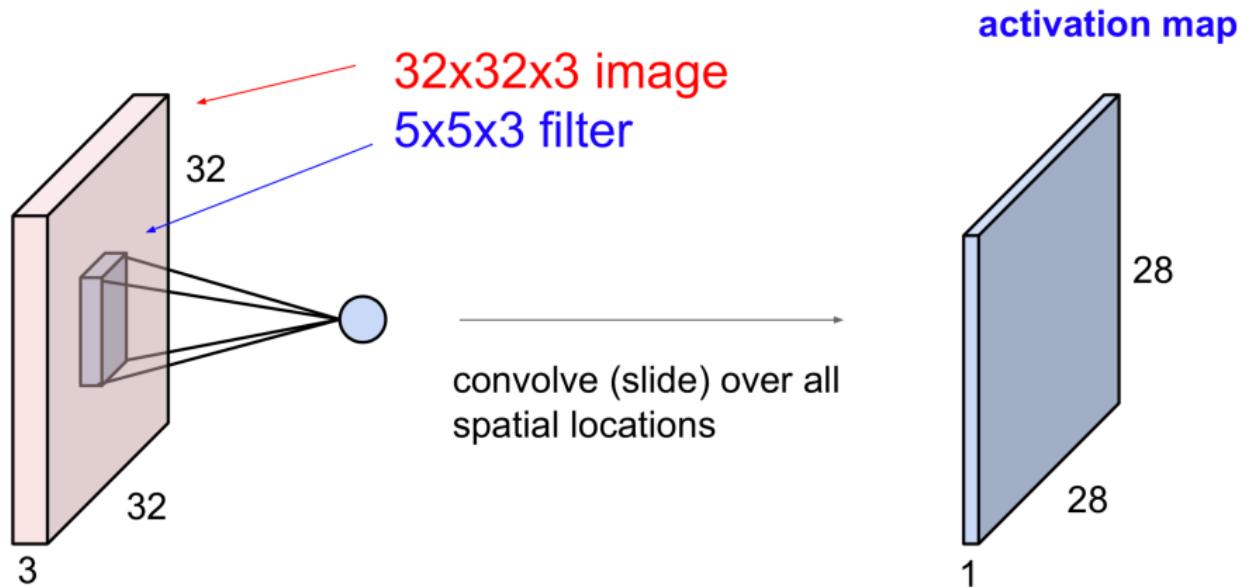
$$\begin{matrix} 2 & 4 & 9 & 1 & 4 \\ 2 & 1 & 4 & 4 & 6 \\ 1 & 1 & 2 & 9 & 2 \\ 7 & 3 & 5 & 1 & 3 \\ 2 & 3 & 4 & 8 & 5 \end{matrix} \times \begin{matrix} 1 & 2 & 3 \\ -4 & 7 & 4 \\ 2 & -5 & 1 \end{matrix} = \begin{matrix} 51 & 66 & 20 \\ 31 & 49 & 101 \\ 15 & 53 & -2 \end{matrix}$$

Image Filter / Kernel Feature

Convolutional layer



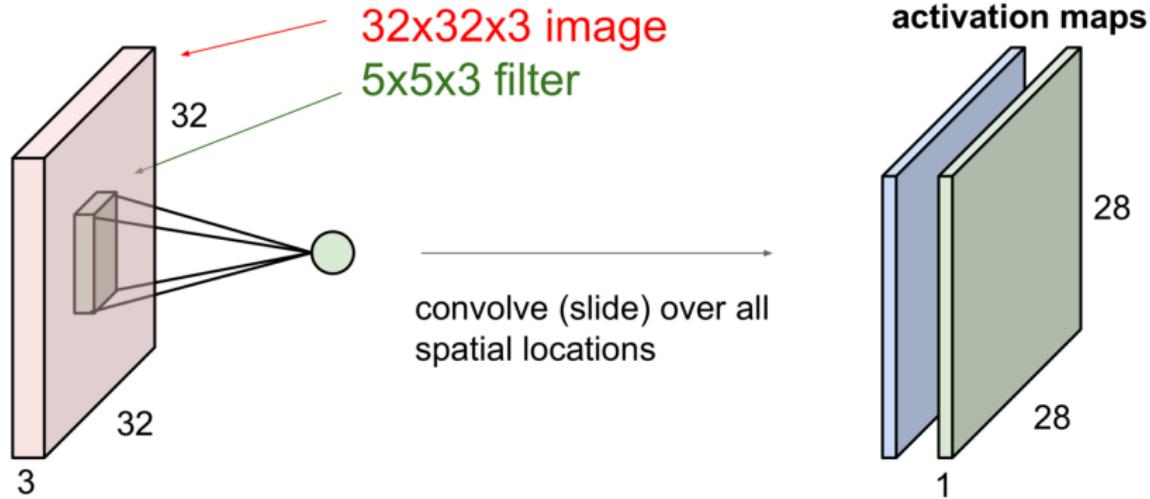
Convolutional layer



Convolutional layer

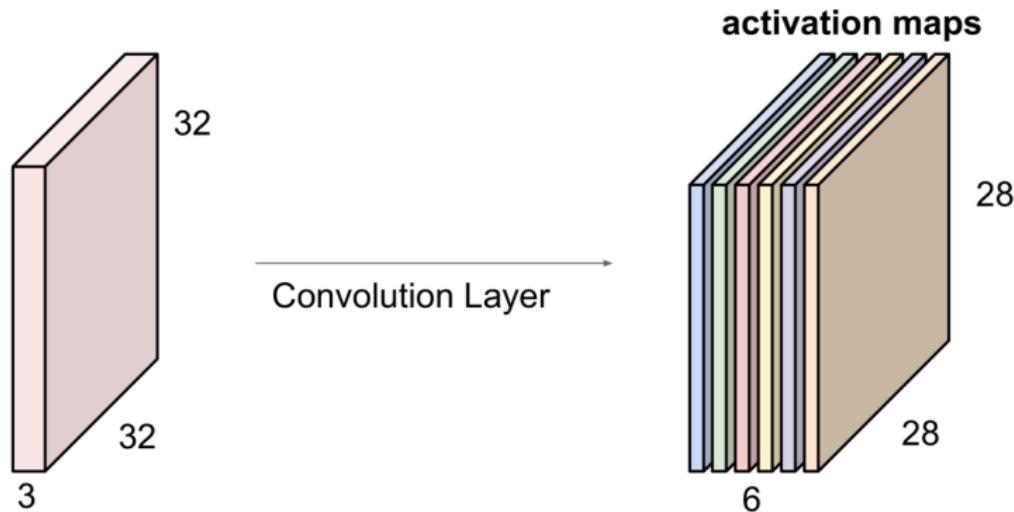
Convolution Layer

consider a second, green filter



Convolutional layer

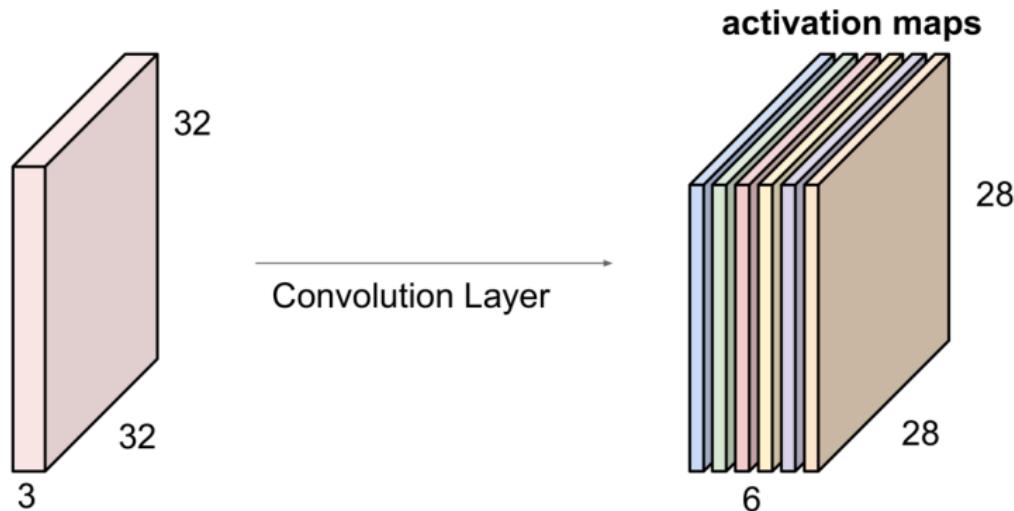
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a “new image” of size $28 \times 28 \times 6$!

Convolutional layer

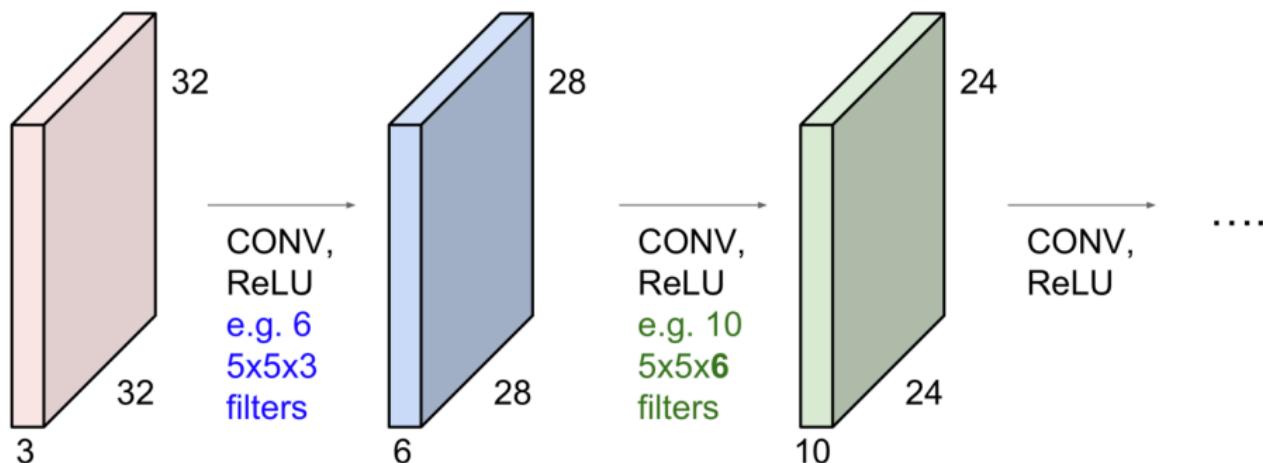
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a “new image” of size $28 \times 28 \times 6$!

Convolutional layer

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



Convolutional layer

$$\begin{array}{|ccccccc|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 2 & 4 & 9 & 1 & 4 & 0 \\ \hline 0 & 2 & 1 & 4 & 4 & 6 & 0 \\ \hline 0 & 1 & 1 & 2 & 9 & 2 & 0 \\ \hline 0 & 7 & 3 & 5 & 1 & 3 & 0 \\ \hline 0 & 2 & 3 & 4 & 8 & 5 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \times \begin{array}{|ccc|} \hline 1 & 2 & 3 \\ \hline -4 & 7 & 4 \\ \hline 2 & -5 & 1 \\ \hline \end{array} = \begin{array}{|ccccc|} \hline 21 & 59 & 37 & -19 & 2 \\ \hline 30 & 51 & 66 & 20 & 43 \\ \hline -14 & 31 & 49 & 101 & -19 \\ \hline 59 & 15 & 53 & -2 & 21 \\ \hline 49 & 57 & 64 & 76 & 10 \\ \hline \end{array}$$

Filter / Kernel

Feature

Image

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

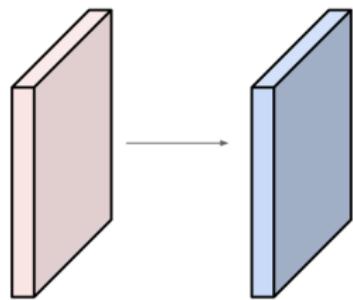
Convolutional layer

Examples time:

Input volume: **32x32x3**

10 5x5 filters with stride 1, pad 2

Output volume size: ?

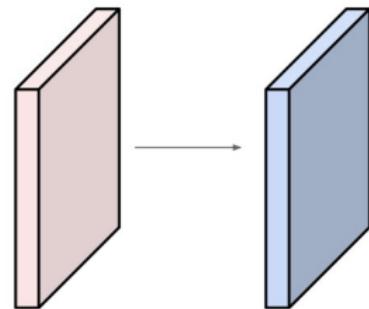


Convolutional layer

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

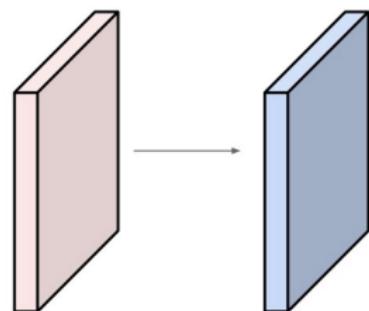
32x32x10

Convolutional layer

Examples time:

Input volume: **32x32x3**

10 **5x5** filters with stride 1, pad 2



Number of parameters in this layer?

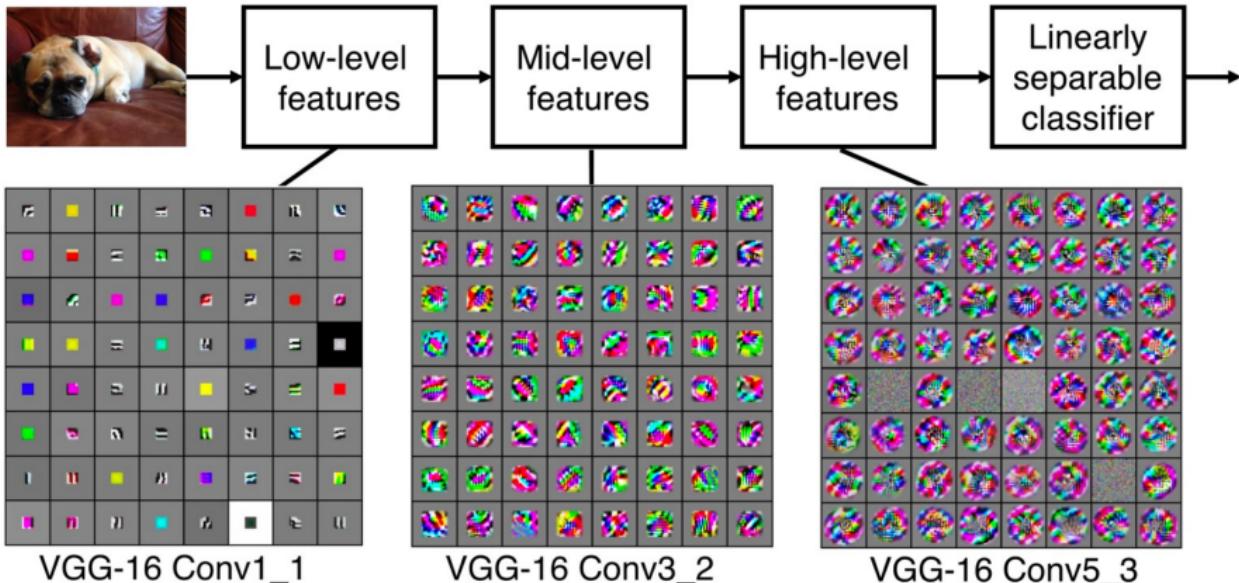
each filter has $5 \times 5 \times 3 + 1 = 76$ params (+1 for bias)
=> **76 * 10 = 760**

Convolutional layer

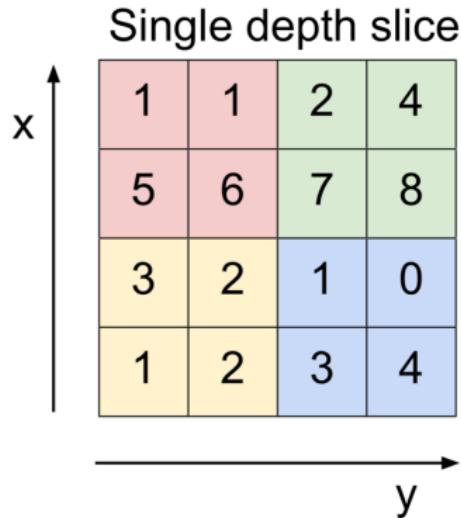
Preview

[Zeiler and Fergus 2013]

Visualization of VGG-16 by Lane McIntosh. VGG-16 architecture from [Simonyan and Zisserman 2014].



Pooling layer



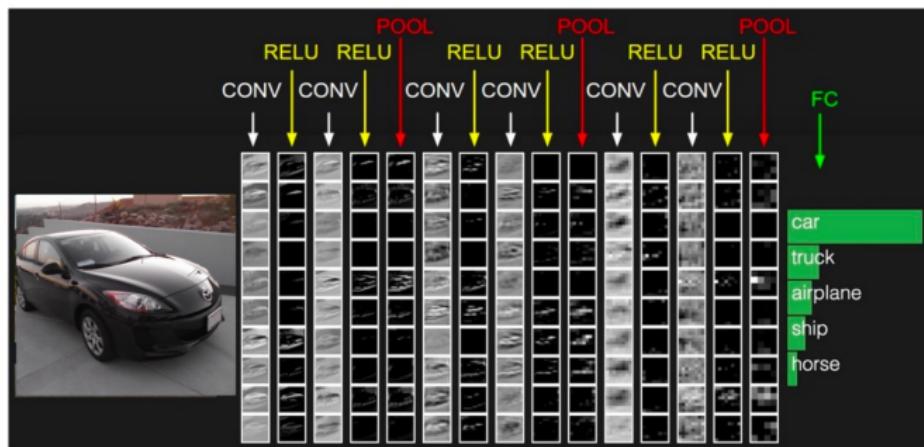
max pool with 2x2 filters
and stride 2

The output matrix is a 2x2 grid with two columns labeled x and two rows labeled y. The cells are colored in a repeating pattern: top row (y=1) has colors pink, light green; bottom row (y=2) has colors yellow, light blue. The matrix contains the following values:

6	8
3	4

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