Lecture 10:
Deep Learning

COURSE: BIOMEDICAL DATA SCIENCE

Parisa Rashidi

FALL 2019

Deep Neural Network

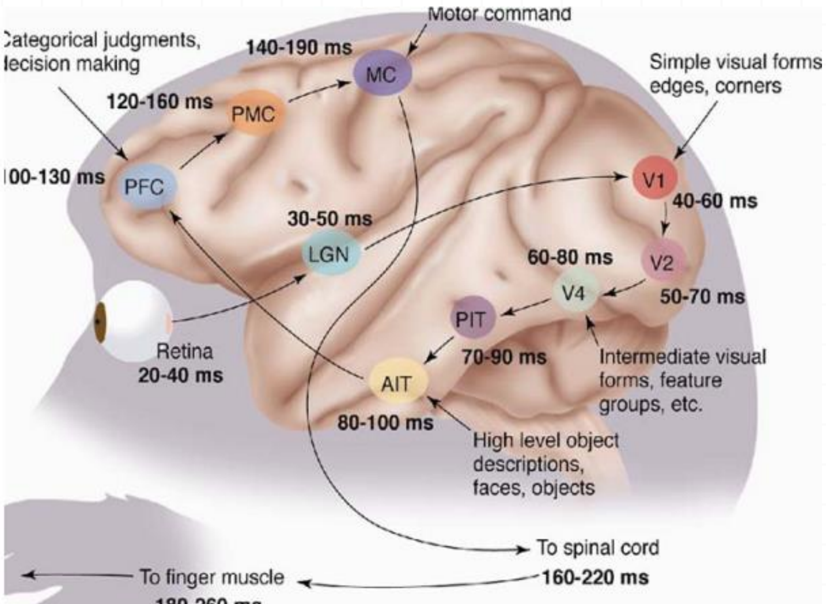
Material partially based on:

-Raschka, Sebastian. Python Machine Learning (p. 18). Packt Publishing.

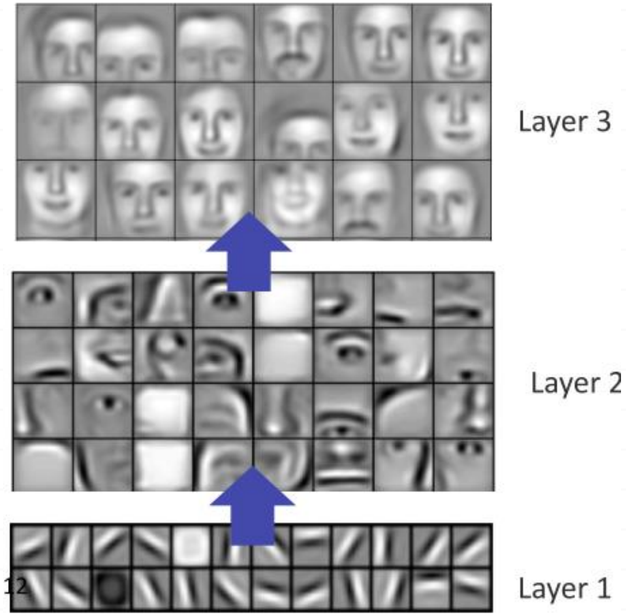
-Stanford CS231n: Convolutional Neural Networks for Visual Recognition, 2019.

Learning Features instead of Feature Engineering

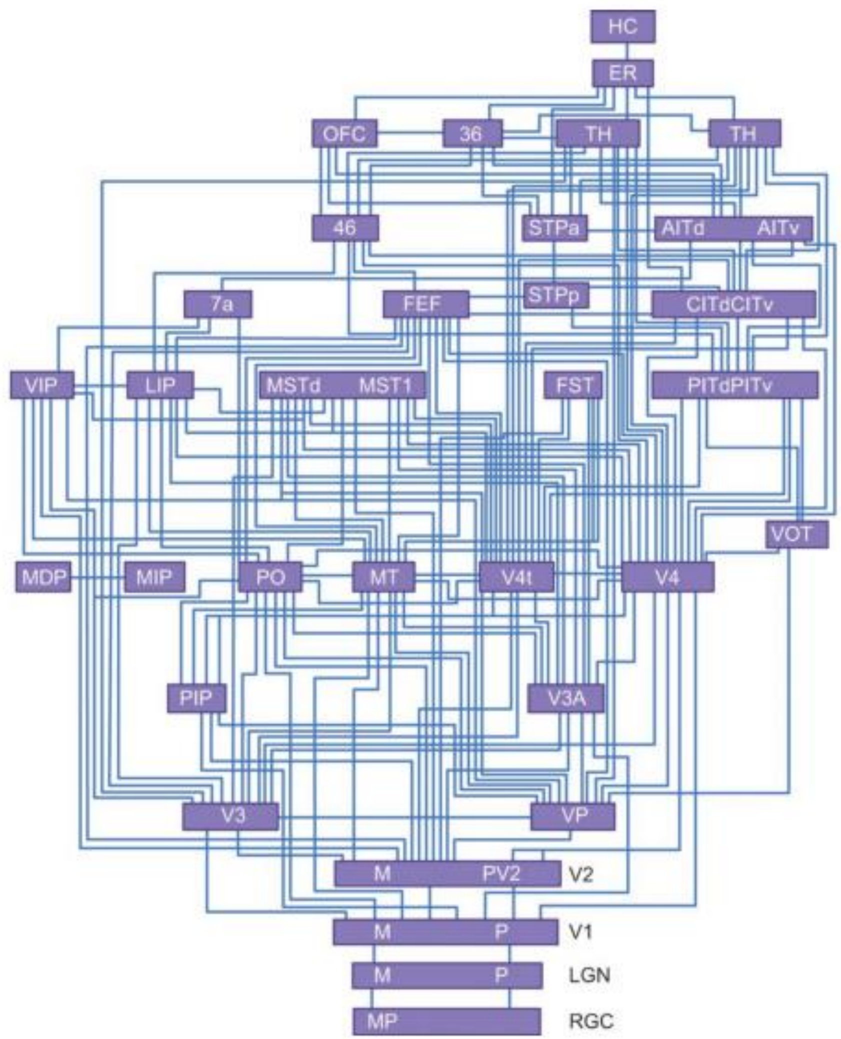
- The visual system is deep (around 10 layers)
- What is the learning algorithm of the neo-cortex?



[picture from Simon Thorpe]



Visual Pathway is Complex!



Banich, Marie T.; Compton, Rebecca J.. Cognitive Neuroscience (Kindle Location 7070). Cambridge University Press.

A bit of history:

Hubel & Wiesel,

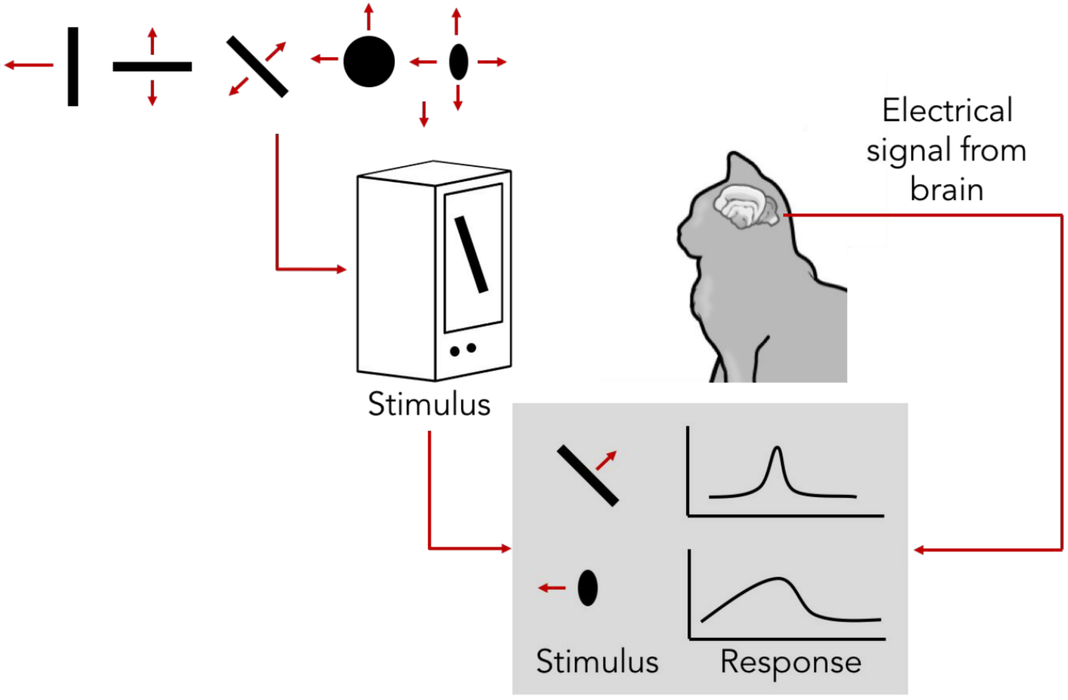
1959

RECEPTIVE FIELDS OF SINGLE NEURONES IN
THE CAT'S STRIATE CORTEX

1962

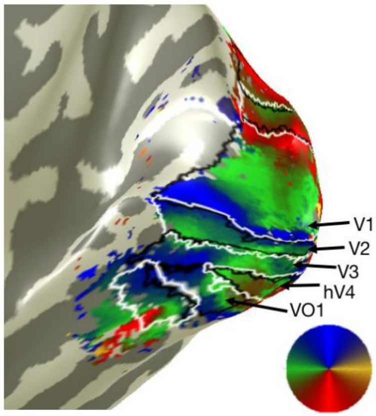
RECEPTIVE FIELDS, BINOCULAR INTERACTION
AND FUNCTIONAL ARCHITECTURE IN
THE CAT'S VISUAL CORTEX

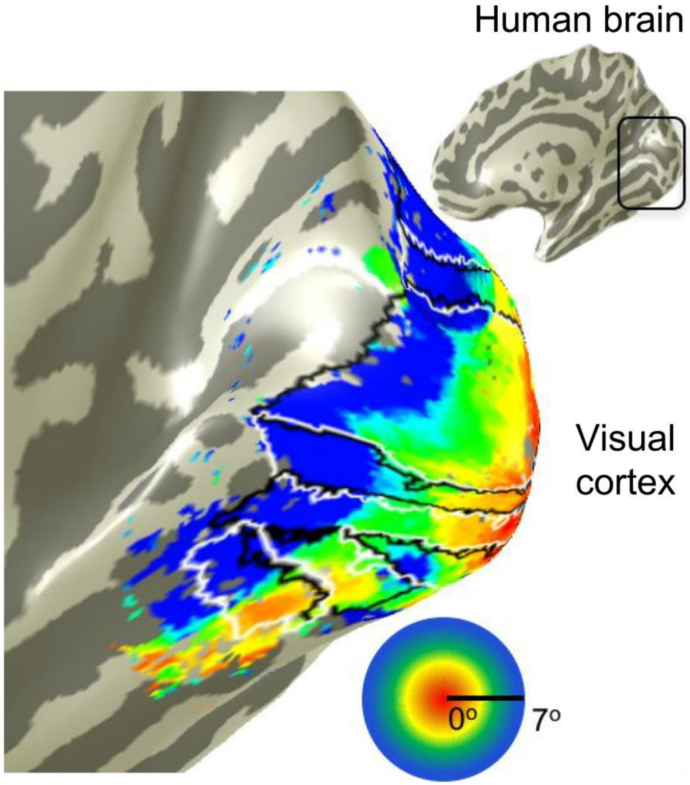
1968...



A bit of history

Topographical mapping in the cortex:
nearby cells in cortex represent
nearby regions in the visual field





Retinotopy images courtesy of Jesse Gomez in the Stanford Vision & Perception Neuroscience Lab.

Hierarchical organization

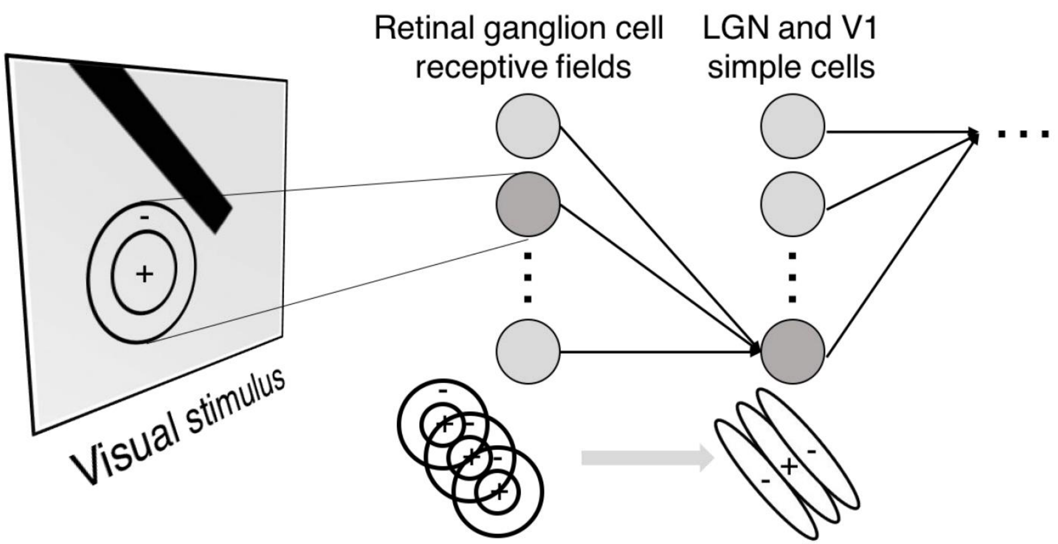
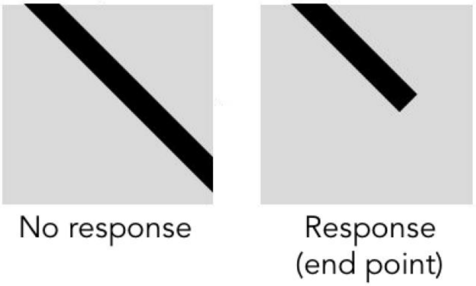


Illustration of hierarchical organization in early visual pathways by Lane McIntosh, copyright CS231n 2017

Simple cells:
Response to light orientation

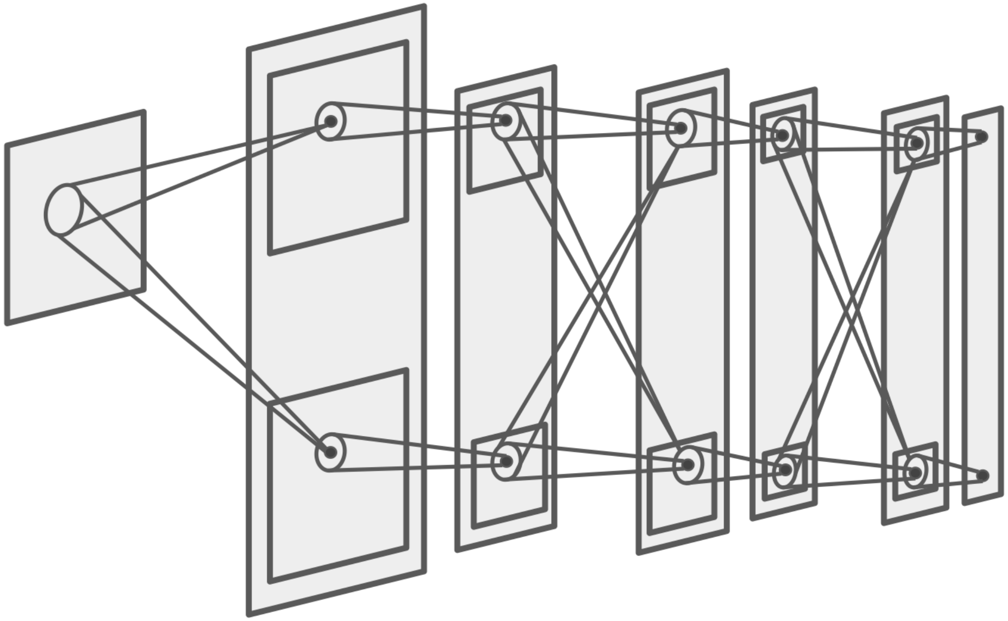
Complex cells:
Response to light orientation and movement

Hypercomplex cells:
response to movement with an end point



A bit of history:

Neocognitron
[Fukushima 1980]

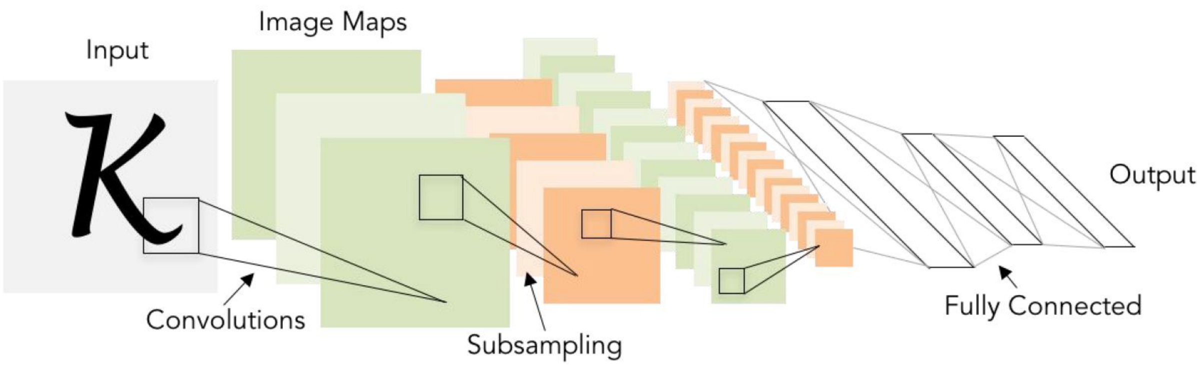


"sandwich" architecture (SCSCSC…)
simple cells: modifiable parameters
complex cells: perform pooling

A bit of history:

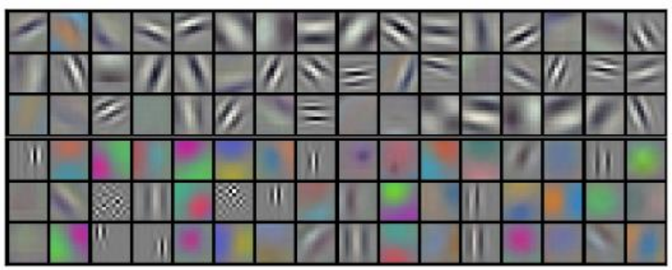
Gradient-based learning applied to document recognition

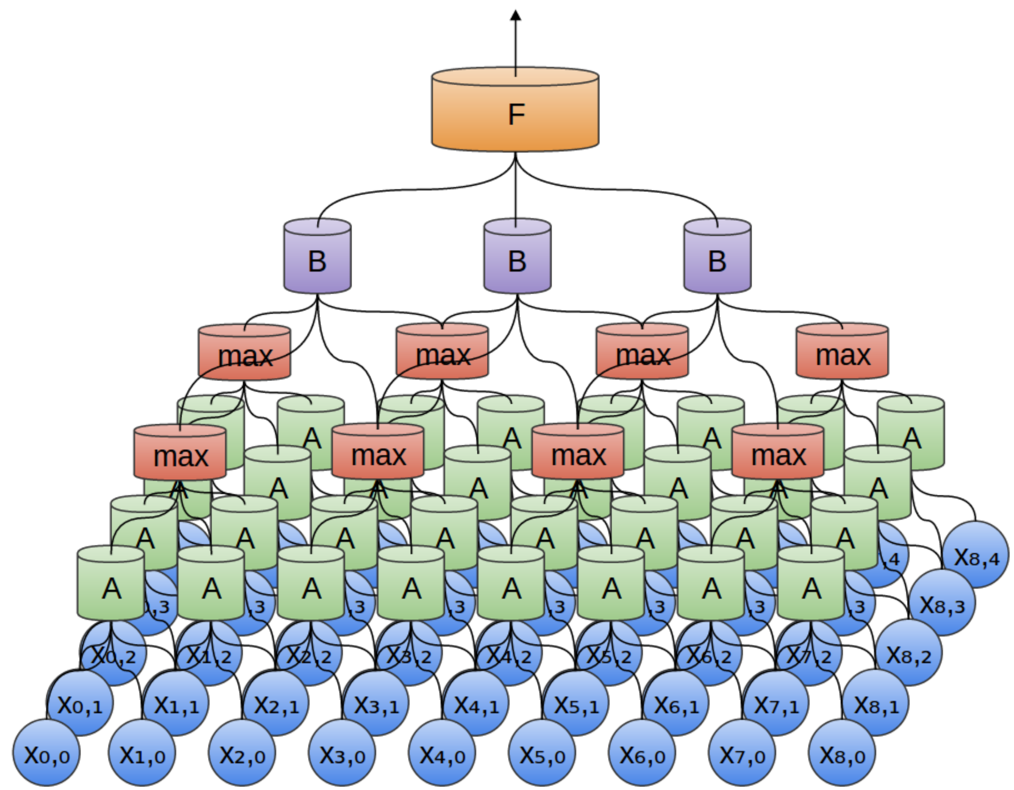
[LeCun, Bottou, Bengio, Haffner 1998]



LeNet-5

Convolutional NN





http://colah.github.io/posts/2014-07-Conv-Nets-Modular/

Why Deep Learning Works

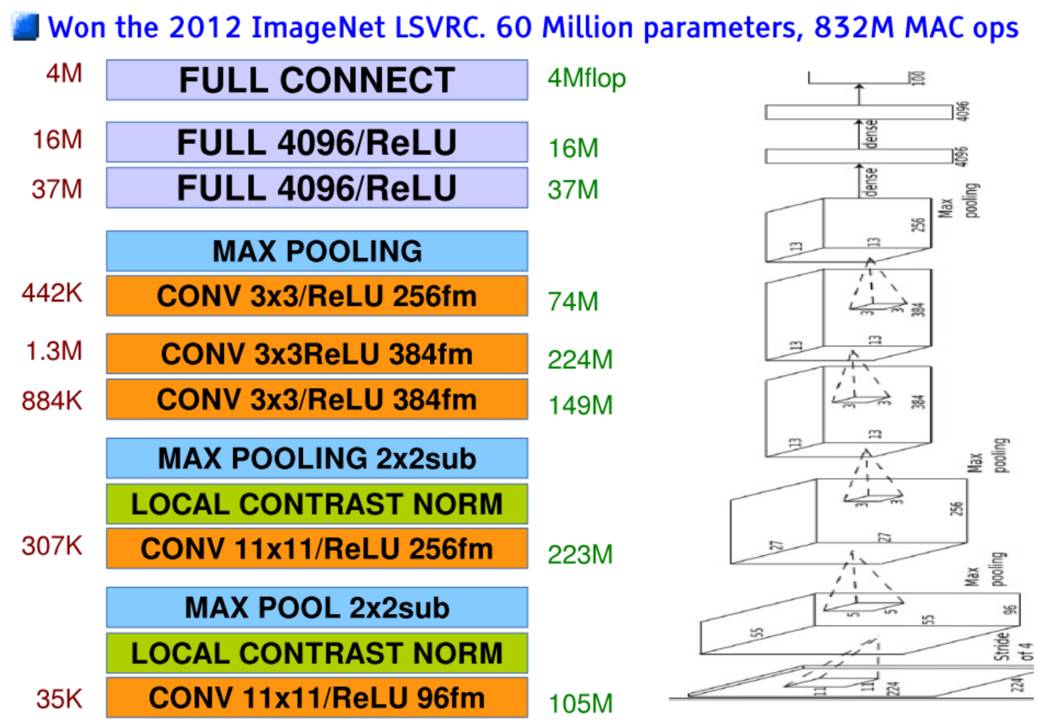
- Some tricks
  - GPU utilization
  - Dropout
  - Pre-training each layer (not popular anymore)

Notable Architectures

ImageNet Challenge: 1000 classes, 1.5M training images

AlexNet by Krizhevsky et al. 2012

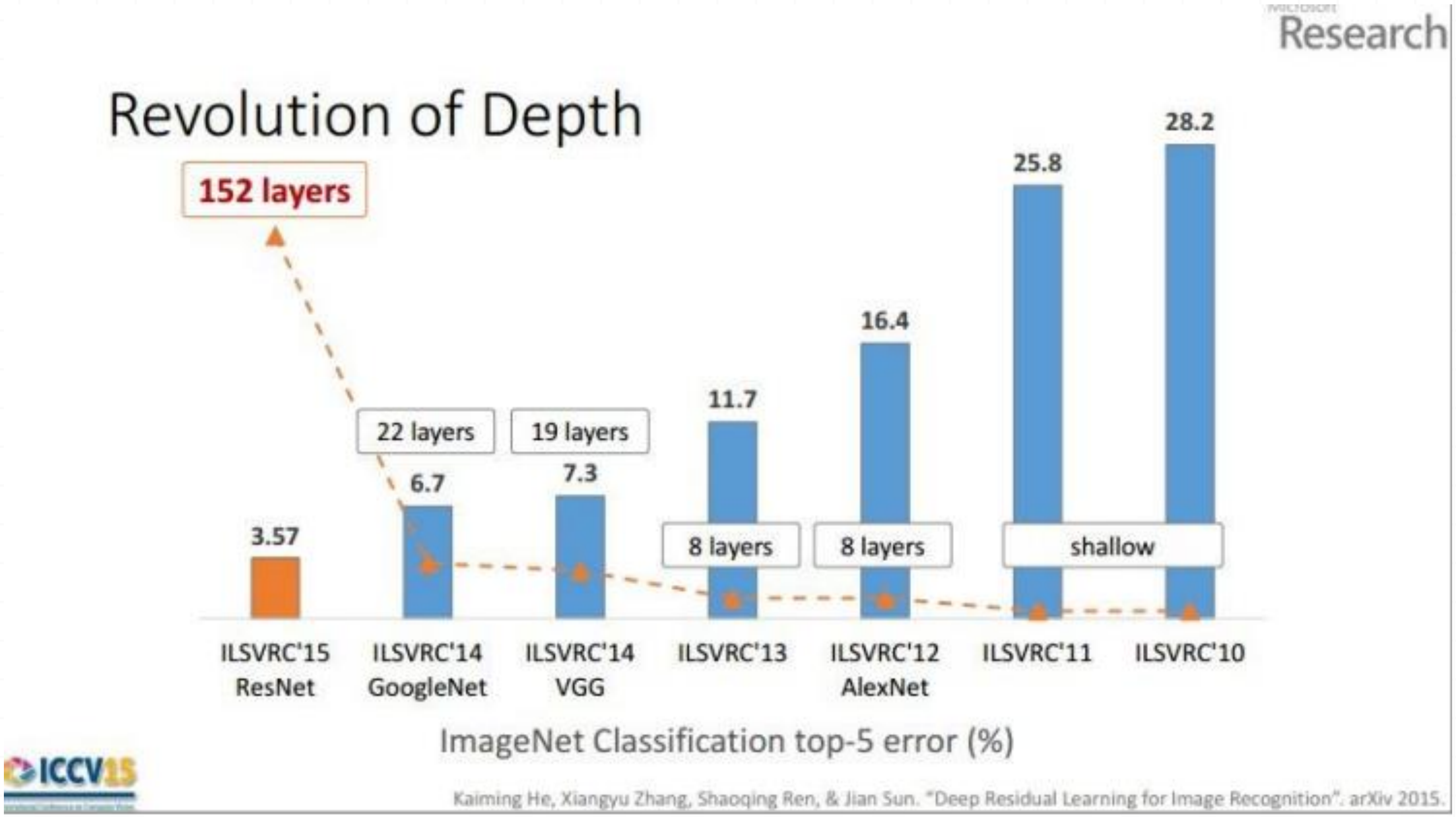
- 650K neurons, 832M synapses, 60M parameters
- Error rate 15%



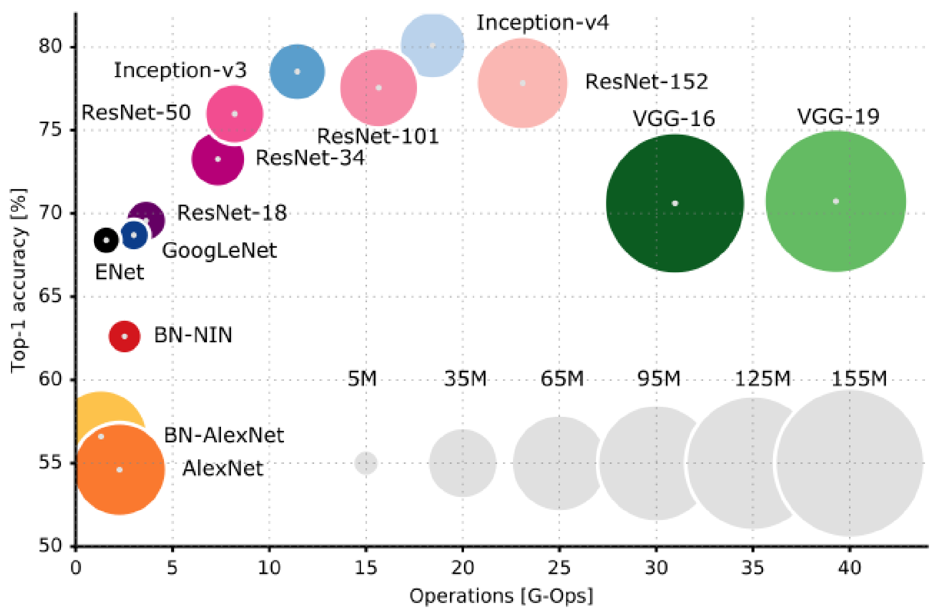
Object Recognition [Krizhevsky, Sutskever, Hinton 2012]

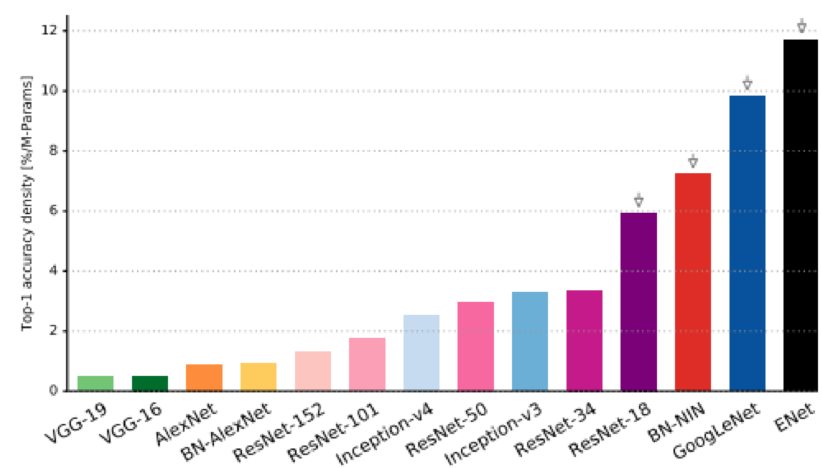
Improvement Over Time

- ResNet: 152 layers, 2015



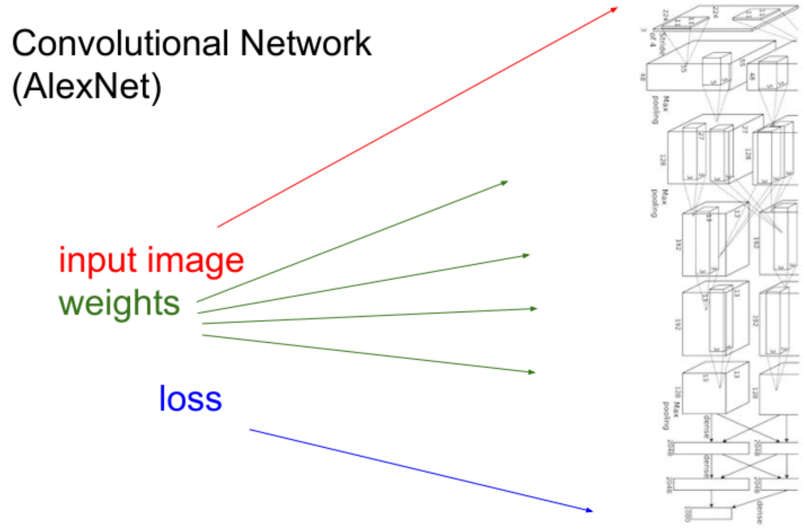
Improvement over Time





Big Picture

- Remember the 4 steps (sample, forward, backprop, update)
  - Learning rate
  - Activation function



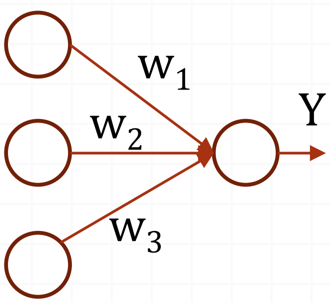
Weight Initialization

- All set to zero?
  - No asymmetry
- Random numbers?
  - Variance is important
  - If the weights are initialized to be too small, then the output from each layer gets smaller and smaller.
  - If the weights are initialized to be too large, then output from each layer gets larger and larger.

Weight Initialization

- The variance of the output from a randomly initialized neuron grows with the number of inputs.
  - Fine for small networks, it can lead to non-homogeneous distributions of activations across the layers of a network.

$$\mathrm{Var}(Y) = \mathrm{Var}(W_1X_1 + W_2X_2 + \cdots + W_nX_n) = \boxed{n\mathrm{Var}(W_i)}\mathrm{Var}(X_i)$$



$$\mathrm{Var}(W_i) = \frac{1}{n} = \frac{1}{n_{\text{in}}}$$

$$\mathrm{Var}(W_i) = \frac{1}{n_{\text{out}}}$$

Weight Initialization

- We can initialize the weights using a distribution with mean zero and variance as obtained in the previous slides.
- Two popular methods:
  - He initialization
  - Xavier (or Glorot) initialization

  - Or use batch normalization after every layer (an advanced topic)

Regularization

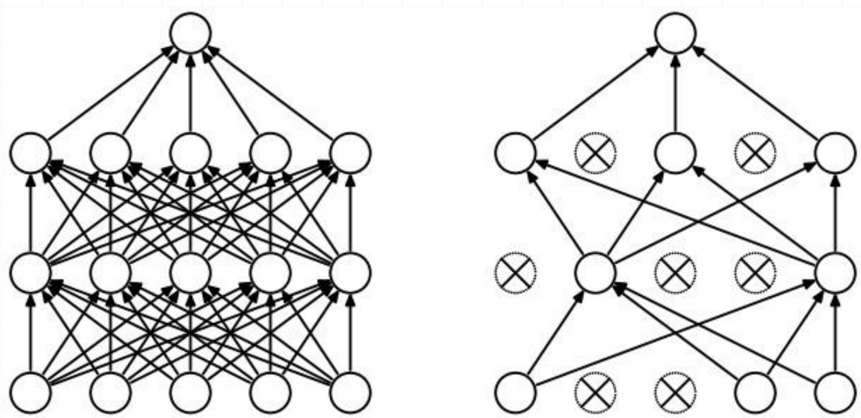
- Minimize (*Loss* + λ *Penalty*)

- L2 regularization is the most common form of regularization.
  - encouraging the network to use all of its inputs a little rather that some of its inputs a lot (diffused weights)
- L1 regularization allows the weight vectors to become sparse.

- In practice, if you are not concerned with explicit feature selection, use L2 regularization.

Dropout

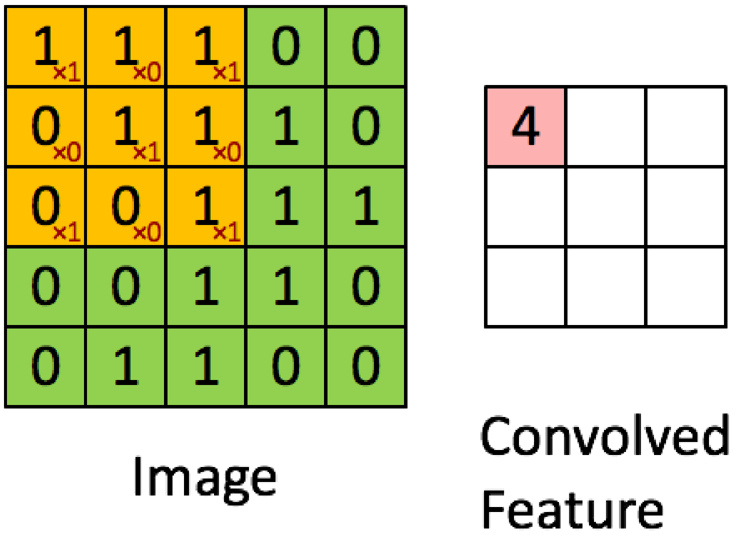
- Randomly set some neurons to zero in each forward pass



(a) Standard Neural Net (b) After applying dropout.

Convolutional Filters

- A group operator
  - Goes back to conventional vision techniques



[Link](https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1) https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1

Convolutional Filters

- Example: edge detection

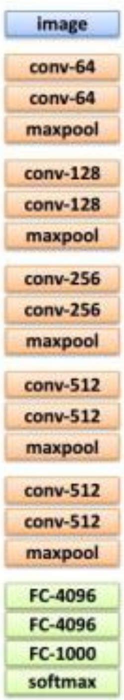
Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

Link

https://en.wikipedia.org/wiki/Kernel_(image_processing)

Transfer Learning

Transfer Learning with CNNs



1. Train on ImageNet

2. If small dataset: fix all weights (treat CNN as fixed feature extractor), retrain only the classifier

i.e. swap the Softmax layer at the end

3. If you have medium sized dataset, **"finetune"** instead: use the old weights as initialization, train the full network or only some of the higher layers

retrain bigger portion of the network, or even all of it.

More

- We will teach the rest of the material using slides here