Deep Learning Overview

Samuel Cheng

With many slides borrowed from

Stanford CS 231n

Hinton’s coursera course

Quick Logistics

- “Homework” ~ 30%
- Programming assignment ~ 30%
- Final Project ~ 40%
- Course website: http://www.samuelcheng.info/deeplearning_2017/index.html

Advisors

- Dr. Refai’s students:
- Dr. Imran’s students:
- Dr. Chan’s students:
- Mine:
- Others:

Research Directions

- Image Processing/Computer Vision/Imaging:
- Telecommunications:
- Natural language processing (NLP):
- Others:

Exposure to deep learning courses

- Hinton’s coursera course
- Stanford CS 231n
- Oxford
- Larochelle’s course
- Others?

Exposure to deep learning packages?

- Caffe:
- Torch:
- Tensorflow:
- Theano:
- Keras:
- Lasagne:
- Matconvnet:
- Mxnet:
- Others:

Programming Languages?

- Python/Numpy (required but easy)
- C/C++
- Lua
- Matlab
- Java
- Others
- None at all

Platforms?

- Linux
- Windows
- Mac
- GPUs?

Logistics

- “Homework”: presentations/review/quizzes/TBD
- **Programming assignment**: (Important)
  - mostly based on Stanford CS231n, maybe more
  - require screenshot submission, preferably screencast
  - “sample-ly” graded
- Final Project: Any target problem, can be group project
  - Proposal required near mid semester
  - Partially peer graded, precise scoring mechanism TBD

Coverage (tentative)

- Overview of machine learning
- Overview of supervised learning
- Neural network basics
- Backpropagation algorithm
- Regularization
- Optimization methods
- **CNN**
- Weight visualization
- Recurrent neural networks
- *Deep learning/neural networks packages (mostly presented by you all)*
- Restricted Boltzmann machine, autoencoder, deep belief networks (?)
- Deep reinforcement learning (?)

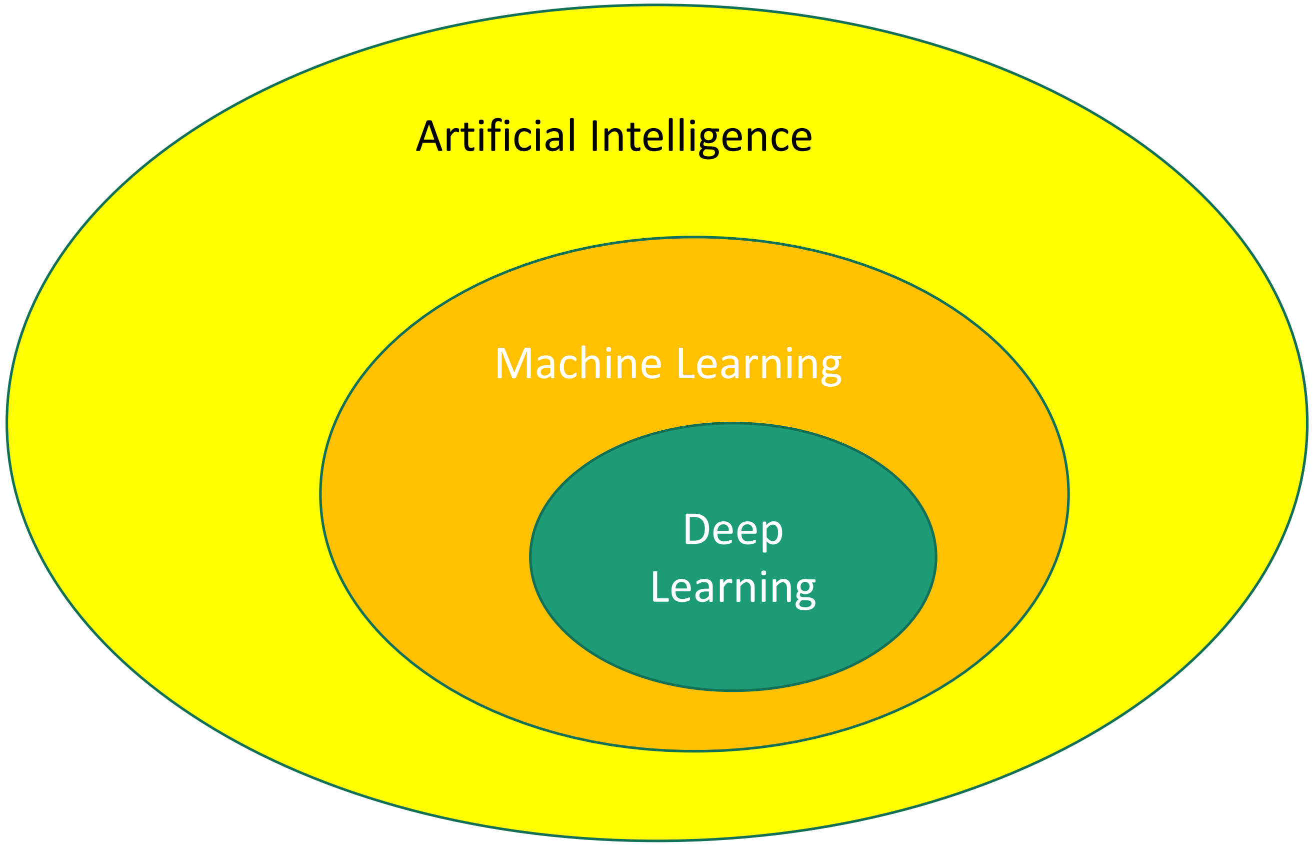
Prerequisite

- Python proficiency
  - We will borrow homework from Stanford CS231n, they are all based on Python and Numpy
  - If you have programmed before and are familiar with one of those high-level languages, you should be fine
  - Check out this for a quick tutorial: http://cs231n.github.io/python-numpy-tutorial/
- College calculus and linear algebra (not a lot though)

Deep learning in 1 slide

- Wide sense
    - (machine) **learning** goes **deep** (with layers of representation)
- Narrow sense
    - (Artificial) neural networks with more than one hidden layers
- Why so popular?
    - Probably the most powerful machine learning technique at this moment
    - Won machine learning competition across wide categories with large margins

AI/Machine Learning/Deep Learning



AI

- Strong-AI: fully human-like
  - Turing test
  - Coffee test
  - *… not what we try to study here*
- Weak-AI
  - Trying to tackle some (very) specific tasks

AI Tasks

- Playing chess :
- Playing strategy games (e.g., Starcraft) :
- Playing GO :
- Solve Sudoku :
- Find a cat in a picture :
- Describe a picture :
- Solve a calculus problem :
- *Understand* a story:

AI Tasks

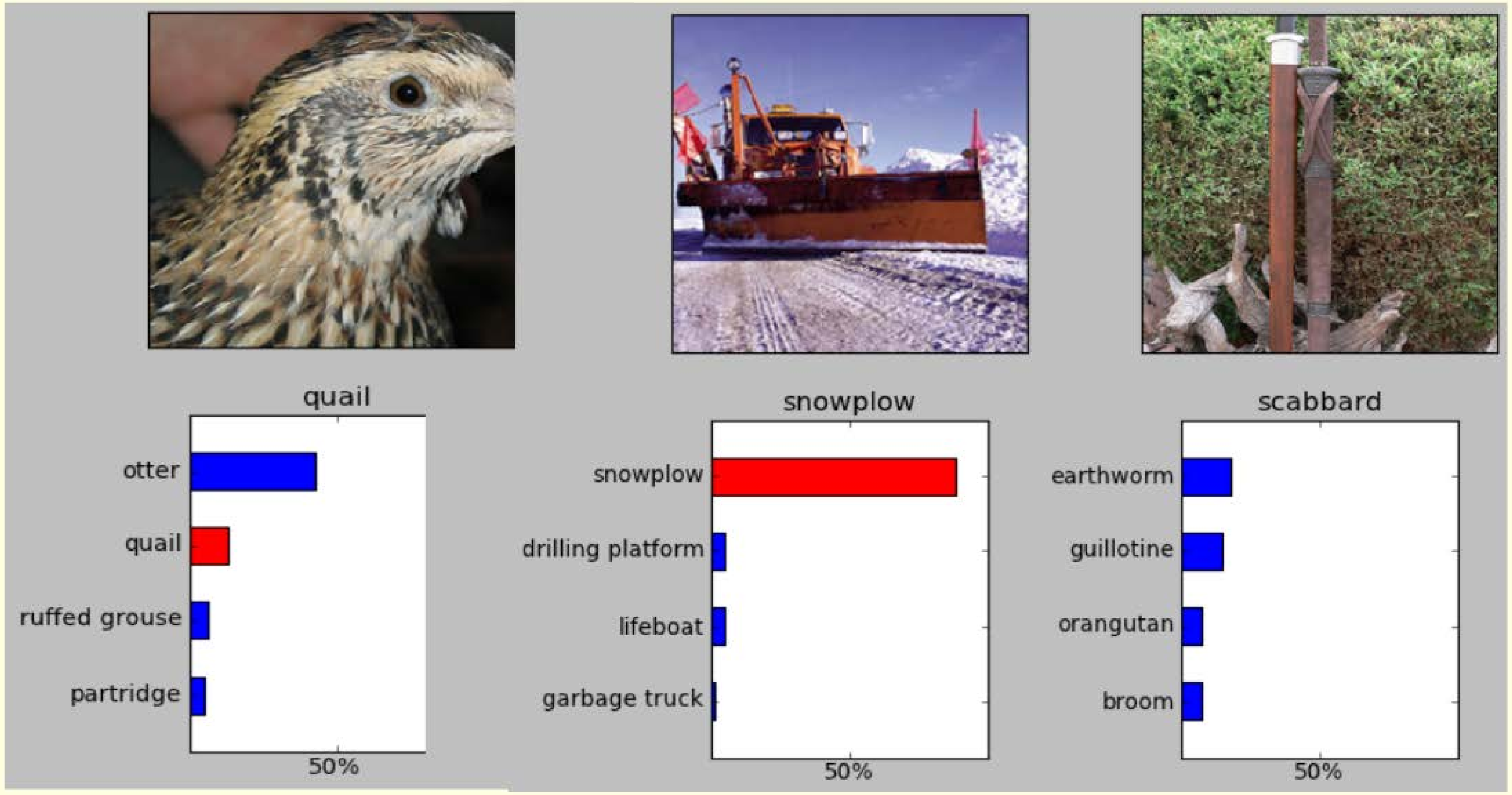
- Playing chess : relatively easy
- Playing strategy games (e.g., Starcraft) : hard
- Playing GO : hard
- Solve Sudoku : easy
- Find a cat in a picture : quite hard if it needs to be perfect
- Describe a picture : hard
- Solve a calculus problem : easy
- *Understand* a story: very hard

AlphaGo

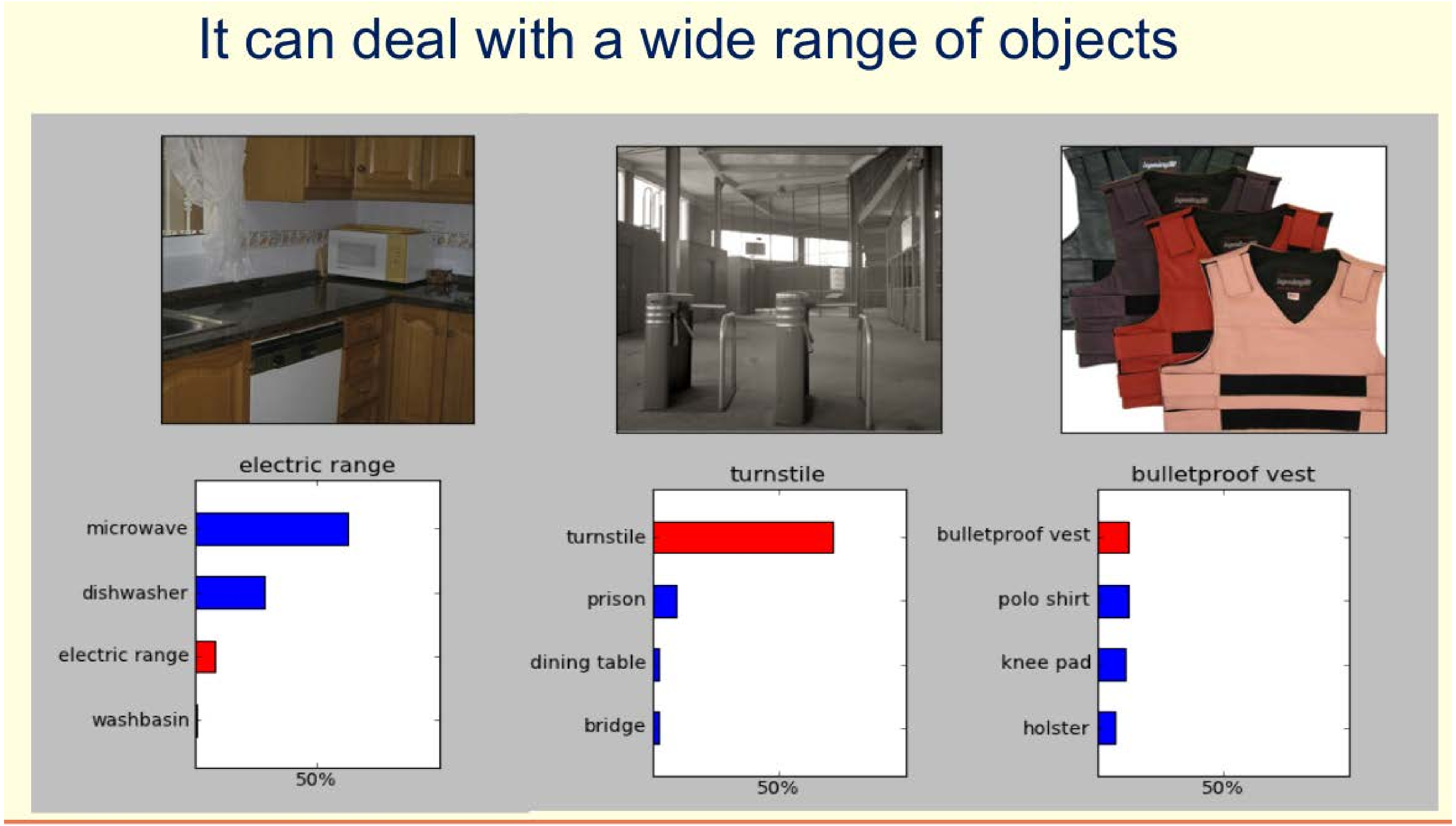
Google DeepMind

- Go was considered one of most difficult broad games
- It was thought that machine wouldn’t be able to beat professional human players for at least another decade
- AlphaGo defeated Fan Hui in October 2015, the then European champion
- It then defeated Sedol Lee in April 2016, ranked second in terms of international titles
- Monte-Carlo tree search + **Deep neural networks**

Some examples from an earlier version of the net

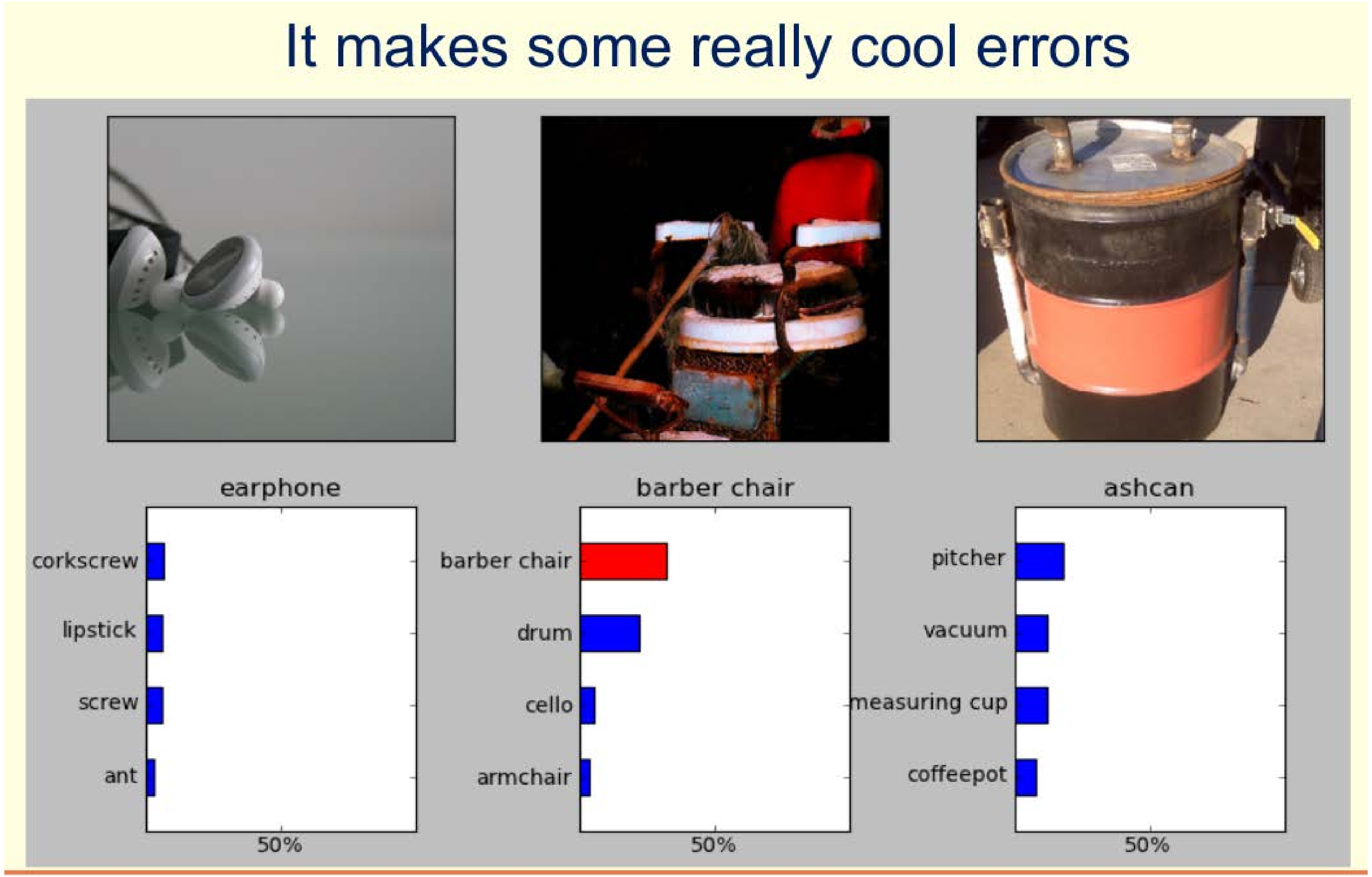


From Hinton’s coursera course



It can deal with a wide range of objects

From Hinton's coursera course

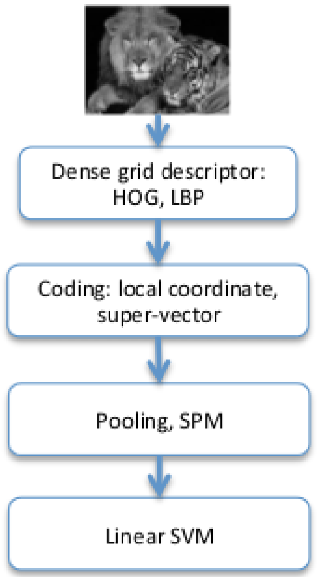


From Hinton's coursera course

IMAGENET Large Scale Visual Recognition Challenge

Year 2010

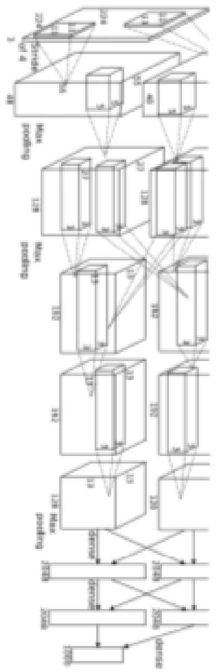
NEC-UIUC


Dense grid descriptor: HOG, LBP
Coding: local coordinate, super-vector
Pooling, SPM
Linear SVM

[Lin CVPR 2011]

Year 2012

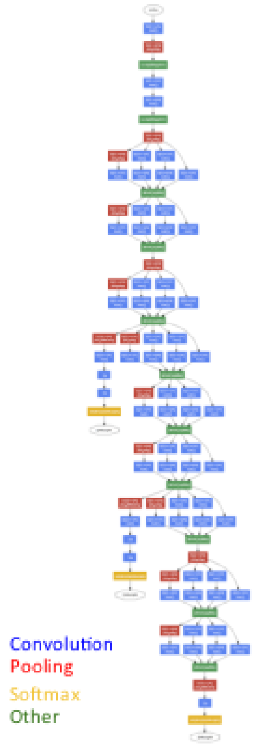
SuperVision



[Krizhevsky NIPS 2012]

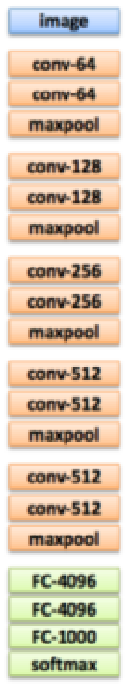
Year 2014

GoogLeNet



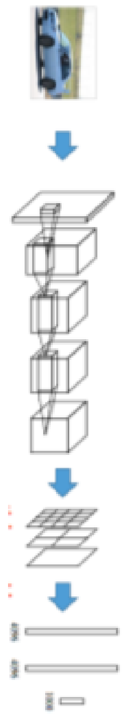
[Szegedy arxiv 2014]

VGG



[Simonyan arxiv 2014]

MSRA



[He arxiv 2014]

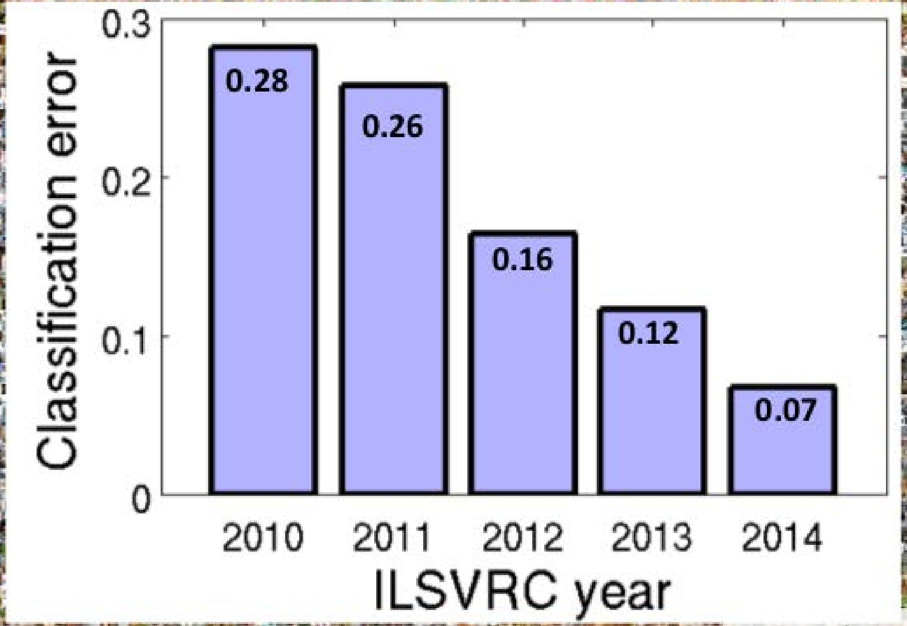
From Stanford CS231n

IMAGENET **Large Scale Visual Recognition Challenge**

The Image Classification Challenge:

1,000 object classes

1,431,167 images

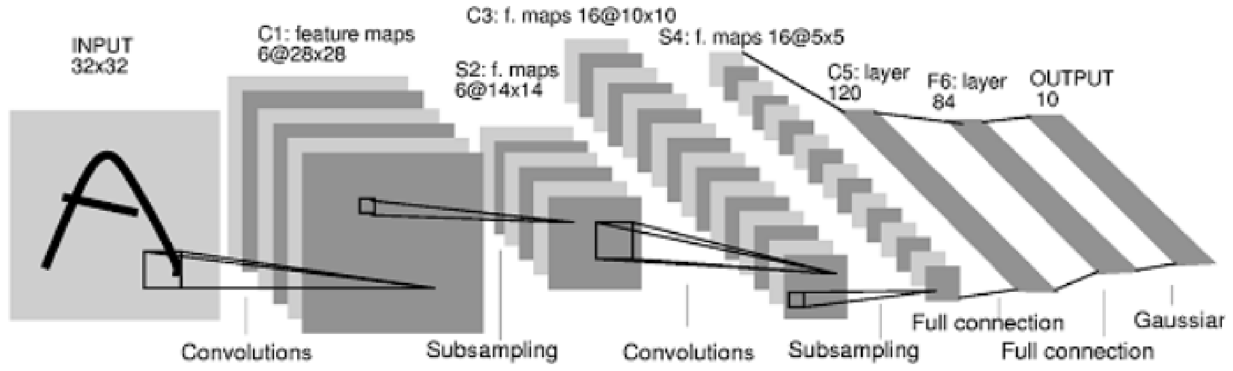


Russakovsky et al. arXiv, 2014

From Stanford CS231n

1998

LeCun et al.



# of transistors

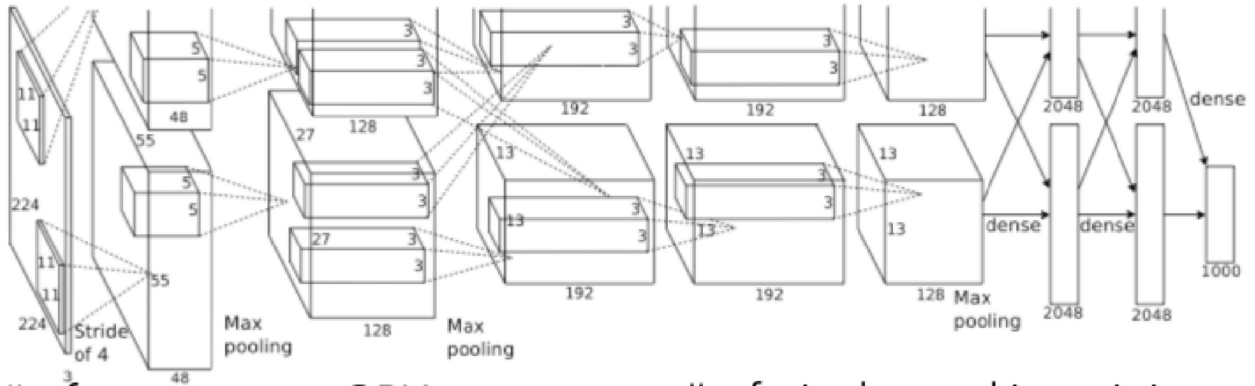
10^6

# of pixels used in training

10^7 NIST

2012

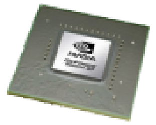
Krizhevsky et al.



# of transistors

10^9

GPUs

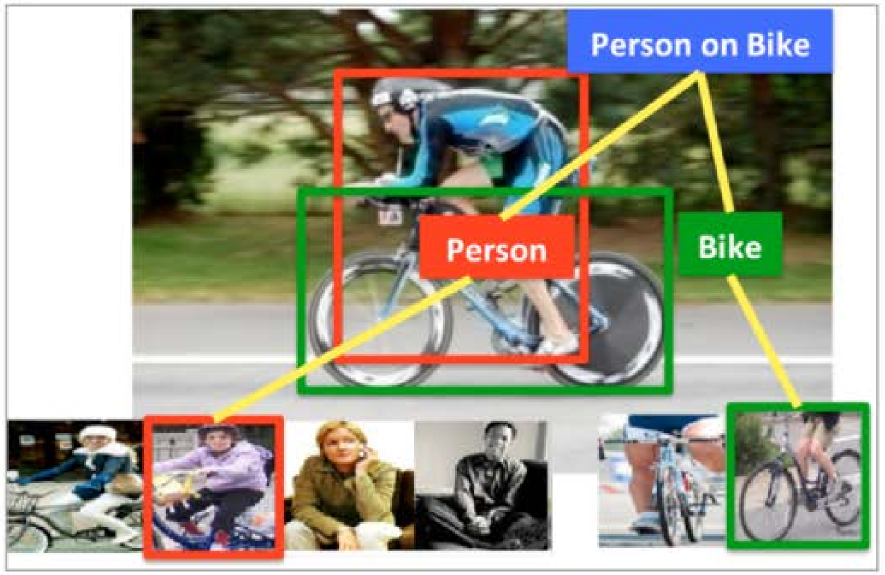


# of pixels used in training

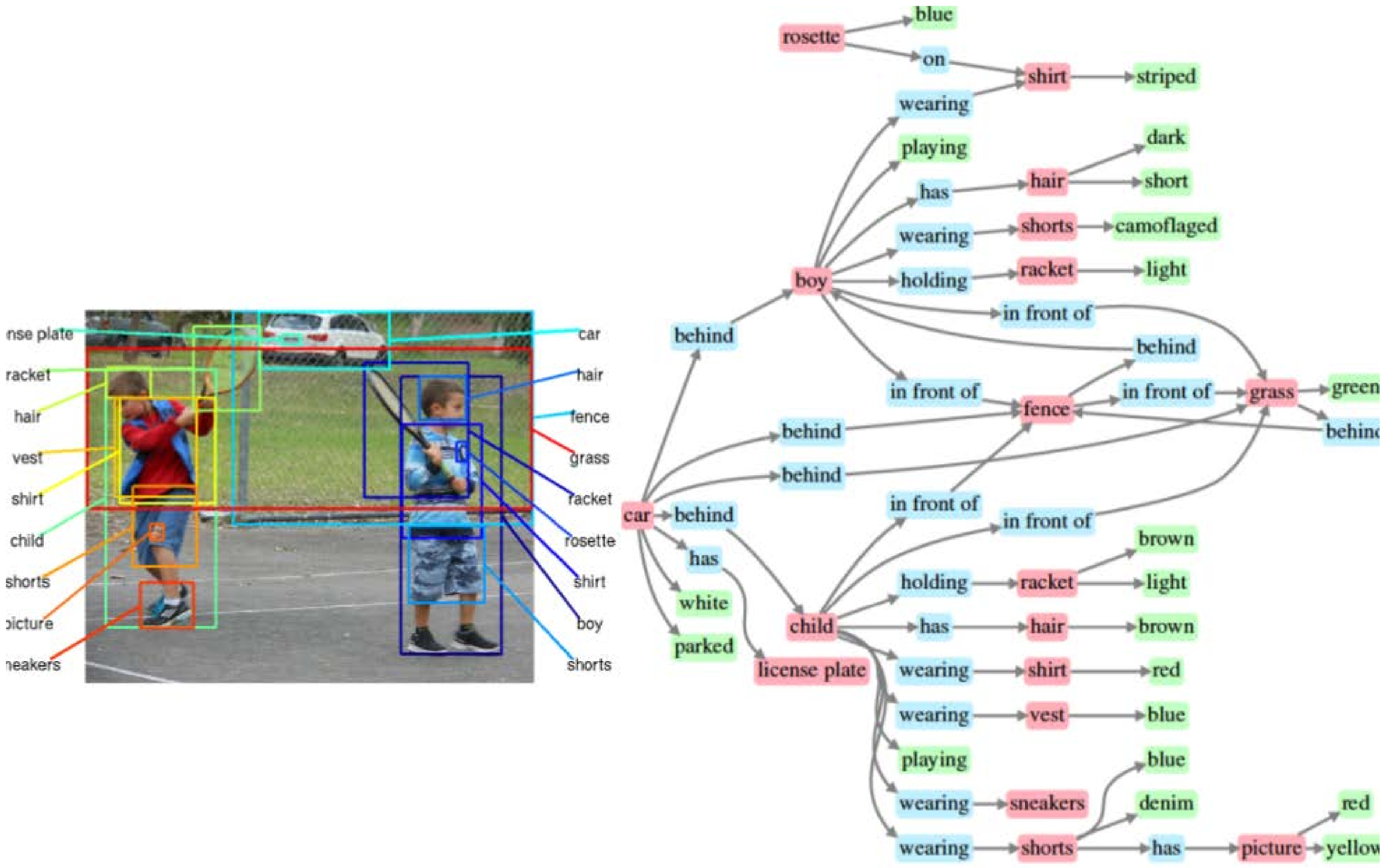
10^{14} IMAGENET

From Stanford CS231n

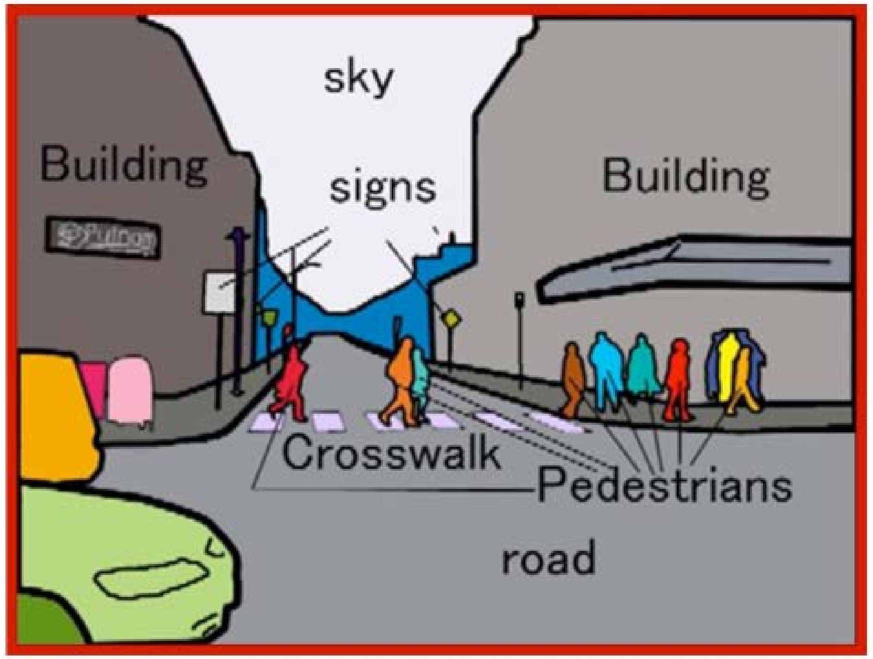
- Object detection
- Action classification
- Image captioning
- …

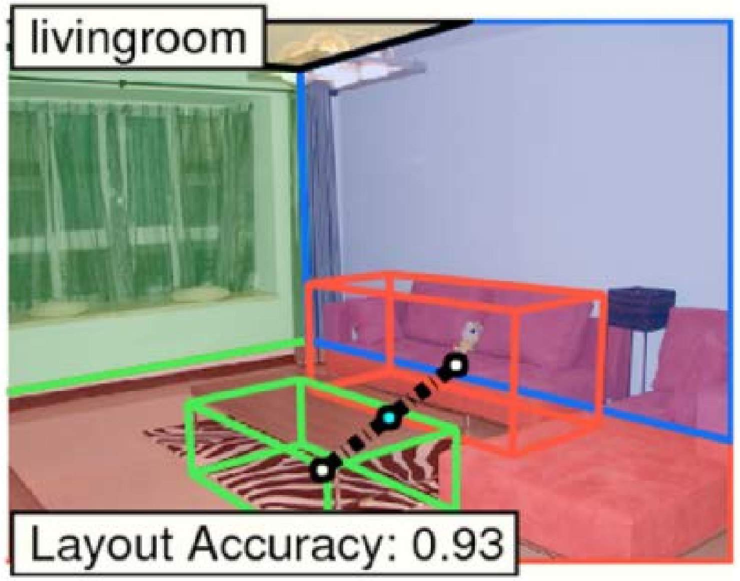


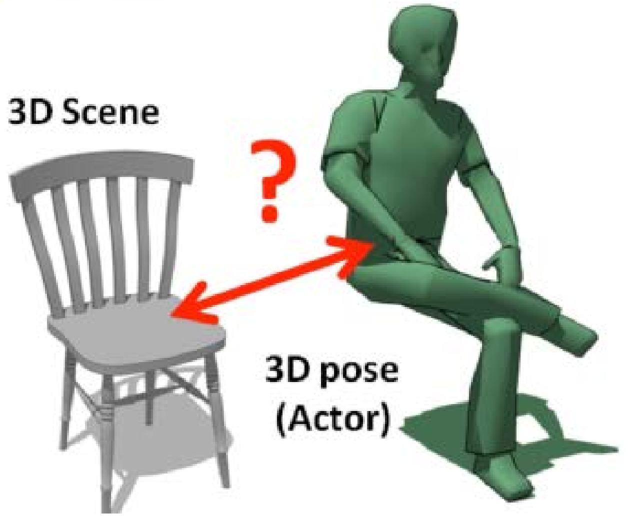
From Stanford CS231n

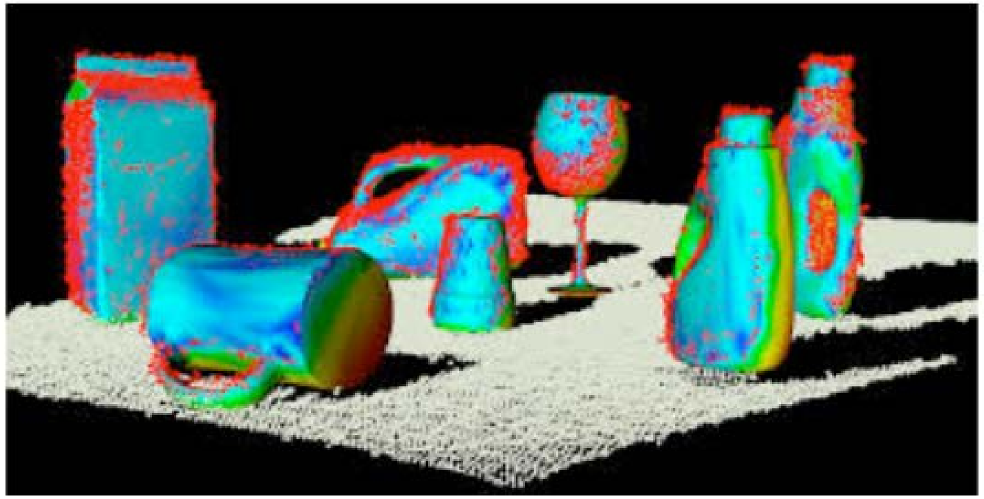


From Stanford CS231n



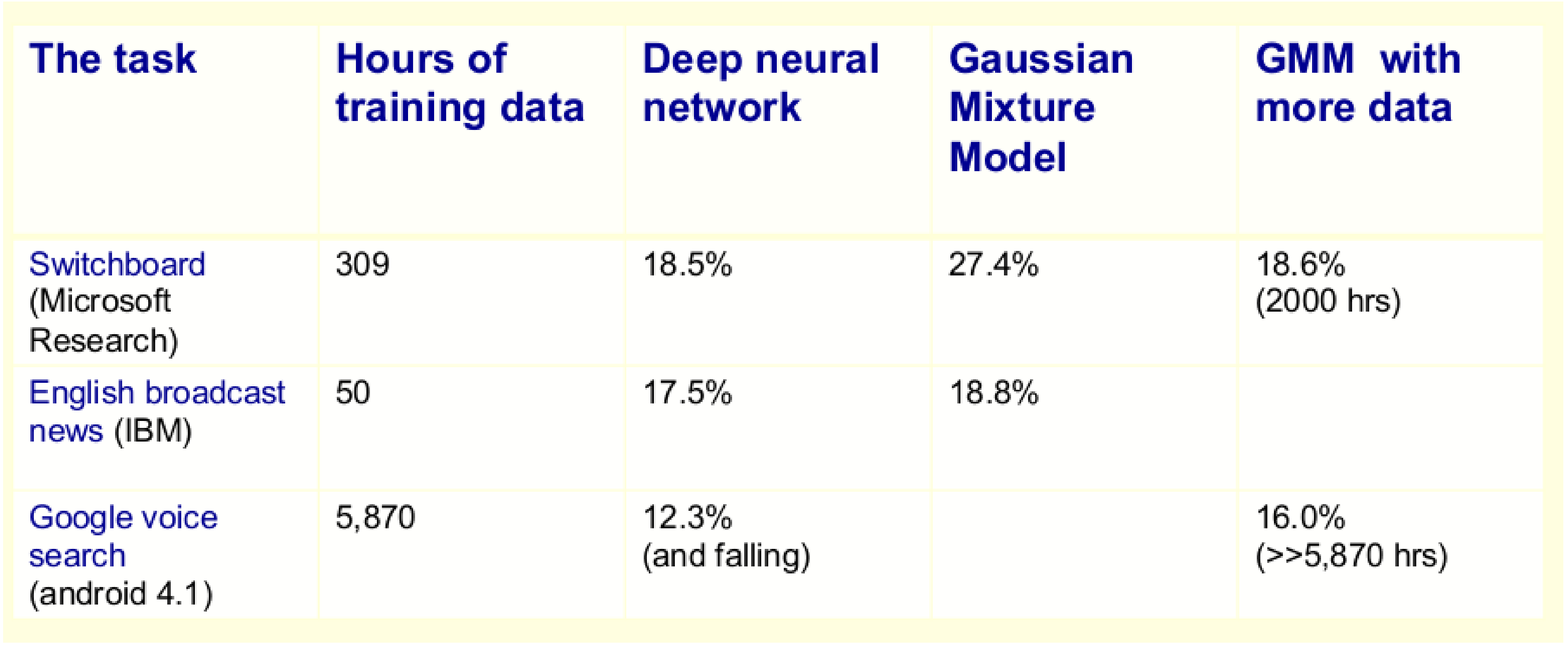






From Stanford CS231n

Natural language processing too…



The task	Hours of training data	Deep neural network	Gaussian Mixture Model	GMM with more data
Switchboard (Microsoft Research)	309	18.5%	27.4%	18.6% (2000 hrs)
English broadcast news (IBM)	50	17.5%	18.8%	
Google voice search (android 4.1)	5,870	12.3% (and falling)		16.0% (>>5,870 hrs)

From Hinton's coursera course

Machine learning overview

- As the name suggests, have **machine learned** (from data)
- Machine learning is only a part of AI
    - Other such as planning and search
    - But machine learning is one key component (tool)

Why and when do we need machine learning?

- Program a machine to recognize a cat
  - We don’t know how we manage to do it
  - So there is no clue how to program such machine
  - Even if we know how to do it, the program is probably extremely complicated
- Compute the probability of a credit card fraud
  - No one simple rule. Need to combine many weak rules
  - It is a moving target

Three types of machine learning

- Supervised learning
  - Learn to predict output when given an input
  - Example pairs (desired output for an input) are given
- Unsupervised learning
  - Learn to discover good internal representation of the input
  - No example output is given
- Reinforcement learning
  - Learn to select an action to maximize payoff
  - No example output (desired action) is given for any input
  - Some hints if you are doing good or bad are given though

Two types of supervised learning

- Regression: the desired output is a real number or a real vector
  - Price of a stock 6 months later
  - Temperature at noon tomorrow
- Classification: the output (label) is discrete
  - Whether a picture includes a cat
  - Types of object in a photo (can be from a set of different labels, a dog, a cat, a raccoon, etc.)

Supervised learning in action

- Pick a model class
  - A model class is just a set of possible mappings f from any valid inputs to any valid outputs
  - We can modify the mapping f by varying some internal parameters W
  - Namely, the predicted output $y = f(x; W)$ with an input x
- Learn to adjust parameters of the class so as to reduce discrepancy between target output t and the produced output y
  - For regression, it is common to use mean square error $\frac{1}{2}(y - t)^2$ to measure the discrepancy
  - For classification, other measures are usually more sensible (TBD)

Reinforcement learning

- In reinforcement learning, output is an action or sequences of action but only the desired outcome instead of the actual action would be told
  - Desired output is usually given as a reward score
  - The learning goal is to select actions respective to inputs to maximize expected sum of future rewards
- Reinforcement learning is difficult
  - Rewards are typically delayed and so it is hard to know where we went wrong (or right)
  - A scalar reward does not supply much information

Unsupervised learning

- It has vague definition
  - Many researchers just restrict it to mean clustering
  - According to Hinton, its aim is to find good internal representation and can be subsequently used for supervised or reinforcement learning
- Some examples
  - Produce a compact, low-dimensional representation
    - E.g., map a 100×100 image patch to just 100 dimensional vector
    - Principal component analysis (PCA) is a most commonly used technique
  - Clustering
    - Can be considered as the extreme low-dimensional representation (with only 1D output)
  - Produce a high-dimensional (but economical) representation
    - A long vector with most components are zero (sparse)

(Artificial) neural networks

- A computing model inspired by the biological brain
- Flexible and very powerful
- Can be adjusted for different learning tasks (supervised, unsupervised, etc.)

Reasons to study neural networks

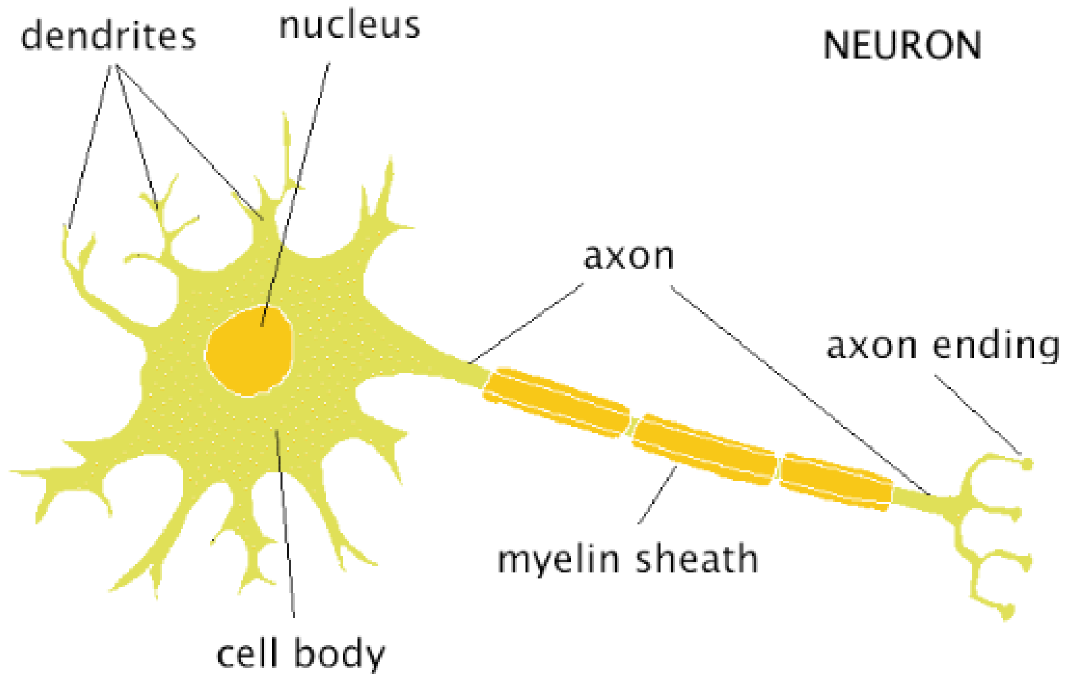
- We usually cannot directly study brain
  - We can with fMRI and EEG but not everyone can afford it
  - There are limitations (yet)
- To understand what we are likely good (bad) at
  - Supposed to be good at things that we are good at: e.g., vision
  - And bad at things that we are bad at: e.g., arithmetic
- A powerful paradigm to solve real-world problem (this course)

Our brain

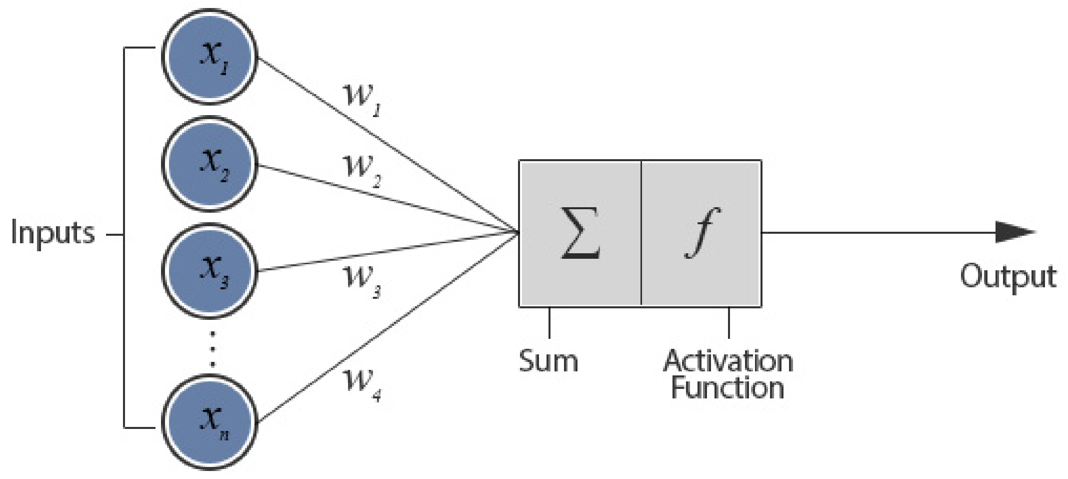
- Average weight 1.4 kg, about 2% of total body weight
- Responsible ~20% of our energy consumption!
  - ~12.6 Watts
  - It sounds a like but is extremely efficient. A typical GPU server requires ~1000 Watts
- Composed of neurons interconnected to each other

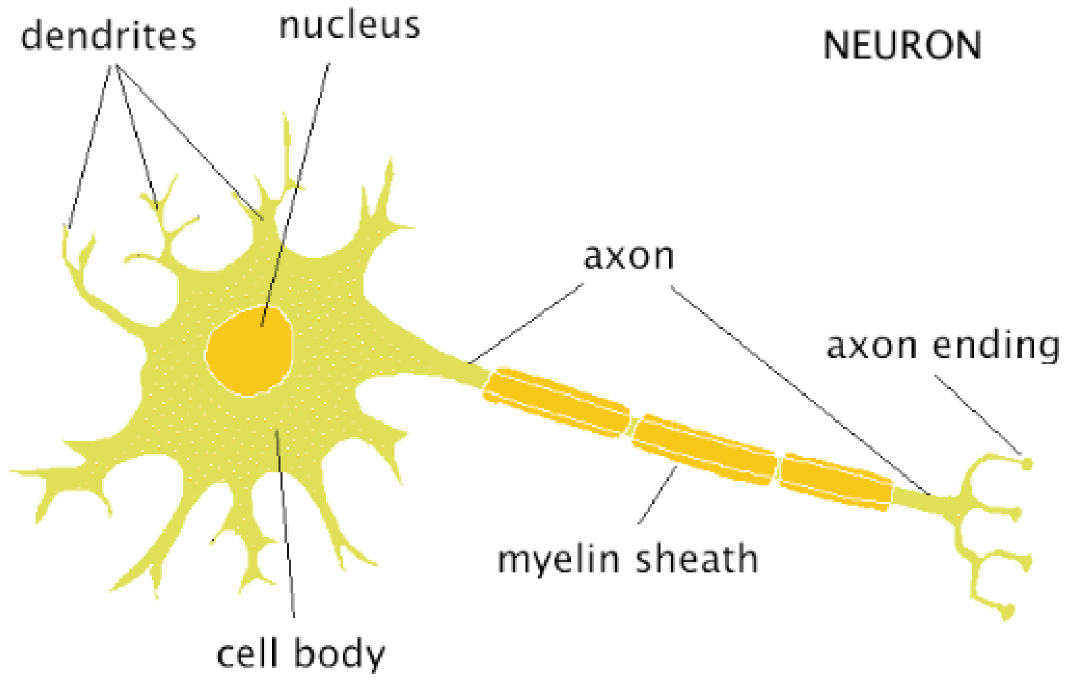
Biological neurons

- Dendrites collect chemicals
- Neuron may “fire” based on the chemical input
- Axon ending will generate chemicals those will in turn be consumed by other neurons



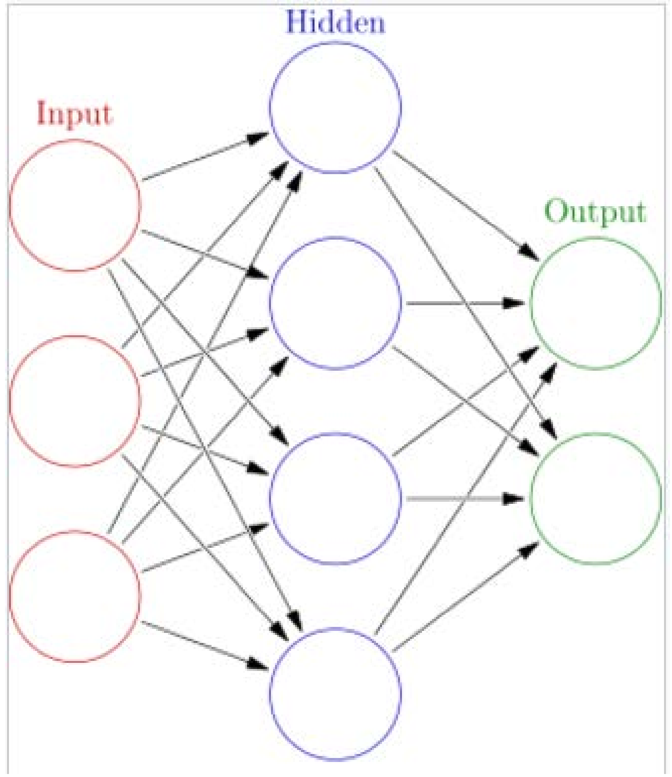
Biological neurons



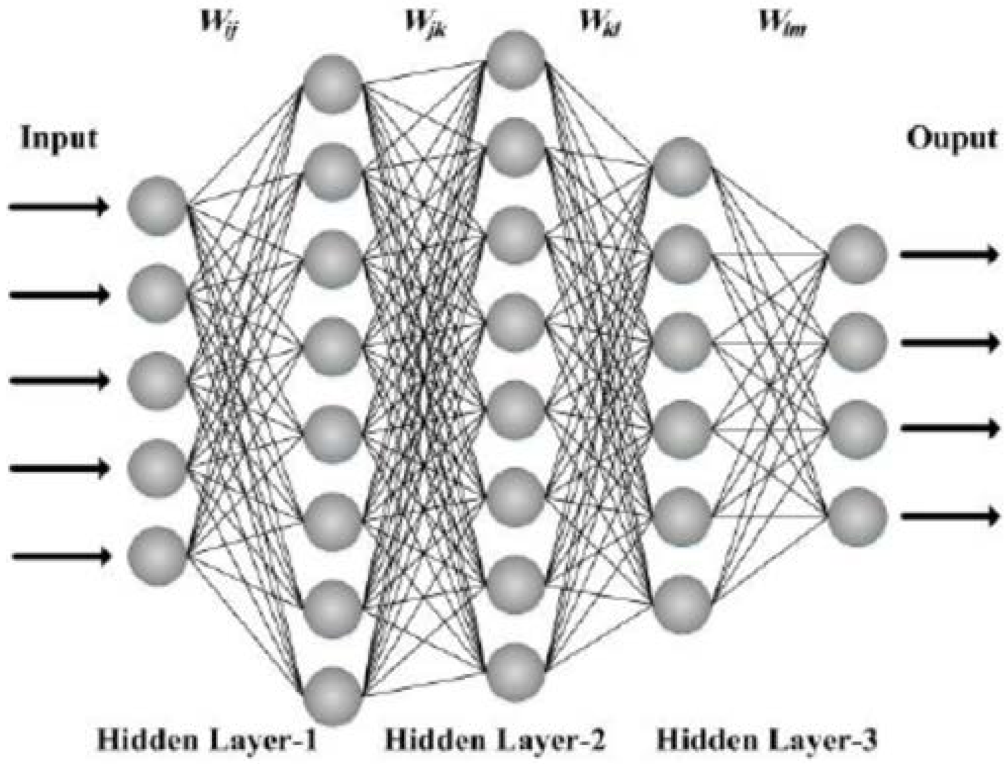


Neural networks

Single layer network



Multiple layer (deep) network



A very brief history of ANN

- Warren McCulloch and Walter Pitts introduced Threshold Logic Unit as a computational model of neuron in 1943
- Donald Hebb created a hypothesis of learning (Hebbian theory) based on neural plasticity (increase synaptic efficacy by repeated and persistent stimulation) in late 1940's
- Frank Rosenblatt created perceptron in 1958 but no concrete learning method appeared to be described
- Bernard Widrow and Marcian Hoff of Stanford developed "ADALINE" and "MADALINE" (Mulitple ADAptive LiINear Elements) in 1959. First neural network to tackle real world problem. It is still used in air traffic control systems (really?)

A very brief history of ANN (con’t)

- Perceptron got high expectation in early year but neural science does not catch up. And the traditional von Neumann and Turing architecture took over the computing scene. Ironically, von Neumann himself suggested to imitate neuron function with telegraph relays or vacuum tubes
- Neural network research was hit hard by the introduction of the book Perceptrons by Marvin Minsky and Seymour Papert. It proved the limitation of perceptrons but combined with the initial hype, people lost trust of the potential of neural networks and we entered the “first dark age” of neural networks

A very brief history of ANN (con’t)

- The interest to neural network slowly revived with the invention of the backpropagation algorithm by Paul Webos in 1974. The algorithm was reinvented by many other such as Parker (1985) and LeCun (1985)
- Hopfield popularizes a form of bi-directional networks (Hopfield networks) in the 1980’s and neural network research was blooming in that decade
- But as the support vector machine (SVM) by Vladmir Vapnik was popularized in late 1980’s and early 1990’s. Neural network slowly lost favors. SVM took hold instead of neural networks because it has more elegant math and worked better at that time
  - Less parameters to tune
  - Computers were too slow then and labeled datasets were too small

A very brief history of ANN (con't)

- Neural networks got a come-back for the last decade as recurrent neural networks and deep feedforward neural networks won numerous competitions in both pattern recognition and machine learning domains
- Fast GPUs were a key for the come-back. Relatively cheap and powerful computing resources are now widely available. And the wide spread of large public labeled datasets helped a whole lot too

Conclusions

- Machine learning allows us to automatically generate programs to solve various tasks by learning from data
- Three main machine learning types:
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning
- Deep learning is just neural networks (TBD) with many layers (hence deep)

Example: image recognition

- http://vision.stanford.edu/teaching/cs231n/slides/lecture2.pdf

Next time…

- More on supervised learning
  - Loss functions
  - Regularizations
  - Examples
- Optimization methods (?)