[70240413 Machine Learning, Spring, 2019]

Deep Learning
(deep neural nets)

Hang Su
suhangss@mail.tsinghua.edu.cn
http://www.suhangss.me
State Key Lab of Intelligent Technology & Systems
Tsinghua University

March 27th, 2019

What is Deep Learning?

Artificial Intelligence

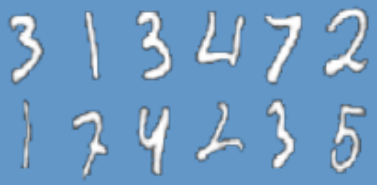
Any technique that enables computers to mimic human behavior

Machine Learning

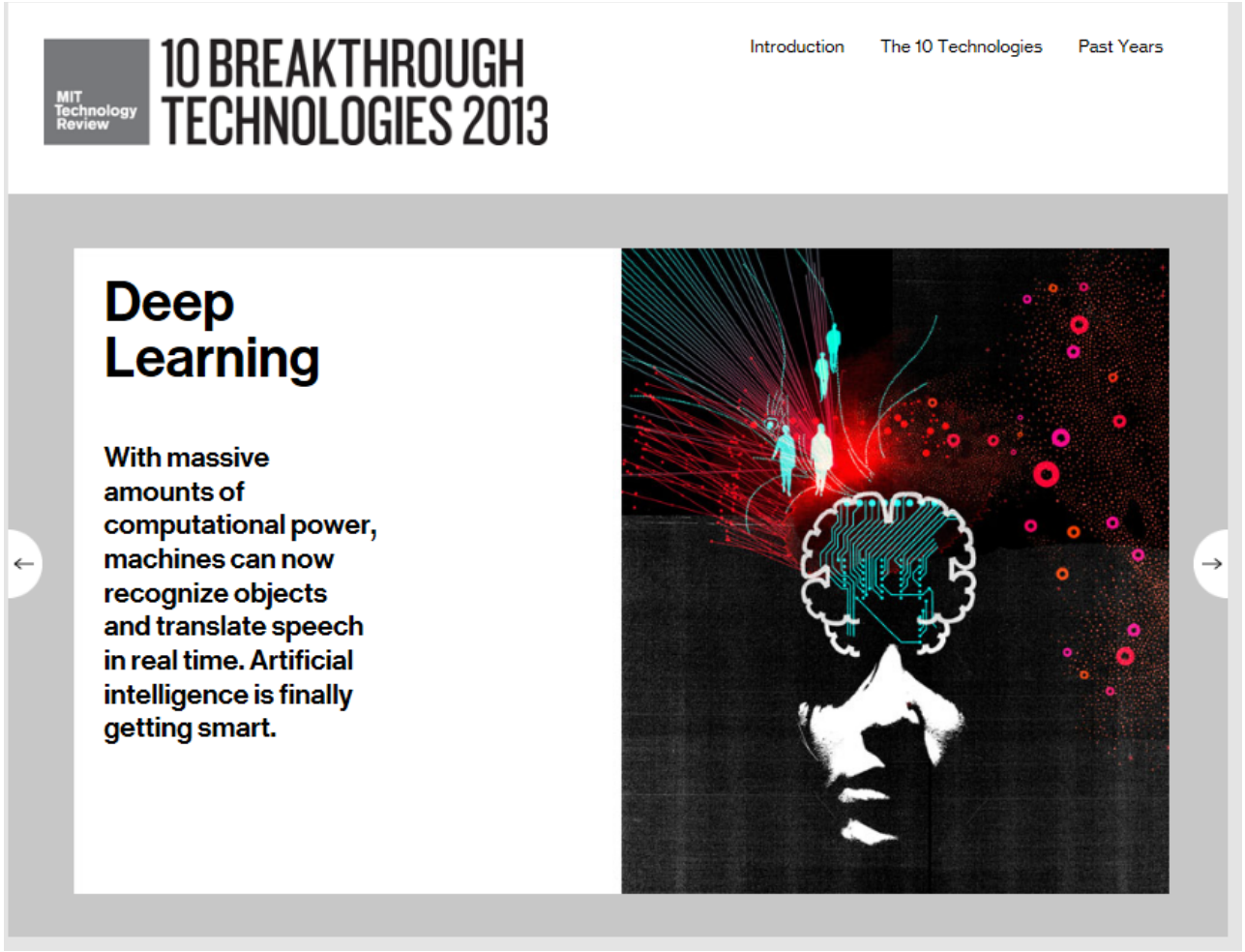
Ability to learn without explicitly being programmed

Deep Learning

Learn underlying features in data using neural networks



MIT 10 Breakthrough Tech 2013



http://www.technologyreview.com/featuredstory/513696/deep-learning/

Deep Learning in industry

Driverless car

Face identification

Speech recognition

Web search

...

Google

阿里巴巴 Alibaba.com

HUAWEI

intel

...

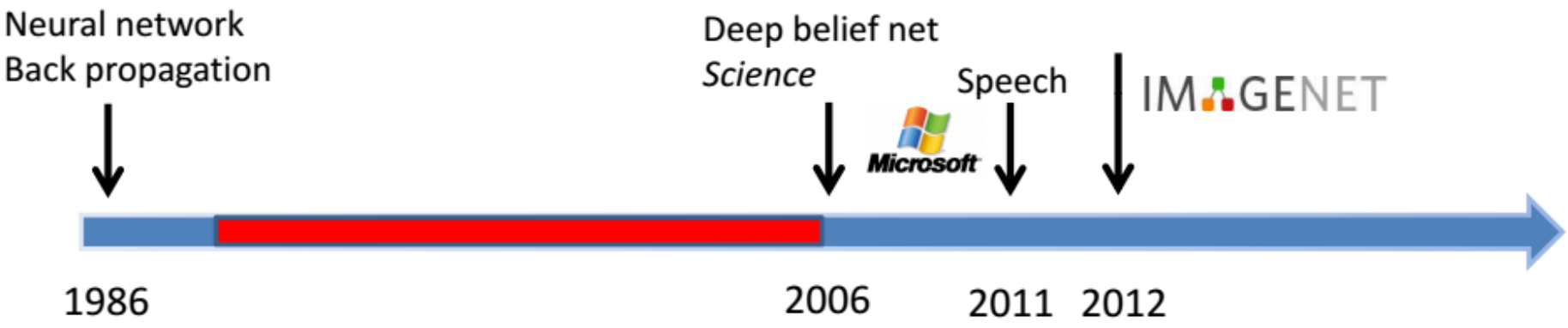
Microsoft

Baidu 百度

Tencent 腾讯

AMD

Deep Learning Success: Vision



- ImageNet 2013 – image classification challenge

Rank	Name	Error rate	Description
1	NYU	0.11197	Deep learning
2	NUS	0.12535	Deep learning
3	Oxford	0.13555	Deep learning

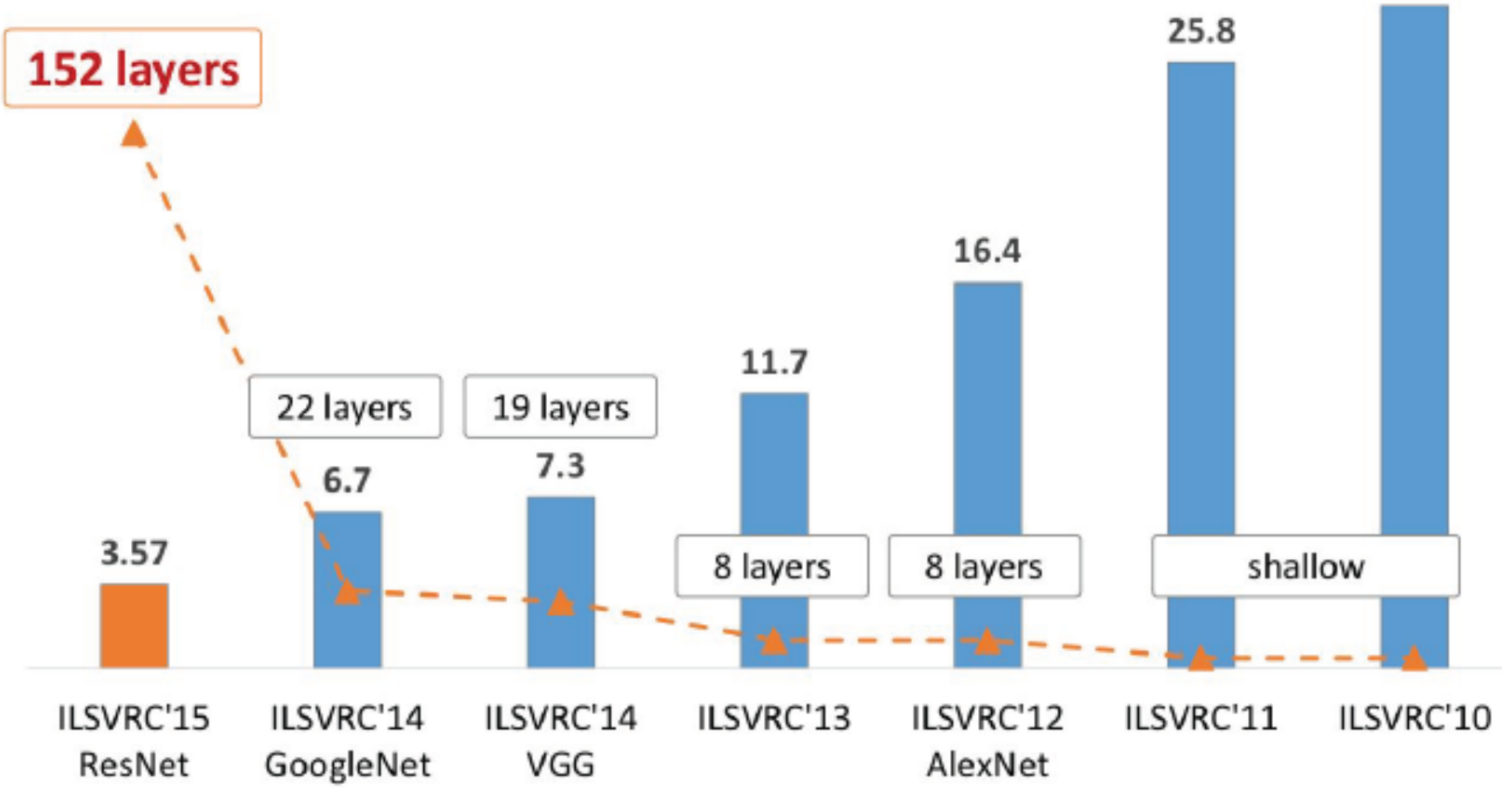
MSRA, IBM, Adobe, NEC, Clarifai, Berkley, U. Tokyo, UCLA, UIUC, Toronto …. Top 20 groups all used deep learning

- ImageNet 2013 – object detection challenge

Rank	Name	Mean Average Precision	Description
1	UvA-Euvision	0.22581	Hand-crafted features
2	NEC-MU	0.20895	Hand-crafted features
3	NYU	0.19400	Deep learning

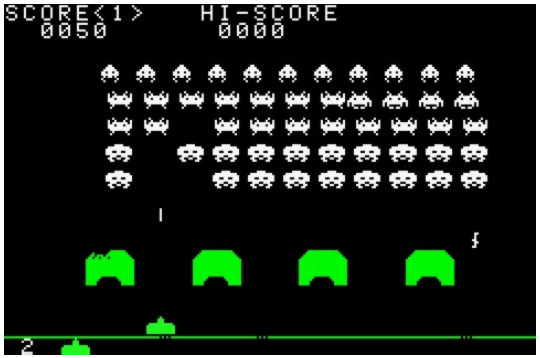
Resolution in Image Classification

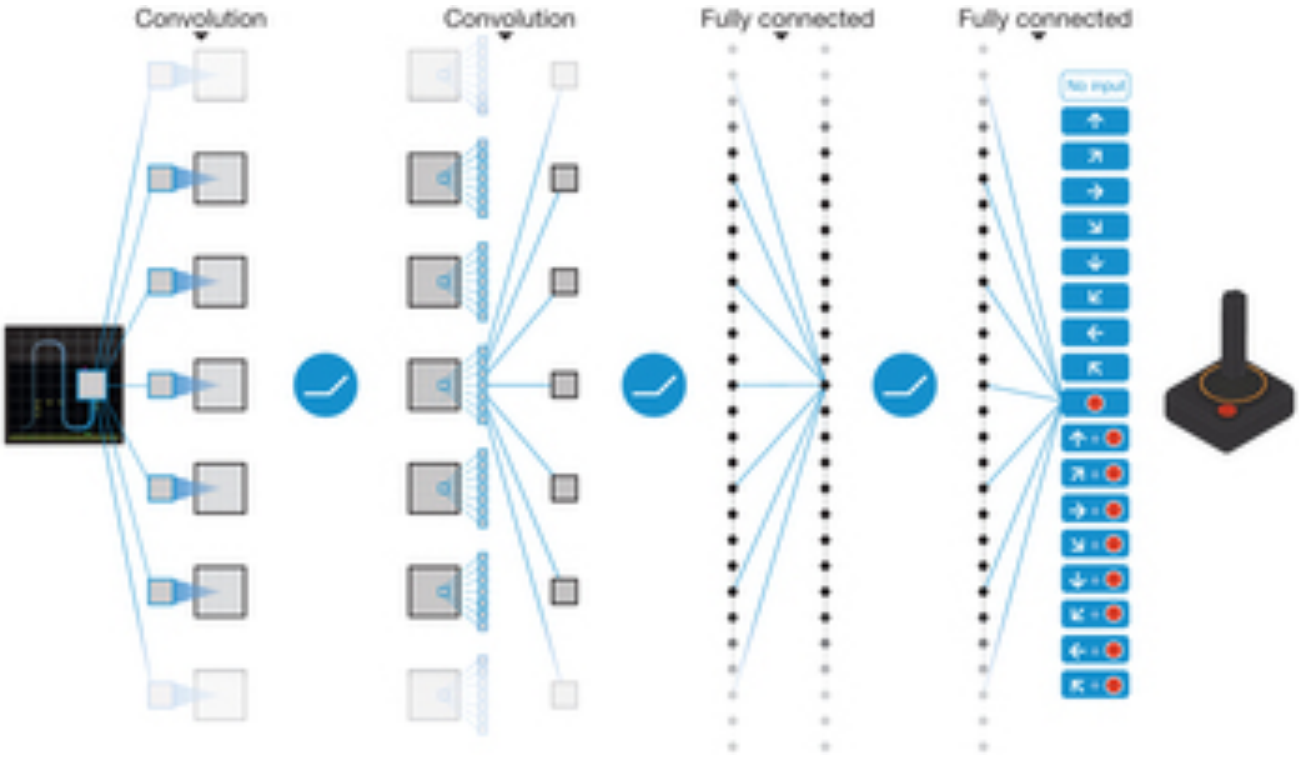
- ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)



Human-Level Control via Deep RL

- Deep Q-network with human-level performance on Atari games

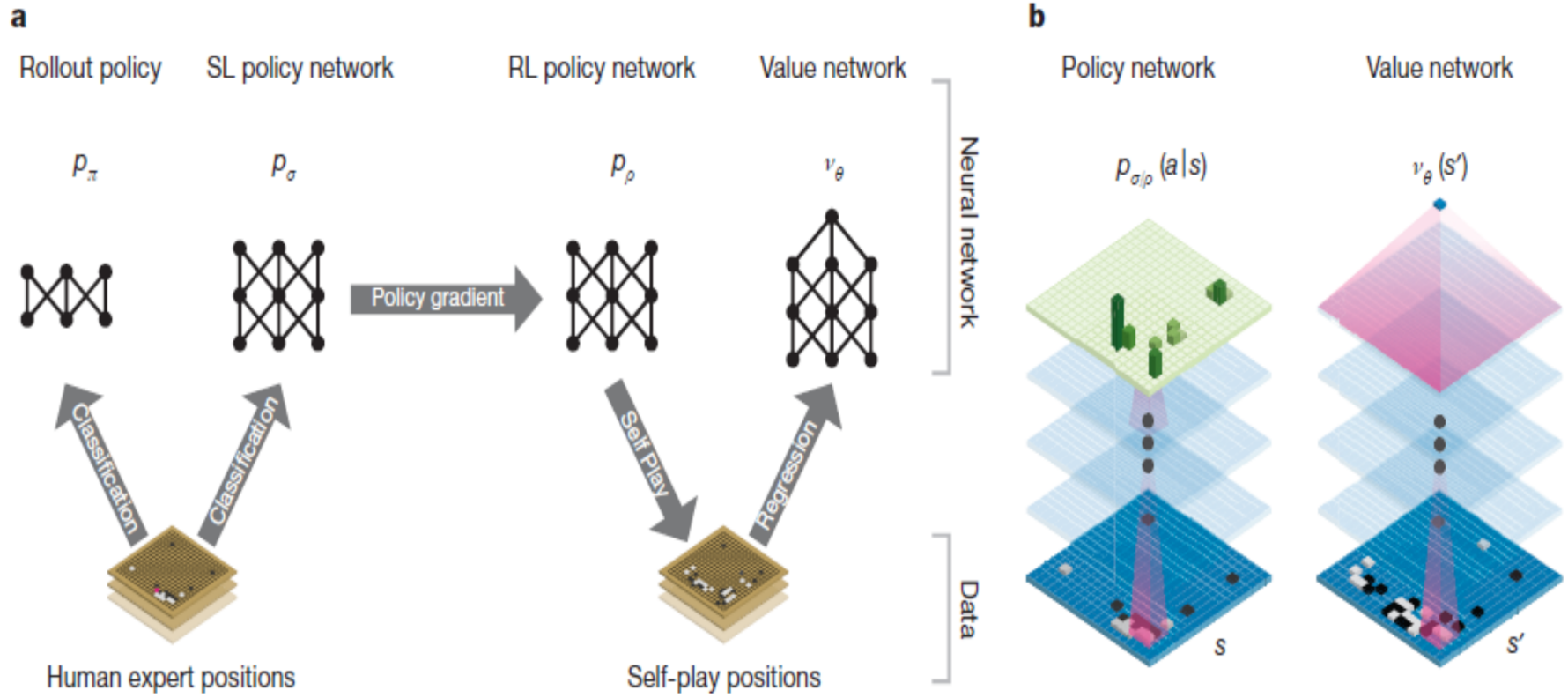




[Mnih et al., Nature 518, 529–533, 2015]

AlphaGo

- Neural network training pipeline and architecture



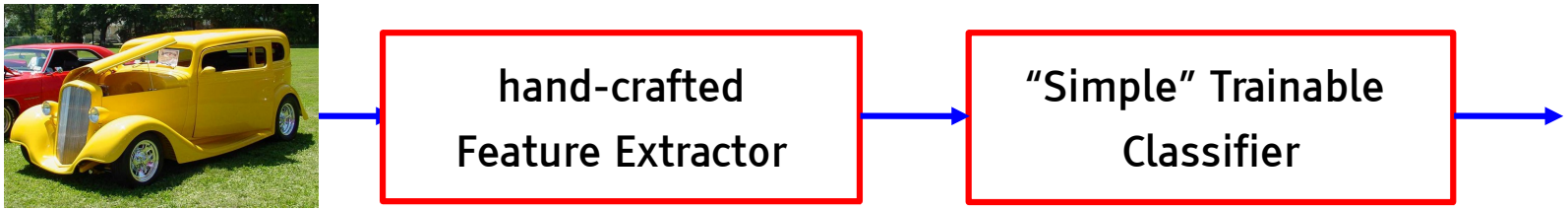
[Silver et al., Mastering the game of Go with deep neural networks and tree search. Nature, 484(529), 2016]

Content

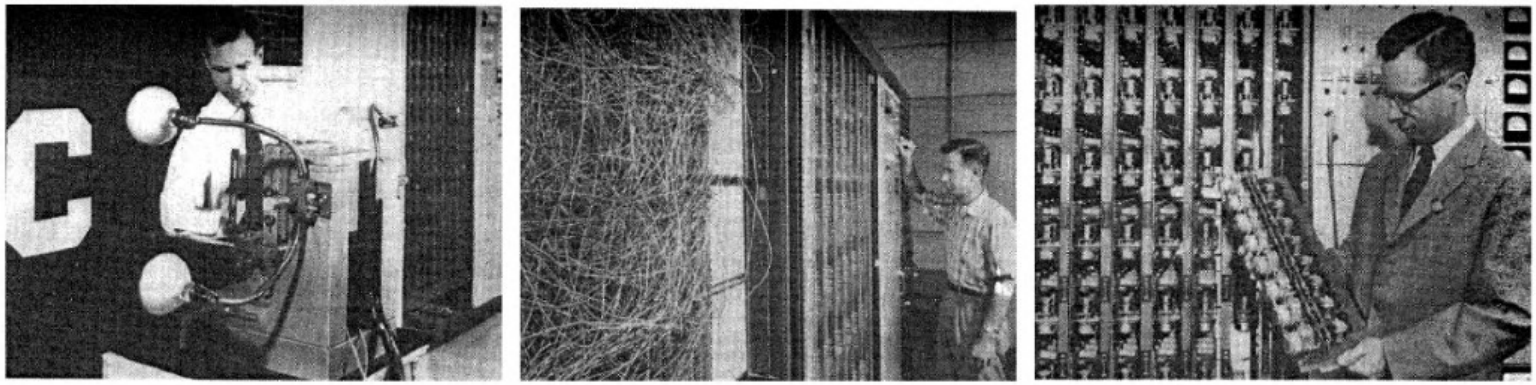
- Why deep Learning?
- Convolutional Neural Networks
- Discussion of Deep Learning

The Traditional Model of Machine Learning

- The traditional model of machine learning(since the late 50's)
  - Fixed/engineered features + trainable classifier

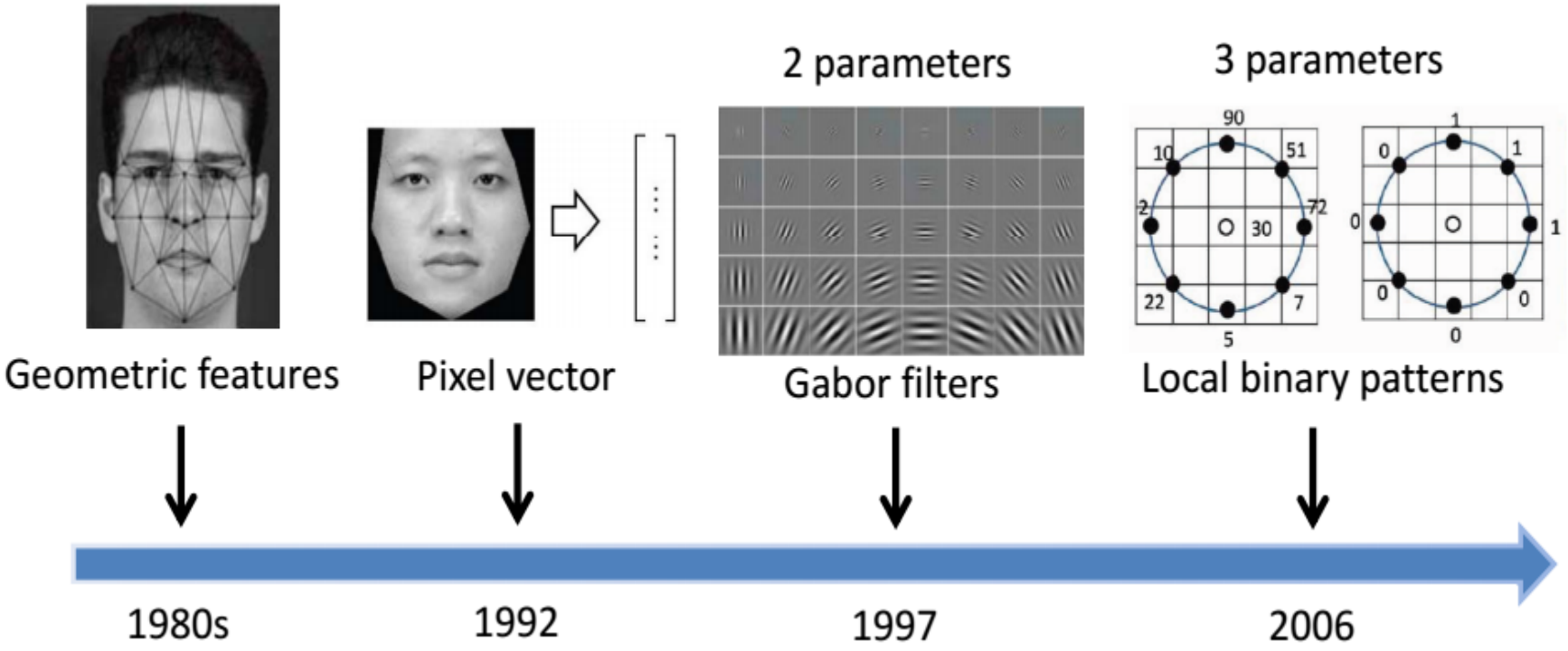


- Perceptron (Cornell University, 1957)



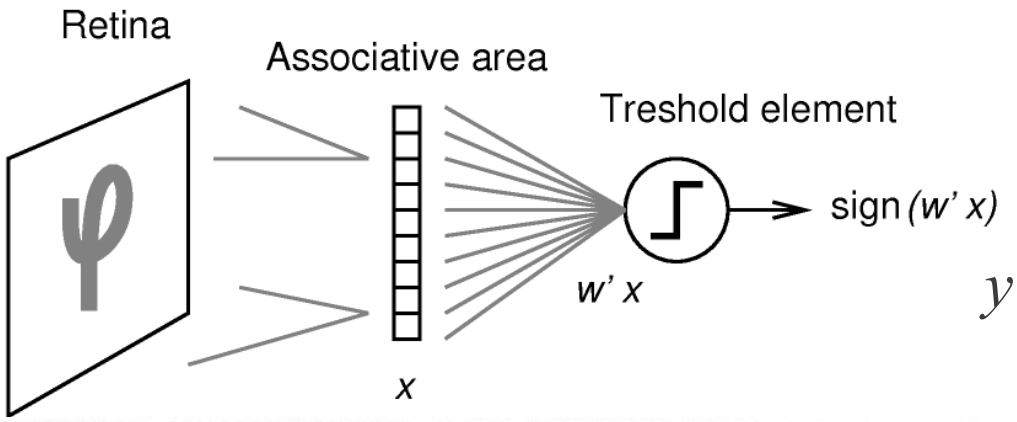
Feature Engineering

- Handcrafted Features for Face Recognition



1957: The Perceptron (the first learning machine)

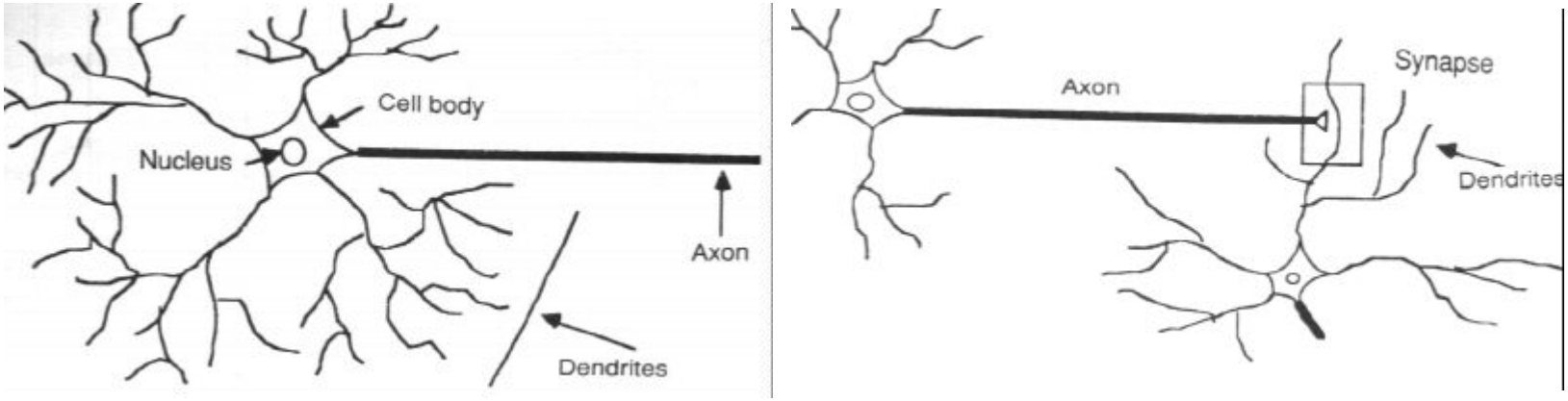
- A simple simulated neuron with adaptive "synaptic weights"
  - Computes a weighted sum of inputs
    Output is +1 if the weighted sum is above a thresold, -1 otherwise.



$$y = sign\left(\sum_{i=1}^{N} W_i X_i + b\right)$$

How the human brain learns?

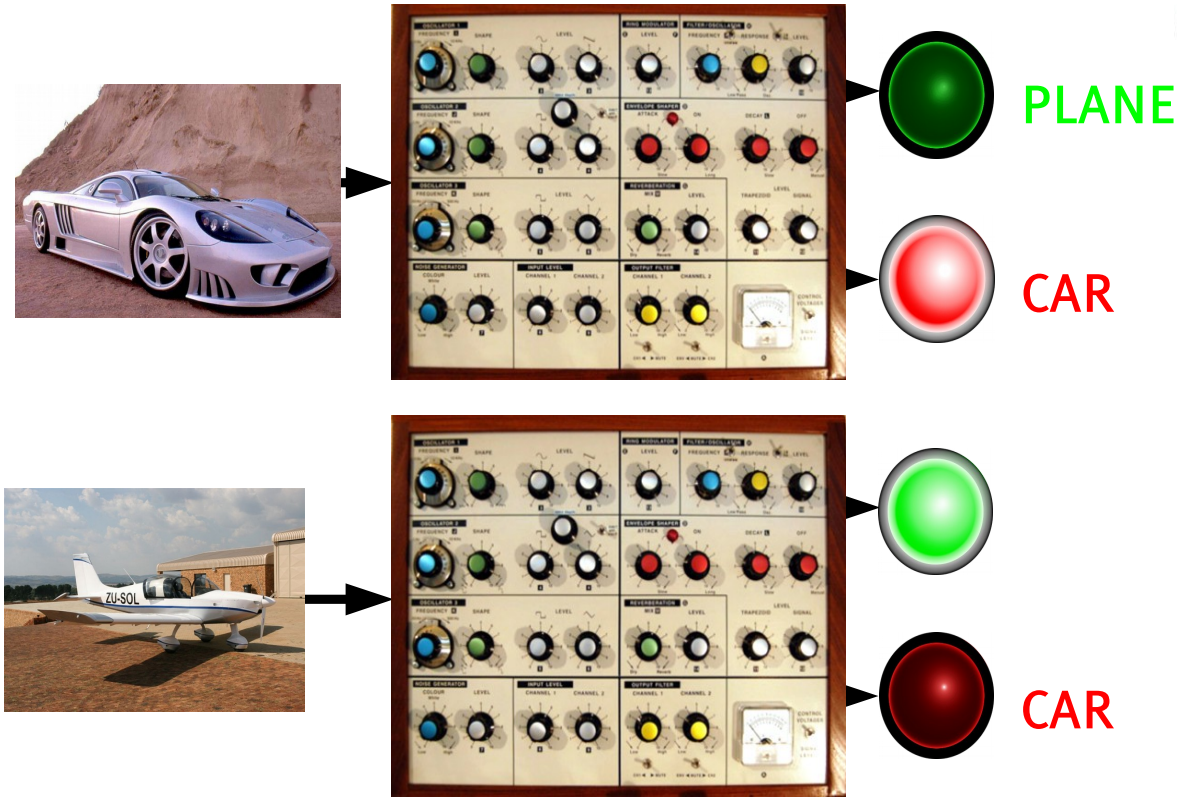
- A typical neuron



- Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes

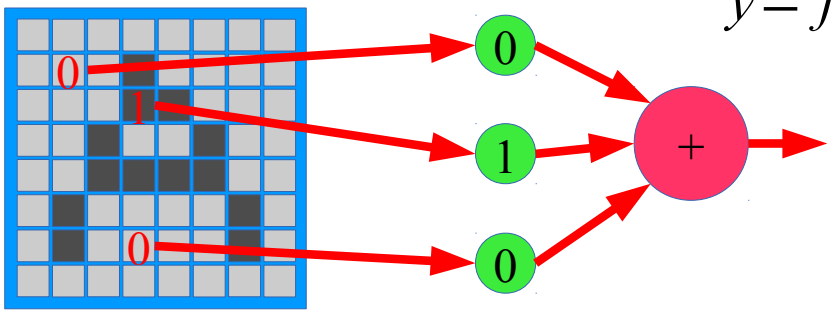
Supervised Learning

- We can train a machine on lots of examples of cars and planes



Error-correction Learning

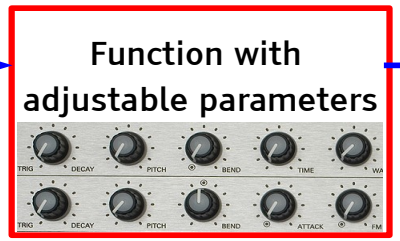
- Given training dataset: $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$
- Take one sample (x_i, y_i), if the desired output is +1 but the actual output is -1
  - Increase the weights whose input is positive
  - Decrease the weights whose input is negative
- If the desired is -1 and actual is +1, do the converse.
- If desired and actual are equal, do nothing

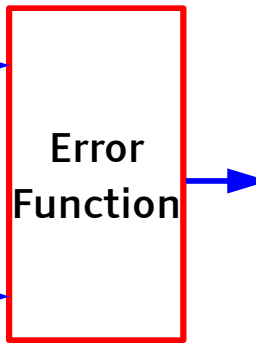


$$y = f\left(\sum_{i=1}^{N} w_i x_i + w_0\right) = f\left(W' X\right)$$

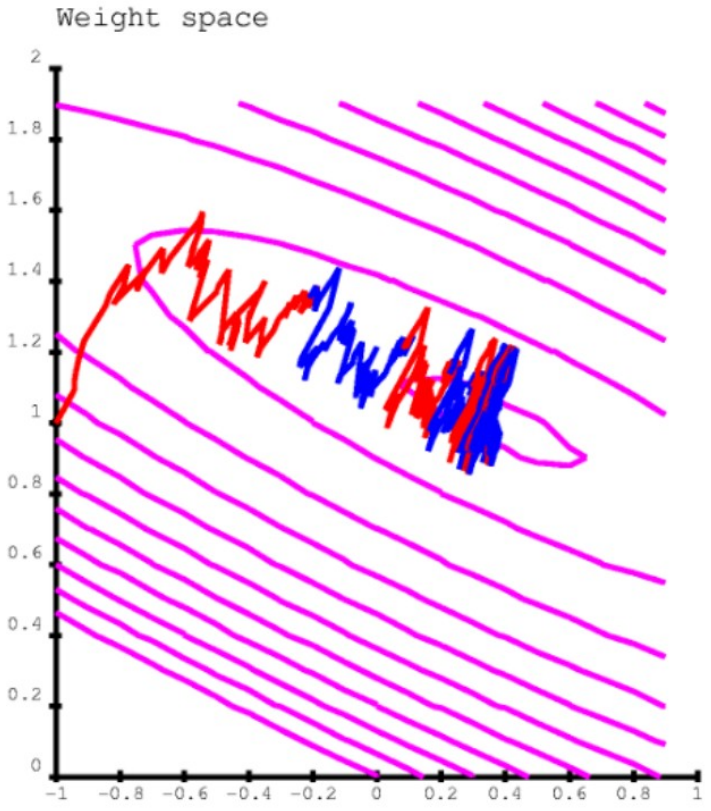
Supervised Learning

Function Optimization：Error-correction learning





traffic light: -1

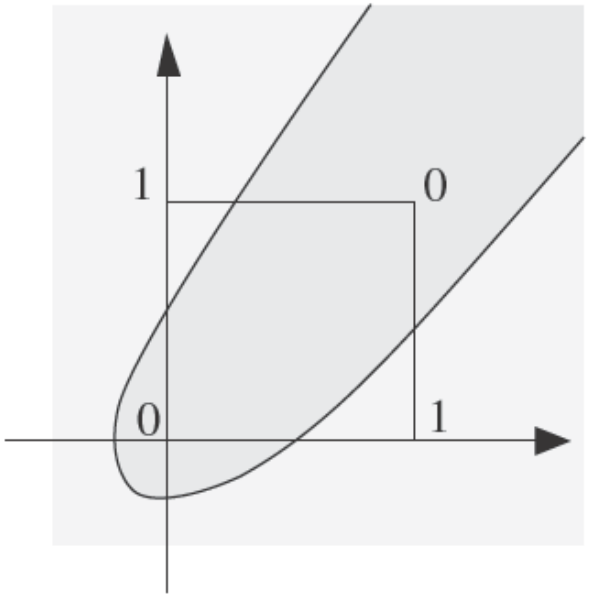


- It's like walking in the mountains in a fog and following the direction of steepest descent to reach the village in the valley
- Each sample gives us a noisy estimate of the direction.

$$W_i \leftarrow W_i - \eta \frac{\partial L(W, X)}{\partial W_i}$$

Stochastic Gradient Descent (SGD)

XOR Problem



- Single-layer perceptron can't solve the problem

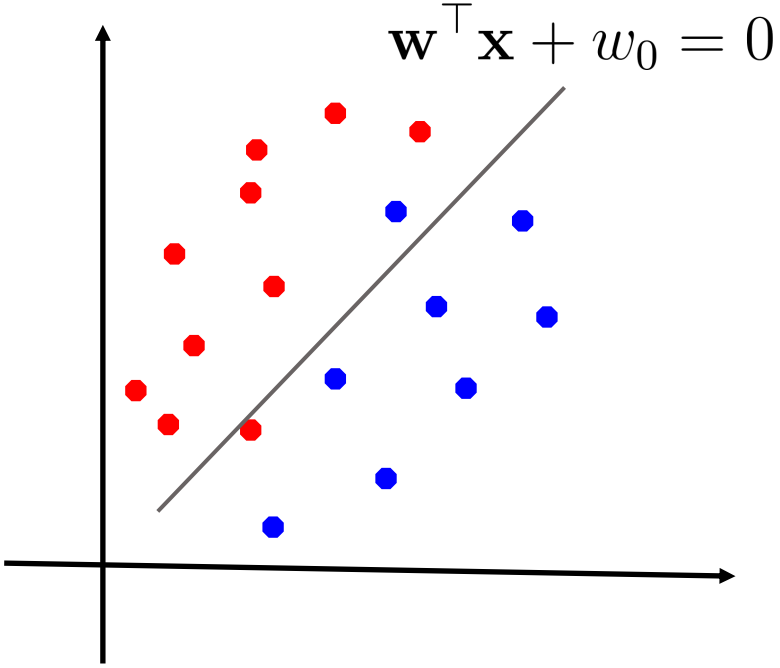
Connection with Logistic Regression

- What's the decision boundary of logistic regression? (linear or nonlinear?)

$$P(y=1|\mathbf{x}) = \frac{1}{1+\exp(-(w_0+\mathbf{w}^\top\mathbf{x}))}$$

$$\log\frac{P(Y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = 0$$

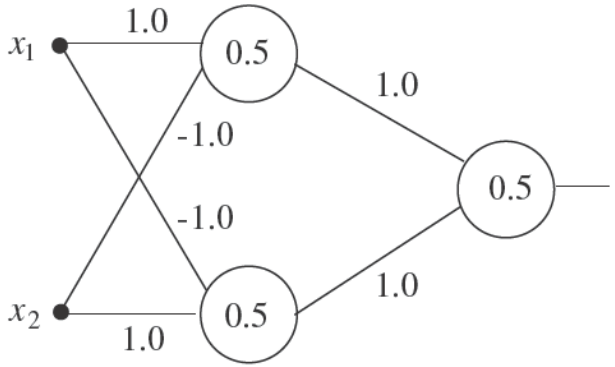
$$\mathbf{w}^\top\mathbf{x} + w_0 = 0$$



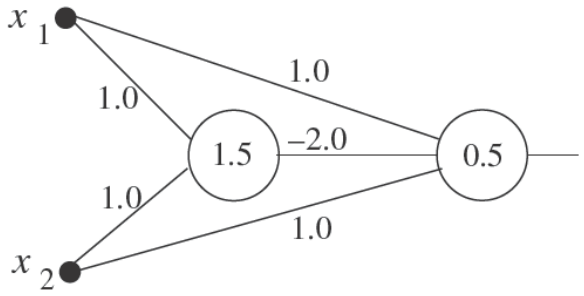
Logistic regression is a linear classifier!

XOR Problem

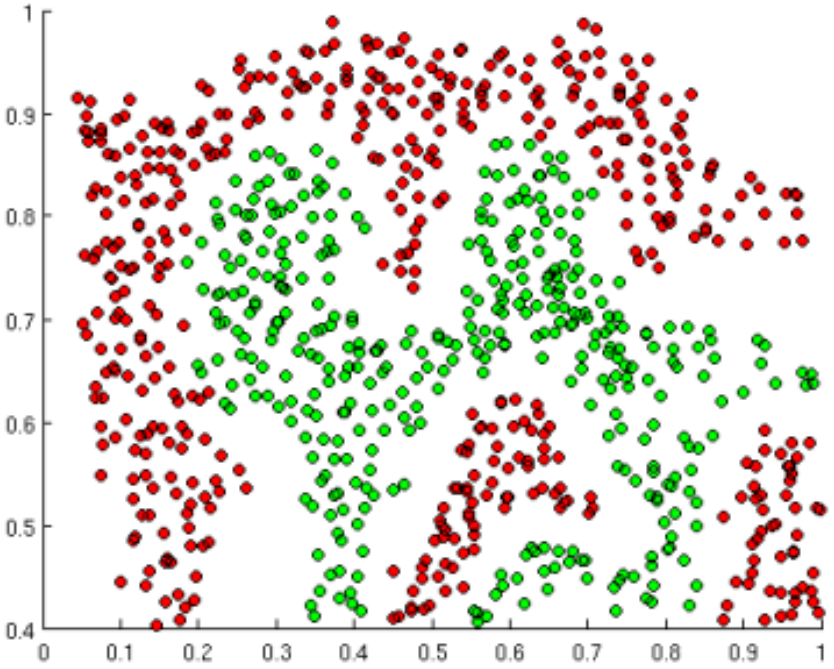
- A network with 1-layer of 2 neurons works for XOR:
  - threshold activation function



  - Many alternative networks exist (not layered)



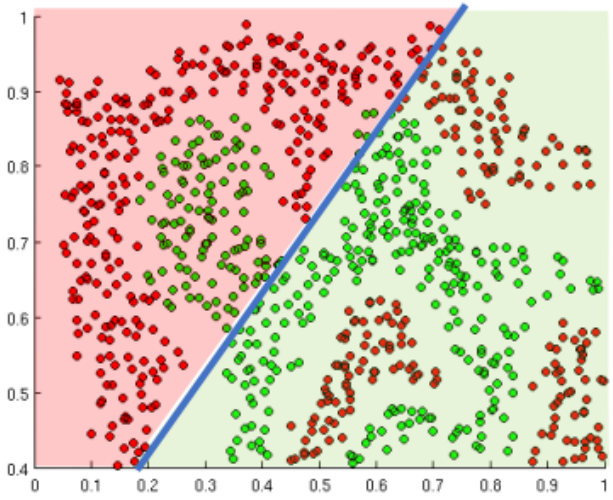
Importance of Activation Functions



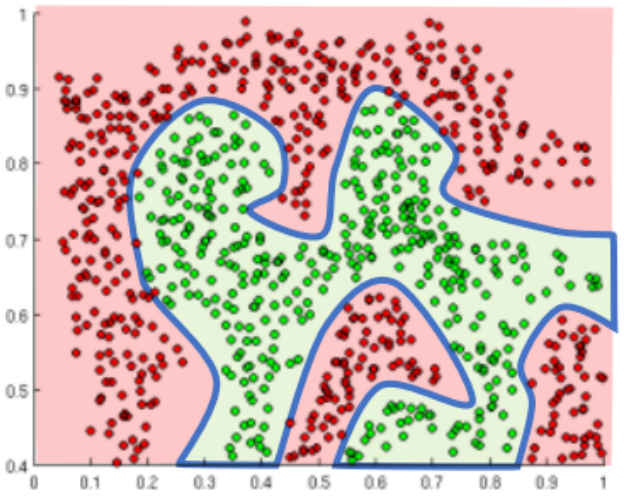
What if we wanted to build a Neural Network to distinguish green vs red points?

Importance of Activation Functions

The purpose of activation functions is to ***introduce non-linearities*** *into the network*

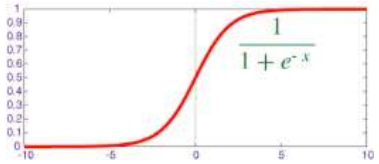


Linear Activation functions produce linear decisions no matter the network size



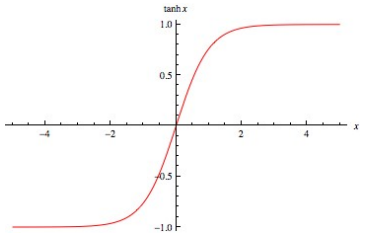
Non-linearities allow us to approximate arbitrarily complex functions

Activations and their derivatives



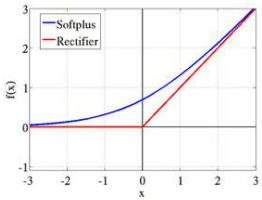
$$f(z) = \frac{1}{1 + \exp(-z)}$$

- Approximation of the step function
- Vanishing gradient



$$f(z) = \tanh(z)$$

- Extend to negative values and avoiding bias in the gradients
- Vanishing gradient



$$f(z) = \begin{cases} 0, & z < 0 \\ z, & z \geq 0 \end{cases}$$

$$f(z) = \log(1 + \exp(z))$$

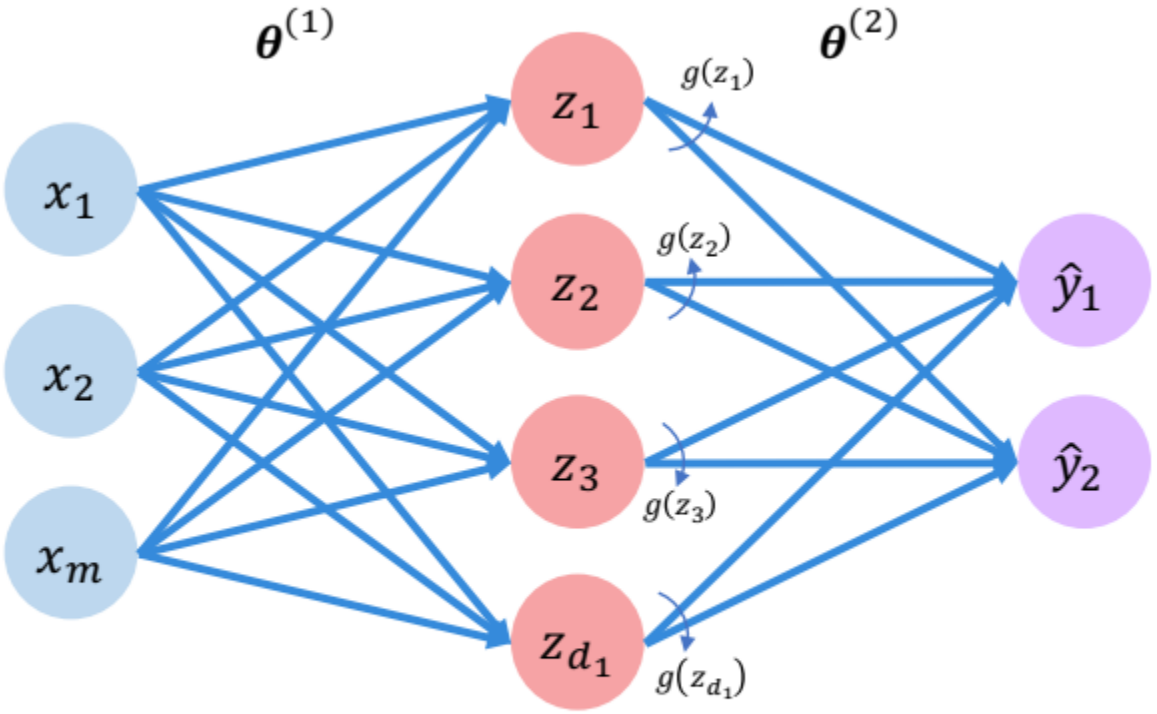
- Promote a reduced likelihood of encountering vanishing gradient
- Constant gradient results in faster learning
- Increasing the sparsity of the network

Some popular activation functions and their derivatives!

Supervised Learning

- Hundreds of millions of "knobs" (or weights)
- Thousands of categories
- Millions of training samples
- Recognizing each sample may take billions of operations
  - But these operations are simple multiplications and additions

Single Layer Neural Network

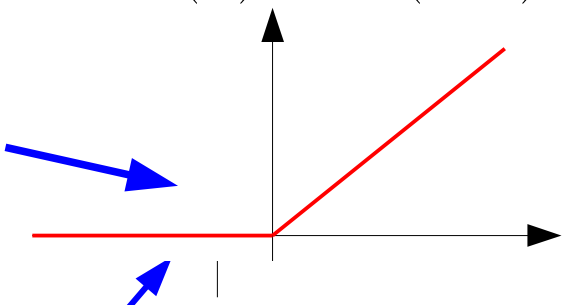


$$z_i = \theta_{0,i}^{(1)} + \sum_{j=1}^{m} x_j \, \theta_{j,i}^{(1)} \qquad \hat{y}_i = \theta_{0,i}^{(2)} + \sum_{j=1}^{d_1} z_j \, \theta_{j,i}^{(2)}$$

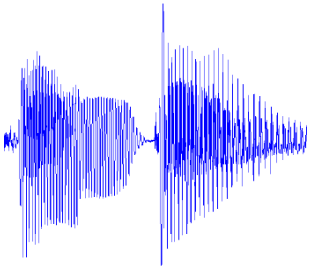
Multi-Layer Neural Nets

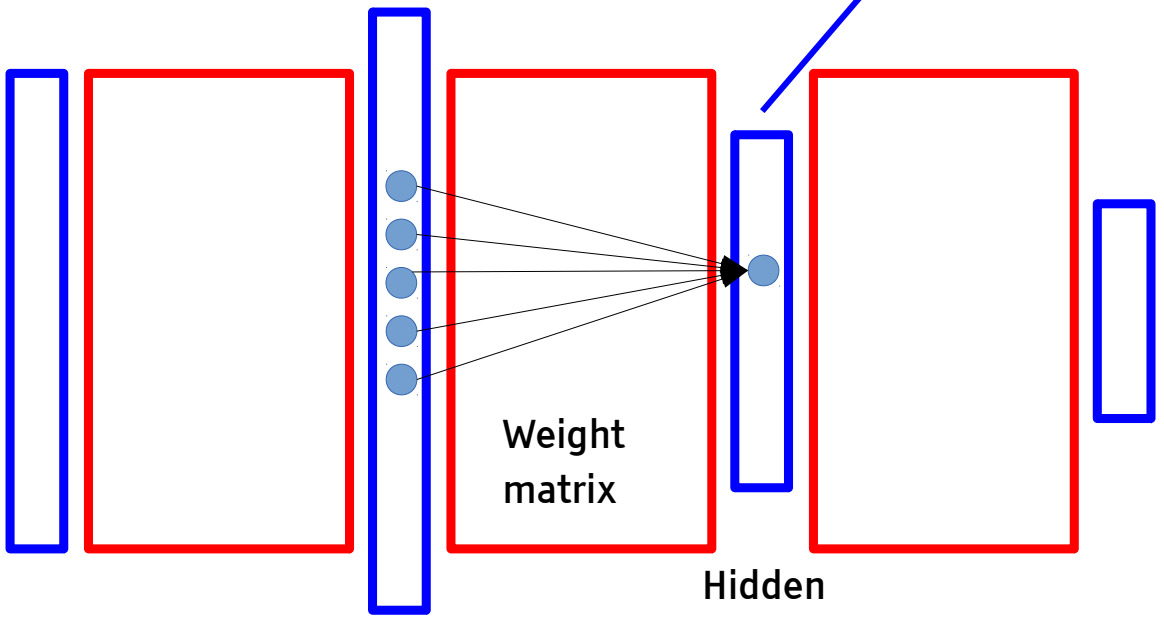
- Multiple Layers of simple units
- Each units computes a weighted sum of its inputs
- Weighted sum is passed through a non-linear function
- The learning algorithm changes the weights

$$ReLU(x) = max(x, 0)$$



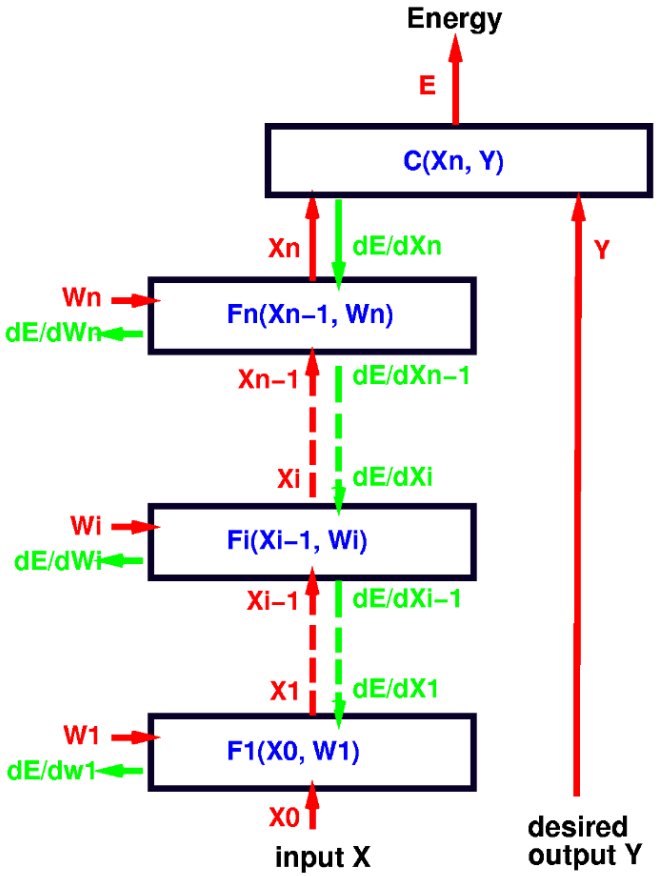
Ceci est une voiture





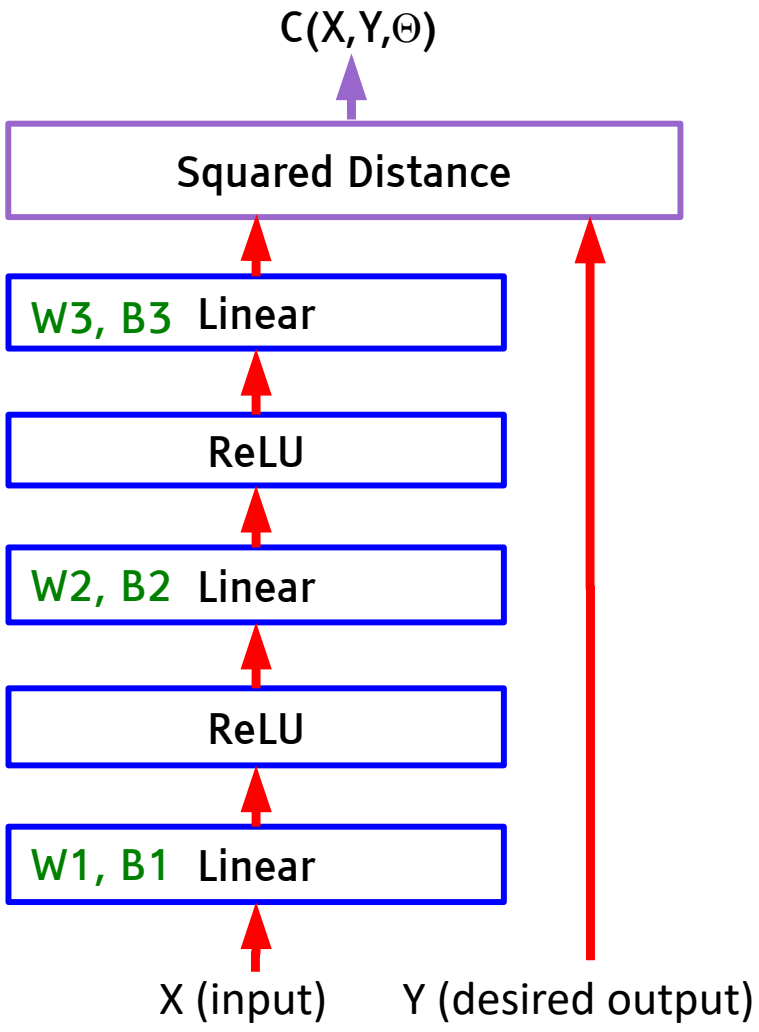
Back Propagation

To compute all the derivatives, we use a backward sweep called the **back-propagation algorithm** that uses the recurrence equation for $\frac{\partial E}{\partial X_k}$



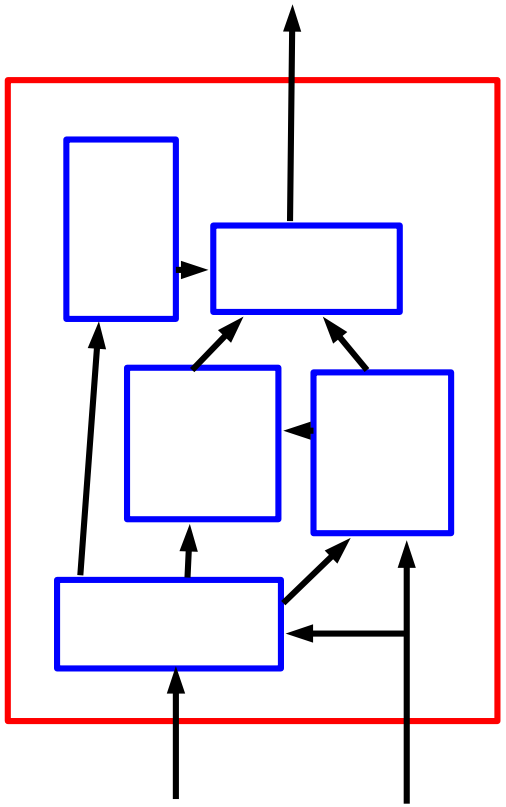
- $\frac{\partial E}{\partial X_n} = \frac{\partial C(X_n, Y)}{\partial X_n}$
- $\frac{\partial E}{\partial X_{n-1}} = \frac{\partial E}{\partial X_n} \frac{\partial F_n(X_{n-1}, W_n)}{\partial X_{n-1}}$
- $\frac{\partial E}{\partial W_n} = \frac{\partial E}{\partial X_n} \frac{\partial F_n(X_{n-1}, W_n)}{\partial W_n}$
- $\frac{\partial E}{\partial X_{n-2}} = \frac{\partial E}{\partial X_{n-1}} \frac{\partial F_{n-1}(X_{n-2}, W_{n-1})}{\partial X_{n-2}}$
- $\frac{\partial E}{\partial W_{n-1}} = \frac{\partial E}{\partial X_{n-1}} \frac{\partial F_{n-1}(X_{n-2}, W_{n-1})}{\partial W_{n-1}}$
- ....etc, until we reach the first module.
- we now have all the $\frac{\partial E}{\partial W_k}$ for $k \in [1, n]$.

Typical Multilayer Neural Net Architecture



- Complex learning machines can be built by assembling modules into networks
- Linear Module
  - Out = W.In+B
- ReLU Module (Rectified Linear Unit)
  - $Out_i = 0$ if $In_i < 0$
  - $Out_i = In_i$ otherwise
- Cost Module: Squared Distance
  - $C = ||In1 - In2||^2$
- Objective Function
  - $L(\Theta) = 1/p \sum_k C(X^k, Y^k, \Theta)$
  - $\Theta = (W1, B1, W2, B2, W3, B3)$

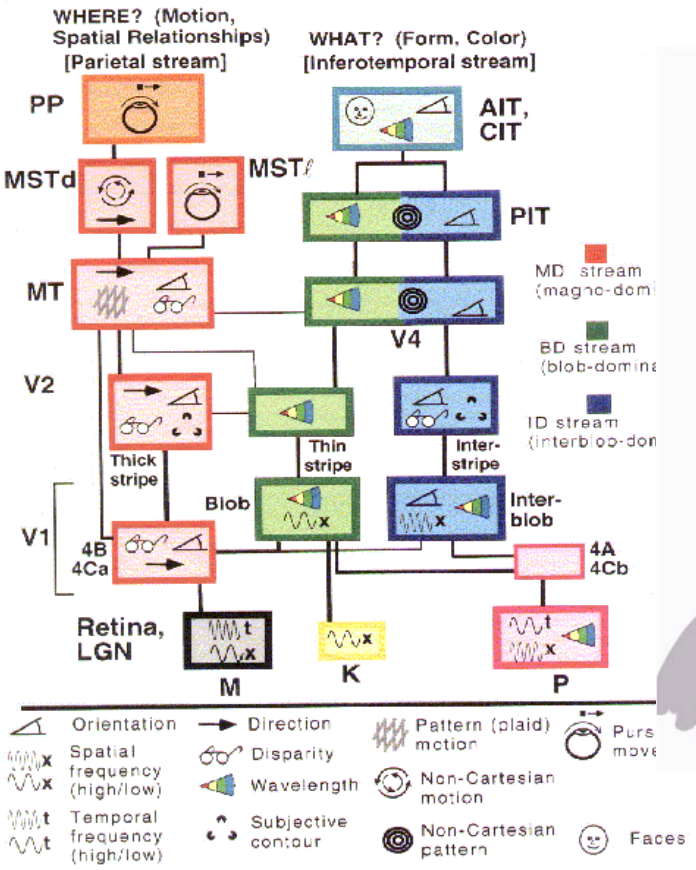
Any Architecture works

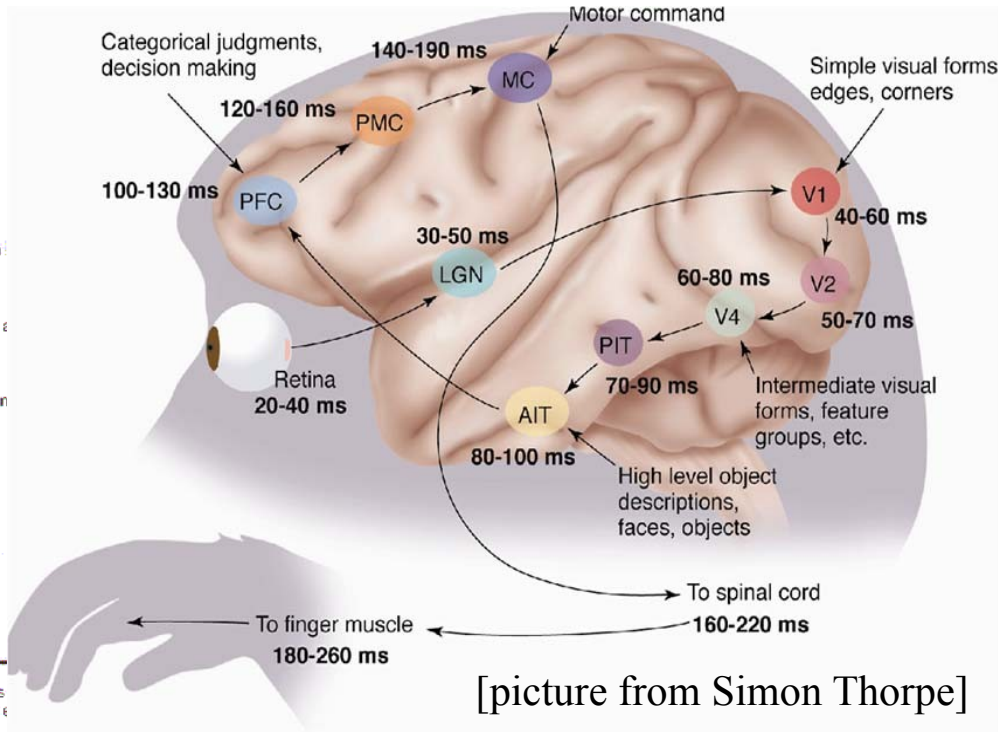


- Any connection graph is permissible
  - Directed acyclic graphs (DAG)
  - Networks with loops must be "unfolded in time".
- Any module is permissible
  - As long as it is continuous and differentiable almost everywhere with respect to the parameters, and with respect to non-terminal inputs.
- Most frameworks provide automatic differentiation
  - Programs are turned into computation DAGs and automatically differentiated.
  - Tensorflow, PyTorch

How does the brain interprets images?

- The ventral (recognition) pathway in the visual cortex has multiple stages
- Retina - LGN - V1 - V2 - V4 - PIT - AIT ….





[picture from Simon Thorpe]

[Gallant & Van Essen]

Let's be inspired by nature, but not too much

- It's nice imitate Nature, but we also need to understand
  - How do we know which details are important?
  - Which details are merely the result of evolution, and the constraints of biochemistry?
- For airplanes, we developed aerodynamics and compressible fluid dynamics.
- We figured that feathers and wing flapping weren't crucial
- QUESTION: What is the equivalent of aerodynamics for understanding intelligence?

Machine Learning has become Alchemy?

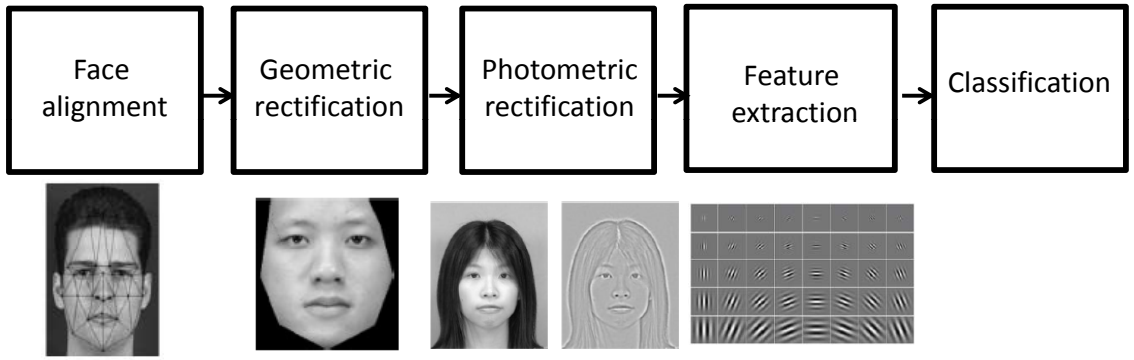
- Ali Rahimi (winner of the Test-of-Time award at NIPS)
  - Contemporary machine learning models' successes are plagued with the same issues as alchemy.
  - The inner mechanisms of machine learning models are so complex and opaque
- Tomoso Poggio (Professor of MIT):
  - Deep Learning: from Alchemy to Chemistry

Content

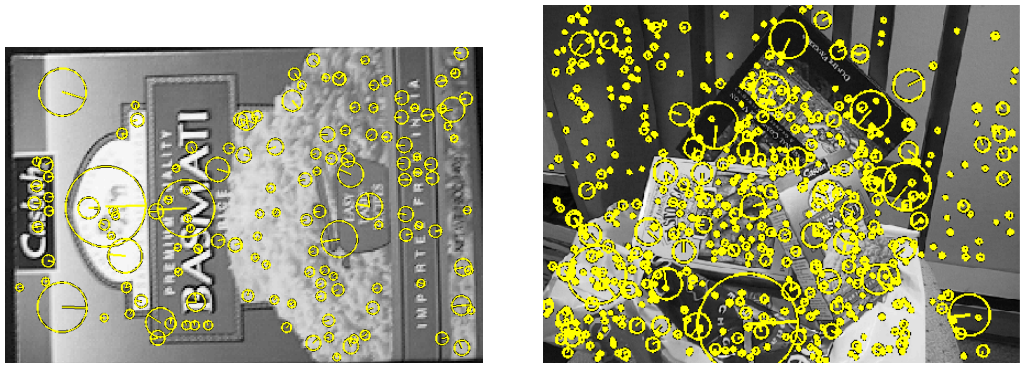
- Why deep Learning?
- Convolutional Neural Networks
- Discussion of Deep Learning

Feature Design

- Face recognition pipeline

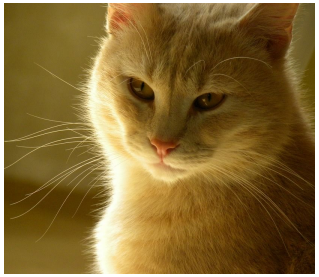


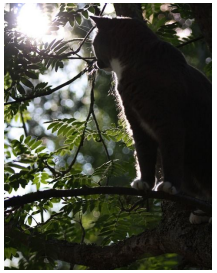
- Feature = interest point + descriptor of a local patch.



source: http://www.cs.ubc.ca/~lowe/keypoints/ + Vedaldi's VLFEAT.

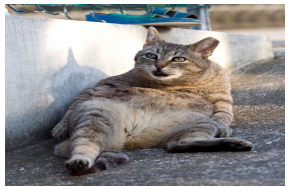
Challenges of Feature Engineering

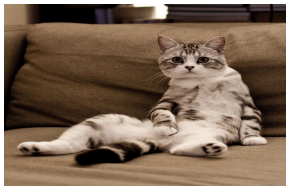




Illumination

Deformation





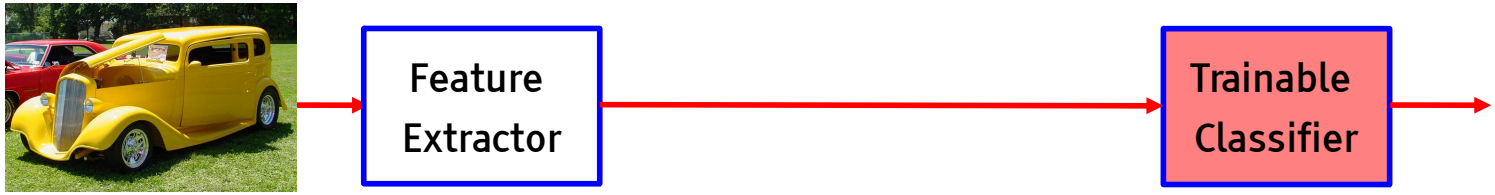
Occlusion


Background Clutter

Intraclass variation

Deep Learning = The Entire Machine is Trainable

- Traditional Pattern Recognition: Fixed/Handcrafted Feature Extractor



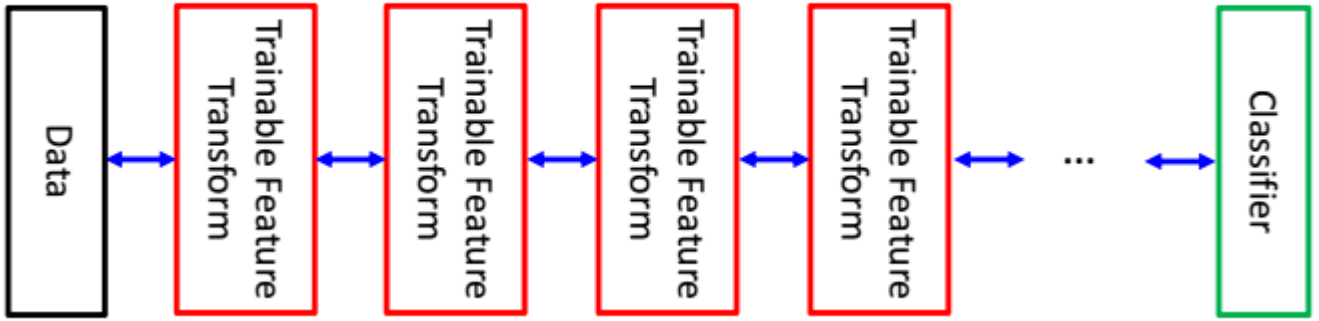
- Mainstream Modern Pattern Recognition: Unsupervised mid-level features

- Deep Learning: Representations are hierarchical and trained

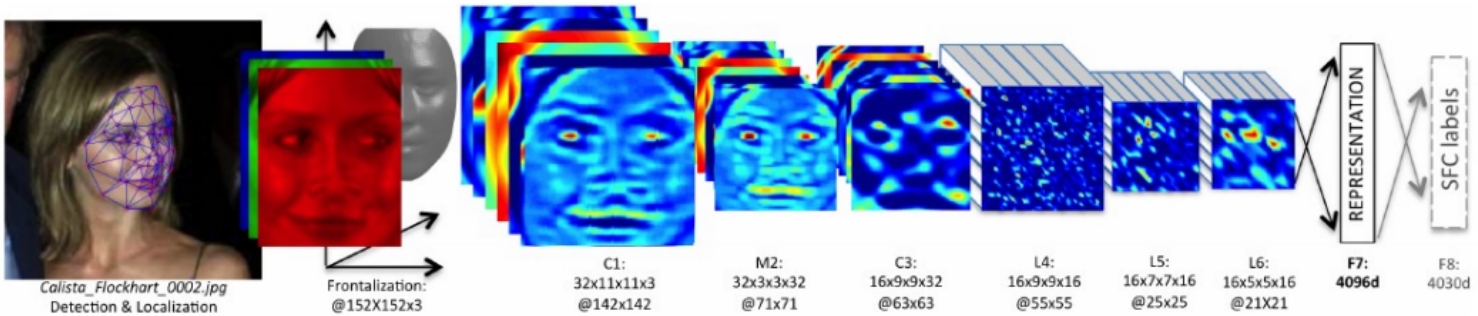
Deep Learning Means Feature Learning

- Deep learning is about learning hierarchical feature representations

$$\mathbf{y} = F(\mathbf{W}^k \cdot F(\mathbf{W}^{k-1} \cdot F(\ldots F(\mathbf{W}^0 \cdot \mathbf{x}))$$

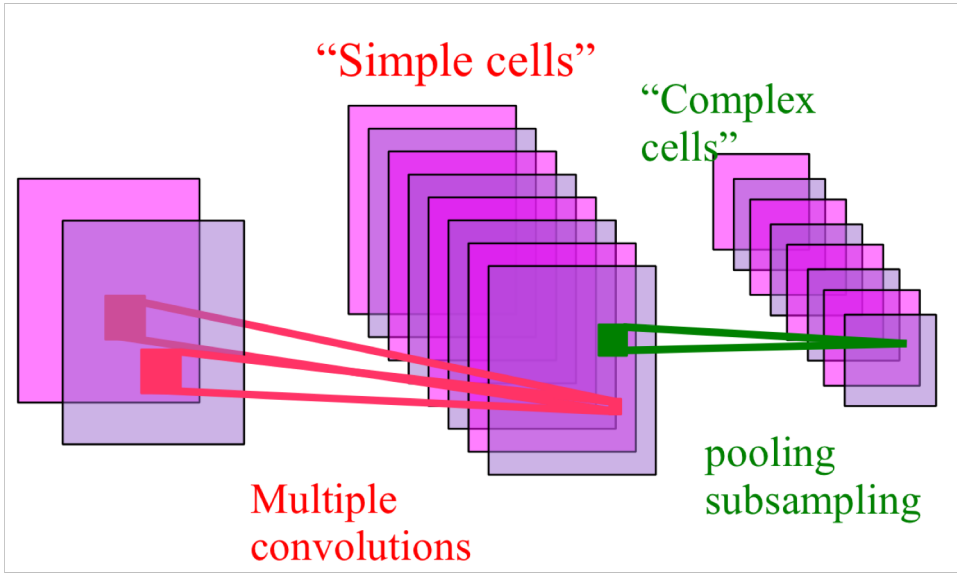


- Deep learning: Hierarchical; nonlinear



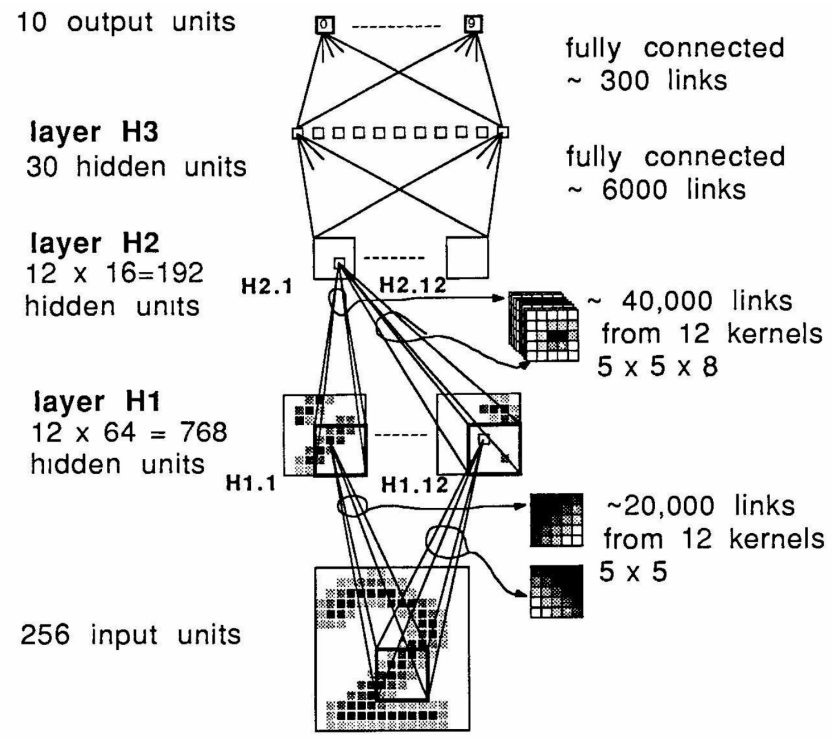
Hubel & Wiesel's Model of the Architecture of the Visual Cortex

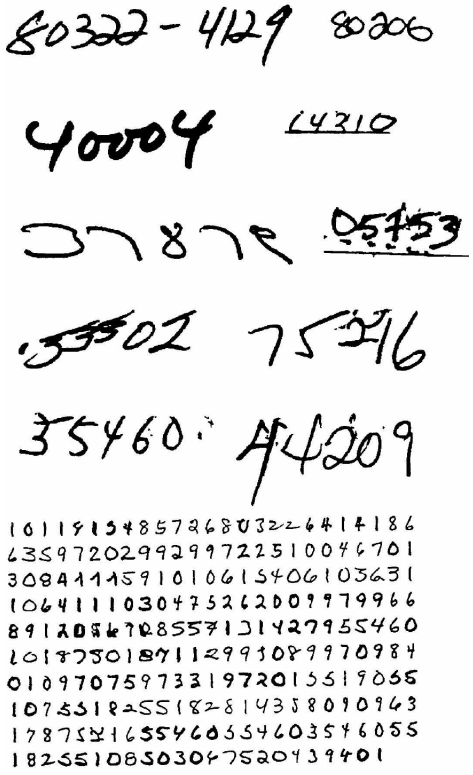
- [Hubel & Wiesel 1962]:
  - Simple cells detect local features
  - Complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.



First "Real" ConvNets at Bell Labs [LeCun et al 89]

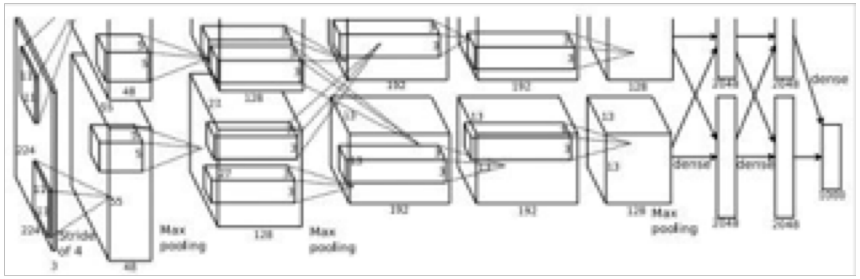
- Trained with Backprop
  - USPS Zipcode digits: 7300 training, 2000 test.
  - Convolution with stride. No separate pooling.



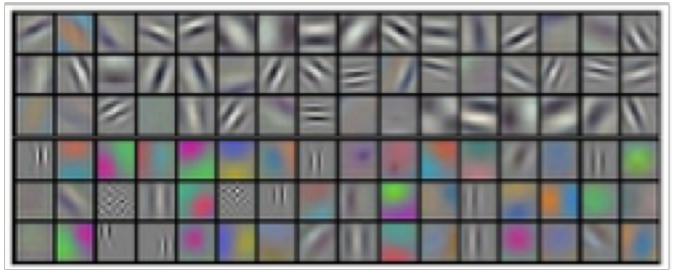


First strong results

- ***Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition***
  George Dahl, Dong Yu, Li Deng, Alex Acero, 2012
- ***Imagenet classification with deep convolutional neural networks***
  Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton, 2012



"AlexNet"



Convolutional Neural Networks

- Convolutional Neural Networks
  - Convolution
  - Subsampling
  - Fully connected

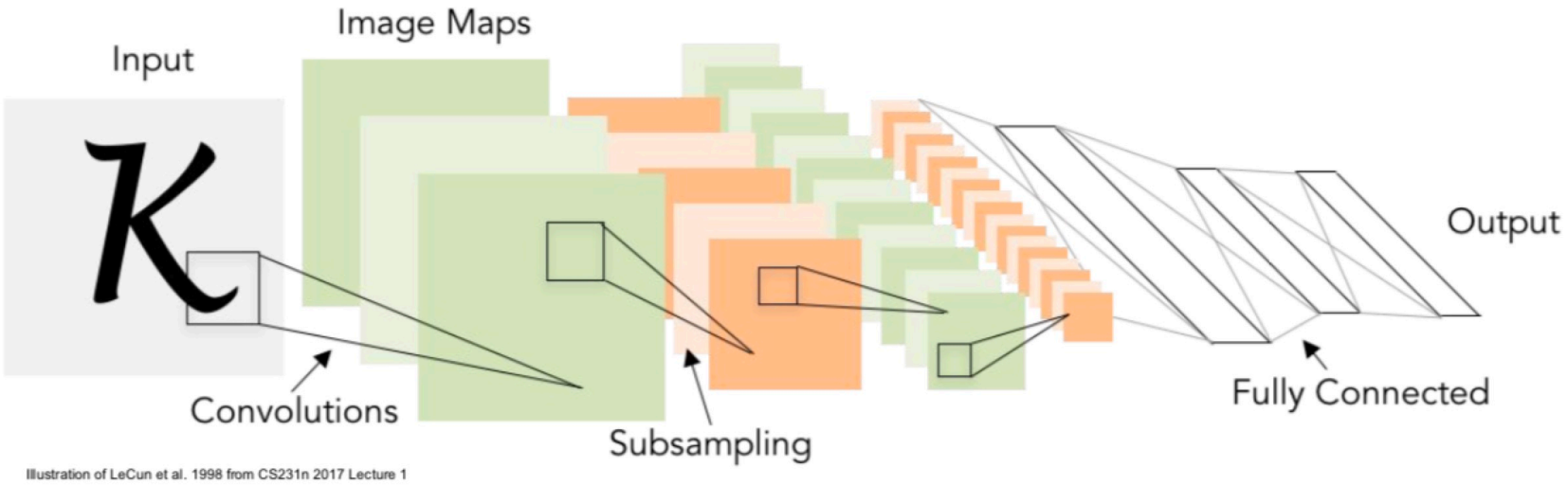
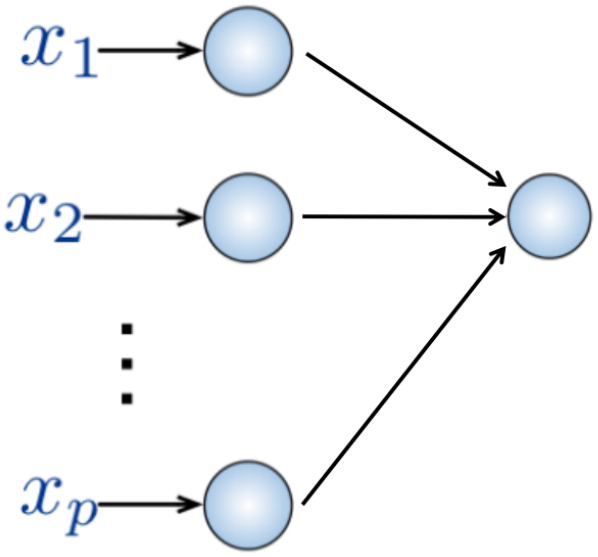


Illustration of LeCun et al. 1998 from CS231n 2017 Lecture 1

Fully Connected Neural Network

Input:

- 2D image
- Vector of pixel values



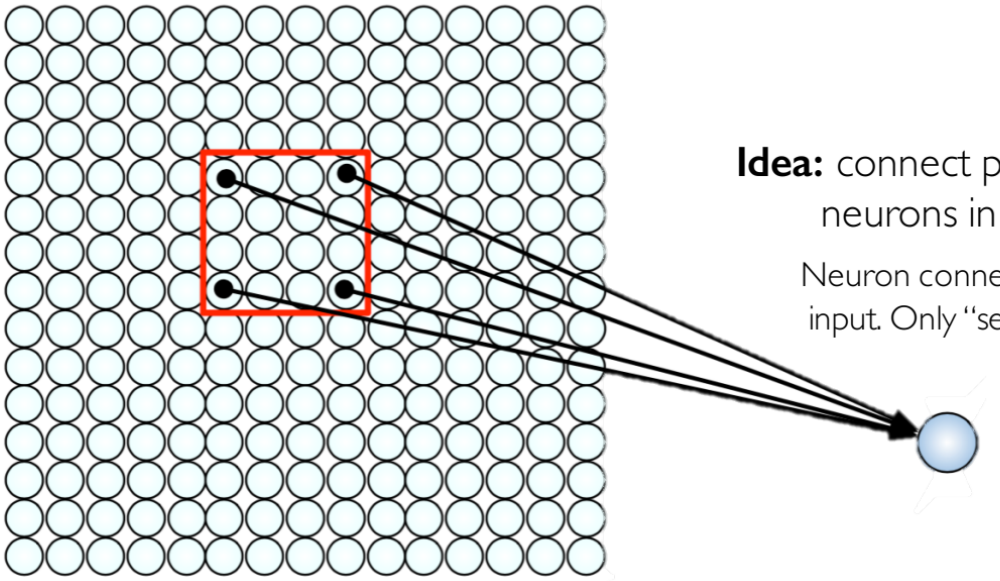
Fully connected:

- Connect neuron in hidden layer to all neurons in input layer
- No spatial information!
- And many, many parameters!

How can we use **spatial structure** in the input to inform the architecture of the network?

Using Spatial Structure

Input: 2D image.
Array of pixel values



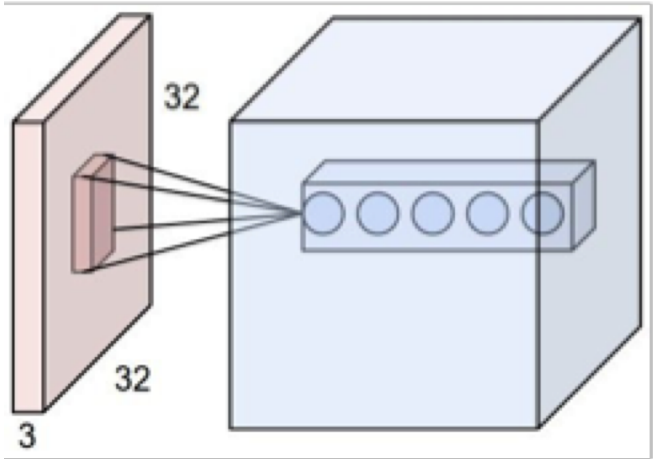
Idea: connect patches of input to neurons in hidden layer.

Neuron connected to region of input. Only "sees" these values.

Connect patch in input layer to a single neuron in subsequent layer.
Use a sliding window to define connections.
How can we weight the patch to detect particular features?

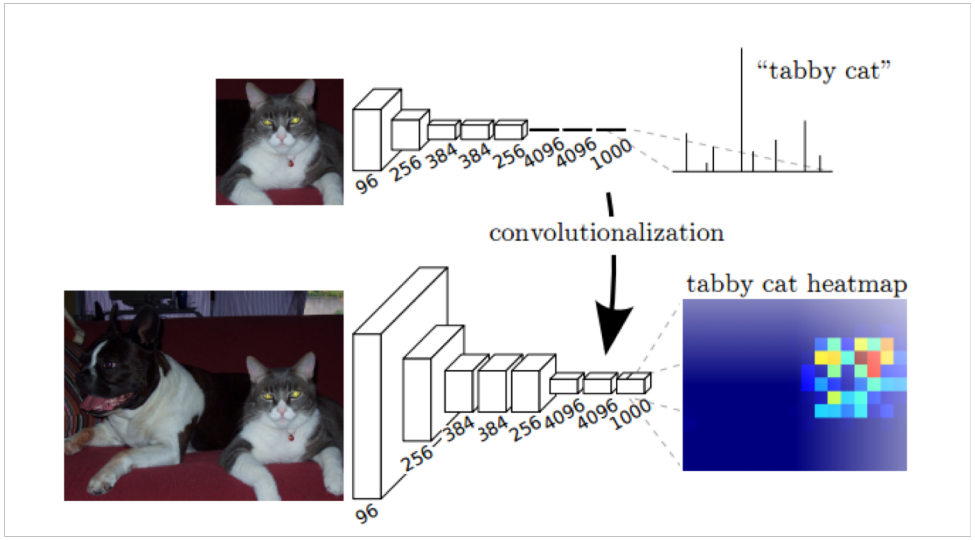
Convolutional Operation

- Convolutions can be thought of as sliding fully connected layers
- When the inputs to a convolutional layer are larger feature maps, outputs are larger feature maps
- Fully connected layers have a fixed number of inputs/outputs, forcing the entire network's input shape to be fixed



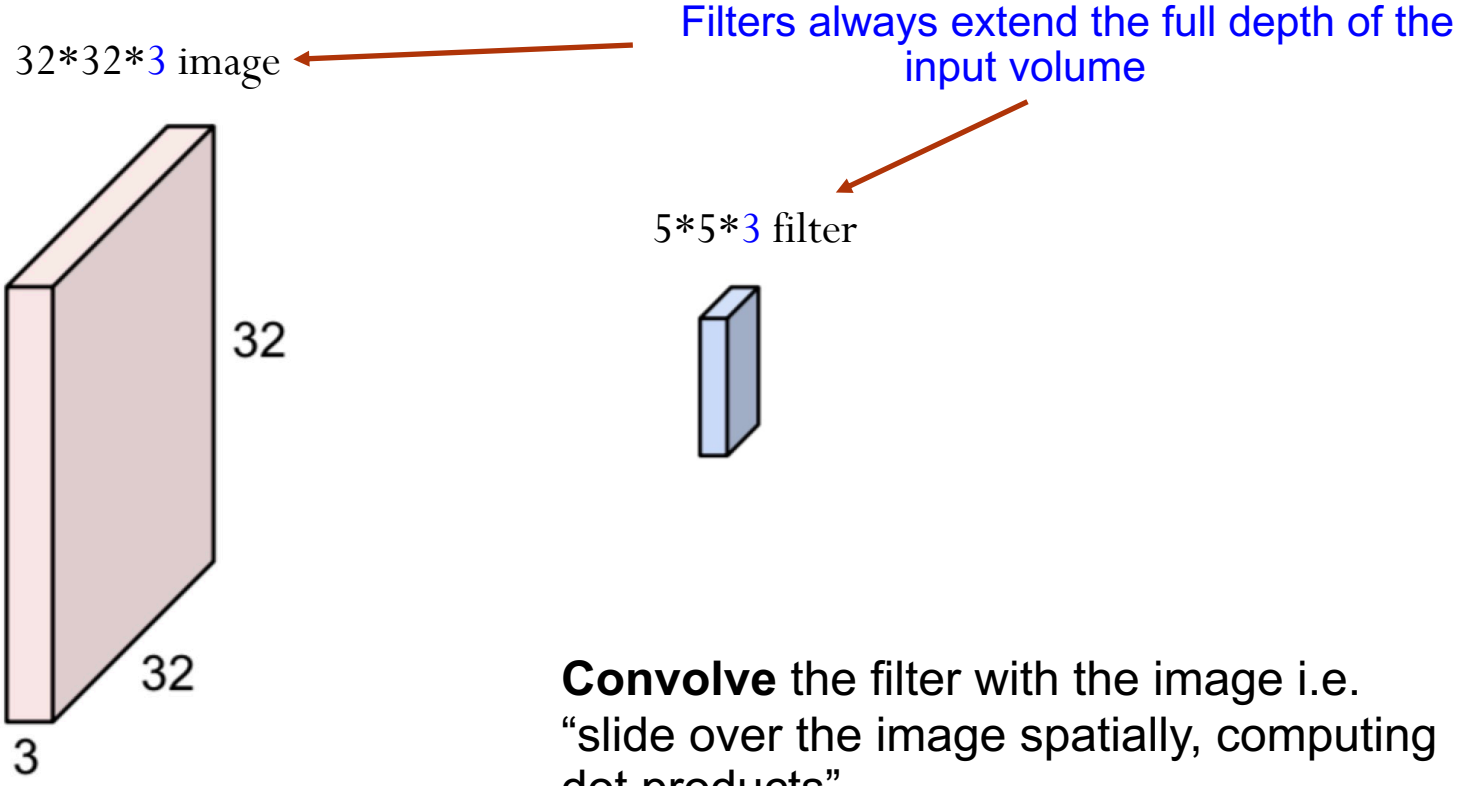
Fully Connected Layers => Convolutions

- Based on the relationships between fully connected layers and convolutions we just discussed, can you think of a way to *convert* fully connected layers to convolutions
  - By replacing fully connected layers with convolutions, we will be able to output heatmaps of class probabilities



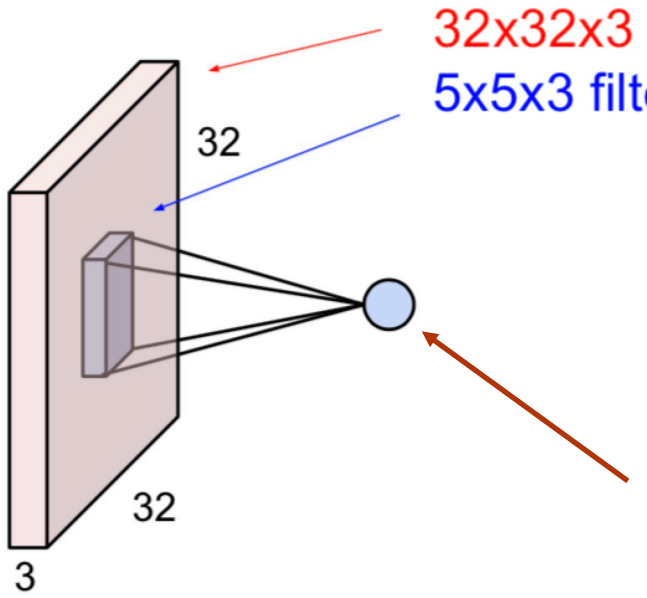
Jonathan Long, Evan Shelhamer, Trevor Darrell, Fully Convolutional Networks for Semantic Segmentation, arXiv preprint 2016

Convolution Layer



Convolve the filter with the image i.e. "slide over the image spatially, computing dot products"

Convolution Layer

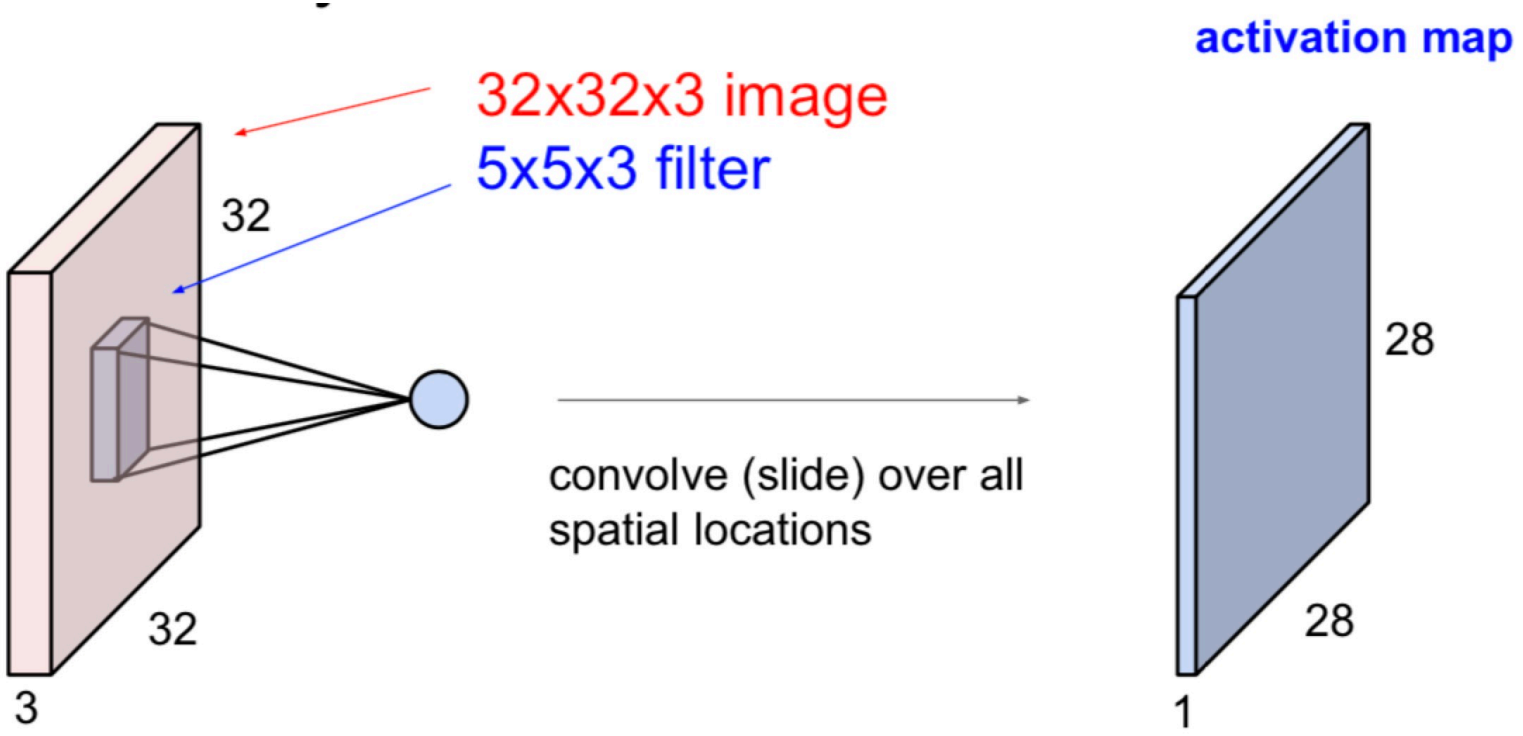


1 number: $w^T x + b$

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)

Convolution Layer

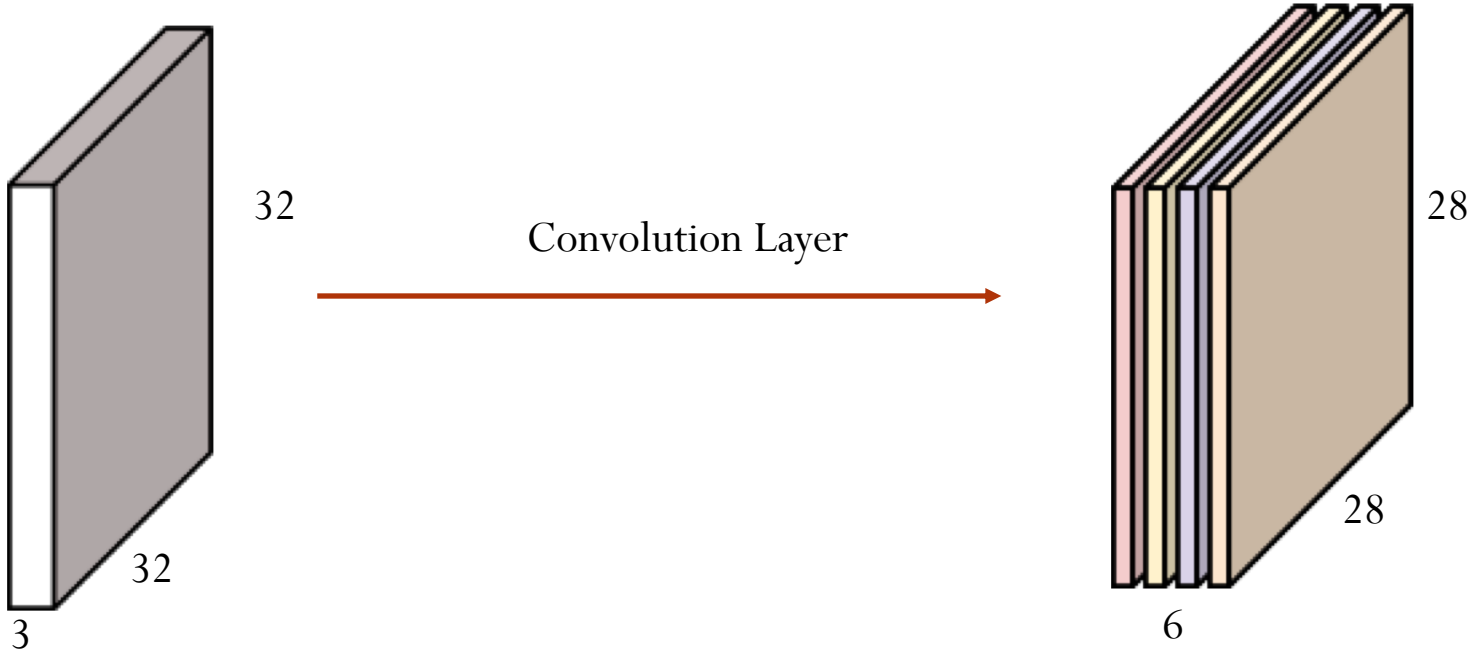


Convolution Layer

Consider a second, green filter

Convolution Layer

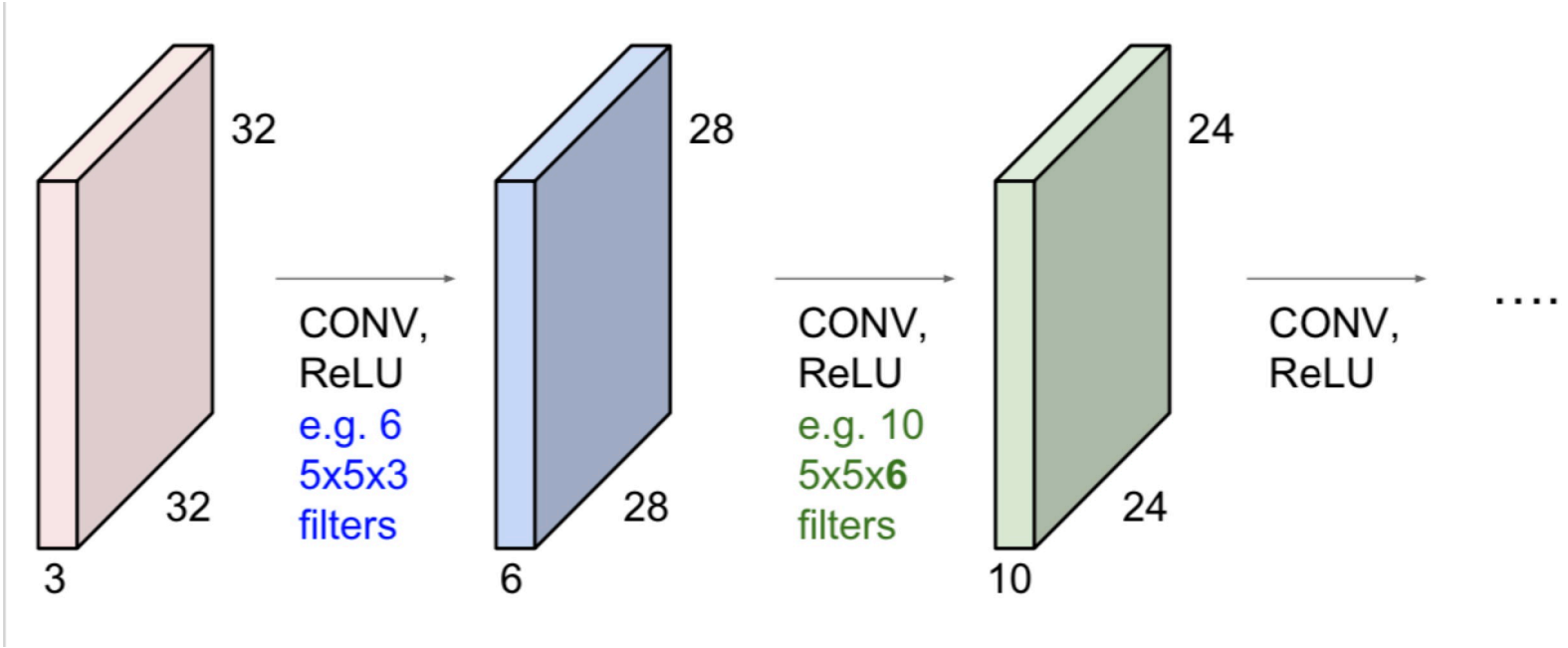
- If we have 6 5x5 filters, we'll get 6 separate activation maps:



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

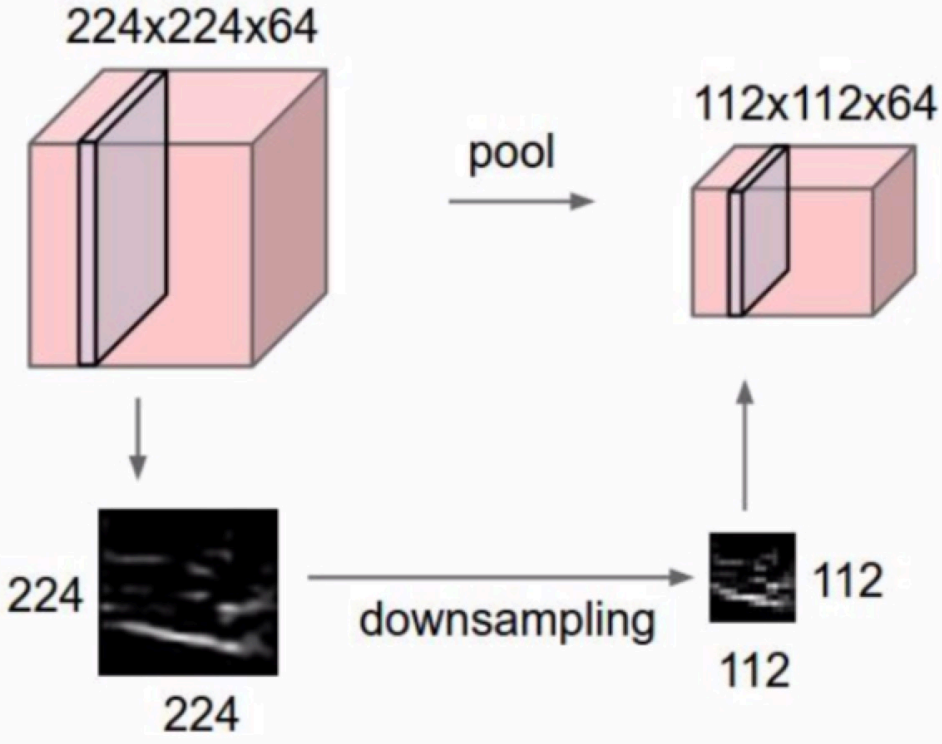
Convolution Layer

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

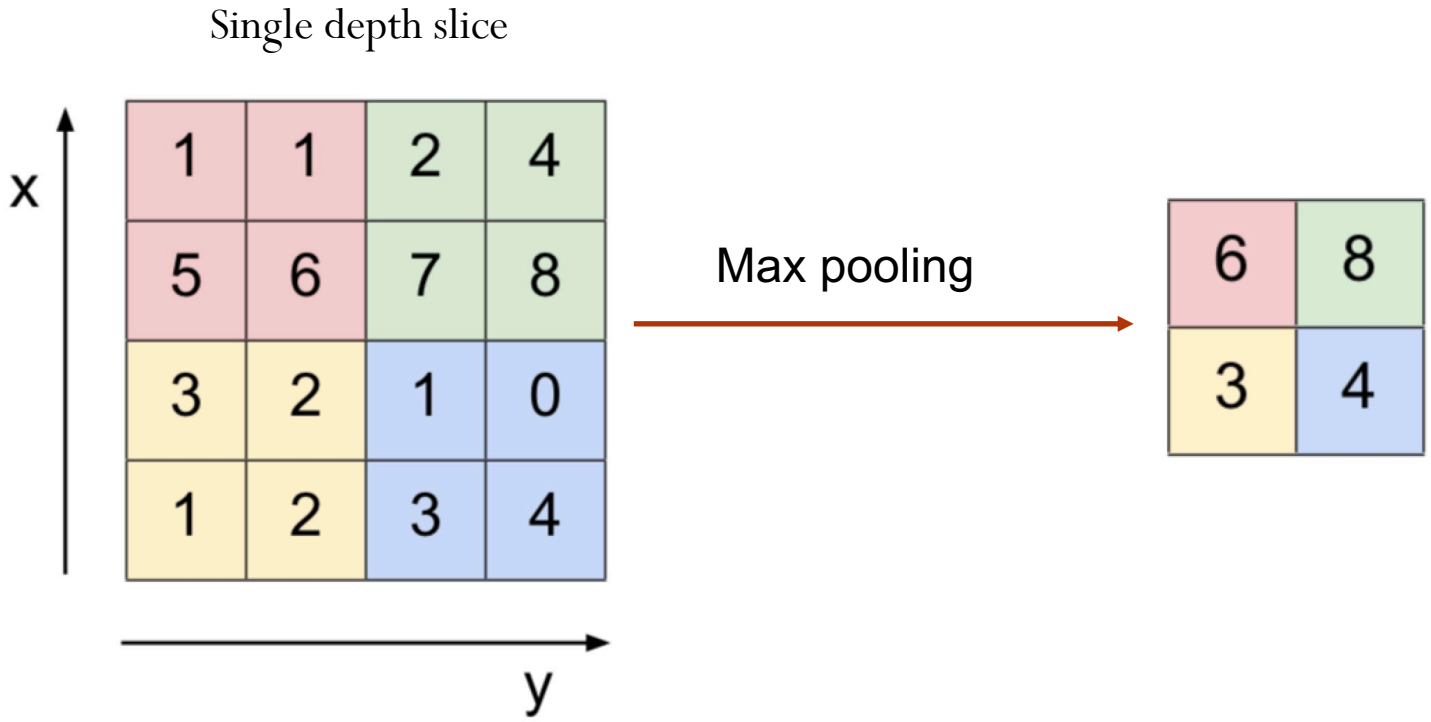


Pooling Layer

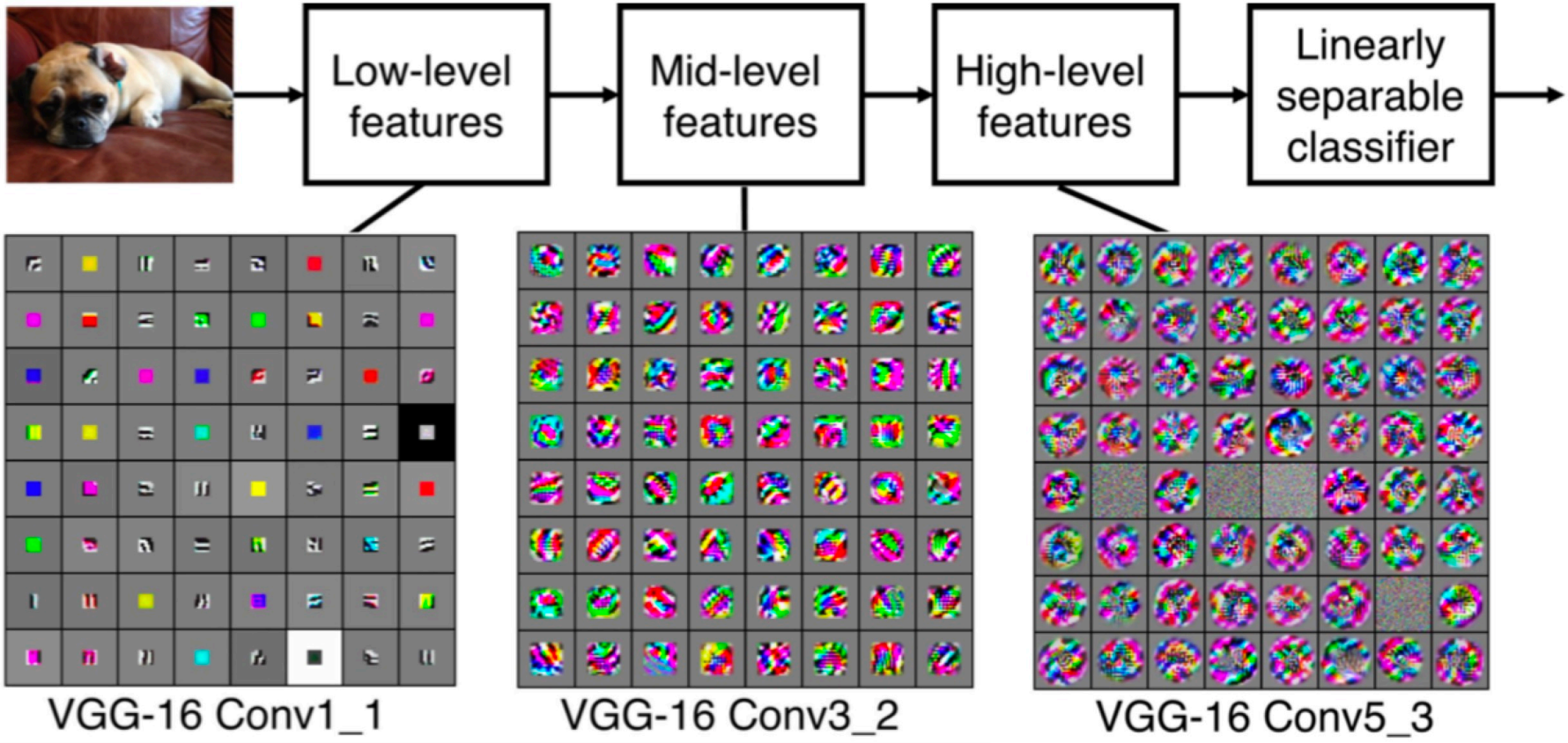
- makes the representations smaller and more manageable
- operates over each activation map independently:



Pooling Layer

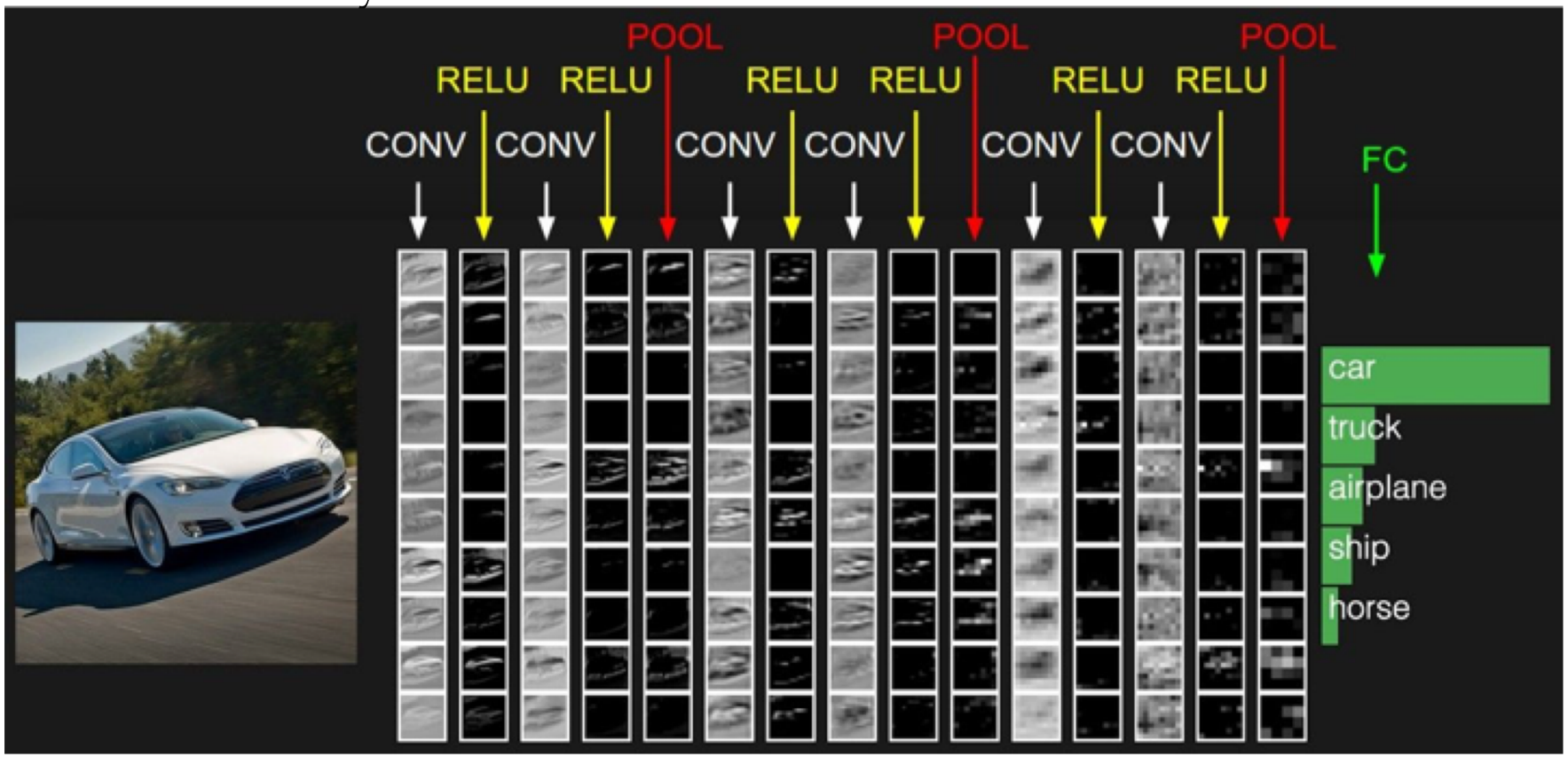


Convolutional Neural Networks



Fully Connected Layer (FC layer)

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

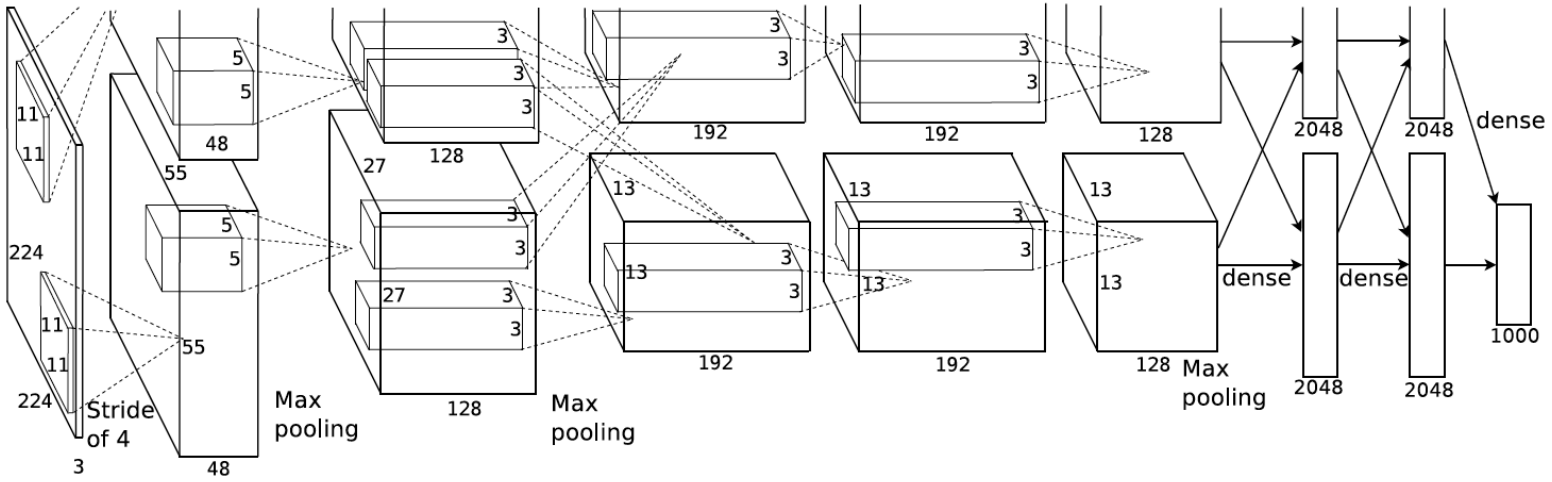


Advanced CNN Architectures

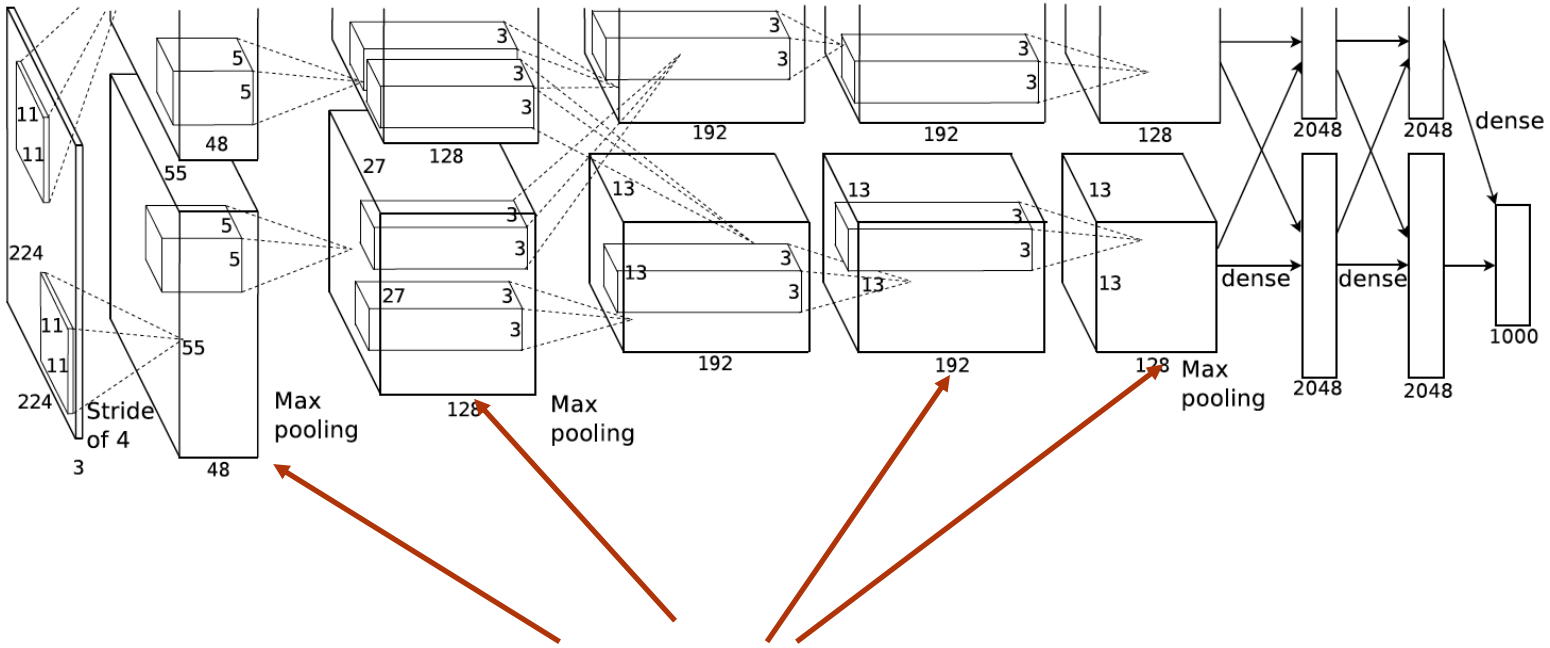
- AlexNet
- Vgg
- GoogleNet
- ResNet

AlexNet

- Architecture:
  - CONV1 MAX POOL1 NORM1
  - CONV2 MAX POOL2 NORM2
  - CONV3 CONV4 CONV5
  - Max POOL3
  - FC6 FC7 FC8

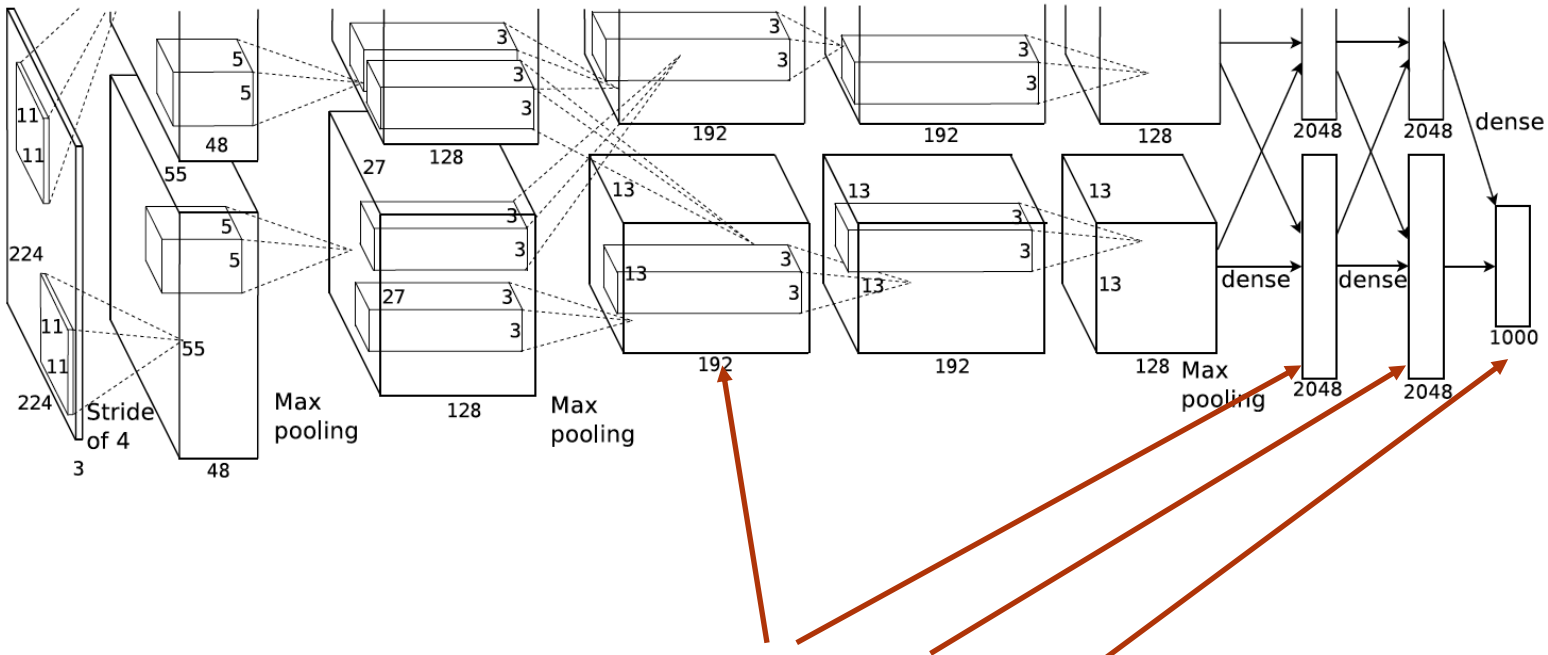


AlexNet



CONV1, CONV2, CONV4, CONV5:
Connections only with feature maps on same GPU

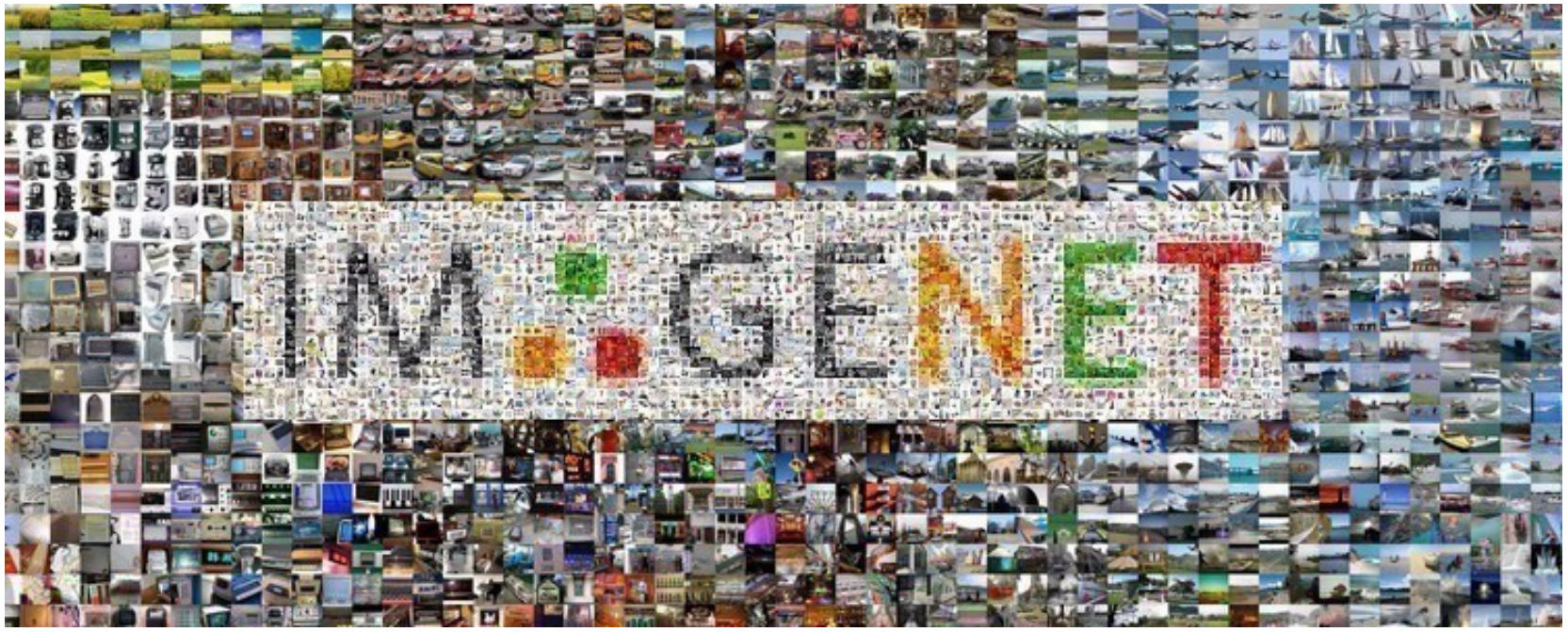
AlexNet



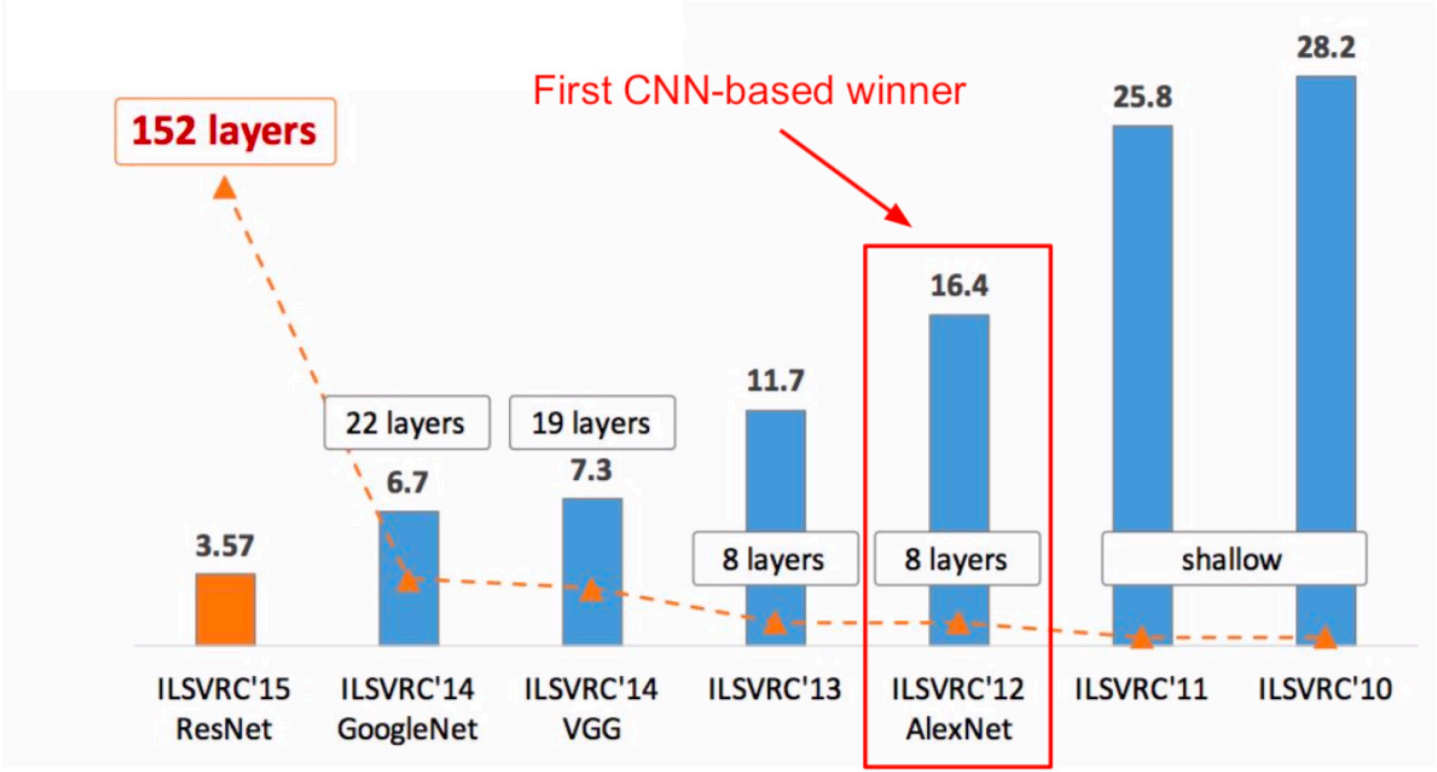
CONV3, FC6, FC7, FC8: Connections with all feature maps in preceding layer, communication across GPUs

ImageNet

- The ImageNet is a large visual database designed for use in visual object recognition research
- More than 14 million images
- more than 20,000 categories



ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



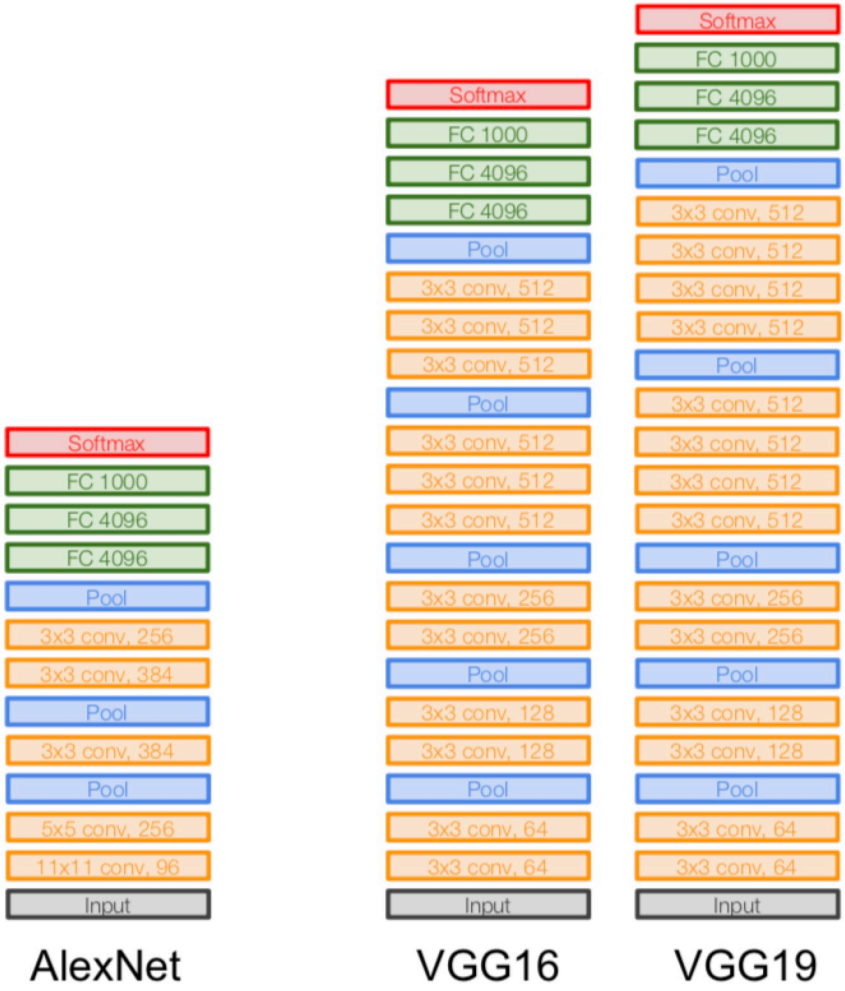
AlexNet

- **Details/Retrospectives:**
  - first use of ReLU
  - used Norm layers (not common anymore)
  - heavy data augmentation
  - 7 CNN ensemble: 18.2% -> 15.4%

VGGNet

[Simonyan and Zisserman, 2014]

- Small filters, Deeper networks
  - 8 layers (AlexNet)->>
    16 - 19 layers (VGG16Net)
  - Only 3x3 CONV stride 1, pad 1 and 2x2 MAX POOL stride 2
  - 15.4% top 5 error for AlexNet
    -> 7.3% top 5 error



VGGNet

[Simonyan and Zisserman, 2014]

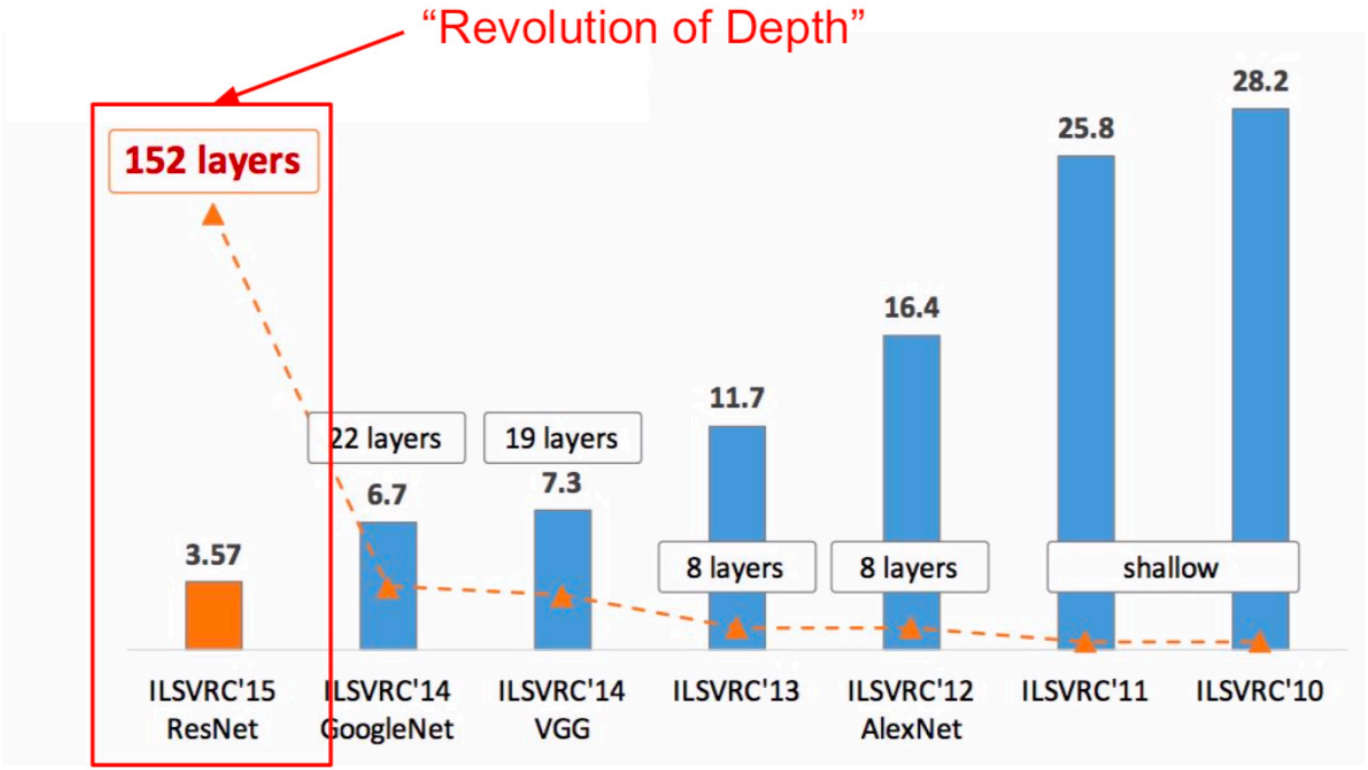
- Q: Why use smaller filters? (3x3 conv)
- Stack of three 3x3 conv (stride 1) layers has same **effective receptive field** as one 7x7 conv layer
- But deeper, more non-linearities
- And fewer parameters: $3 * (3^2C^2)$ vs. 7^2C^2 for C channels per layer

VGGNet

[Simonyan and Zisserman, 2014]

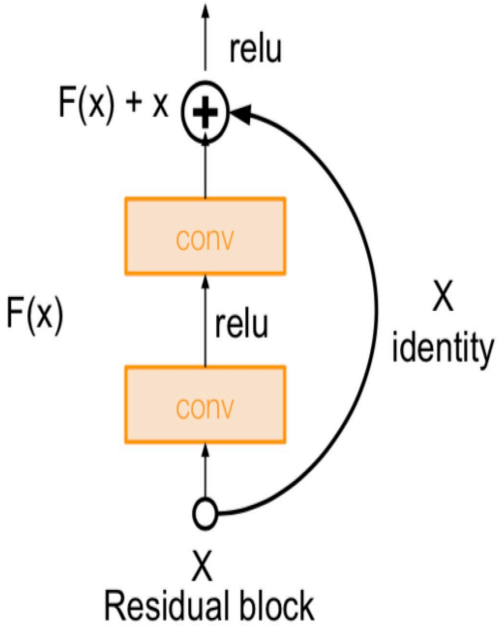
- Details:
  - Use VGG16 or VGG19 (VGG19 only slightly better, more memory)
  - Use ensembles for best results
  - FC7 features generalize well to other tasks

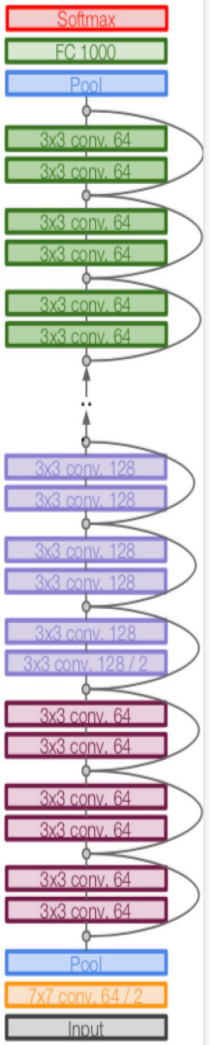
ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



ResNet [He et al., 2015]

- Very deep networks using residua connections
  - 152-layer model for ImageNet
  - -ILSVRC'15 classification winner (3.57% top 5 error)
- Swept all classification and detection competitions in ILSVRC'15 and COCO'15!





ResNet [He et al., 2015]

- What happens when we continue stacking deeper layers on a "plain" convolutional neural network?

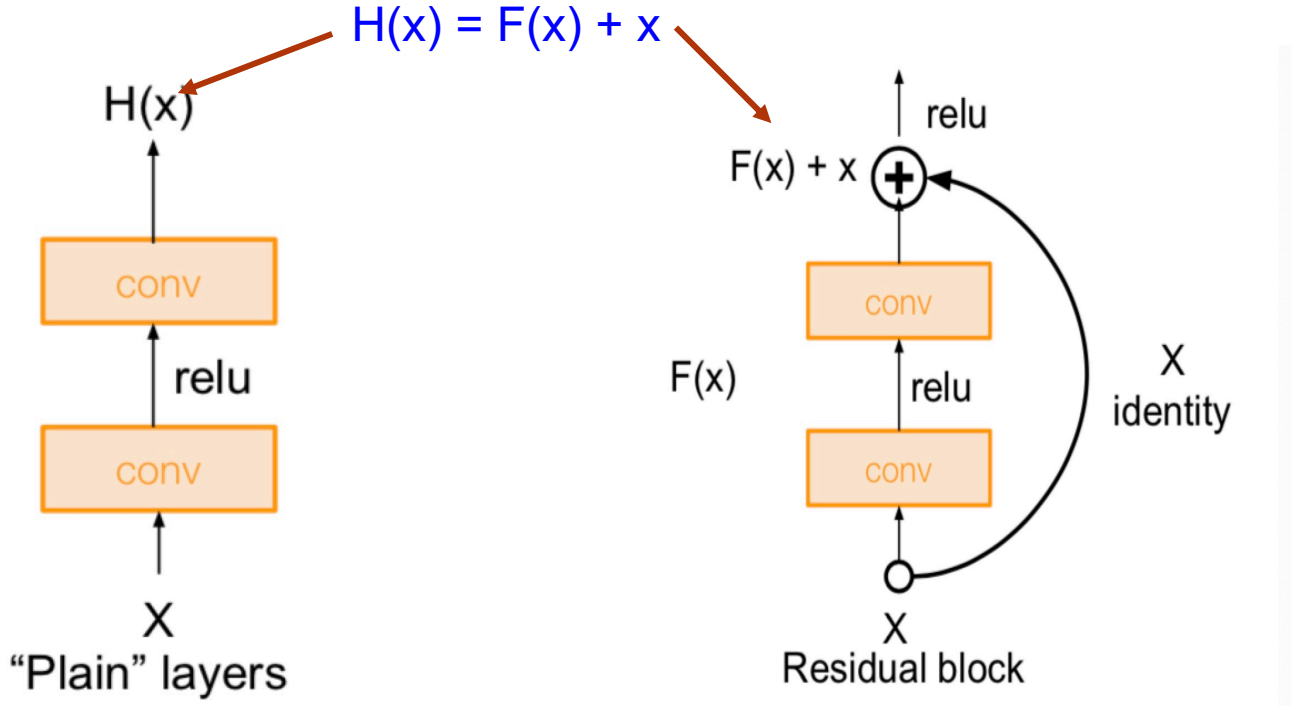
56-layer model performs worse on both training and test error -> The deeper model performs worse, but it's not caused by overfitting!

ResNet [He et al., 2015]

- Hypothesis: the problem is an *optimization* problem, deeper models are harder to optimize
  - The deeper model should be able to perform at least as well as the shallower model.
  - A solution by construction is copying the learned layers from the shallower model and setting additional layers to identity mapping.

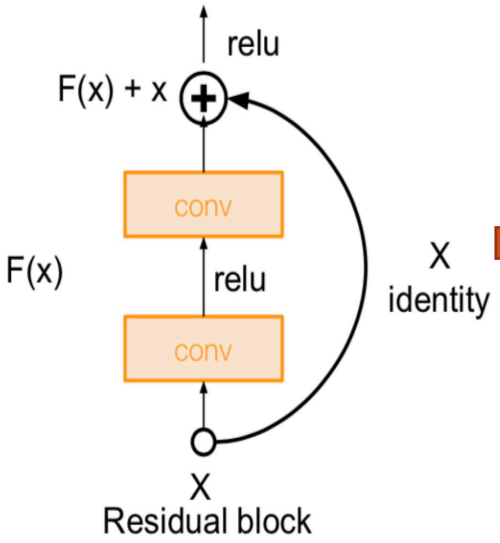
ResNet [He et al., 2015]

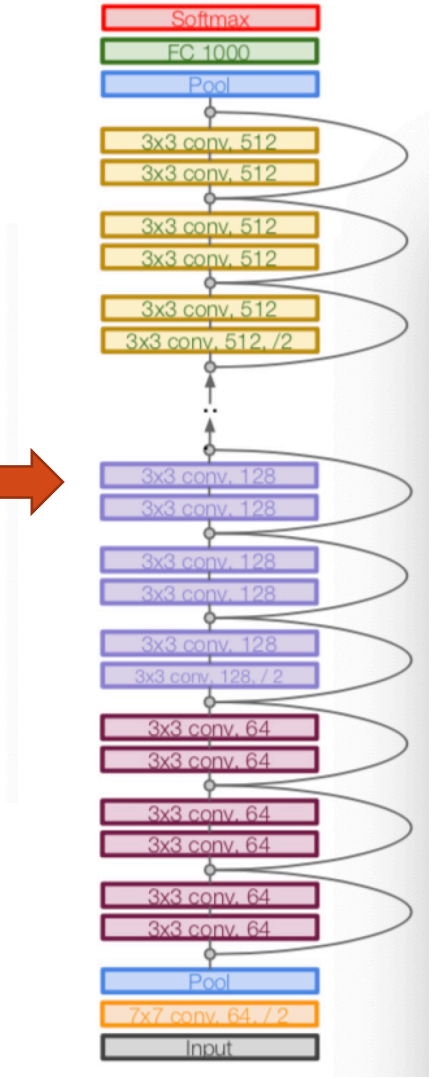
- Solution: Use network layers to fit a residual mapping instead of directly trying to fit a desired underlying mapping



ResNet [He et al., 2015]

- Stack residual blocks
- Able to train very deep networks without degrading
- Deeper networks now achieve lowing training error as expected



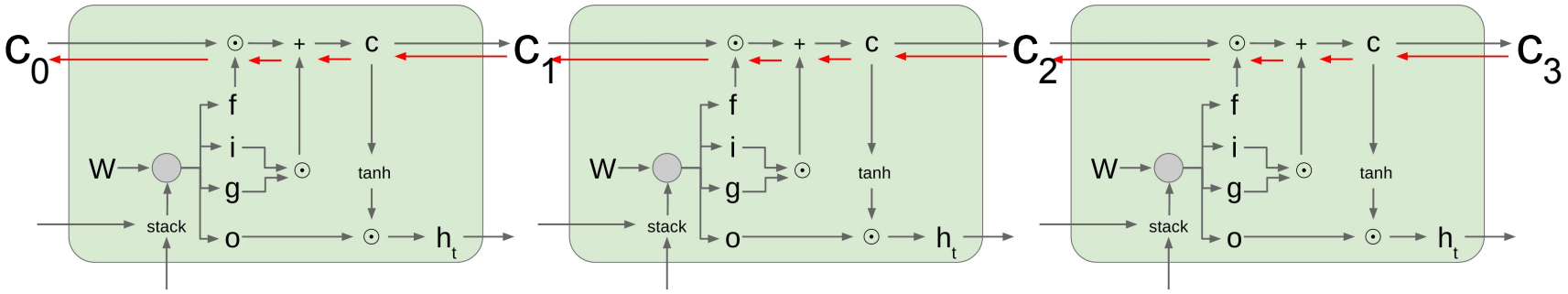


ILSVRC 2015 classification winner (3.6% top 5 error) -- better than “human performance”! (Russakovsky 2014)

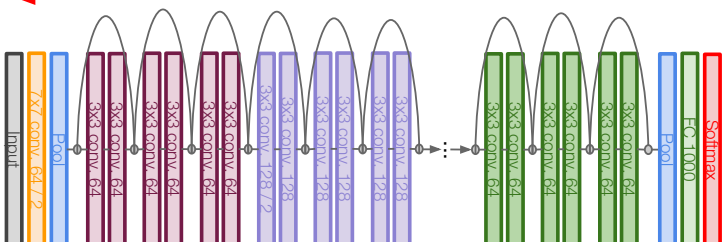
Long Short Term Memory (LSTM): Gradient Flow

- Forget gate gets rid of irrelevant information
- Selectively update cell state
- Output gate returns a filtered version of the cell state

Uninterrupted gradient flow!



Similar to ResNet!



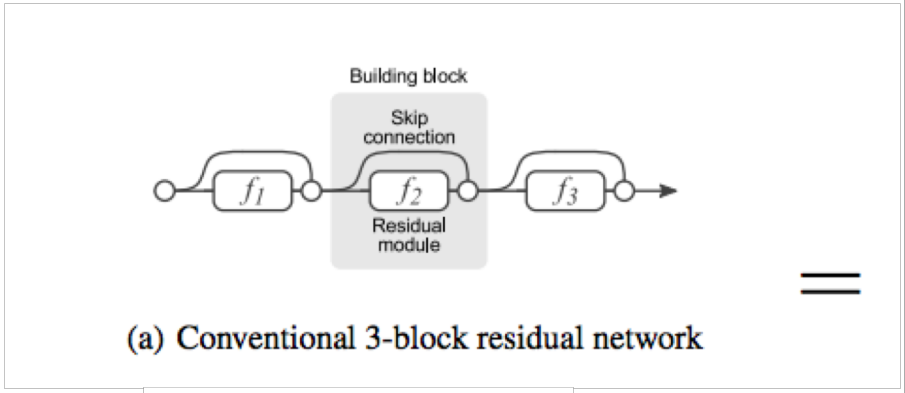
In between:
Highway Networks

$$g = T(x, W_T)$$
$$y = g \odot H(x, W_H) + (1 - g) \odot x$$

Srivastava et al, "Highway Networks", ICML DL Workshop 2015

Why do ResNets work? Some ideas:

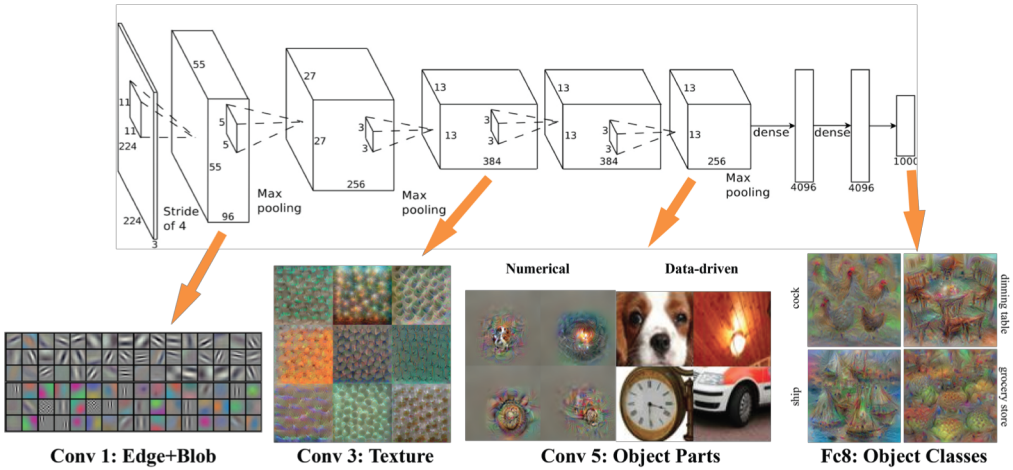
- **They can be seen as implicitly ensembling shallower networks**
- They are able to learn unrolled iterative refinements
- Can model recurrent computations necessary for recognition



Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, arxiv 2016

Why do ResNets work? Some ideas:

- They can be seen as implicitly ensembling shallower networks
- **They are able to learn unrolled iterative estimation**
- Can model recurrent computations necessary for recognition

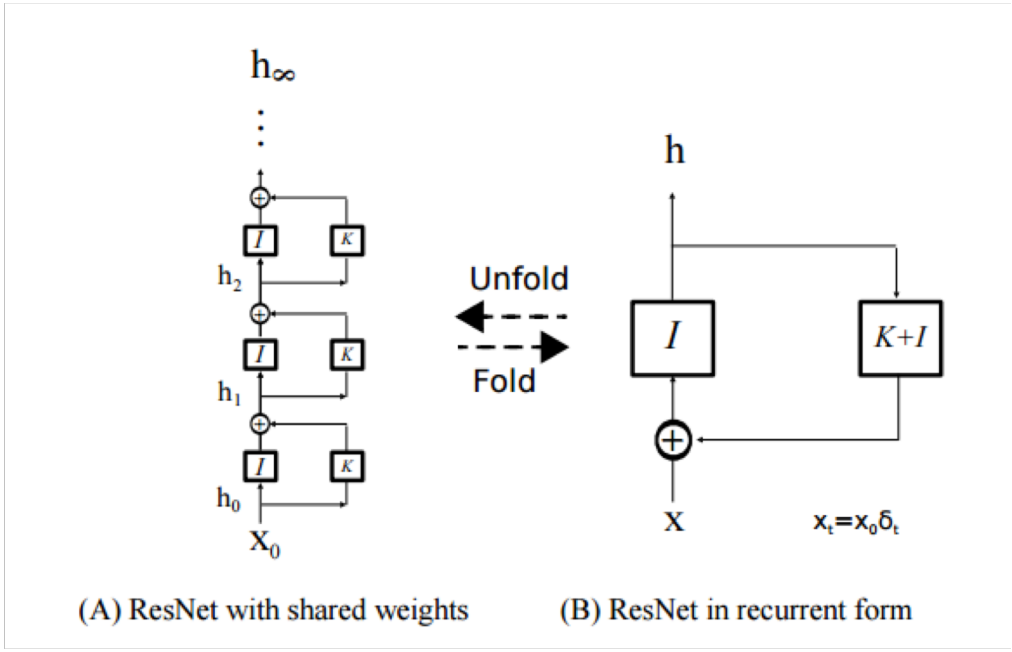


Challenges the "representation view"

Image source: http://vision03.csail.mit.edu/cnn_art/

Why do ResNets work? Some ideas:

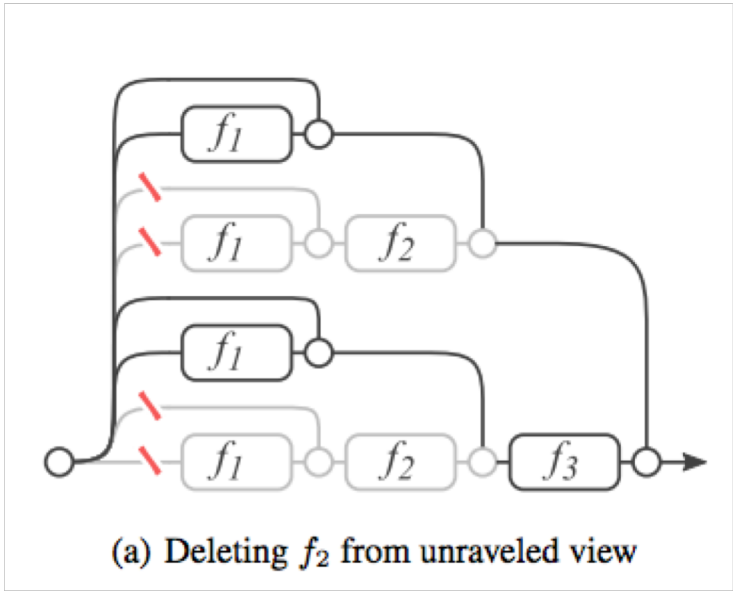
- They can be seen as implicitly ensembling shallower networks
- They are able to learn unrolled iterative estimation
- **Can model recurrent computations useful for recognition**



Qianli Liao, Tomaso Poggio,Bridging the Gaps Between Residual Learning, Recurrent Neural Networks and Visual Cortex,

ResNets as Ensembles

- Can think of ResNets as ensembling subsets of residual modules
- With L residual modules there are 2^L possible subsets of modules
- **If one of modules is removed, there are still 2^{L-1} possible subsets of modules**

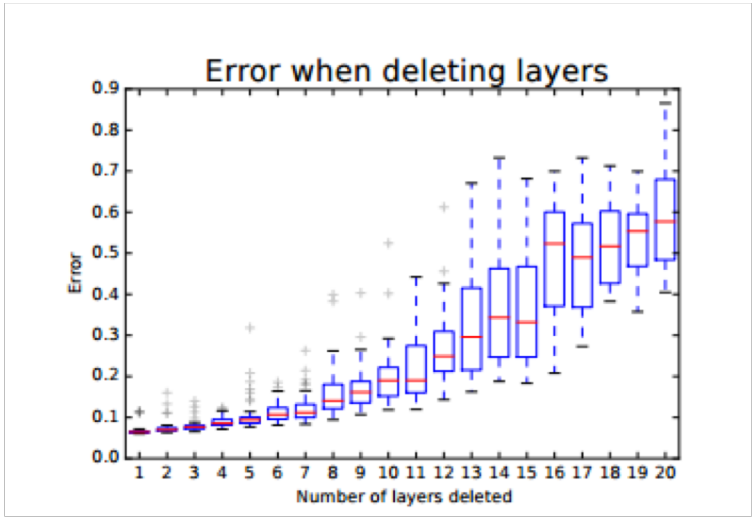


(a) Deleting f_2 from unraveled view

Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, arxiv 2016

ResNets as Ensembles

- Wanted to test this explanation
- **Tried dropping layers**
- Tried reordering layers
- Found effective paths during training are relatively shallow



Performance degrades smoothly as layers are removed

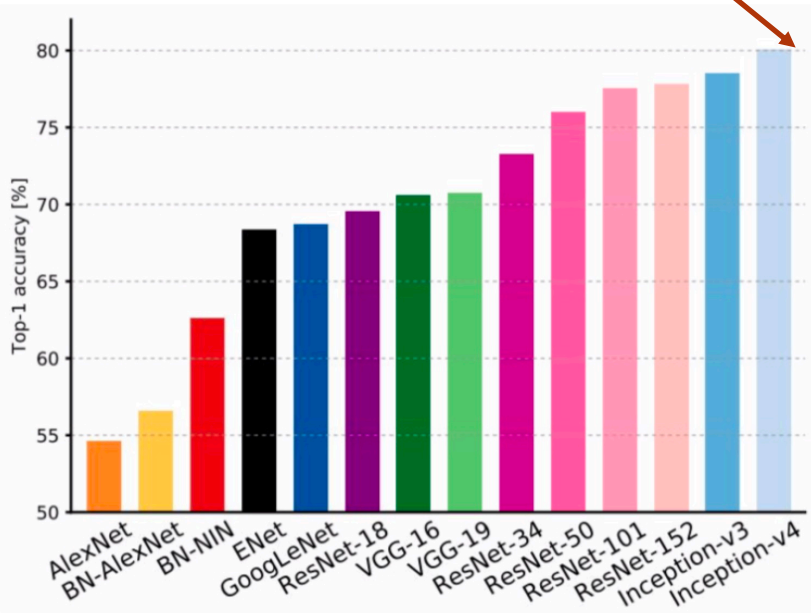
Andreas Veit, Michael Wilber, Serge Belongie, Residual Networks Behave Like Ensembles of Relatively Shallow Networks, arxiv 2016

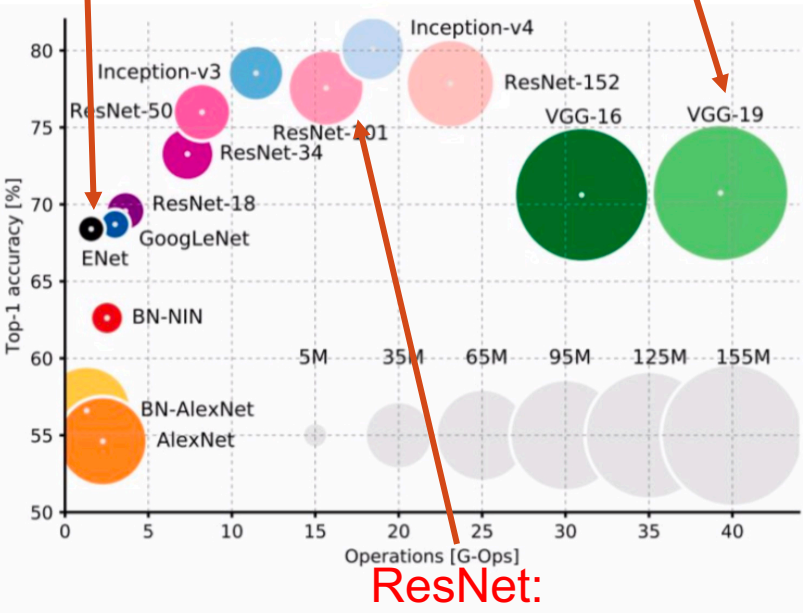
Comparison

GoogleNet: most efficient

Inception-v4: Resnet + Inception!

VGG: Highest memory, most operations





ResNet: Moderate efficiency depending on model, highest accuracy

An Analysis of Deep Neural Network Models for Practical Applications, 2017

Issues with CNN

- Computing the activations of a single convolutional filter is much more expensive than with traditional MLPs

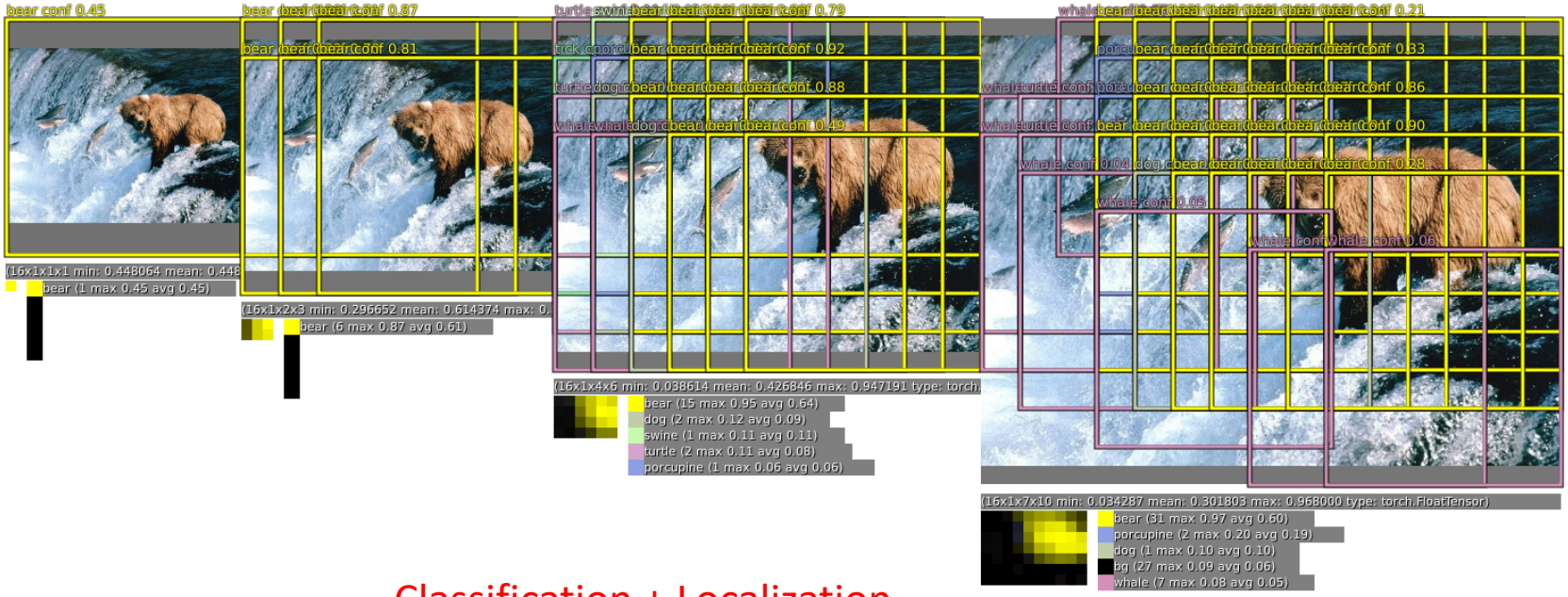
- Many tuning parameters
  - # of filters:
    - Model complexity issue (overfitting vs underfitting)
  - Filter shape:
    - the right level of "granularity" in order to create abstractions at the proper scale, given a particular dataset
    - Usually 5x5 for MNIST at 1st layer
  - Max-pooling shape:
    - typical: 2x2; maybe 4x4 for large images

Application

- Object Detection
- Image Caption
- Segmentation

Object Detection And Localization With ConvNets

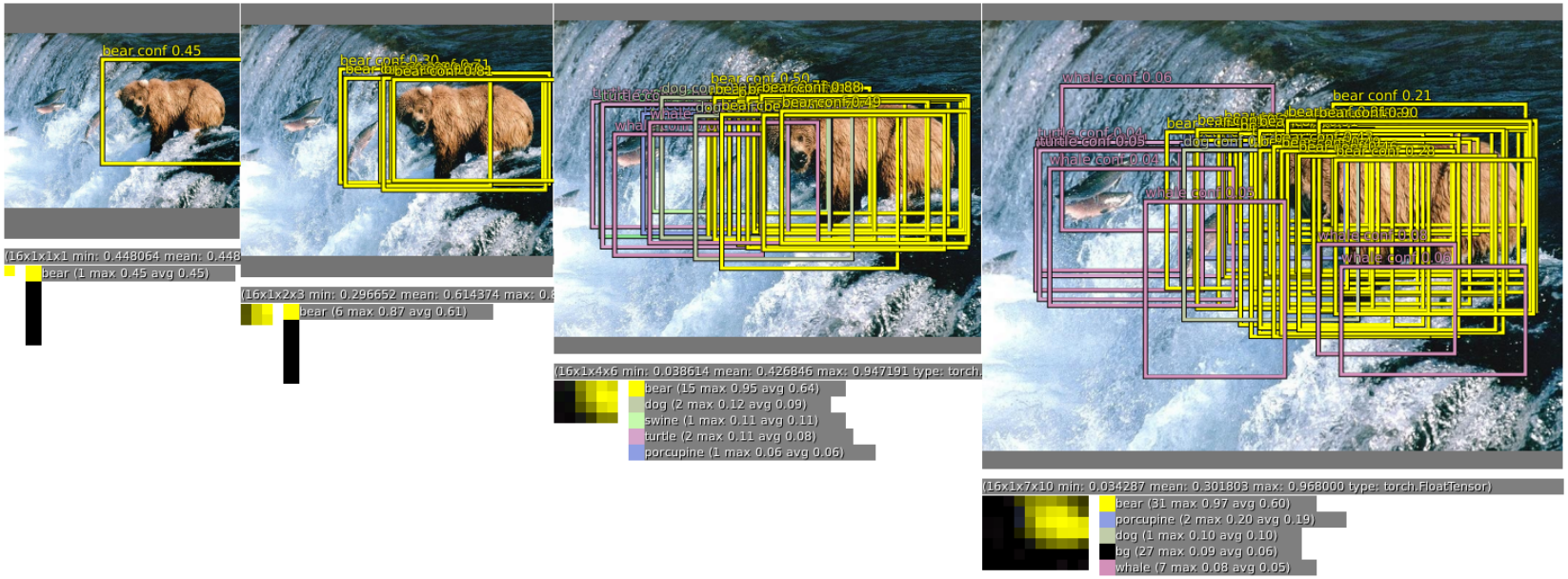
- Apply convnet with a sliding window over the image at multiple scales
- Important note: it's very cheap to slide a convnet over an image
  - Just compute the convolutions over the whole image and replicate the fully-connected layers



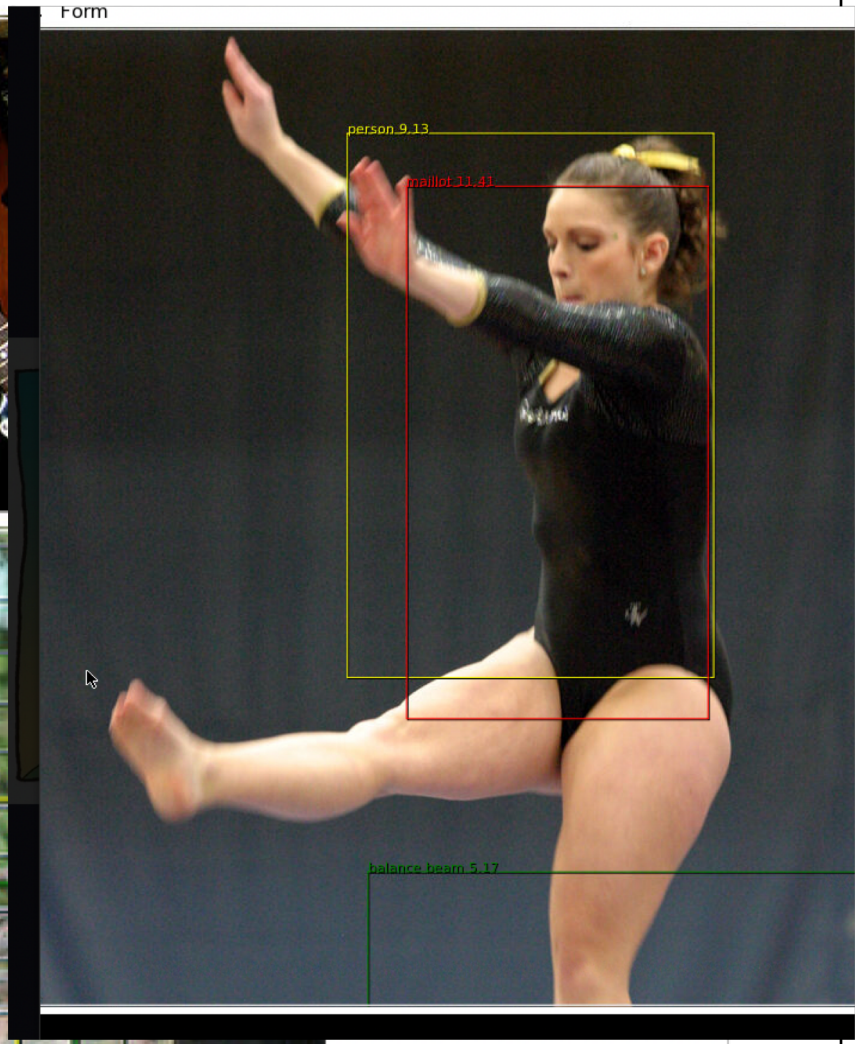
Classification + Localization

Object Detection And Localization With ConvNets

- Apply convnet with a sliding window over the image at multiple scales
- For each window, predict a class and bounding box parameters
- Evenif the object is not completely contained in the viewing window, the convnet can predict where it thinks the object is.

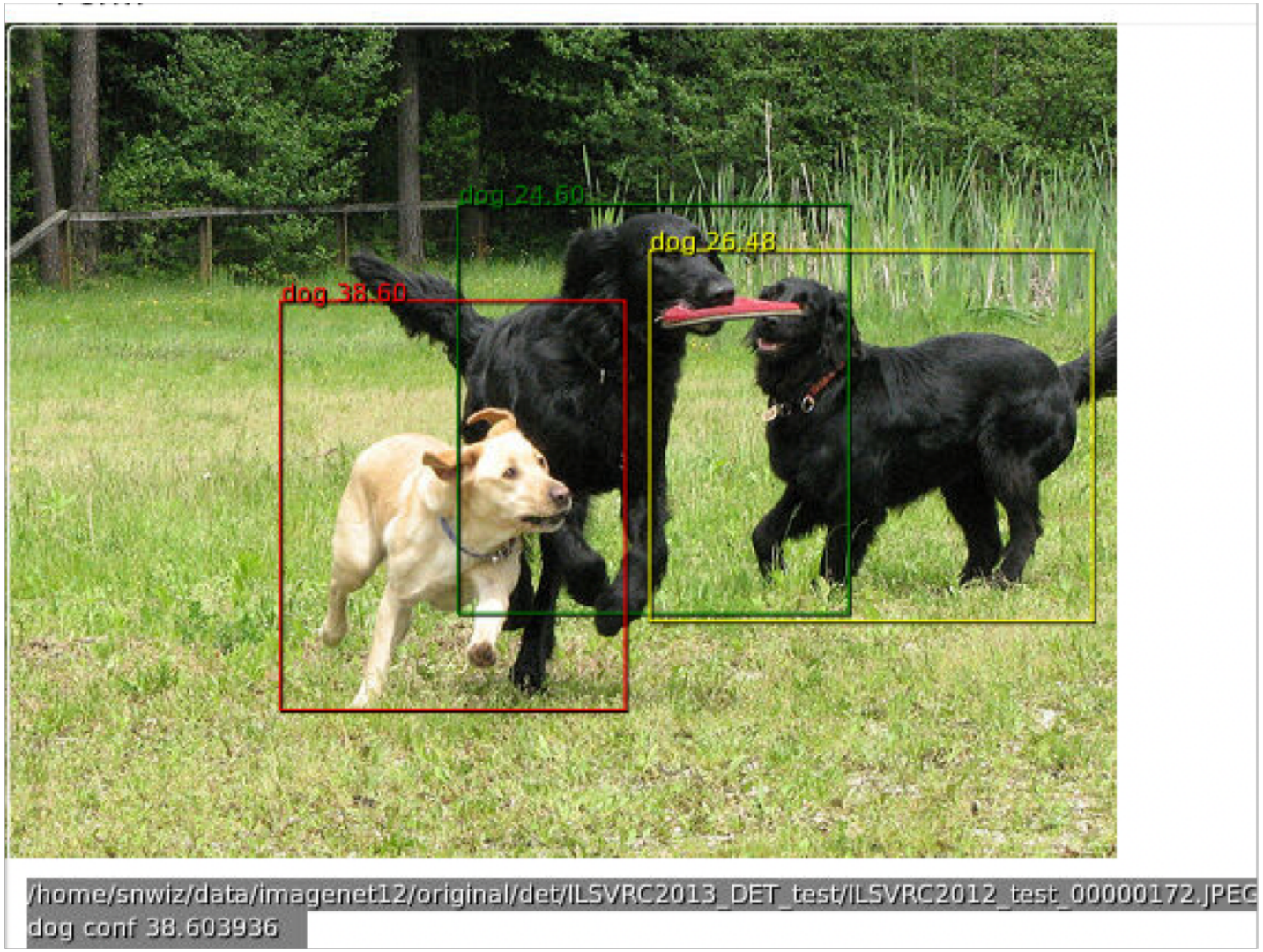


Objection Detection



Object Detection

Object Detection



Segmentation

Summary: CNN Architectures

- VGG, GoogLeNet, ResNet all in wide use, available in model zoos and ResNet current best default
- Trend towards extremely deep networks
- Significant research centers around design of layer / skip
- connections and improving gradient flow
- Even more recent trend towards examining necessity of depth vs. width and residual connection
- Other topic: Inception module

Thank You!