CS109 – Data Science
Deep Learning I - MLPs

Hanspeter Pfister, Mark Glickman, Verena Kaynig-Fittkau

http://www.kdnuggets.com/2015/12/cartoon-data-science-contest-coffeemaker.html

Overview

- Lot’s of practical advice and material
- Can’t do it all at once
- We start with a simple model
- Then introduce updates

- After today you should have the knowledge to train a simple MLP

Next Lectures

- Advance from MLP to CNN
- Upgrades for model components

- Practical Tips and Tricks for training
- Advanced models and optimization techniques
- Swapped 3rd and 4th lecture to help with final project

Deep Learning

Reinventing Social Media: Deep Learning, Predictive Marketing, And Image Recognition Will Change Everything

COOPER SMITH
FEB. 16, 2014, 6:02 PM | 6,821

CULTURE
→Animals. Civil Liberties. Tech. Top Stories

Why Facebook, Google, and the NSA Want Computers That Learn Like Humans

Deep learning could transform artificial intelligence. It could also get pretty creepy.

—By ***Dana Liebelson*** | September/October 2014 Issue

Like Share 454 Tweet 490 Email 71

Scientists See Promise in Deep-Learning Programs

IS "DEEP LEARNING" A REVOLUTION IN ARTIFICIAL INTELLIGENCE?

BY GARY MARCUS

Share Tweet g+1

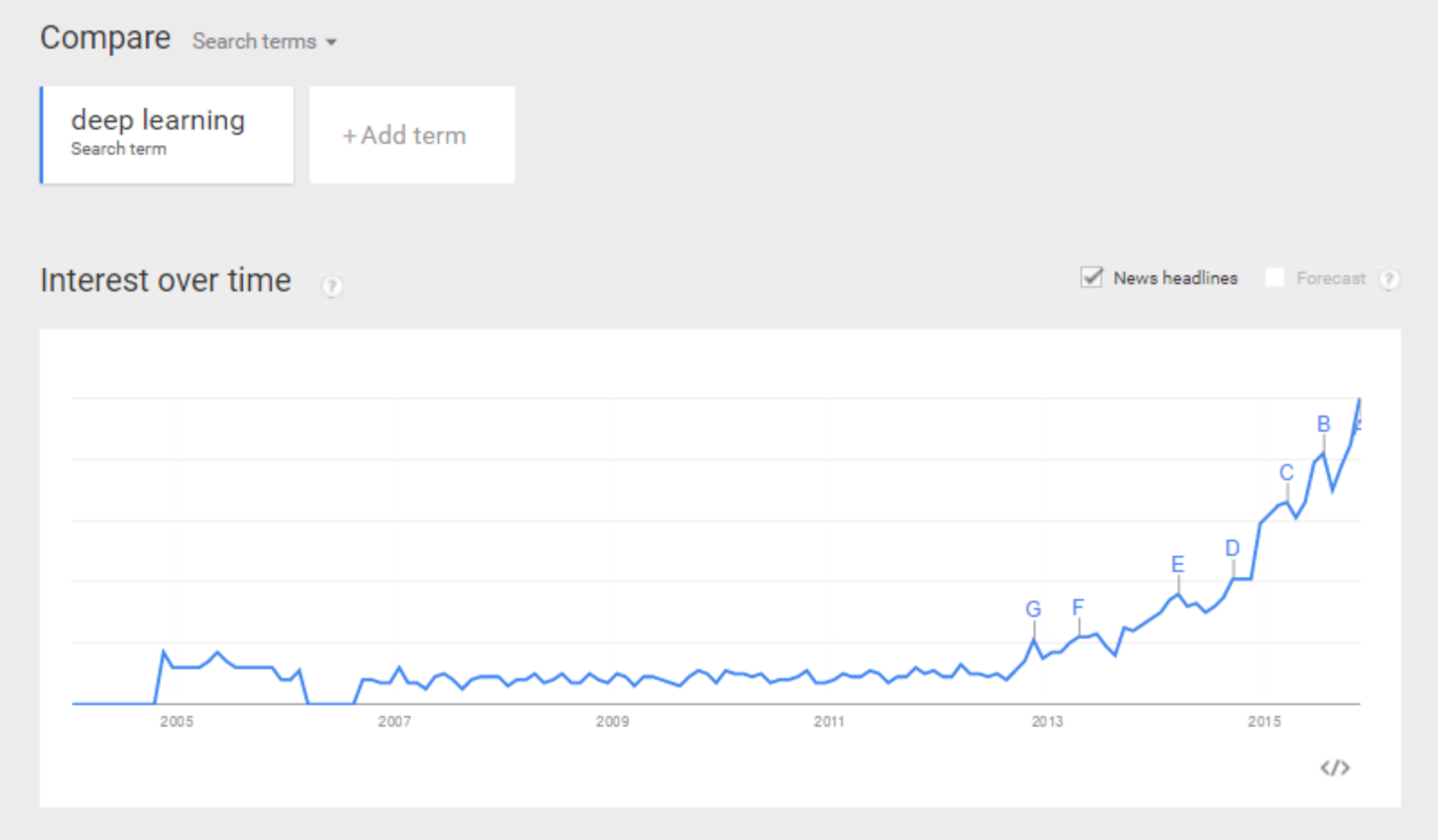
Deep Learning - The Biggest Data Science Breakthrough of the Decade

Motivation

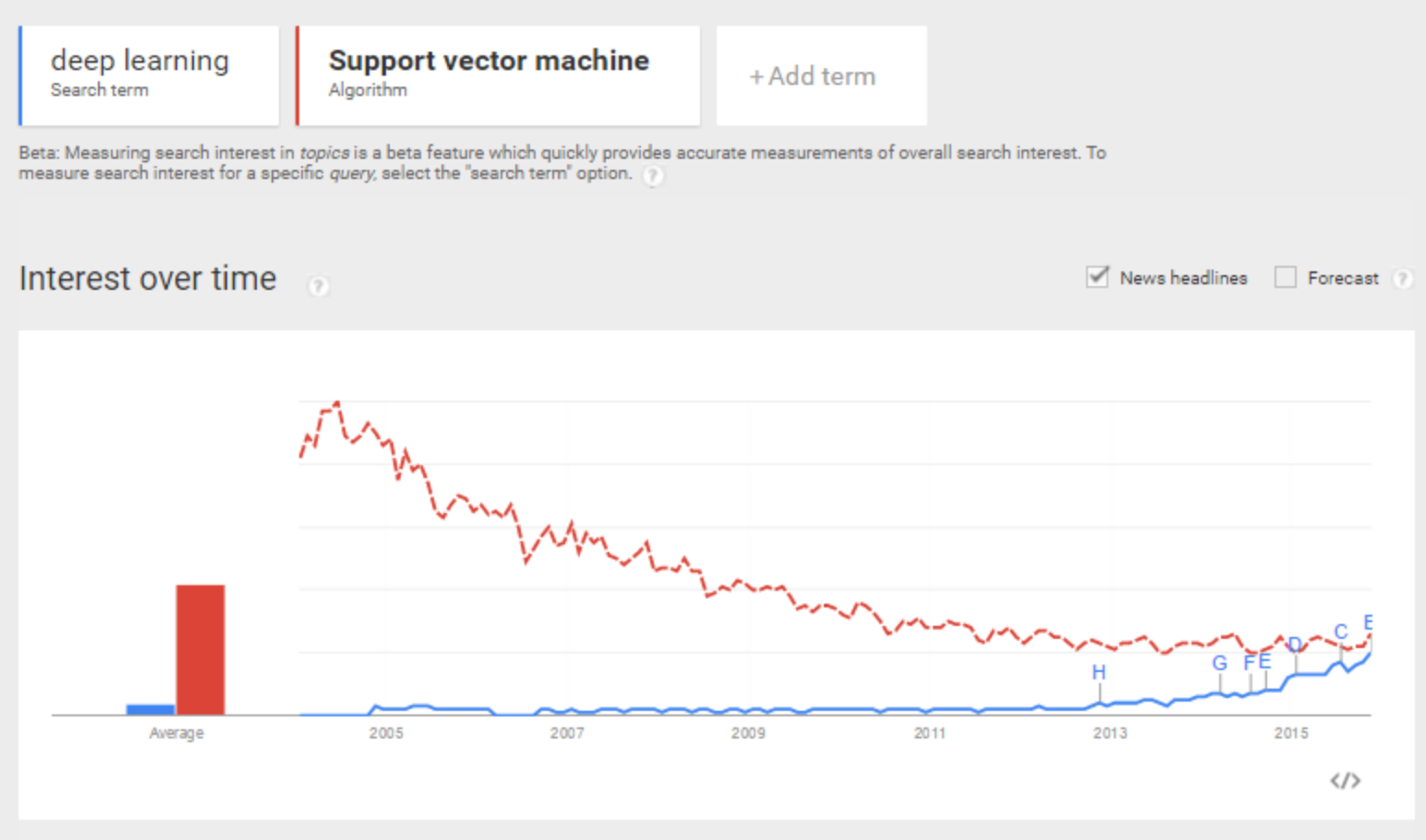
- It works!
- State of the art in machine learning
- Google, Facebook, Twitter, Microsoft are all using it.

- It is fun!
- Need to know what you are doing to do it well.

Google Trends



Google Trends



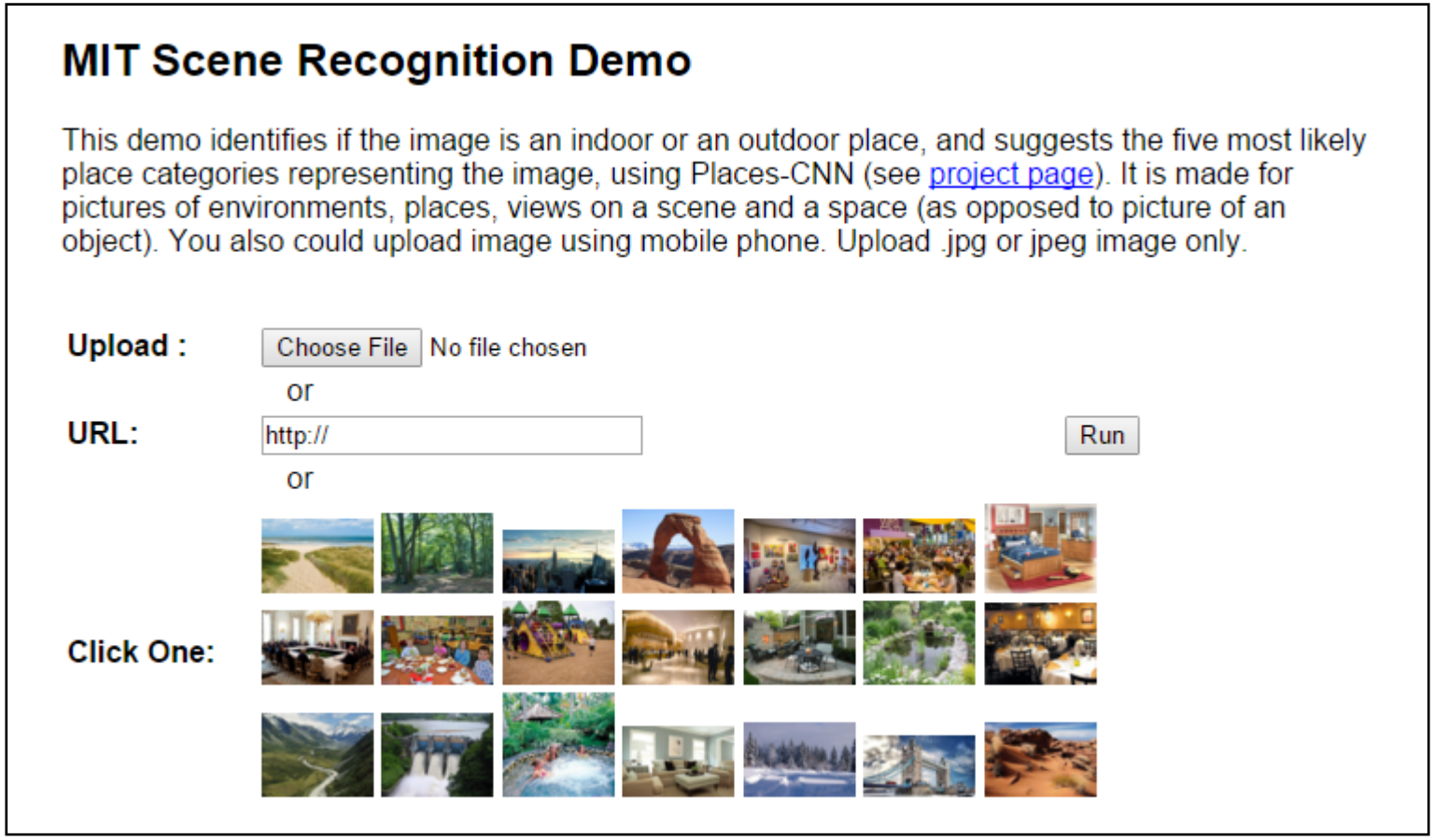
https://www.google.com/trends/explore#q=deep%20learning%2C%20%2Fm%2F0hc2f&cmpt=q&tz=Etc%2FGMT%2B5

Dan Claudiu Cireșan

- **Competitions (all methods use my NN framework)**
- NEW! First place at Assessment of Mitosis Detection Algorithms, MICCAI 2013 Grand Challenge, Nagoya, Japan (with Alessandro Giusti).
- NEW! Best score on test set from Chinese Handwriting Recognition Competition; task: offline characters, ICDAR 2013, Dallas, US - details in Multi-Column Deep Neural Networks for Offline Handwritten Chinese Character Classification - IDSIA Technical Report, August 2013.
- First place at Mitosis Detection in Breast Cancer Histological Images, ICPR 2012, Tsukuba, Japan (with Alessandro Giusti).
- First place at Segmentation of neuronal structures in EM stacks challenge - ISBI 2012, Barcelona, Spain (with Alessandro Giusti). We were the only team with better than human pixel level segmentation performance.
- First place at Offline Chinese Character Recognition (task1: "Offline Chinese Character Recognition") at ICDAR 2011, Beijing, China (with Ueli Meier).
- First place at The German Traffic Sign Recognition Benchmark (both phases) at IJCNN 2011, San Jose, US (with Ueli Meier and Jonathan Masci). We were the only team with better than human performance.

http://people.idsia.ch/~ciresan/

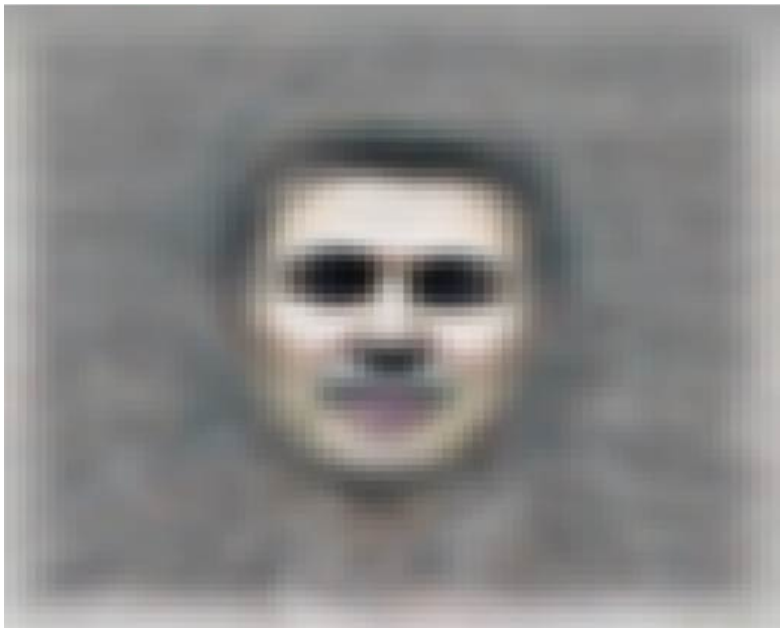
Scene recognition



http://places.csail.mit.edu/demo.html

Google Brain - 2012

What it learned



http://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html?pagewanted=all

Google DeepMind

deep_mind.mp4

https://www.youtube.com/watch?v=V1eYniJ0Rnk

Google Translate

- Early November 2016 Google translate became the No. 1 trend on Japanese Twitter

- Prof. Rekimoto translated the beginning of Hemingway's "The Snows of Kilimanjaro,"
- then ran that passage back through Google into English

https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html?_r=0

Google Translate - Before

Kilimanjaro is 19,710 feet of the mountain covered with snow, and it is said that the highest mountain in Africa. Top of the west, “Ngaje Ngai” in the Maasai language, has been referred to as the house of God. The top close to the west, there is a dry, frozen carcass of a leopard. Whether the leopard had what the demand at that altitude, there is no that nobody explained.

Kilimanjaro is a snow-covered mountain 19,710 feet high, and is said to be the highest mountain in Africa. Its western summit is called the Masai “Ngaje Ngai,” the House of God. Close to the western summit there is the dried and frozen carcass of a leopard. No one has explained what the leopard was seeking at that altitude.

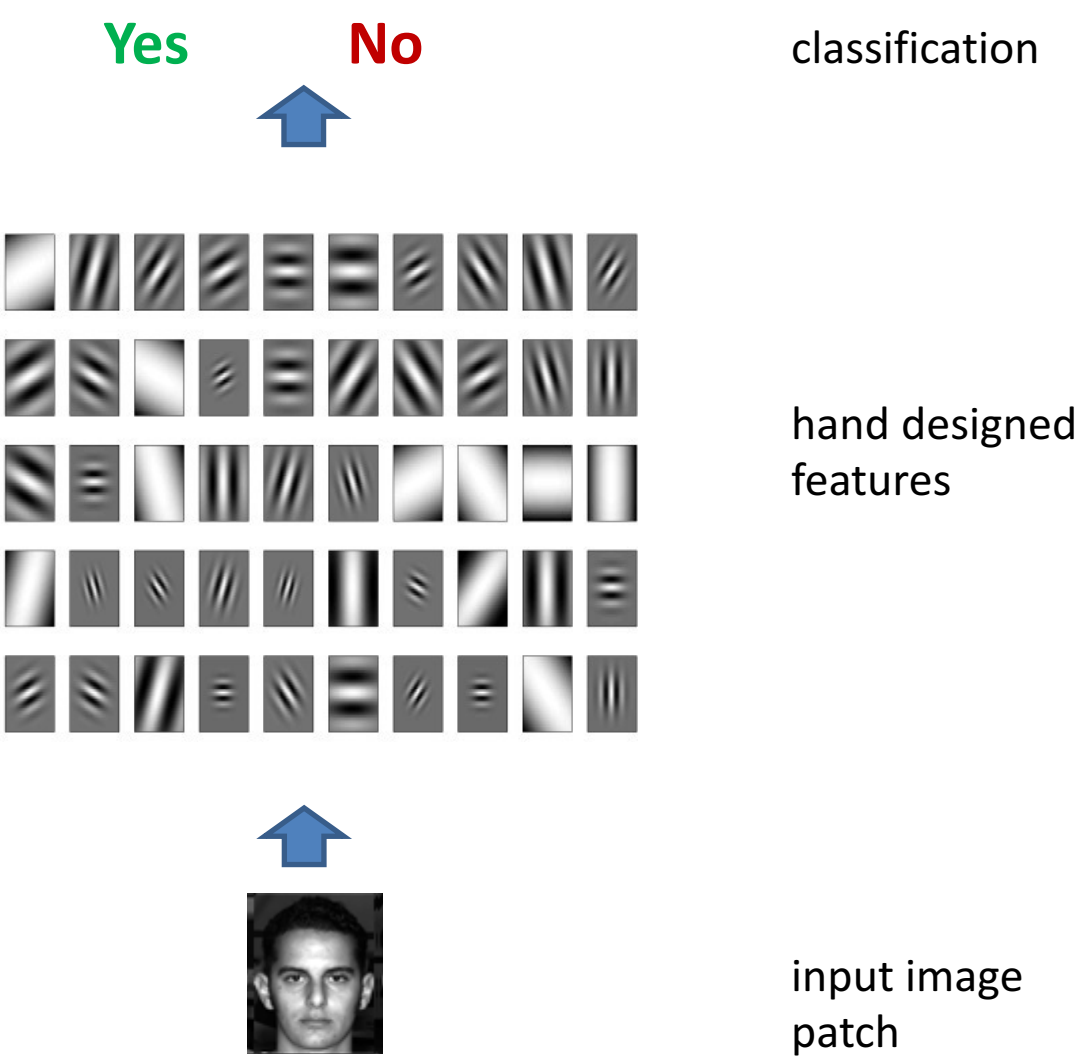
Kilimanjaro is a mountain of 19,710 feet covered with snow and is said to be the highest mountain in Africa. The summit of the west is called “Ngaje Ngai” in Masai, the house of God. Near the top of the west there is a dry and frozen dead body of leopard. No one has ever explained what leopard wanted at that altitude.

What is different?

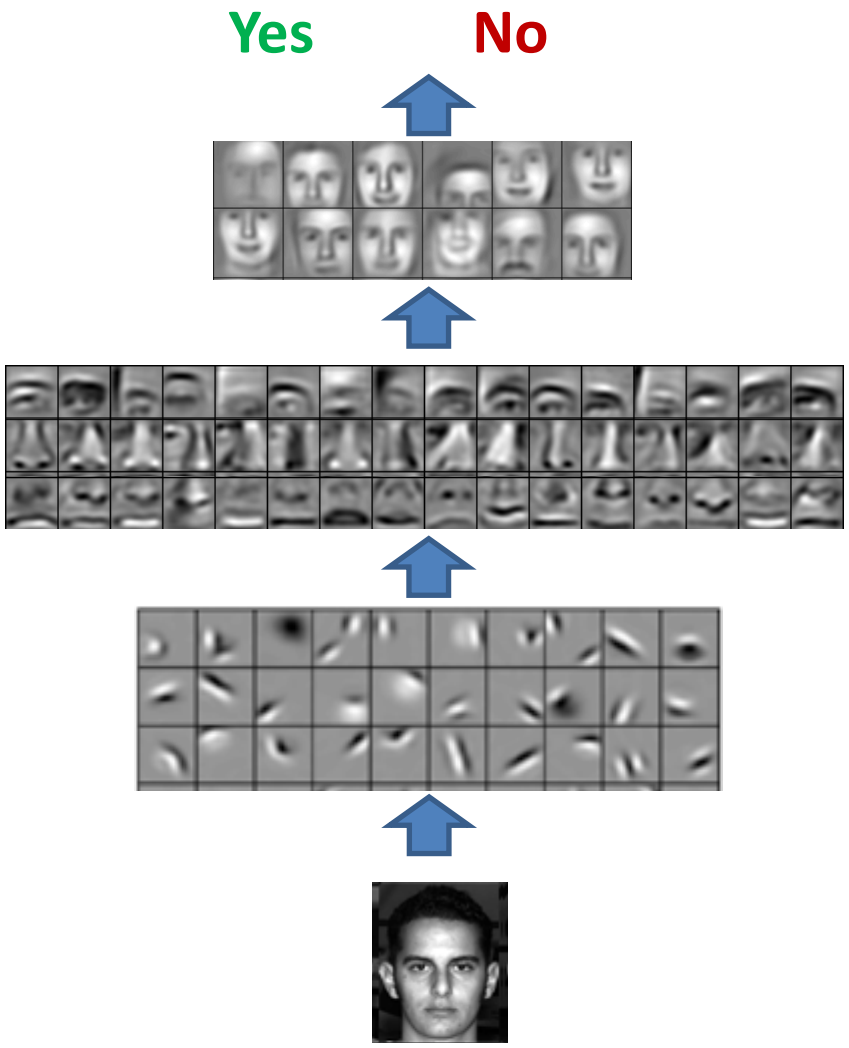
- We have seen a lot of classification methods:
  - SVM, decision trees, boosting, random forest, logistic regression, naïve Bayes

- We needed to hand design the input
- ML algorithm learns the decision boundary

Feature Design



Learned Feature Hierarchy



classification

high level features

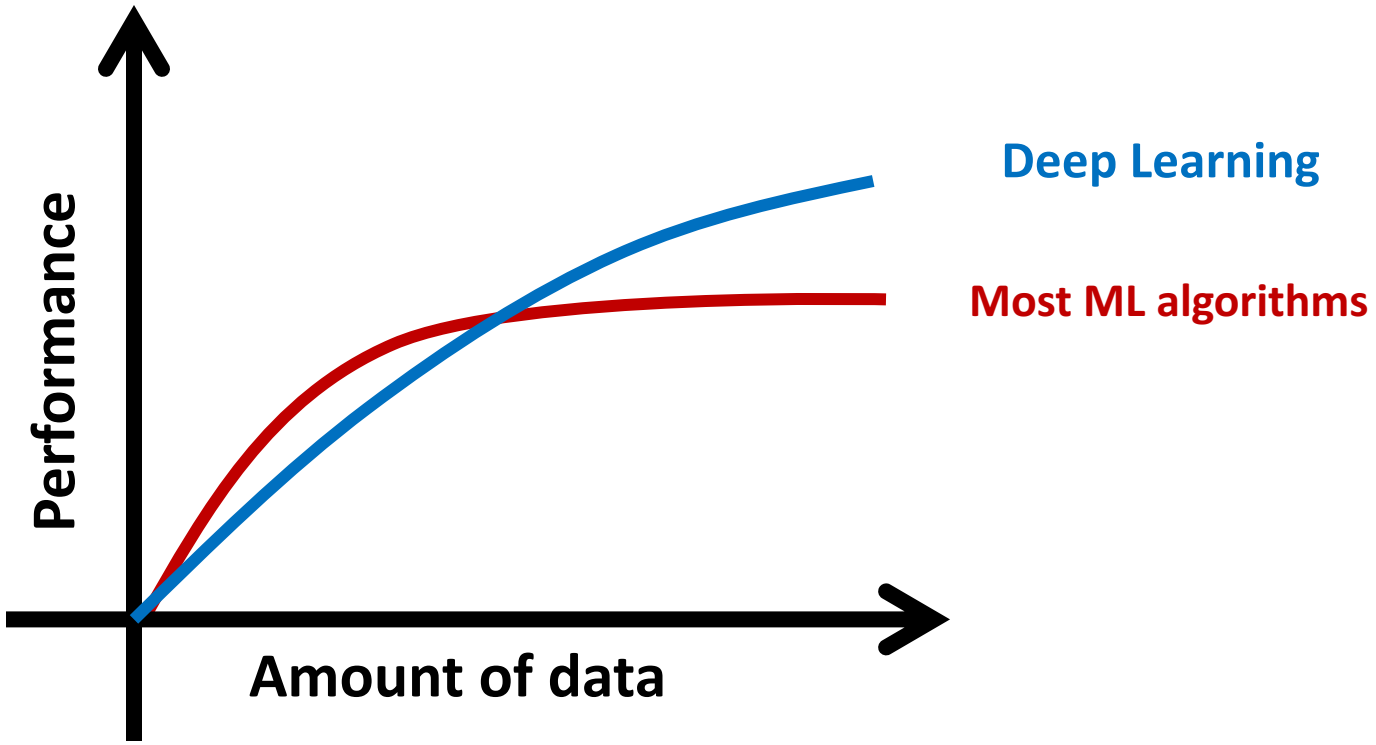
medium level features

low level features

input image patch

[Honglak Lee]

Scaling with Data Size



[Andrew Ng]

Deep Learning Techniques

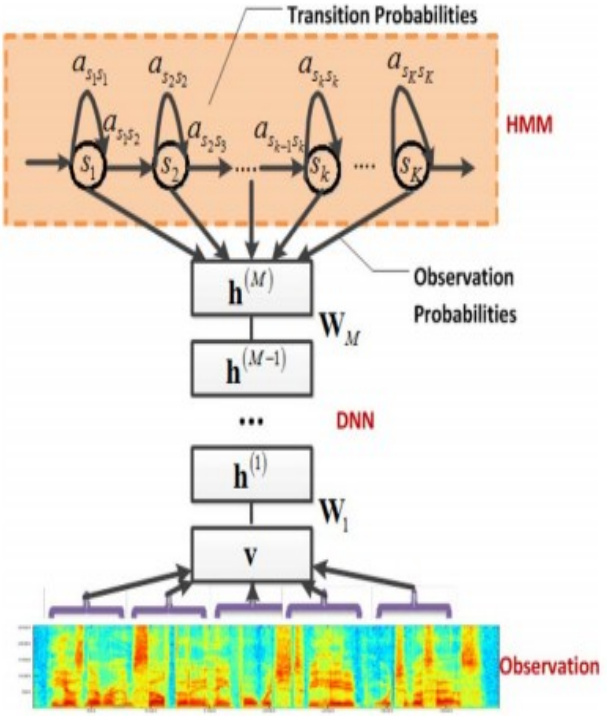
- Artifical neural network
  - Introduced in the 60s
- Convolutional neural network
  - Introduced in the 80s
- Recurrent neural network
  - Introduced in the 80s
- Concepts were there but performance was sub-optimal

Perceptron

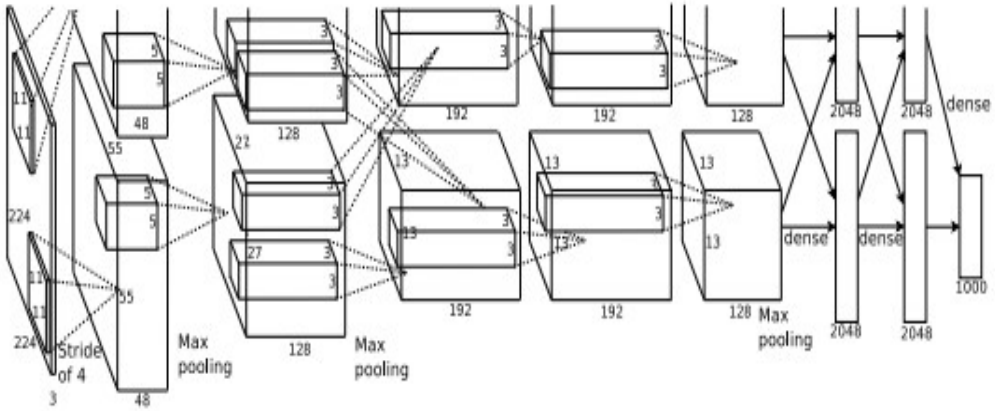
Perceptron.mp4

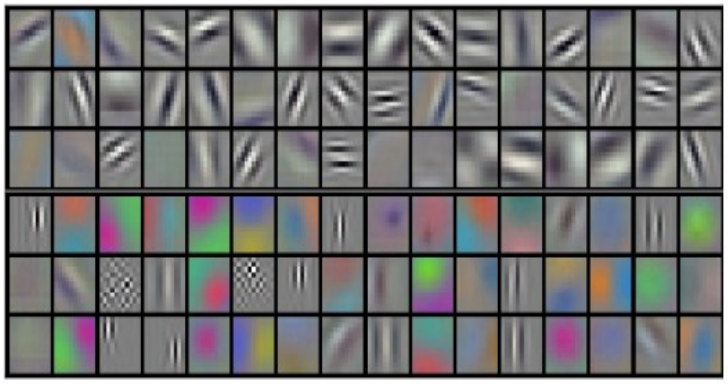
First strong results

Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition
George Dahl, Dong Yu, Li Deng, Alex Acero, 2010



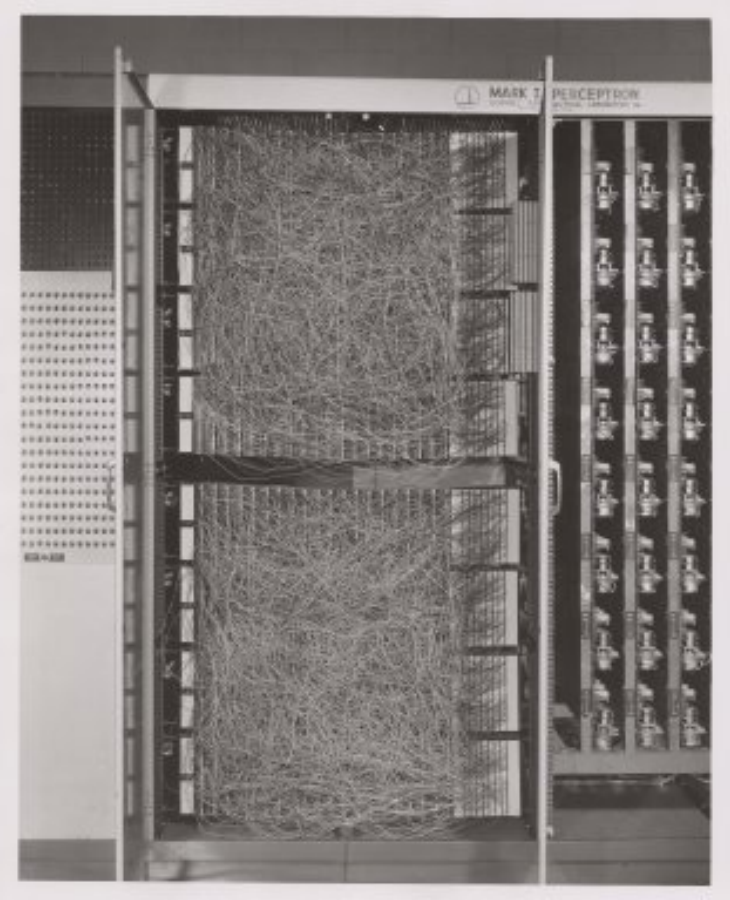
Imagenet classification with deep convolutional neural networks
Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton, 2012

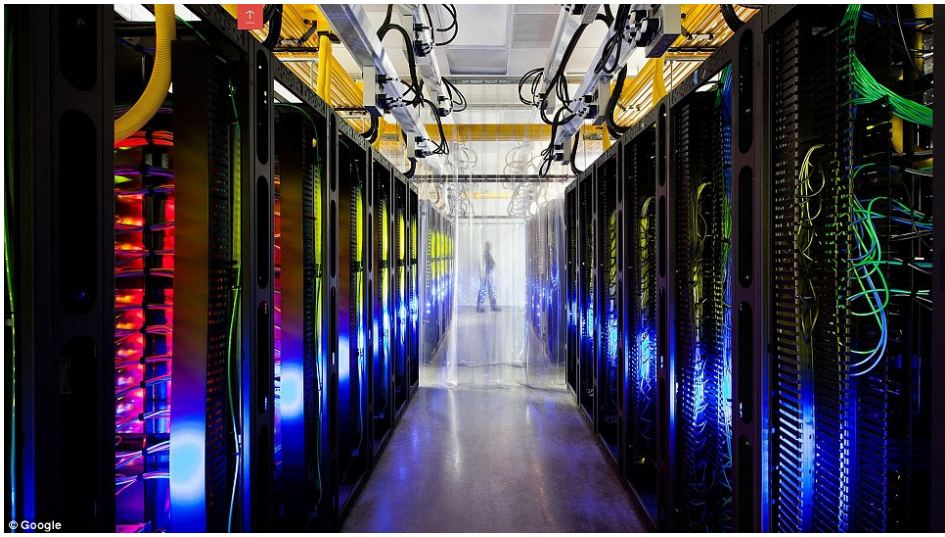




http://cs231n.github.io/

What Changed -Computational Power





I don’t Have a Cluster at Home

http://www.geforce.com/whats-new/articles/introducing-nvidia-geforce-gtx-titan-z

What Changed – Data Size

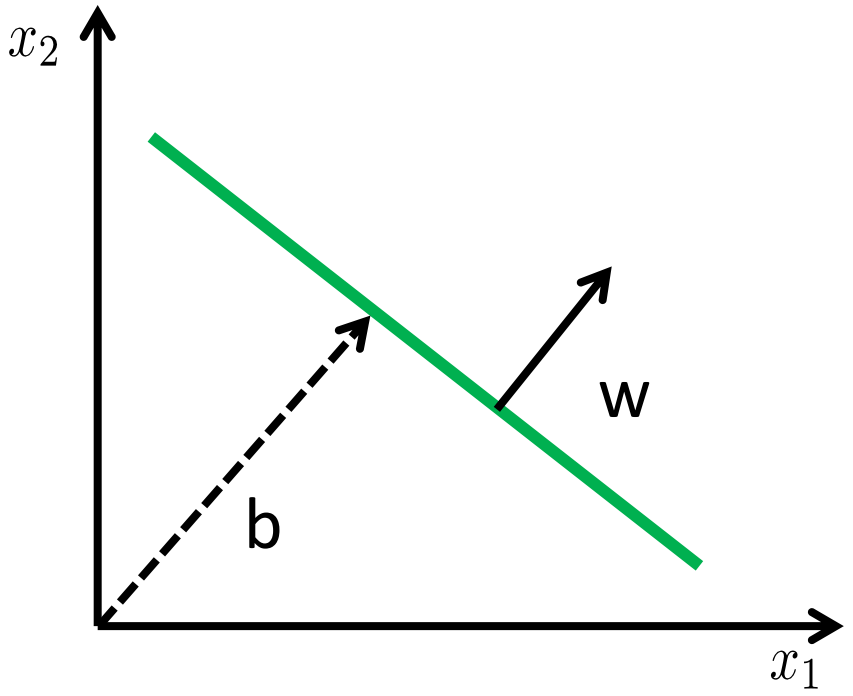
What is a Perceptron?

- the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence. (NYT 1958)
  http://en.wikipedia.org/wiki/Perceptron

- a separating hyperplane

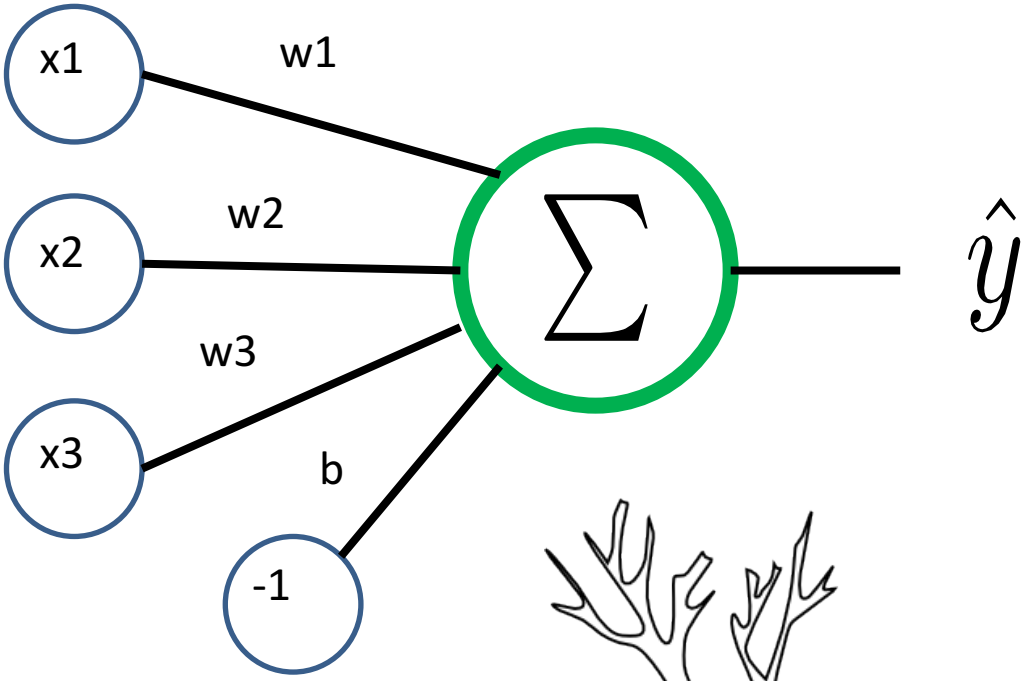
Separating Hyperplane

- x: data point
- y: label $\in \{-1, +1\}$
- w: weight vector
- b: bias

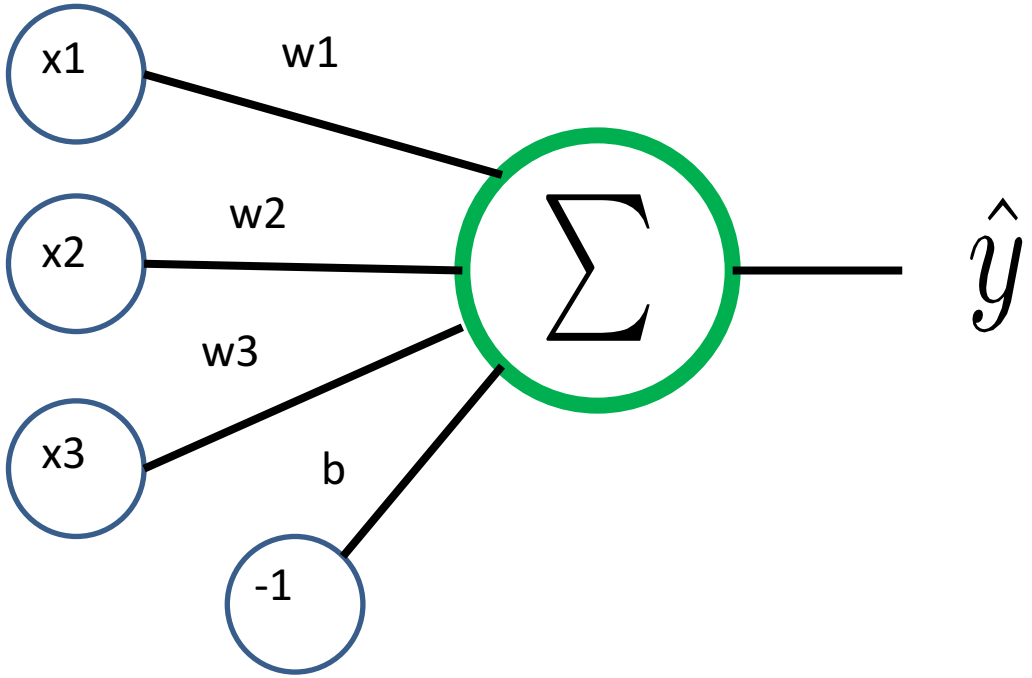


$$w^T x + b = 0$$

Perceptron

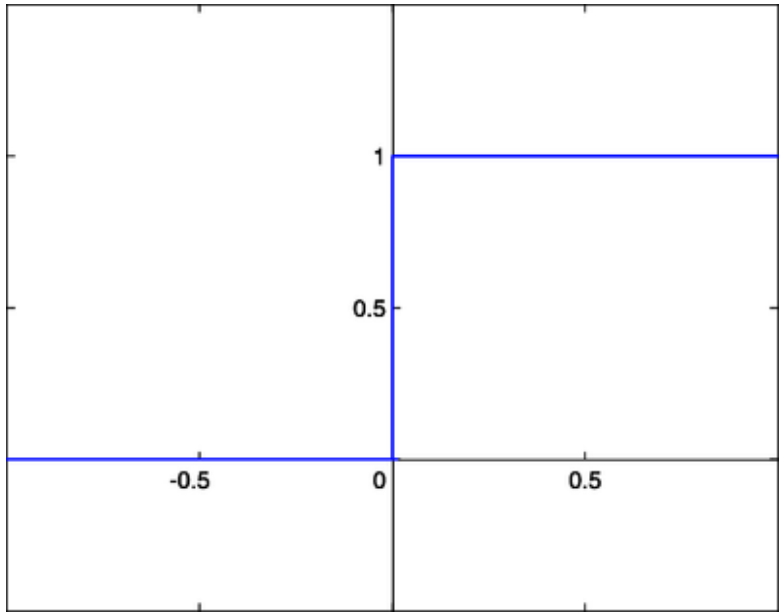


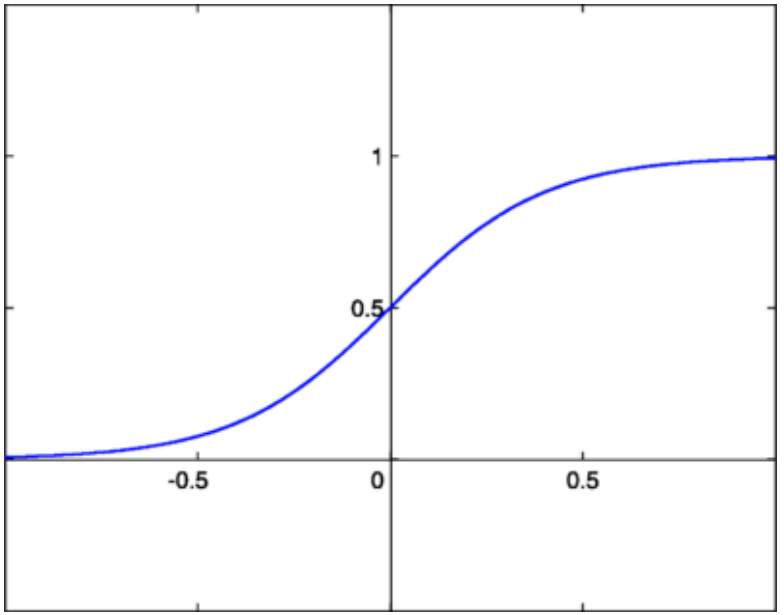
Perceptron



$$s(b + w^T x)$$

Side Note: Step vs Sigmoid Activation





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

Perceptron Fun Facts

- invented 1957
- by Frank Rosenblatt

- the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence. (NYT 1958)

  (http://en.wikipedia.org/wiki/Perceptron

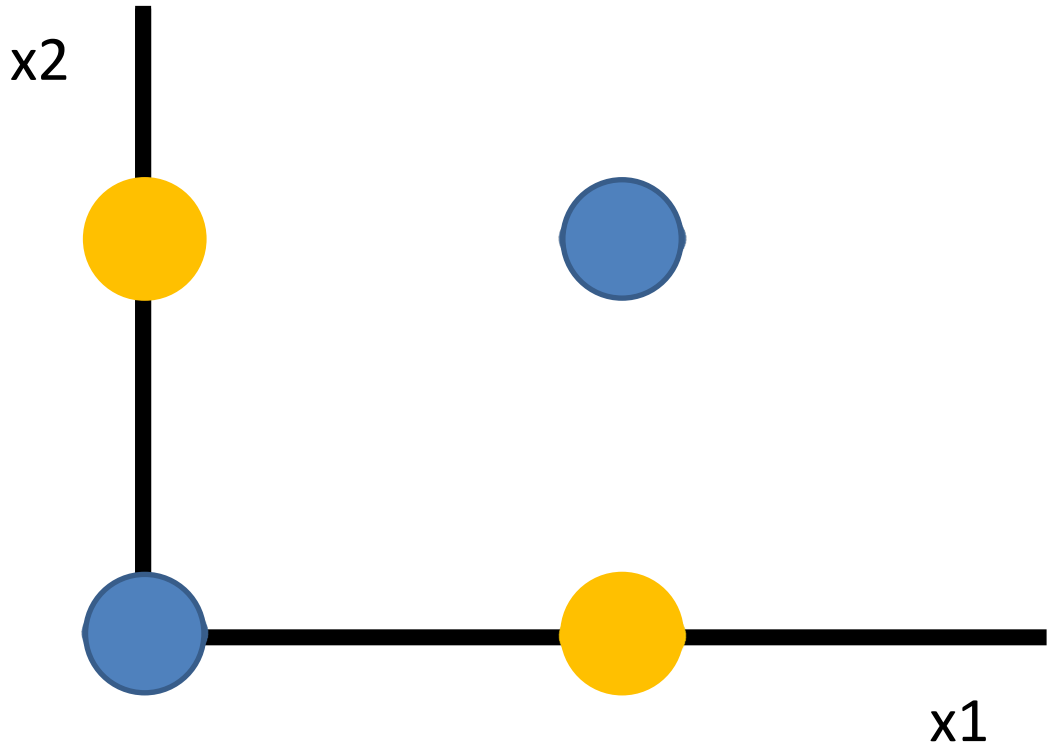
The Critics

- 1969: Minsky and Papert publish their book “Perceptrons”

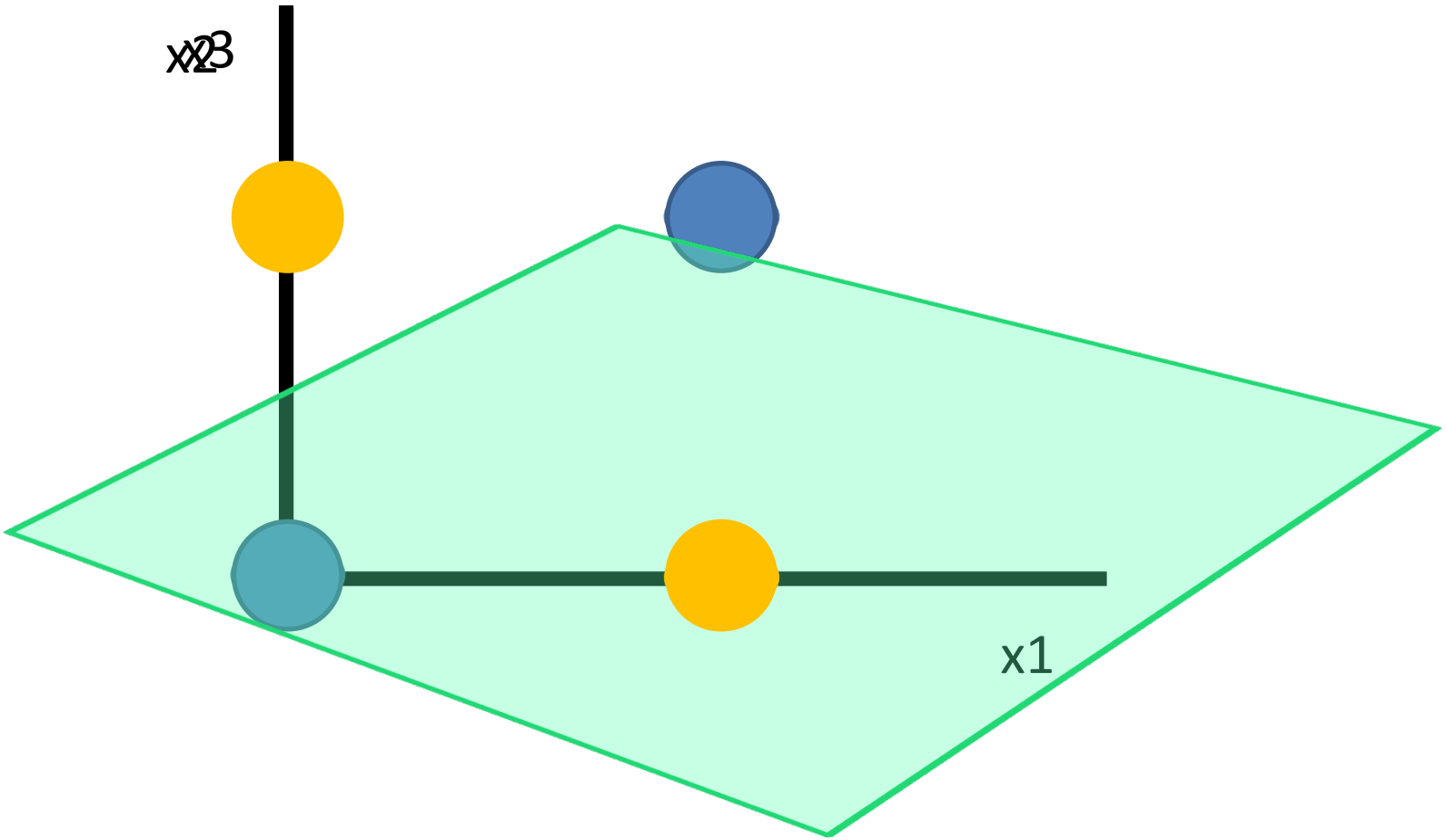
- Very controversial book, some blame the book for causing the whole research area to stagnate.

https://en.wikipedia.org/wiki/Perceptrons_(book)

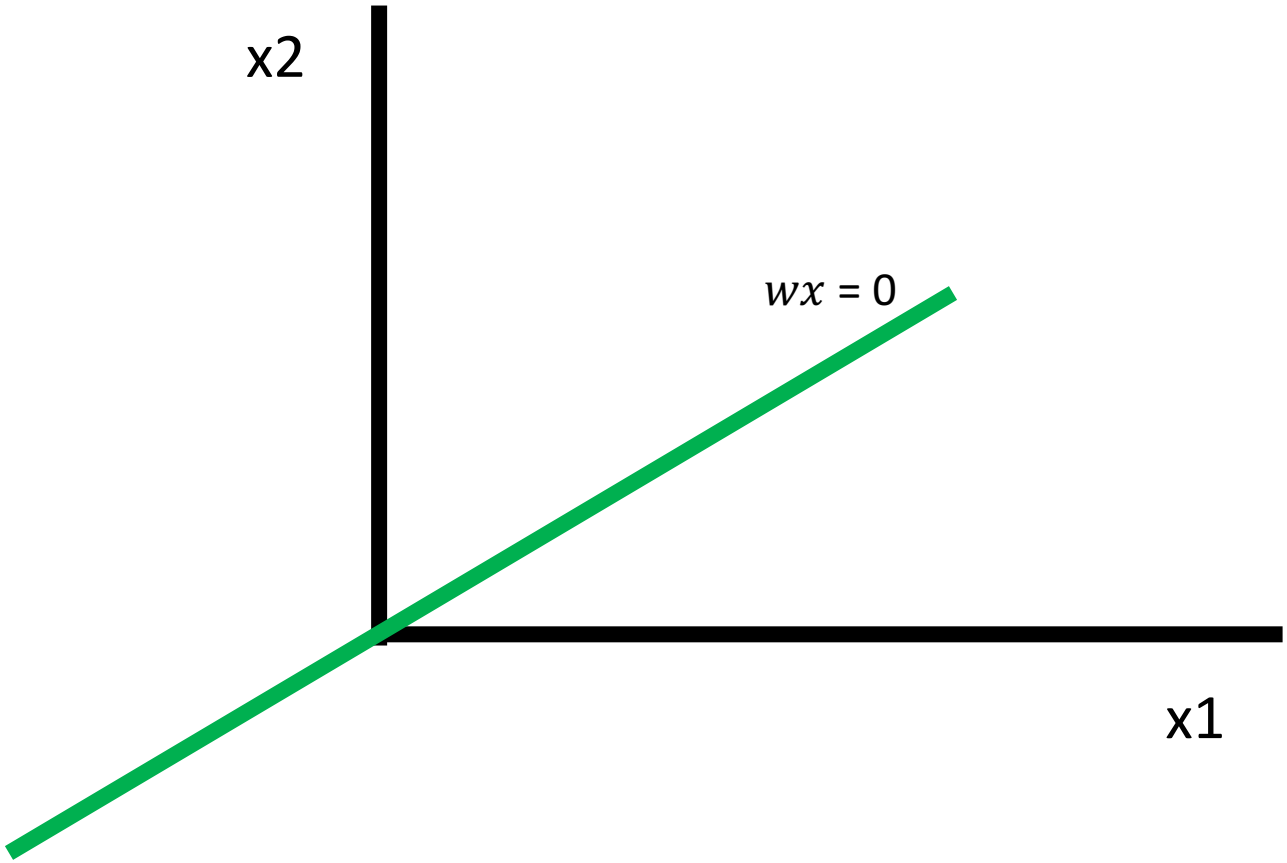
The XOR Problem



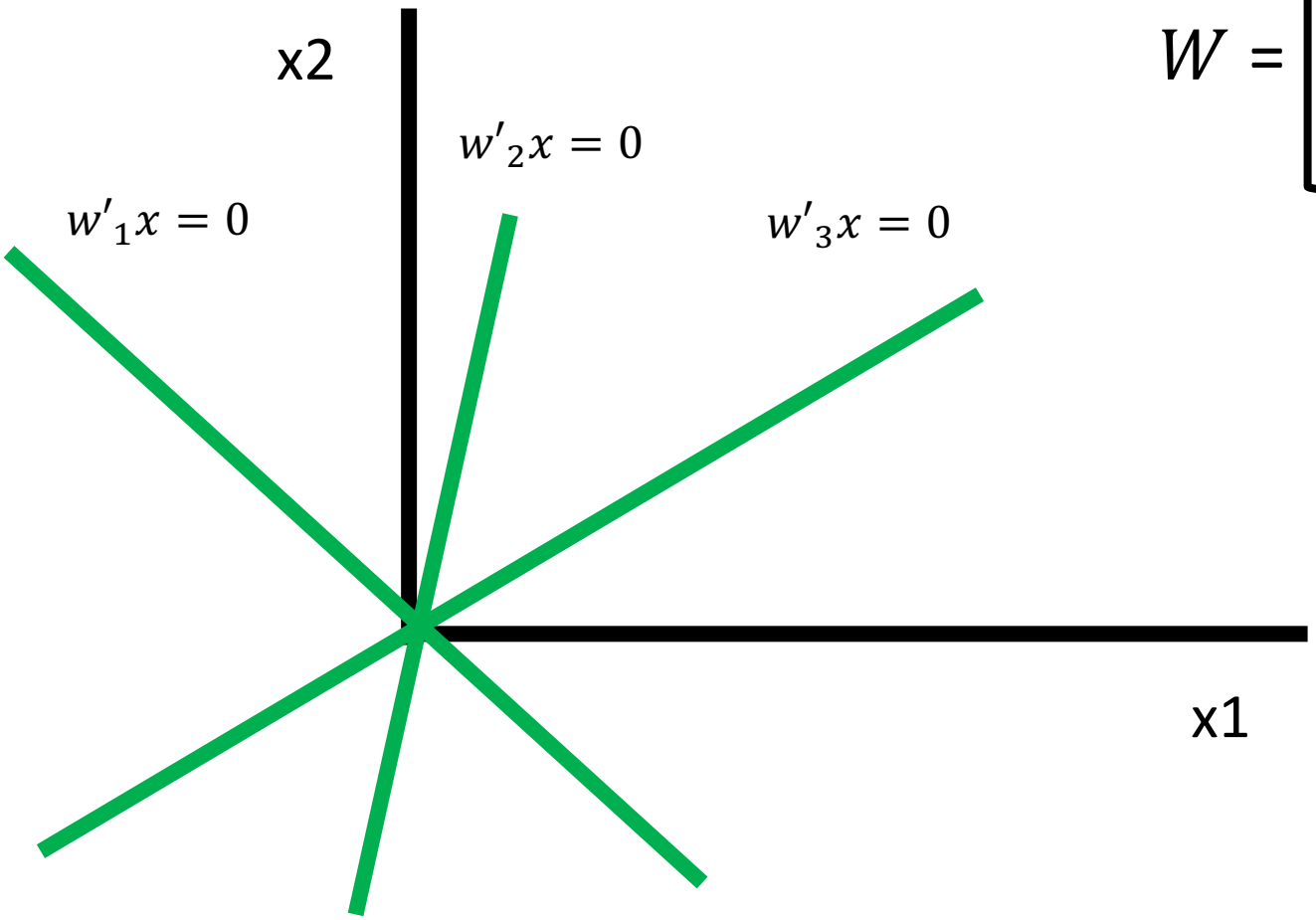
The XOR Problem



Perceptron



Multi-Perceptron

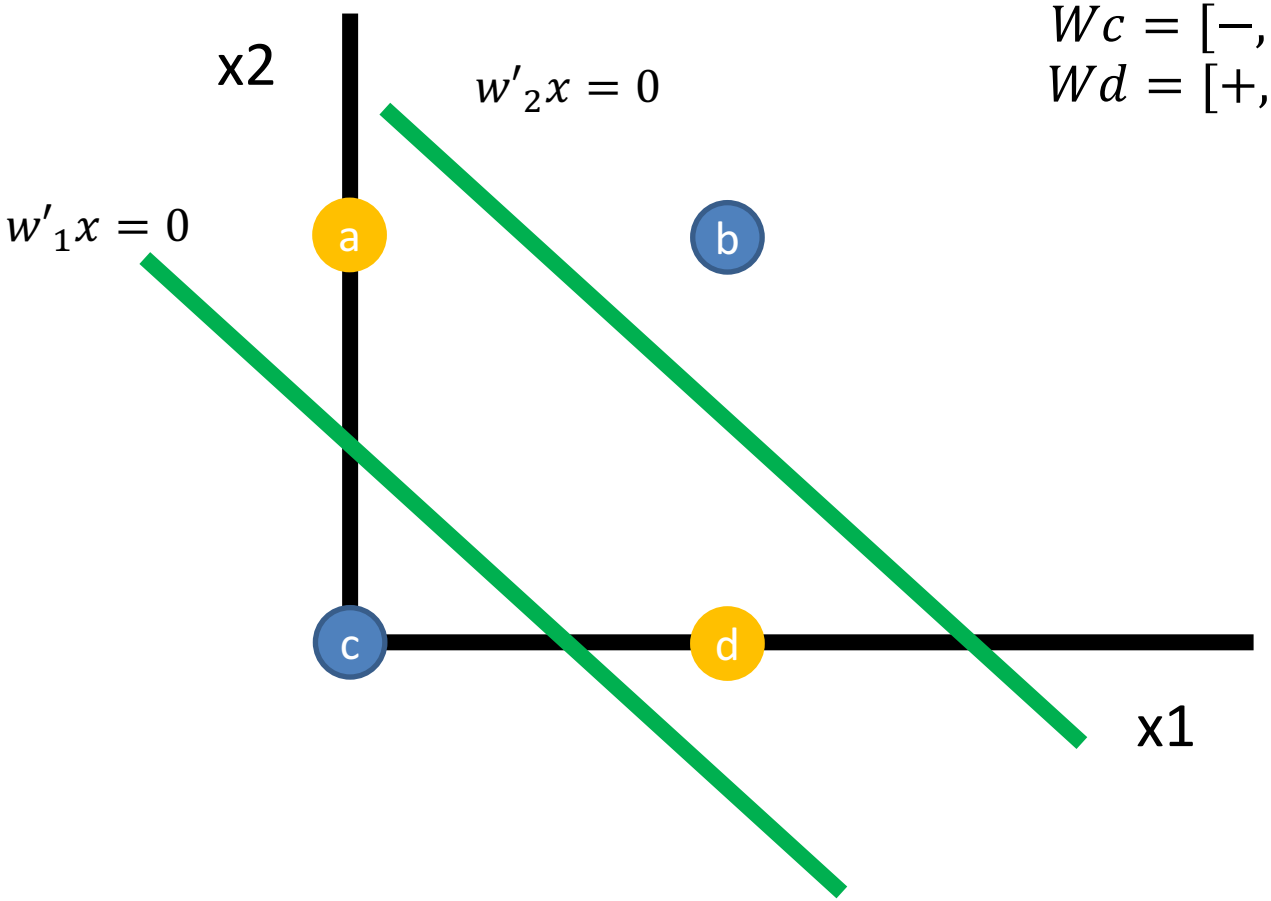


$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix}$$

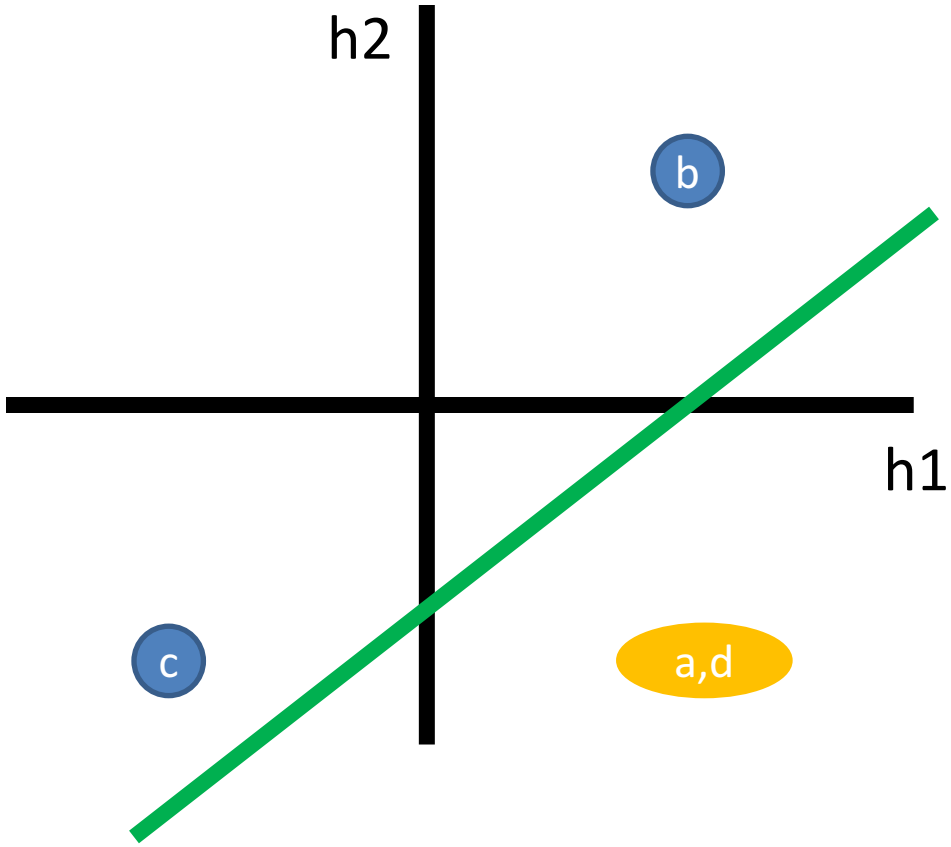
$$Wx = ?$$

Xor Problem

$$Wa = [+, -]$$
$$Wb = [+, +]$$
$$Wc = [-, -]$$
$$Wd = [+, -]$$

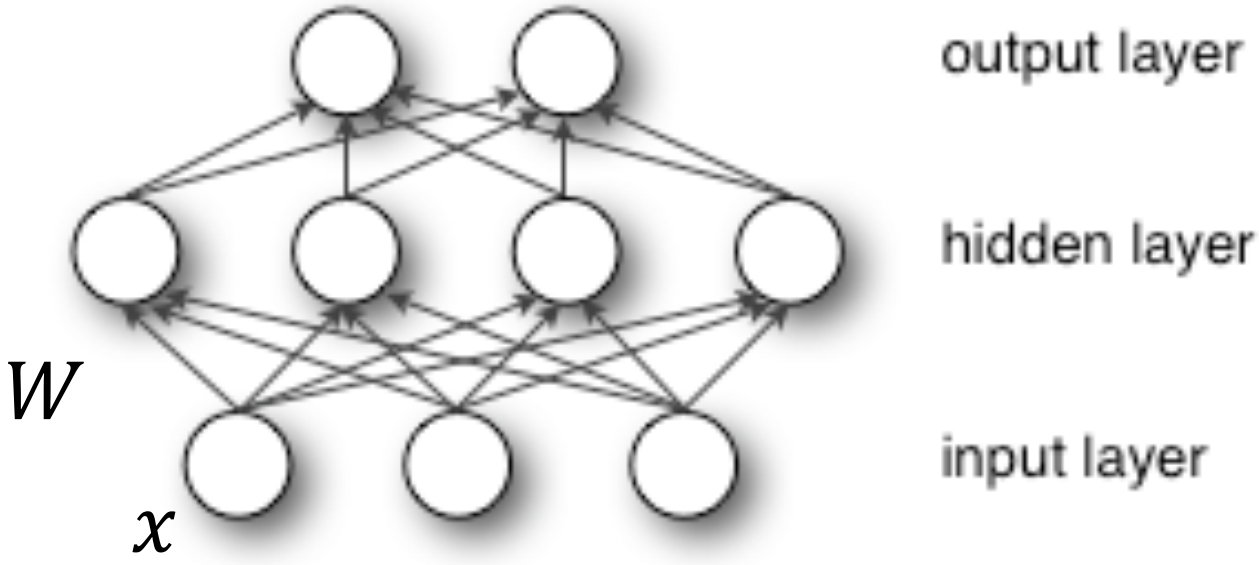


Xor Problem



$$Wa = [+, -]$$
$$Wb = [+, +]$$
$$Wc = [-, -]$$
$$Wd = [+, -]$$

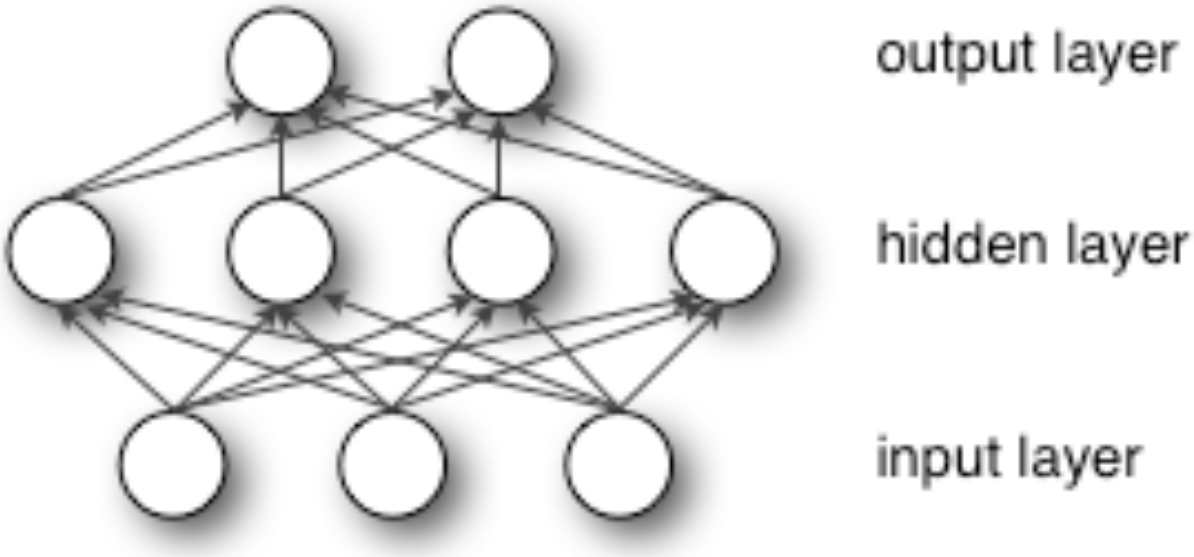
Multi-Layer Perceptron



$$s\left(b^{(1)} + W^{(1)}x\right)$$

http://deeplearning.net/tutorial/mlp.html

Multi-Layer Perceptron

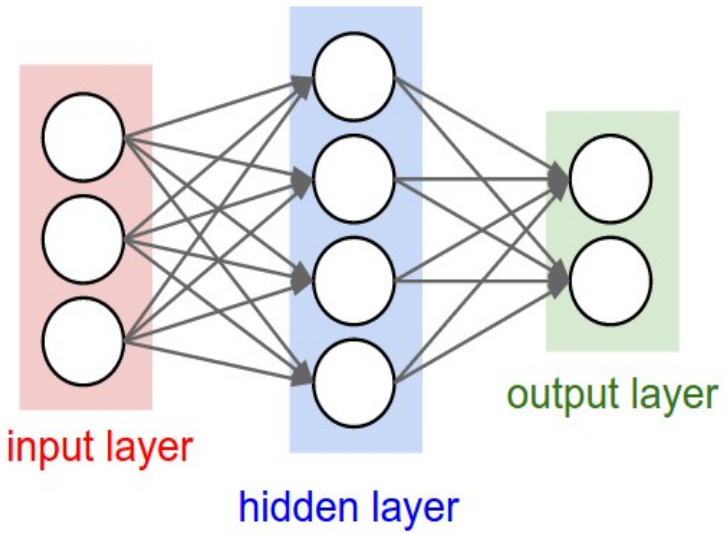


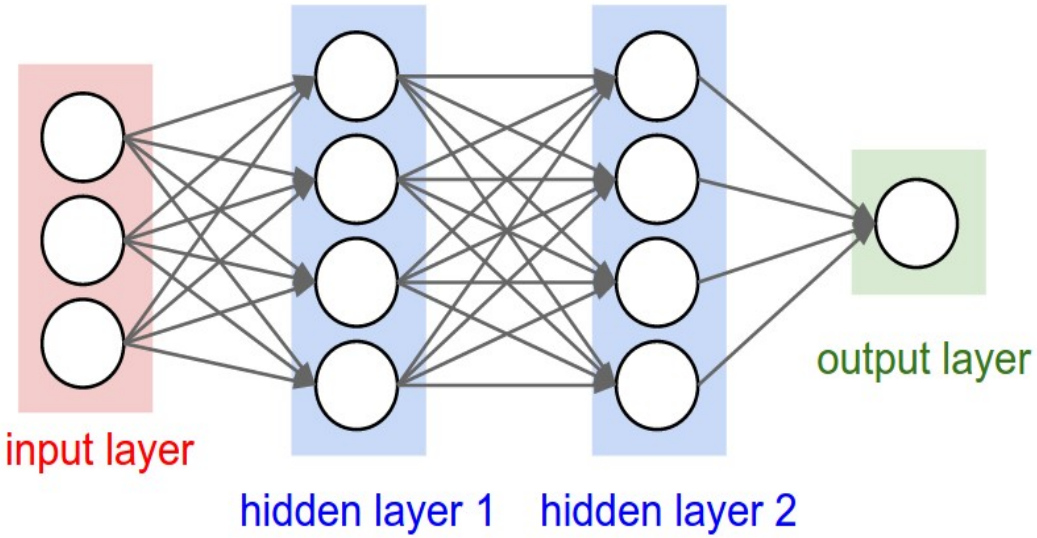
$$f(x) = G(b^{(2)} + W^{(2)}\left(s\left(b^{(1)} + W^{(1)}x\right)\right))$$

G: logistic function, softmax for multiclass

http://deeplearning.net/tutorial/mlp.html

Neural Networks: Architectures



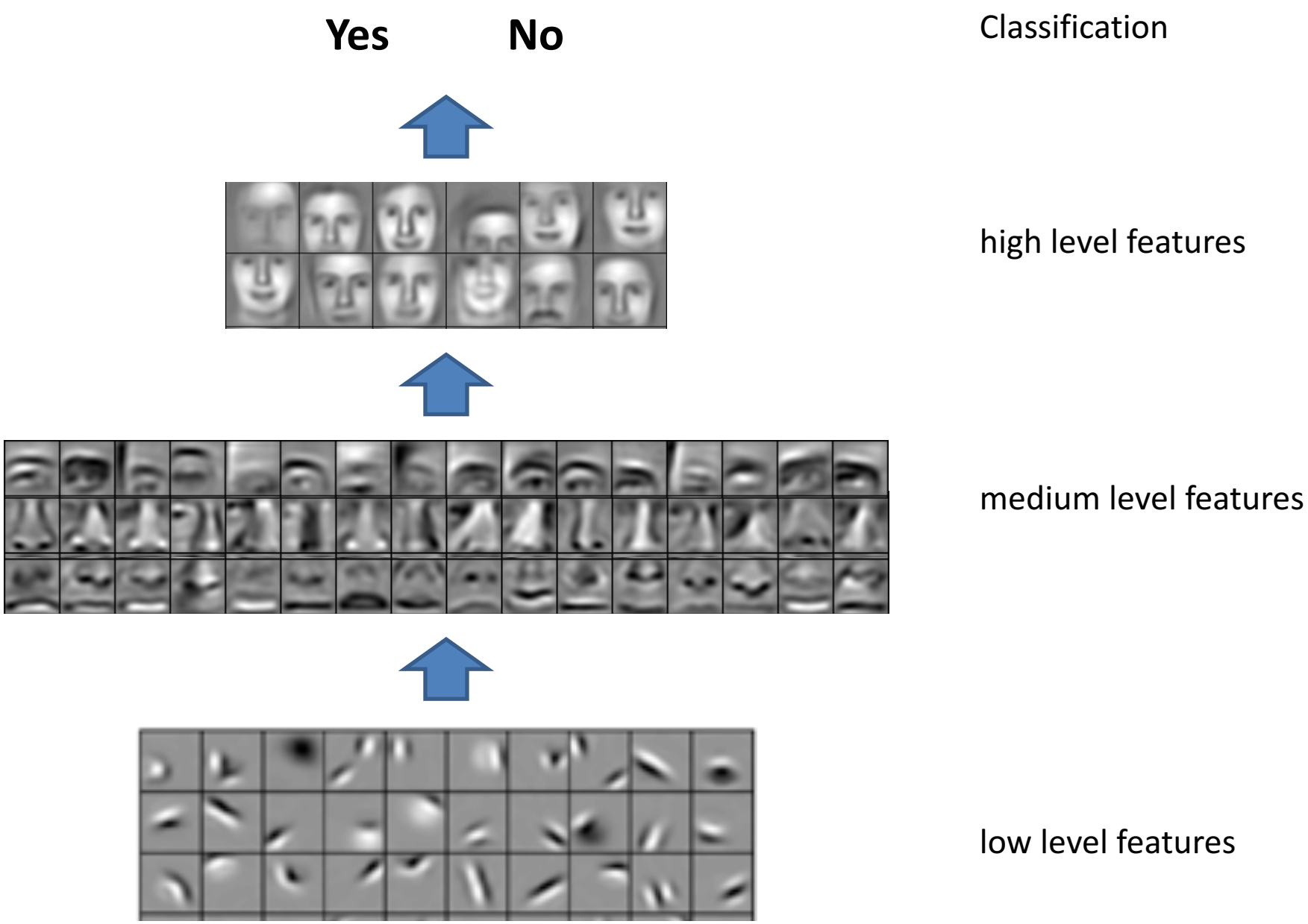


"2-layer Neural Net", or
"1-hidden-layer Neural Net"

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

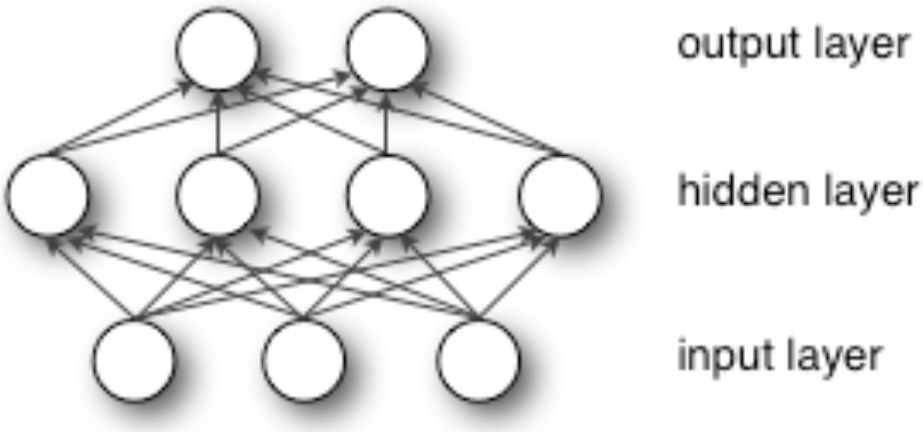
"Fully-connected" layers

http://cs231n.github.io/

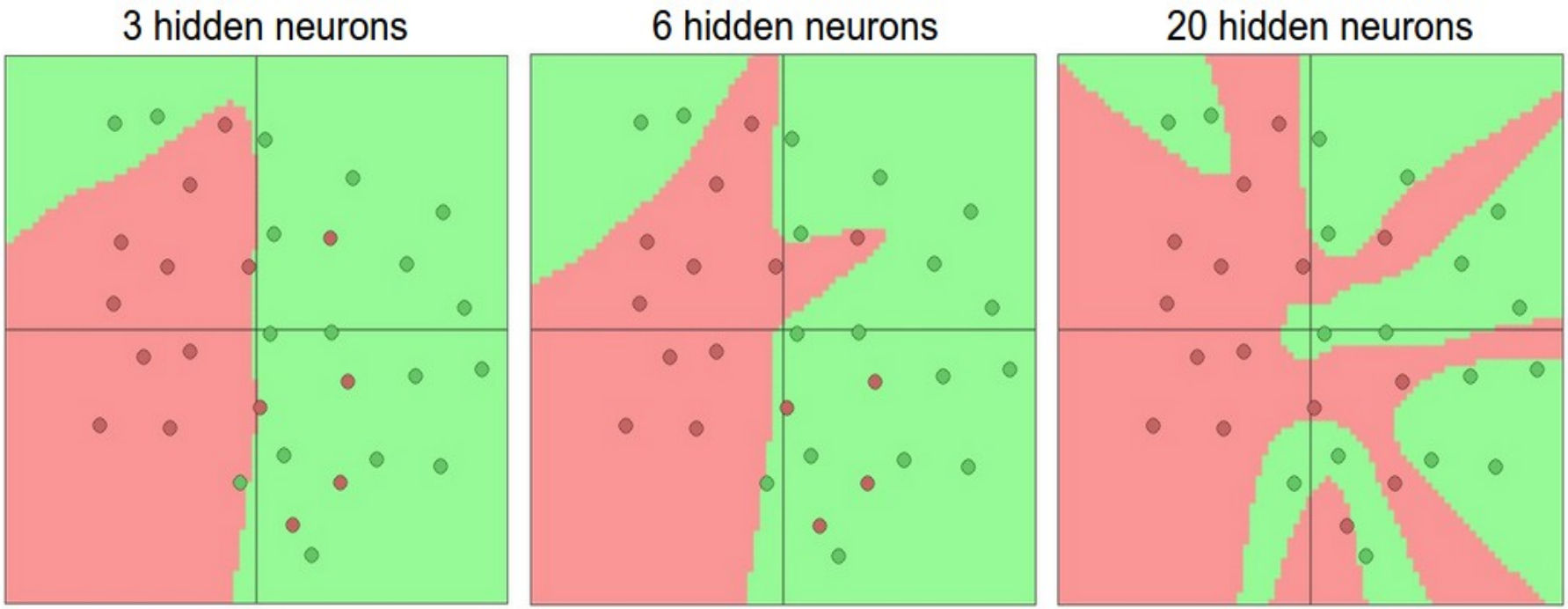

Yes
No
Classification
high level features
medium level features
low level features

To Define a Deep Network Architecture We Need:

- Input layer size
- Number of hidden layers
- Sizes of hidden layers
- Activation functions
- Number of output units



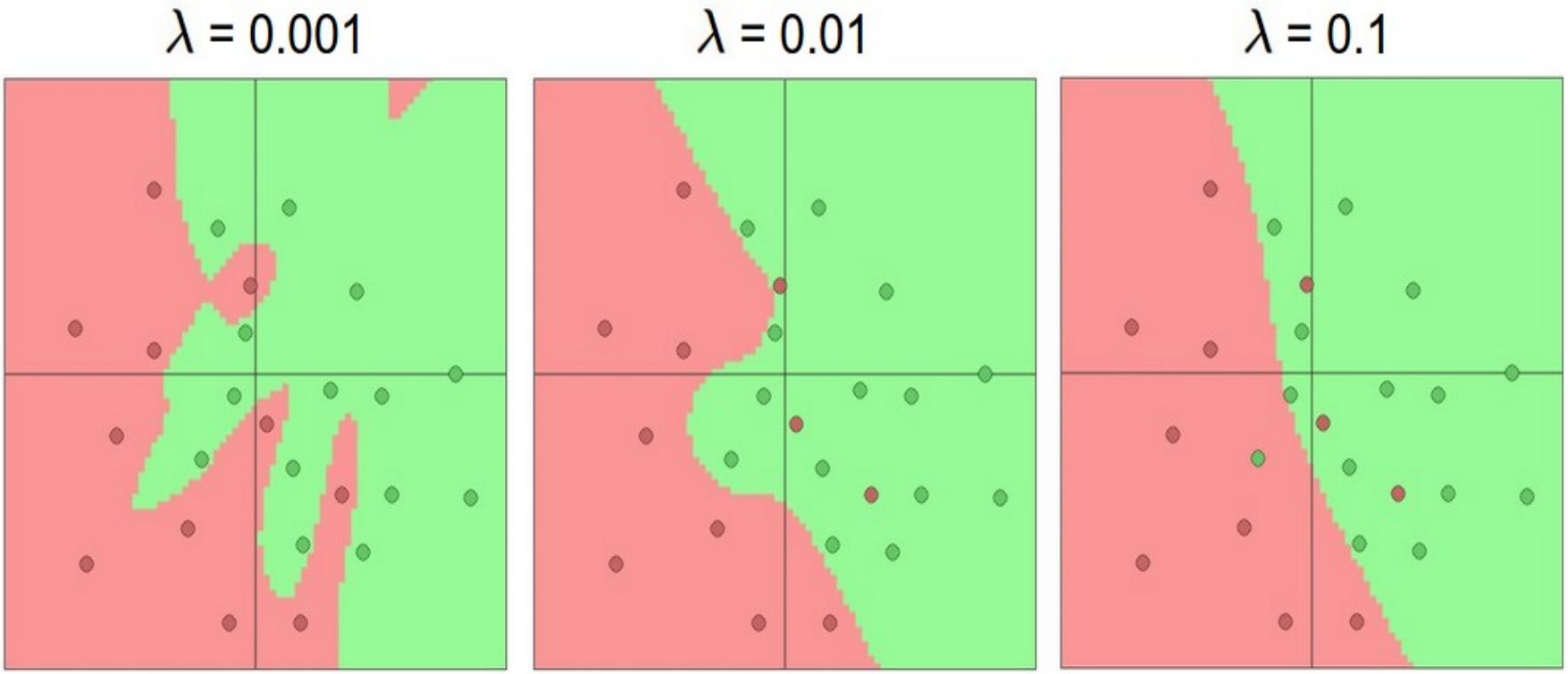
Setting the number of layers and their sizes



more neurons = more capacity

http://cs231n.github.io/

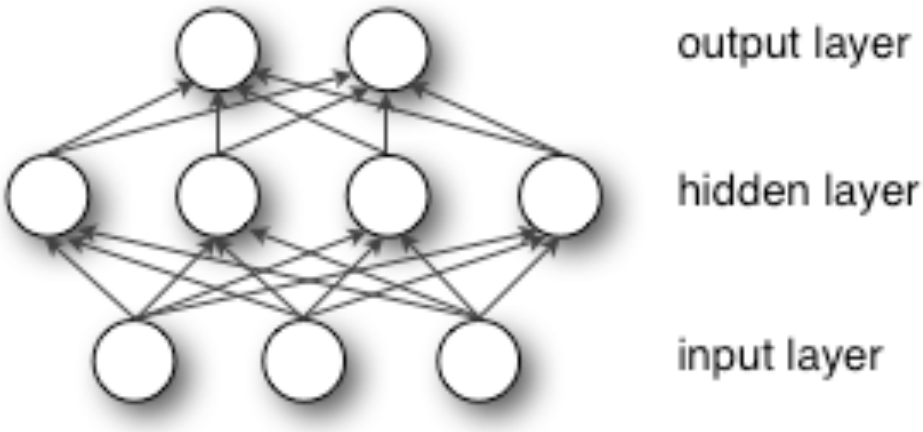
Do not use size of neural network as a regularizer. Use stronger regularization instead:



(you can play with this demo over at ConvNetJS: http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html)

To Define a Deep Network Architecture We Need:

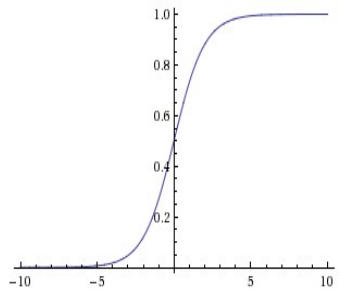
- Input layer size
- Number of hidden layers
- Sizes of hidden layers
- Activation functions
- Number of output units



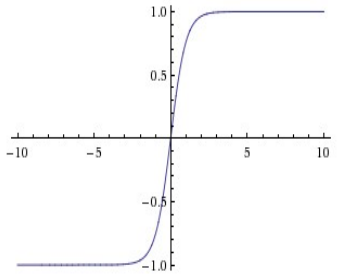
Activation Functions

Sigmoid

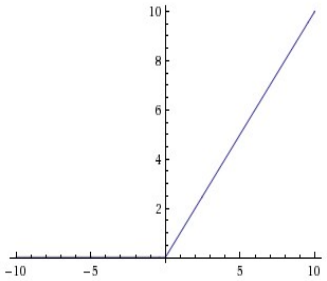
$$\sigma(x) = 1/(1+e^{-x})$$



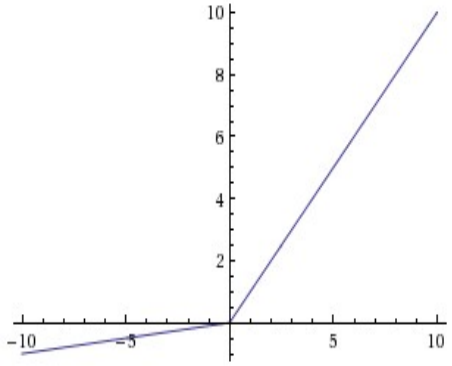
tanh tanh(x)



ReLU max(0,x)



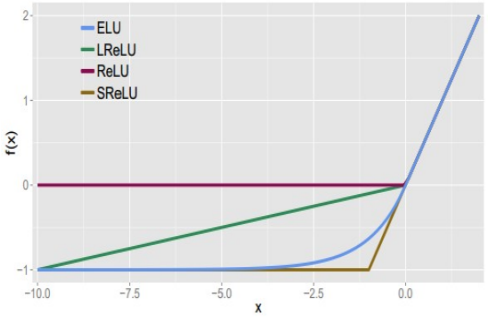
Leaky ReLU
max(0.1x, x)



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

ELU

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha\,(\exp(x) - 1) & \text{if } x \leq 0 \end{cases}$$



http://cs231n.github.io/

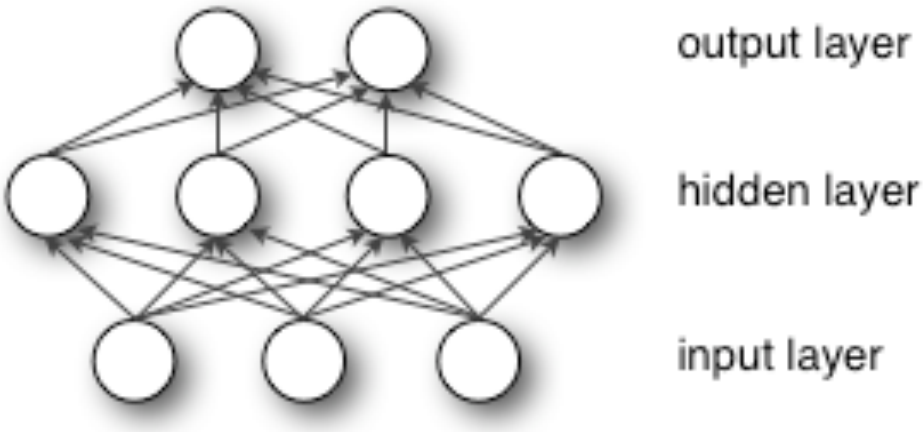
Why non-Linear Activation?

- To build a non-linear classifier
- Without non-linear activation all layers collapse and can be represented by just linear hyperplanes.

- $f(x) = W_2 (W_1 x + b_1) + b_2$
- $f(x) = W_2 W_1 x + W_2 b_1 + b_2$
- $f(x) = W_{21} x + b_{21}$

To Define a Deep Network Architecture We Need:

- Input layer size
- Number of hidden layers
- Sizes of hidden layers
- Activation functions
- Number of output units



Last Layer

- For multiclass typically uses softmax function

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

- Like logistic regression for multiclass
- Values between 0 and 1
- Adding up to 1
- Classes are assumed to be mutually exclusive

Training of Neural Networks

- For training we need:
  - Loss function
  - Optimization method

Loss Function

- quantifies our unhappiness with the scores across the training data
- Training: Find parameters that minimize the loss function

training samples

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{T} \sum_{t} l(f(\mathbf{x}^{(t)}; \boldsymbol{\theta}), y^{(t)})$$

model parameter

training labels

Loss Function

- Classification error would be good, but it's not smooth
- Need to find smooth proxy

- Last layer is basically logistic regression
- The whole network estimates class probabilities

$$P(Y = i|x, W, b) = softmax_i(Wx + b)$$

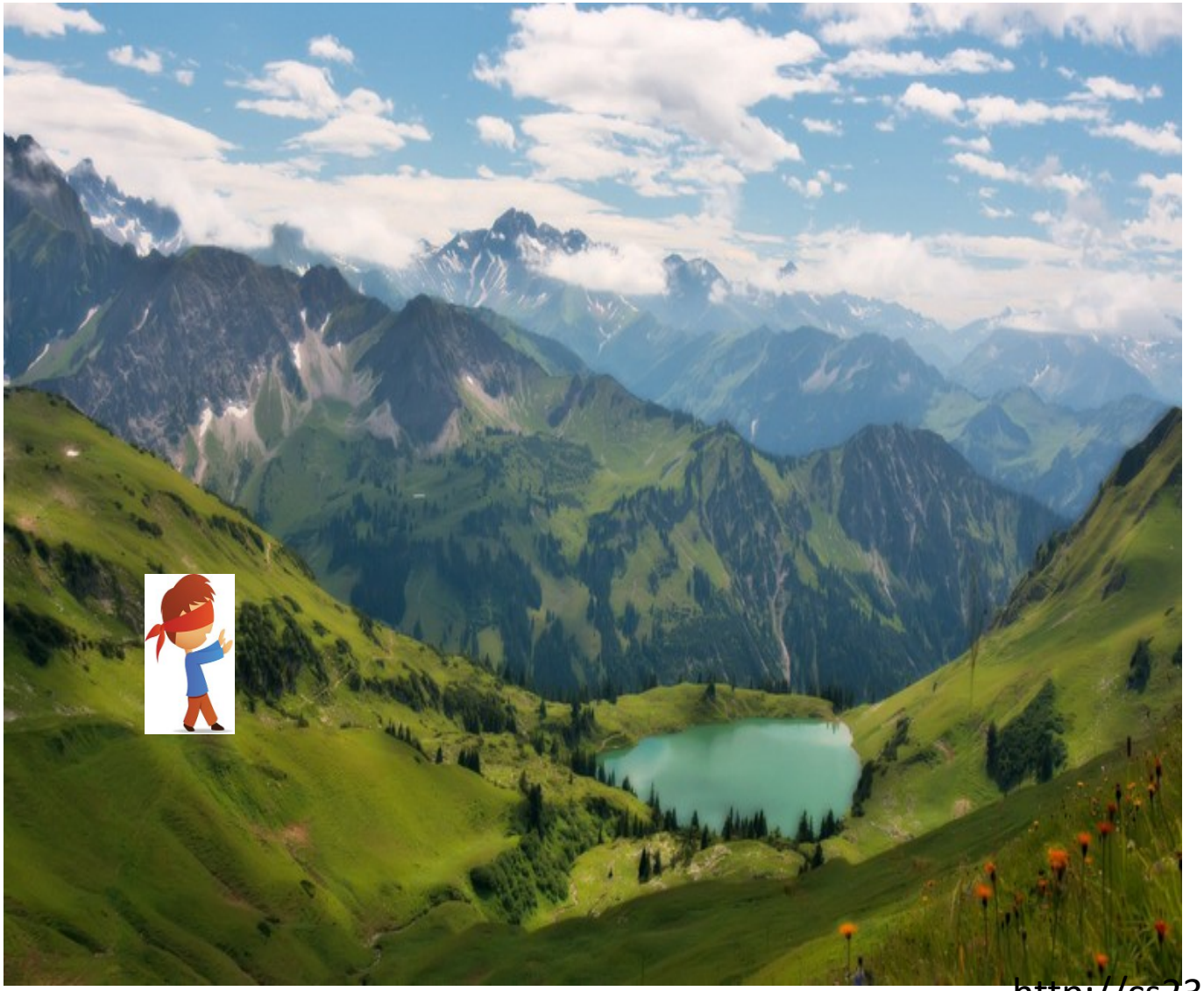
- Maximize the correct class assignment probabilities in the training set

Cross Entropy

$$-\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right]$$

- Here just two class problem
- N is the number of samples
- y_n denotes the true label of sample n

Optimization



http://cs231n.github.io/

Training with SGD

- General optimization problem:
- Find the minimum of $J(\theta)$

- Updates: $\theta = \theta - \eta \nabla_\theta J(\theta)$

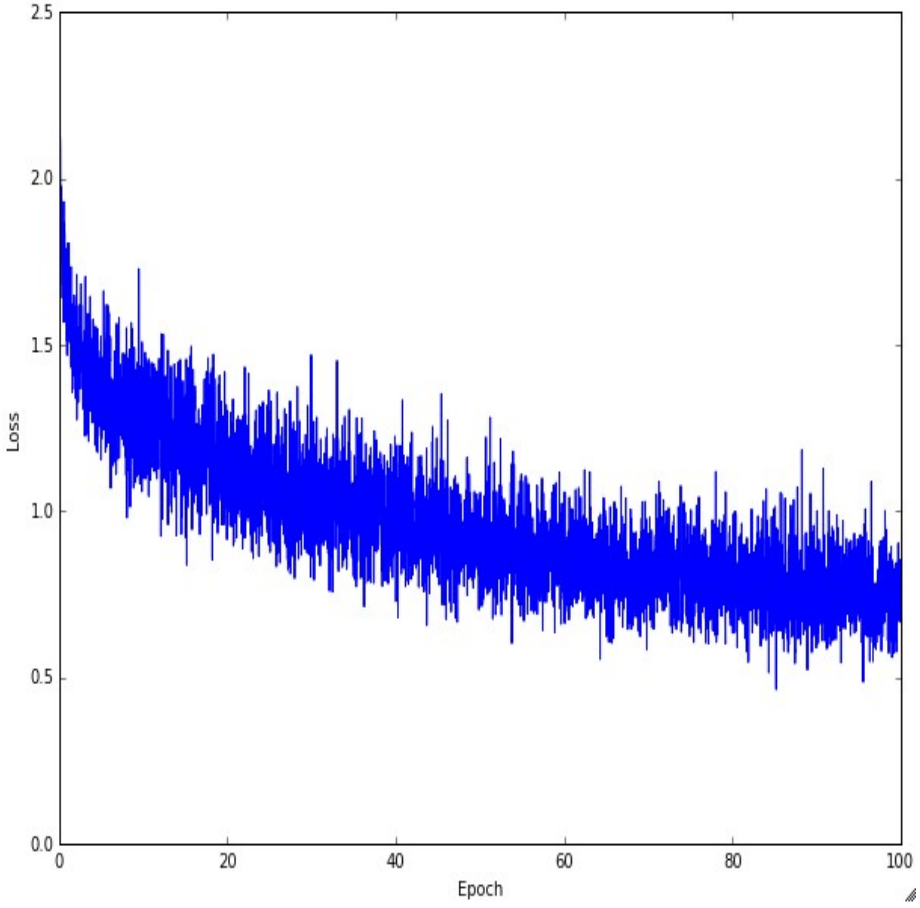
- η is called the learning rate

Put the S in SGD

- Remember our loss function sums over all samples

- We can use just a subset of all samples to estimate the gradient direction

Training with SGD

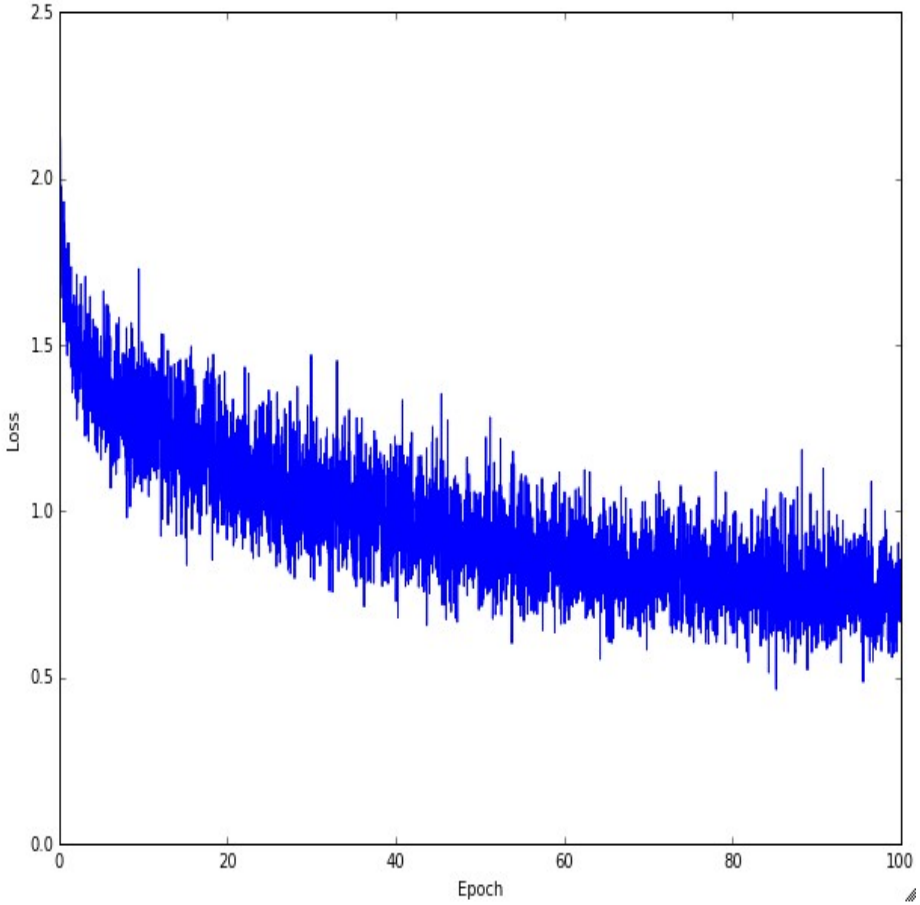


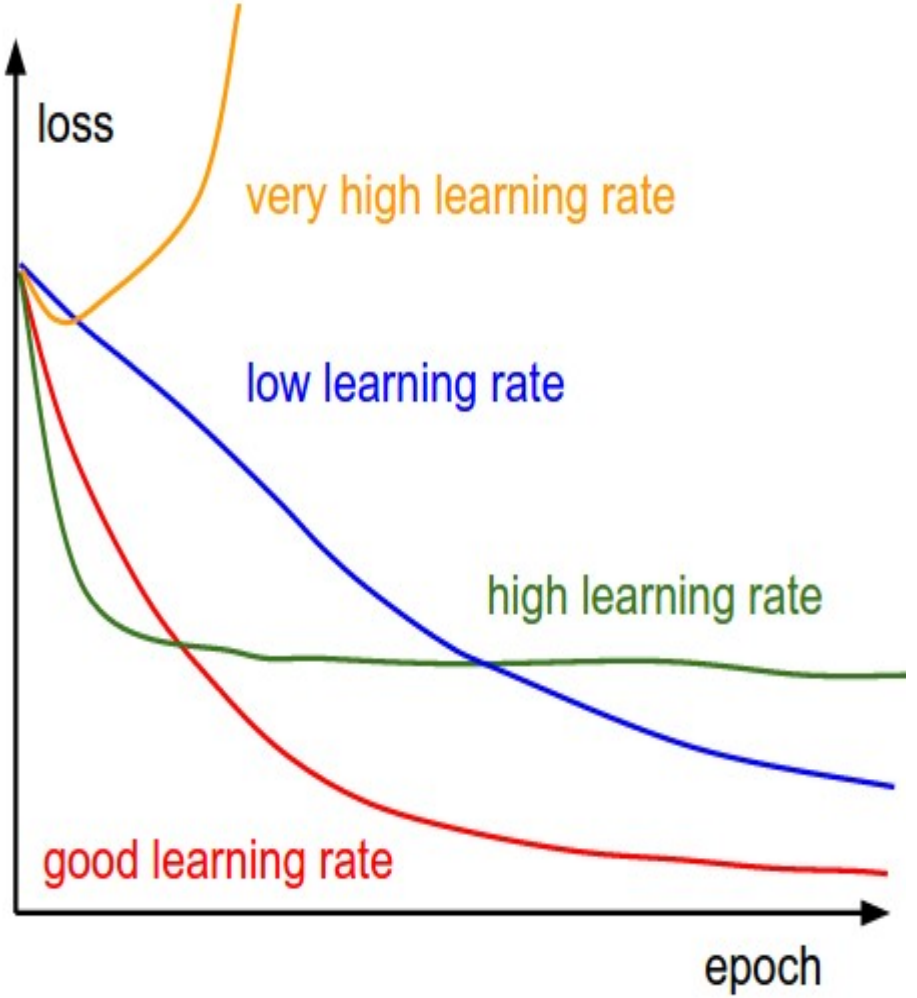
Example of optimization progress while training a neural network.

(Loss over mini-batches goes down over time.)

http://cs231n.github.io/

The effects of step size (or “learning rate”)





http://cs231n.github.io/

Batch Size

Common mini-batch sizes are 32/64/128 examples

e.g. Krizhevsky ILSVRC ConvNet used 256 examples

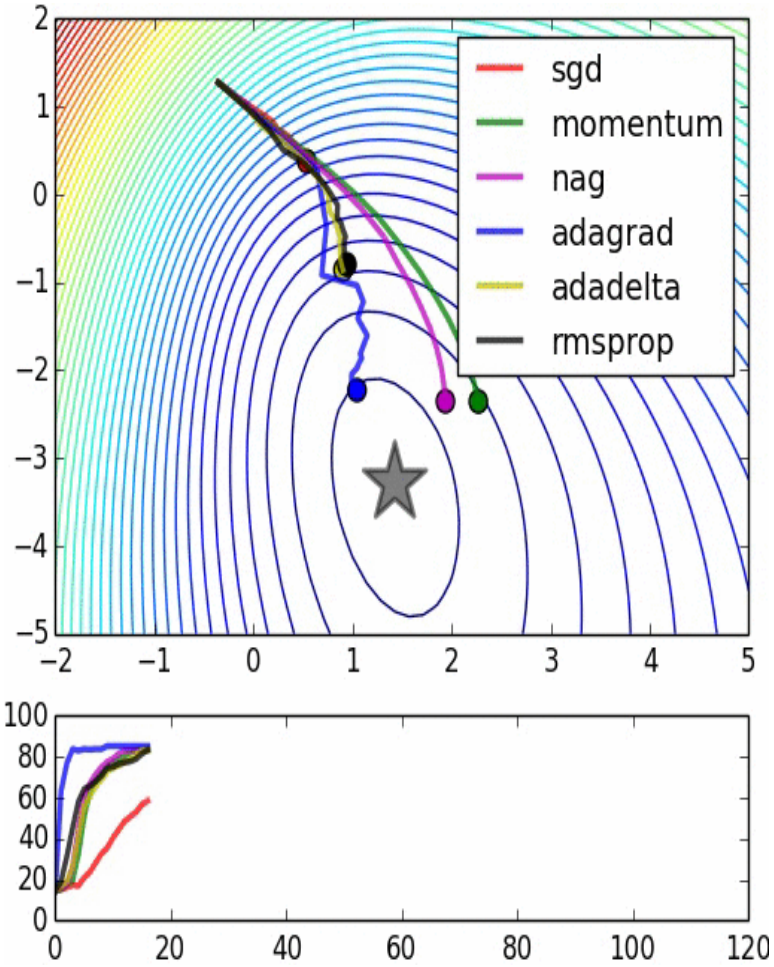
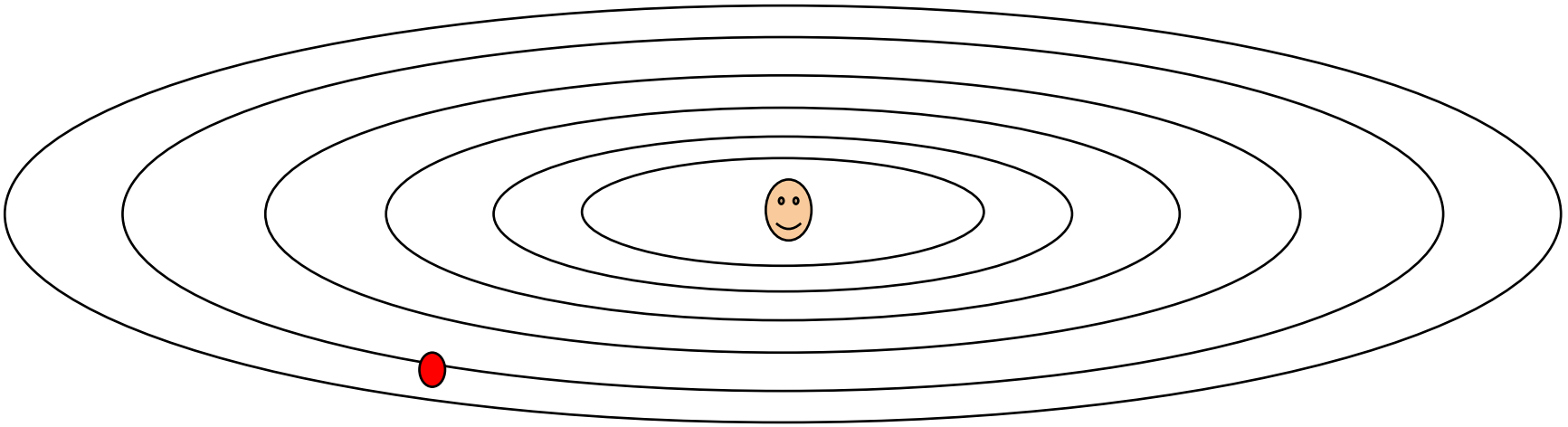


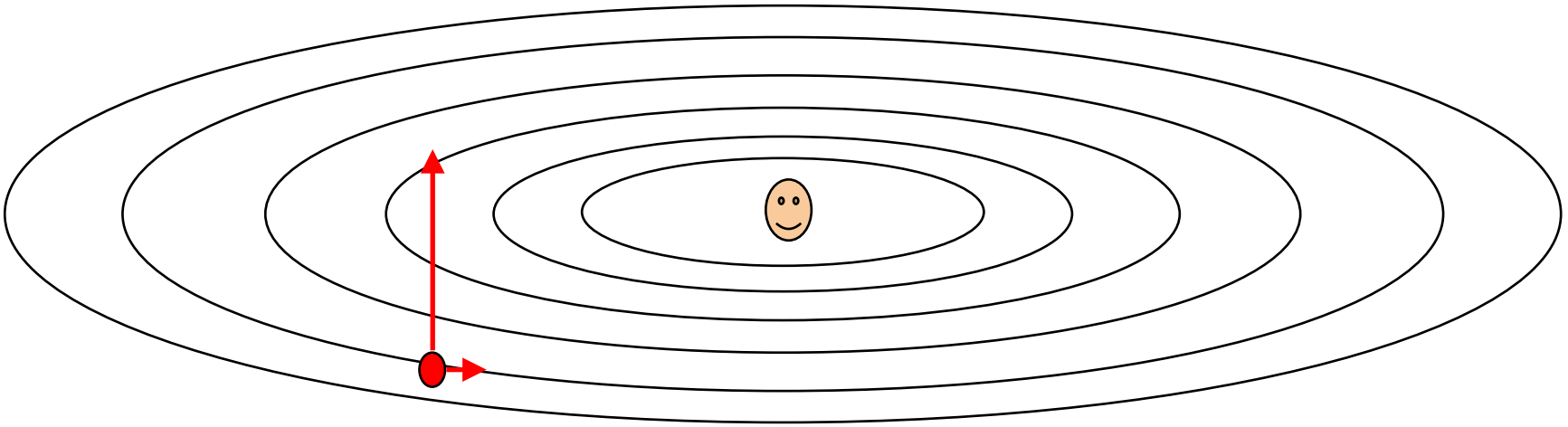
Image credits: Alec Radford

Suppose loss function is steep vertically but shallow horizontally:



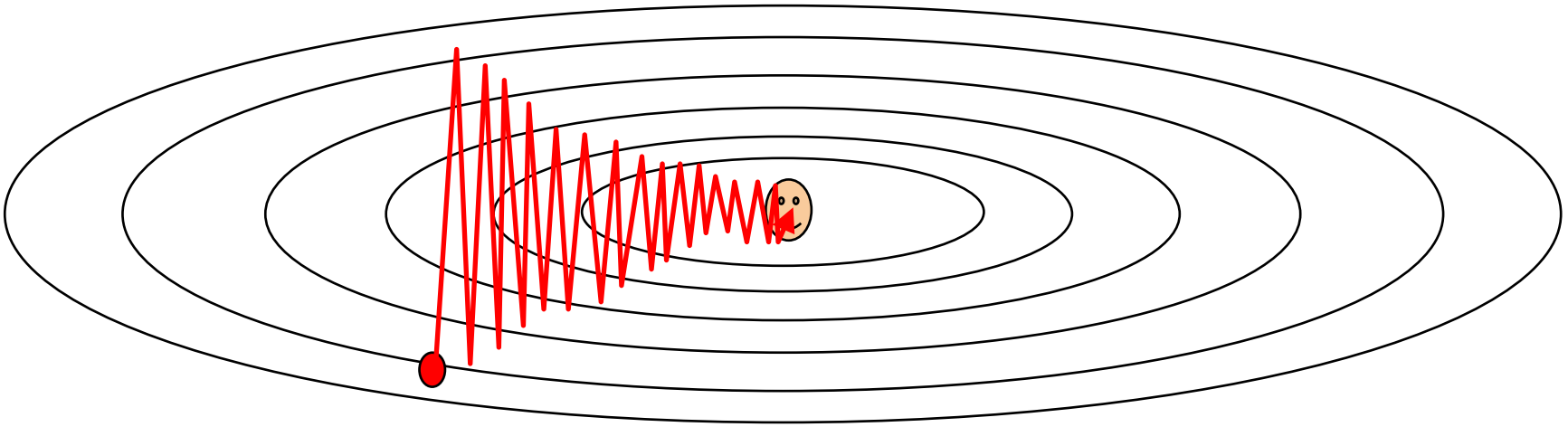
Q: What is the trajectory along which we converge towards the minimum with SGD?

Suppose loss function is steep vertically but shallow horizontally:



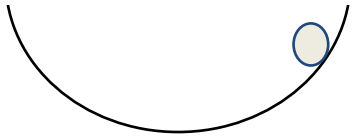
Q: What is the trajectory along which we converge towards the minimum with SGD?

Suppose loss function is steep vertically but shallow horizontally:



Q: What is the trajectory along which we converge towards the minimum with SGD? very slow progress along flat direction, jitter along steep one

Momentum update



- Physical interpretation as ball rolling down the loss function + friction (mu coefficient).
- mu = usually ~0.5, 0.9, or 0.99 (Sometimes annealed over time, e.g. from 0.5 -> 0.99)

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta).$$

$$\theta = \theta - v_t \,.$$

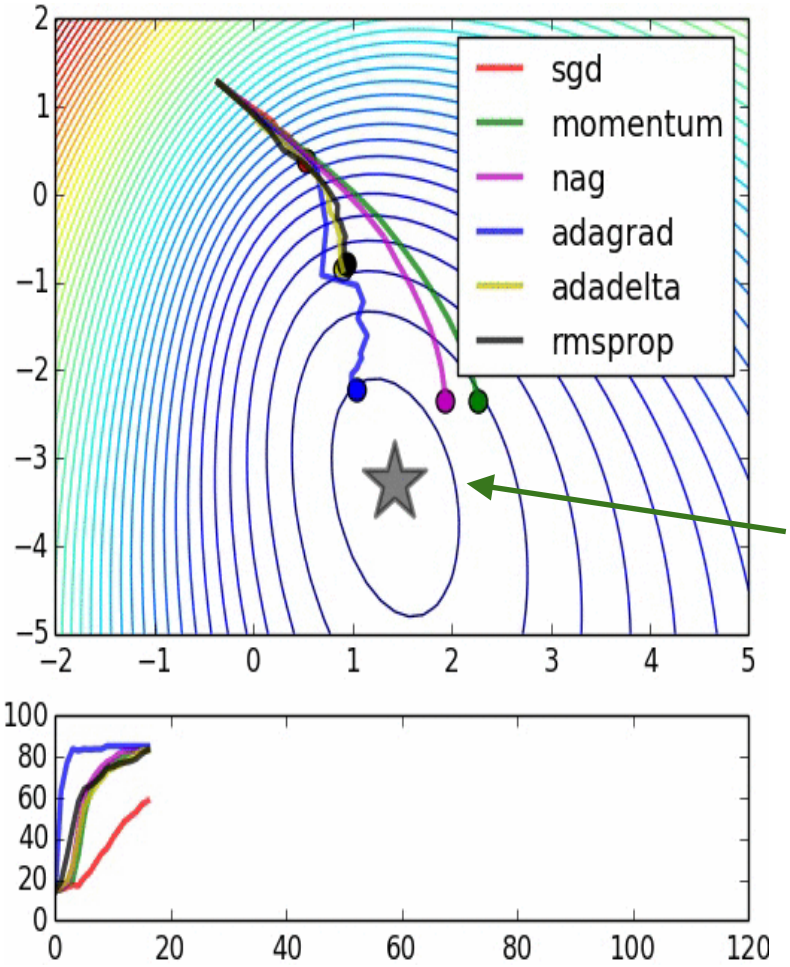
- Allows a velocity to “build up” along shallow directions
- Velocity becomes damped in steep direction due to quickly changing sign

http://cs231n.github.io/

SGD

vs

Momentum



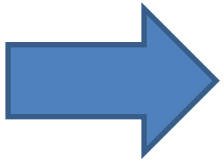
notice momentum overshooting the target, but overall getting to the minimum much faster.

http://cs231n.github.io/

Training a Simple Network

- We have everything we need now to train a simple network!

- Have a look at Keras



https://keras.io/getting-started/sequential-model-guide/

Summary

- Deep learning has been around for decades
- But only now we have the computational tools to make it work
- Once again we just have separating hyperplanes

To Train a Simple Network We Need:

- Input layer size
- Number of hidden layers
- Sizes of hidden layers
- Activation function
- Number of output units

- Loss function
- Optimization method