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

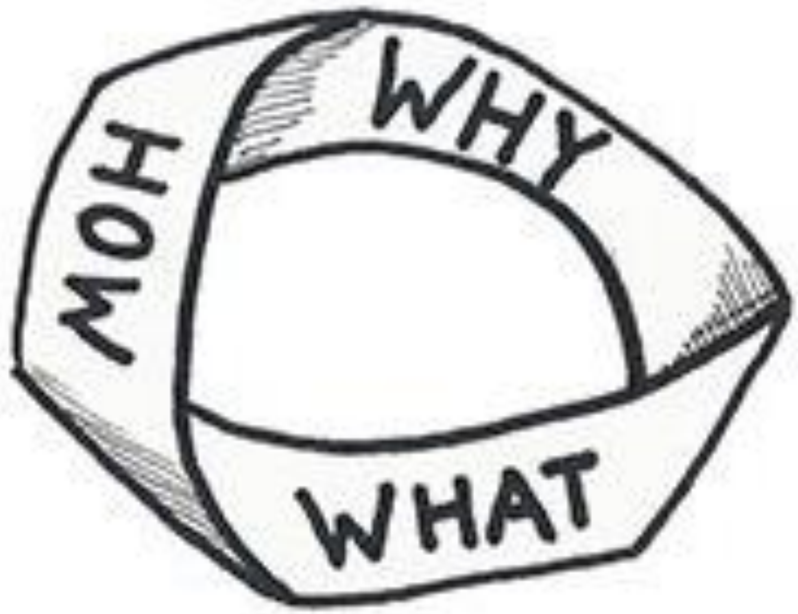
IS "DEEP LEARNING" A REVOLUTION IN ARTIFICIAL INTELLIGENCE?

BY GARY MARCUS

Share Tweet g+1

Deep Learning - The Biggest Data Science Breakthrough of the Decade

Deep Learning



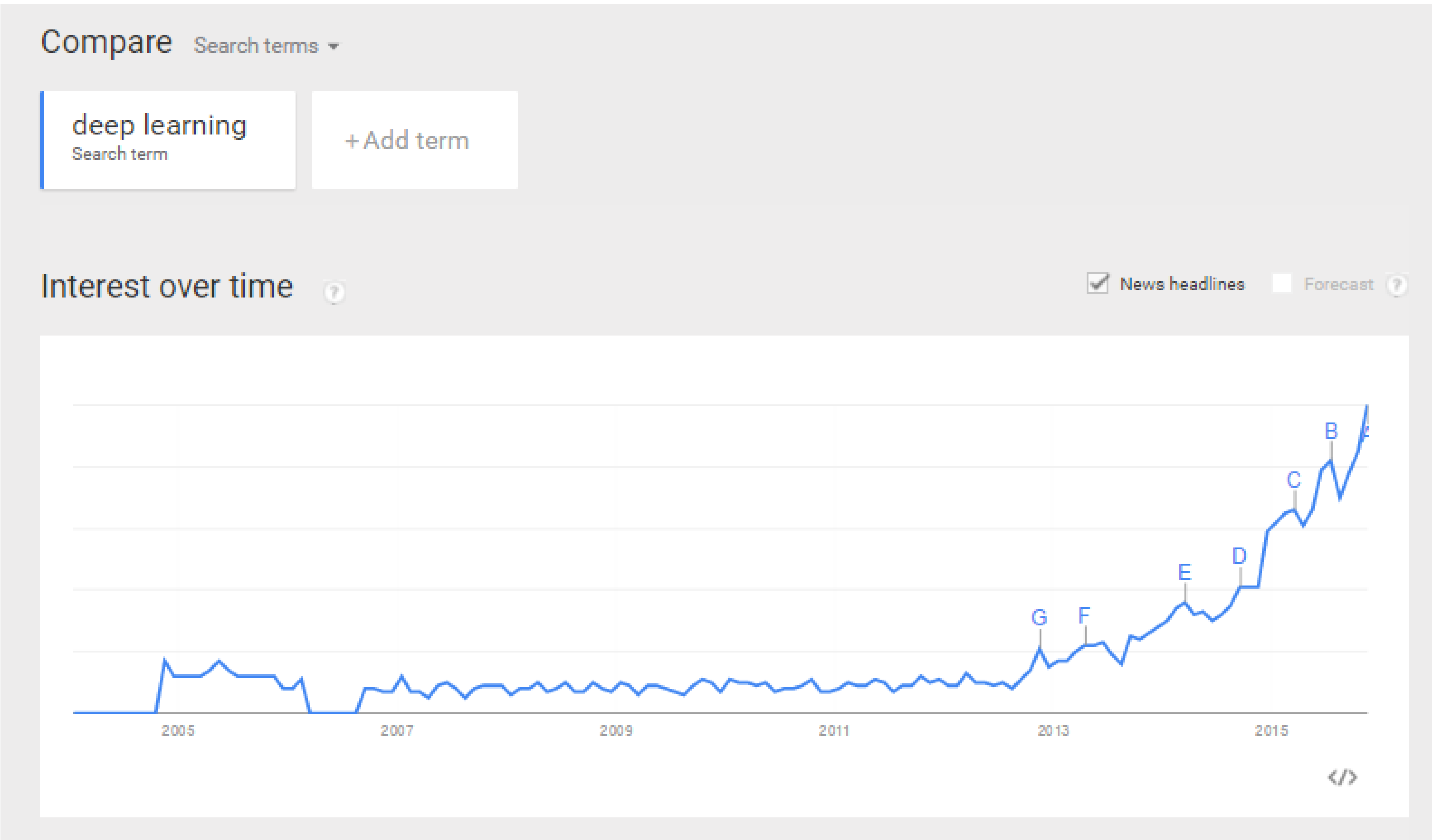
Why do you want to know about deep learning?

Motivation

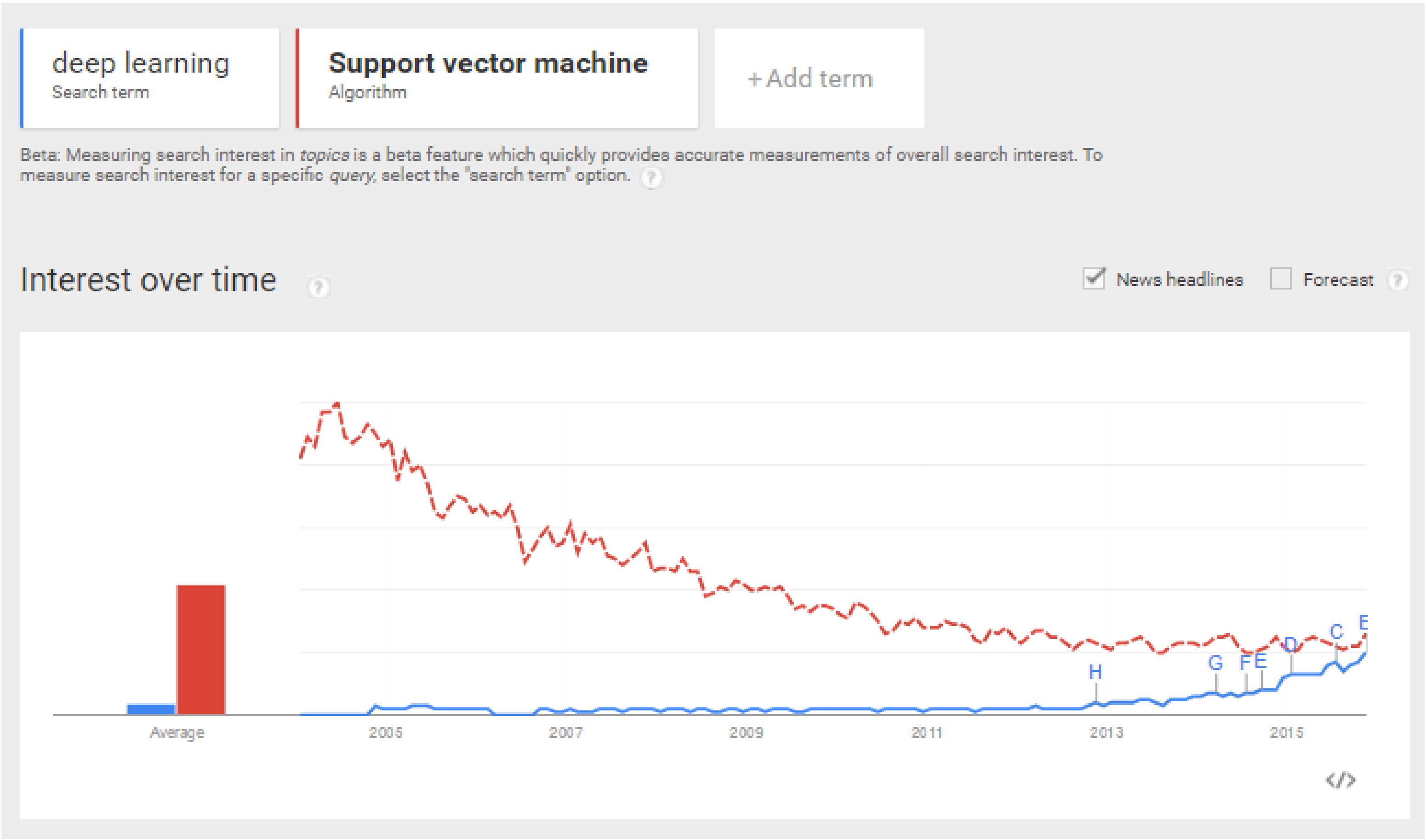
- It works!
- State of the art in machine learning
- Google, Facebook, Twitter, Microsoft are all using it.

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

Google Trends

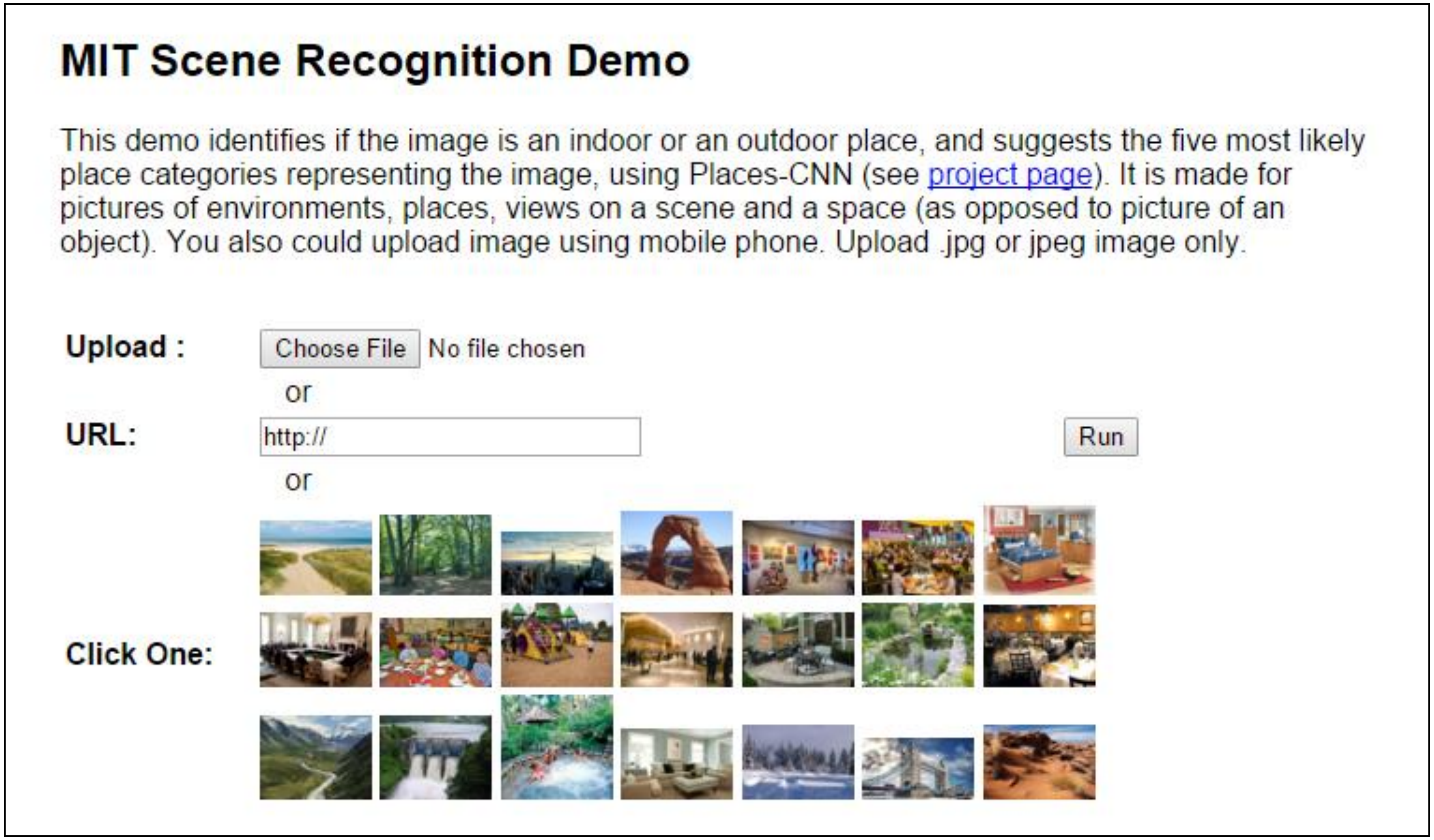


Google Trends



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

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

What is different?

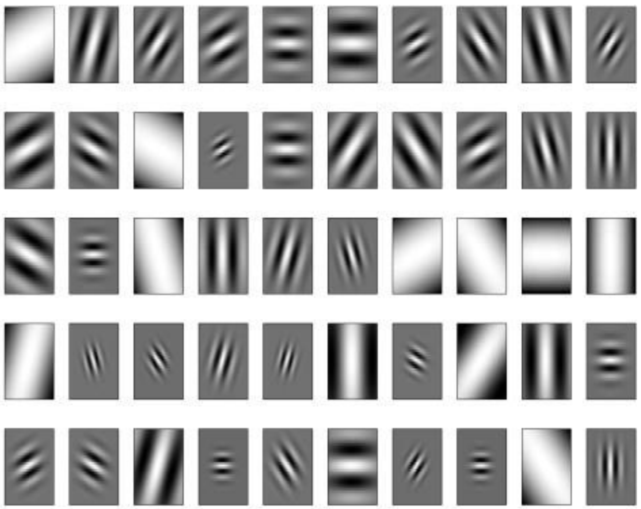
- We have seen ML methods:
  - SVM, decision trees, boosting, random forest

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

Feature Design

Yes No

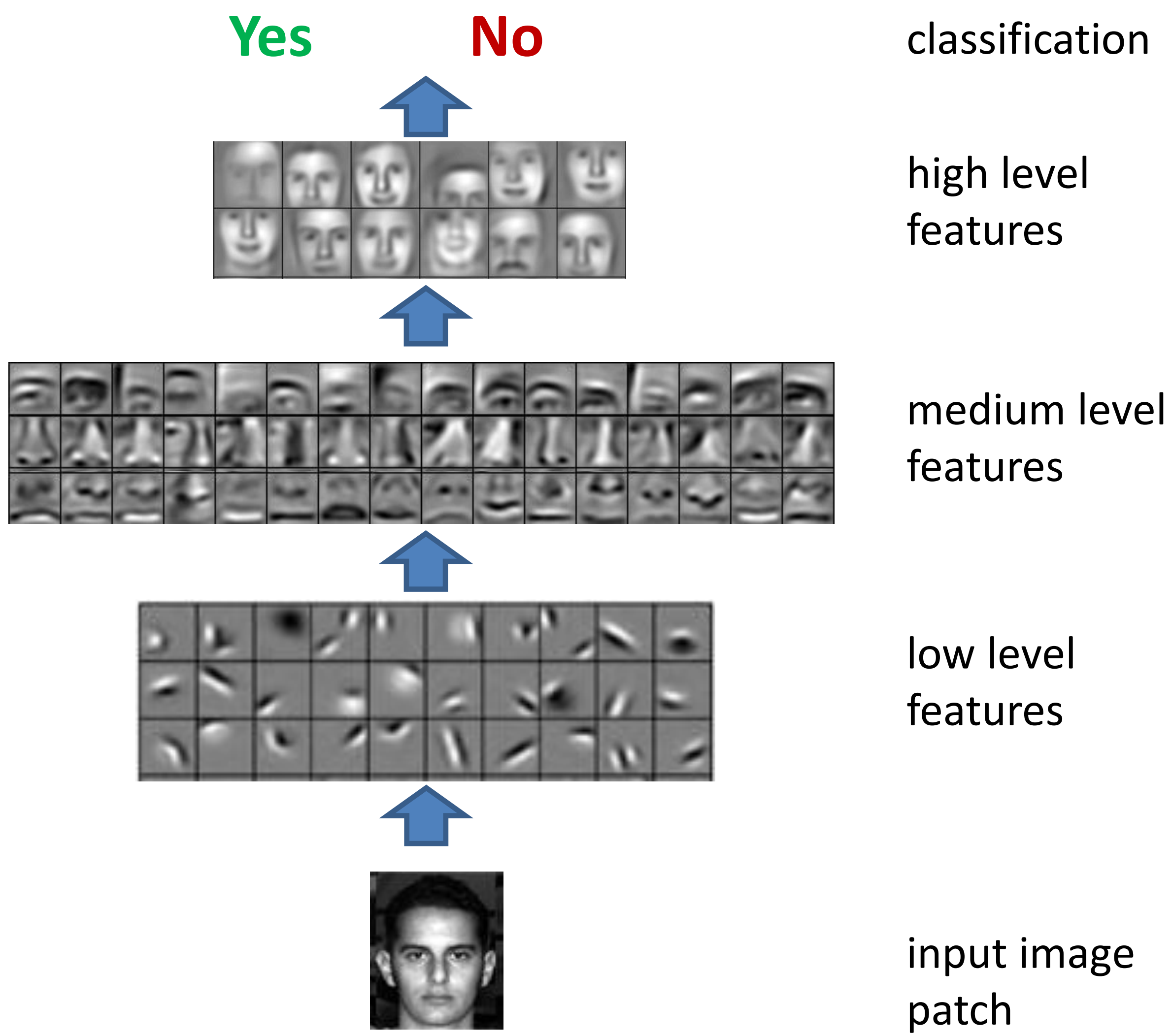
classification



hand designed features

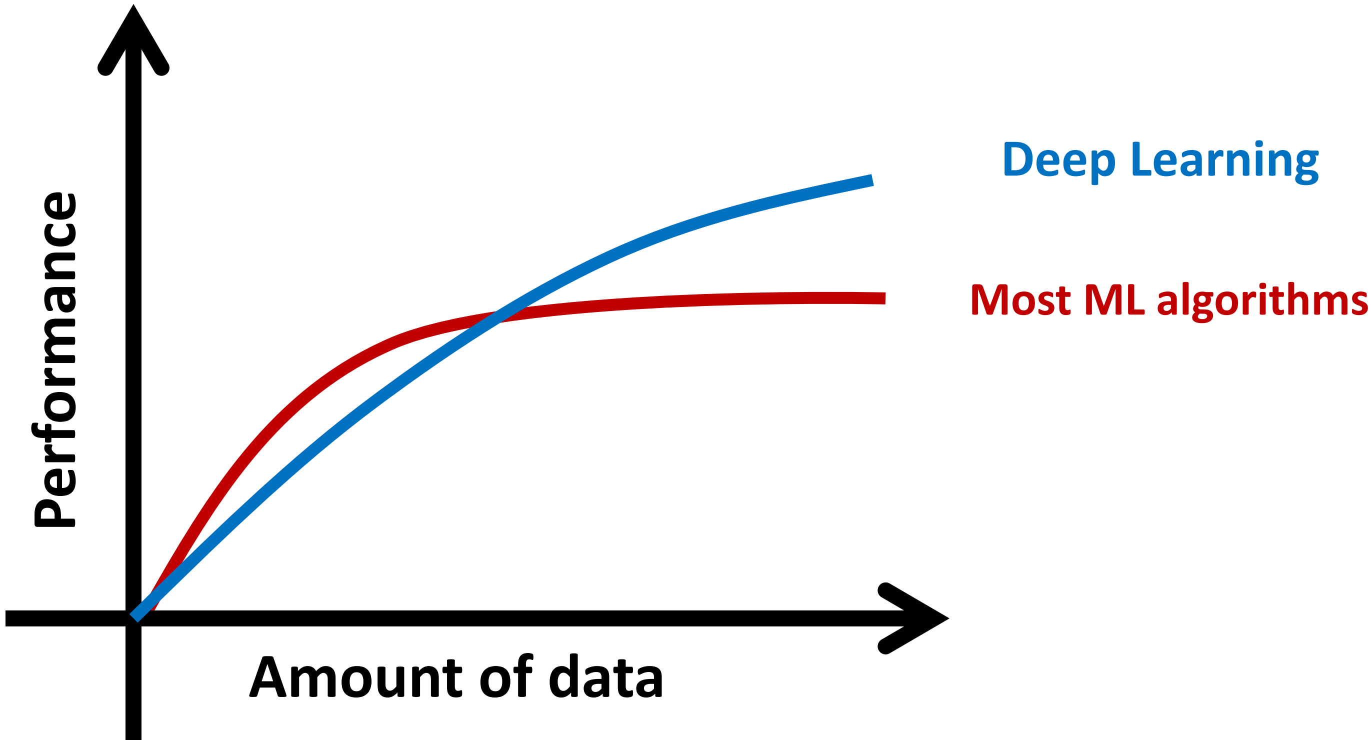
input image patch

Learned Feature Hierarchy



[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

What Changed since the 80s?

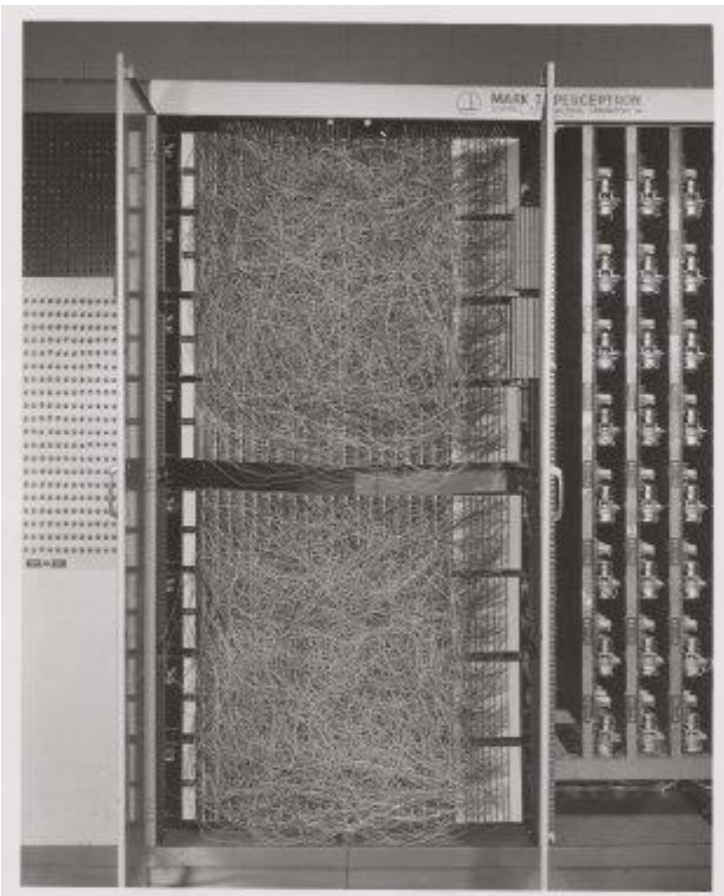
- MLPs and CNNs have been around since the 80s and earlier
- Why did people even bother with SVMs, boosting and co?
- And why do we still care about those methods?

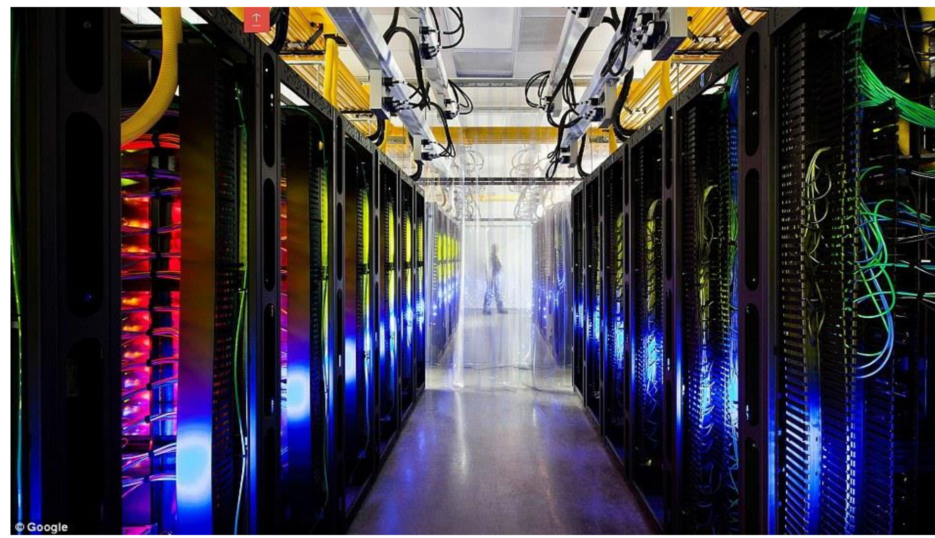
Brain or Rocket

Brain_or_Rocket_Ng.mp4

https://www.youtube.com/watch?v=EczYSl-ei9g

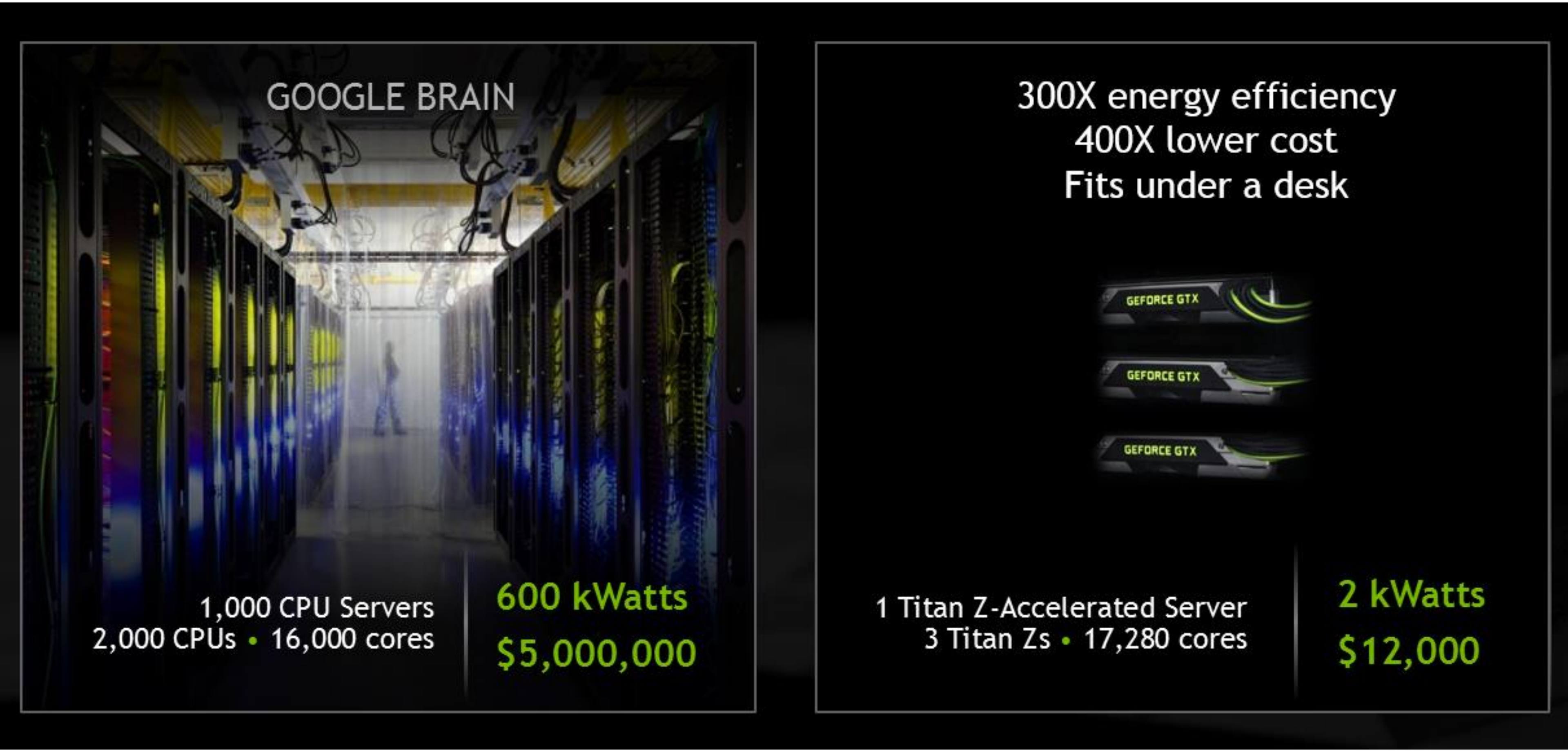
What Changed -Computational Power





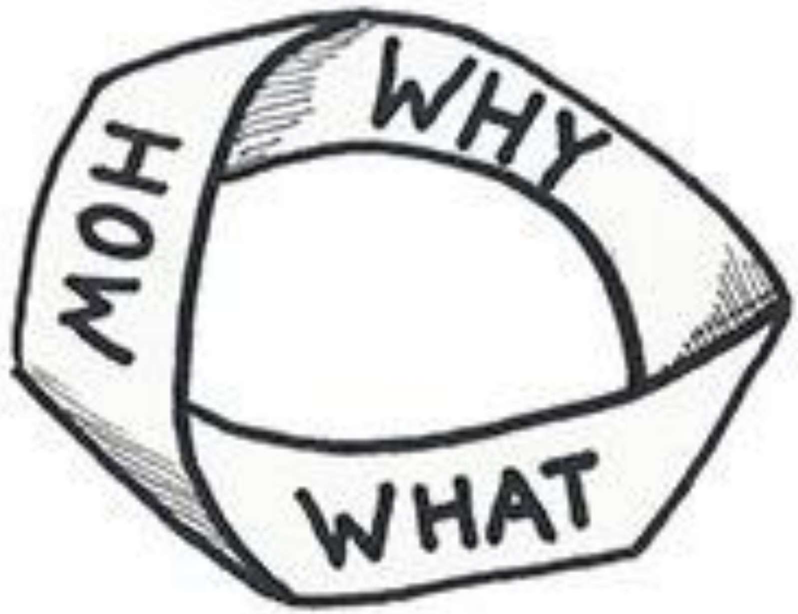
What Changed – Data Size

I don't Have a Cluster at Home



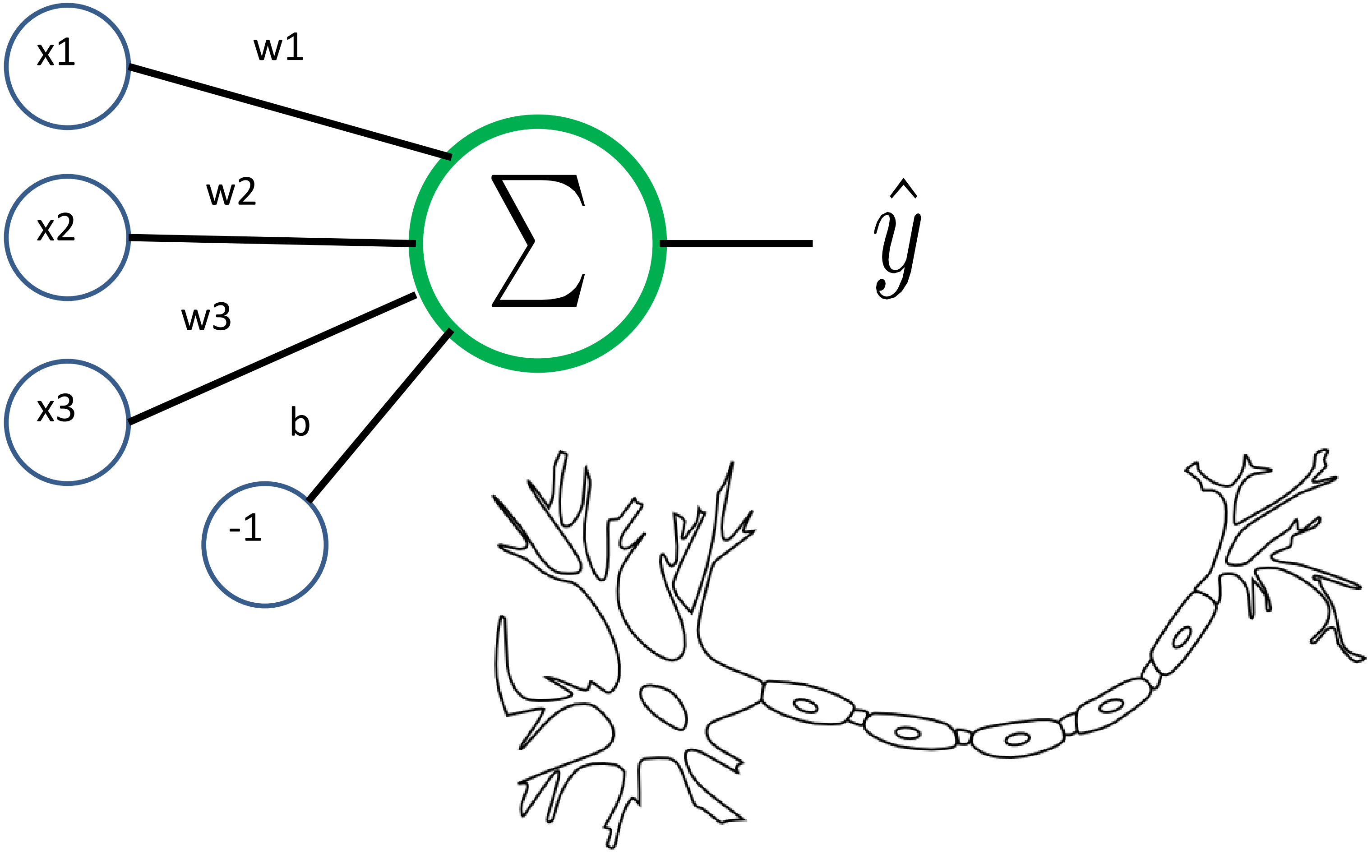
http://www.geforce.com/whats-new/articles/introducing-nvidia-geforce-gtx-titan-z

Deep Learning

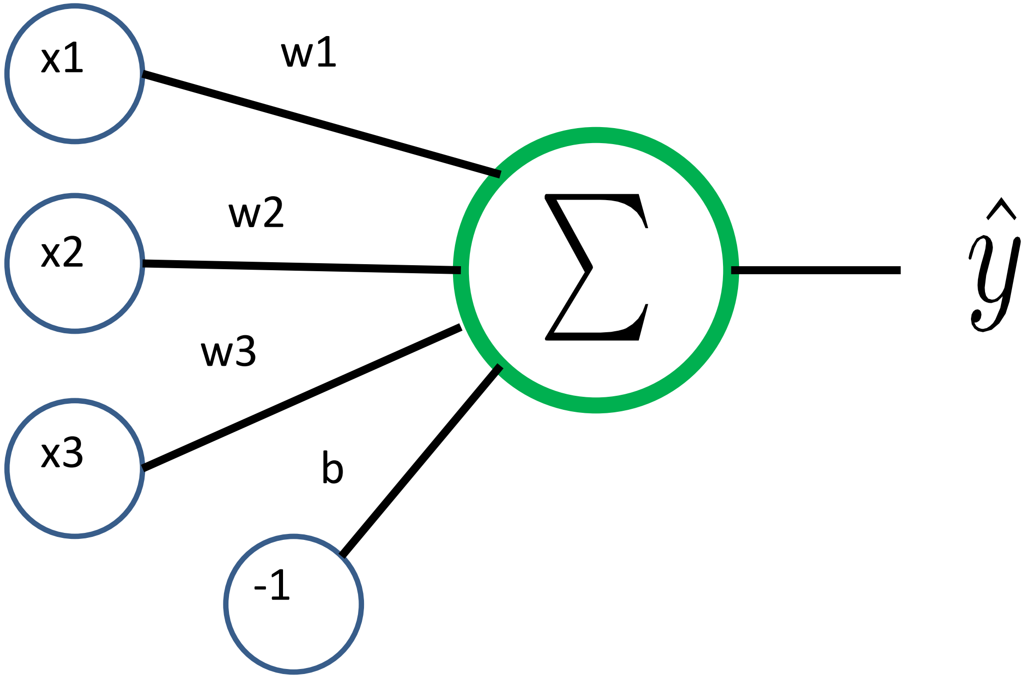


What is deep learning?

Perceptron



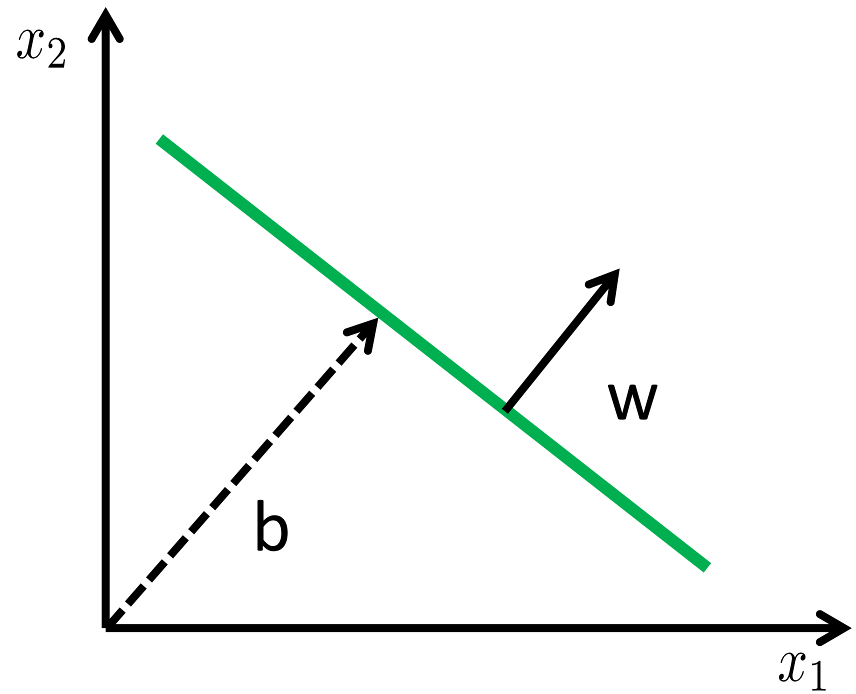
Perceptron



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

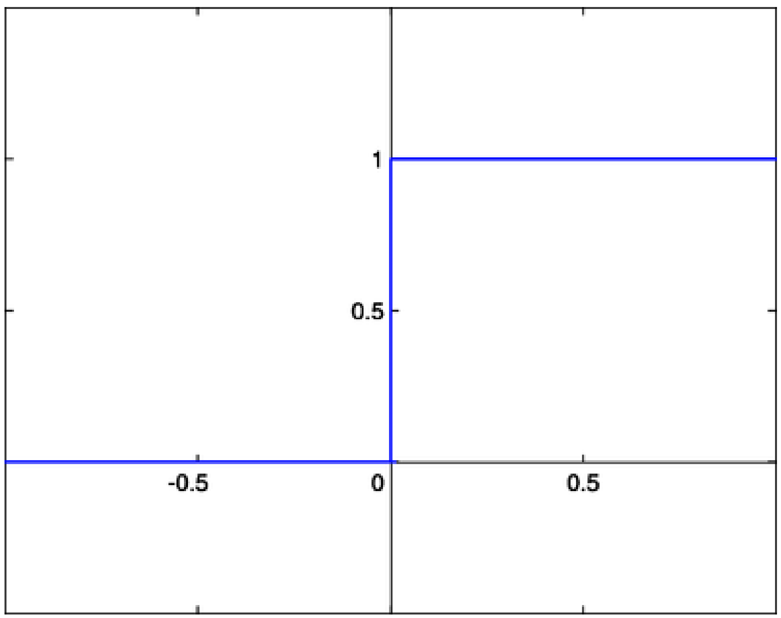
Separating Hyperplane

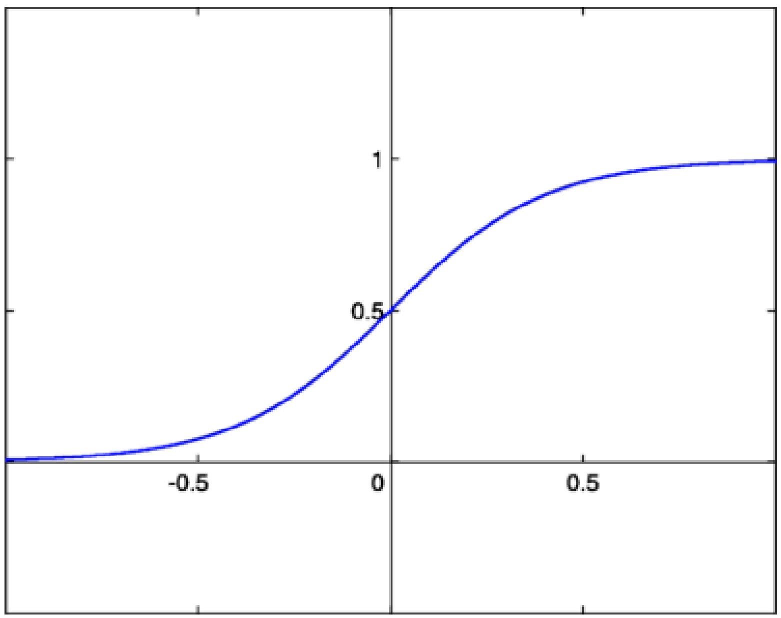
- x: data point
- y: label $\in \{-1, +1\}$
- w: weight vector
- b: bias



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

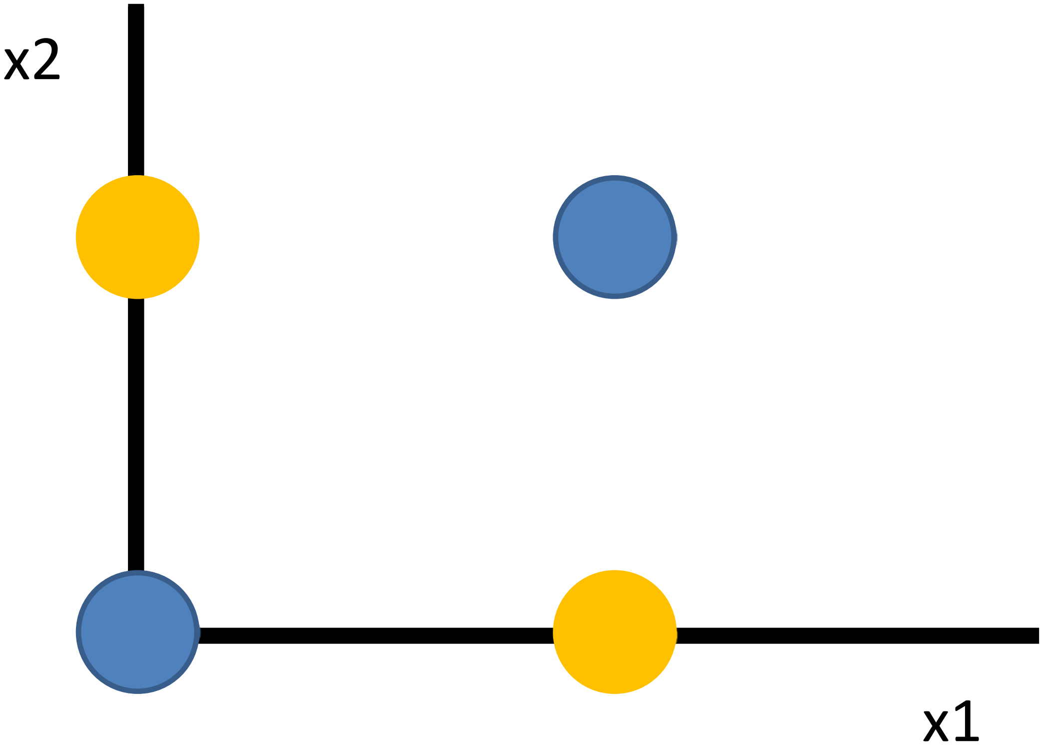
Side Note: Step vs Sigmoid Activation



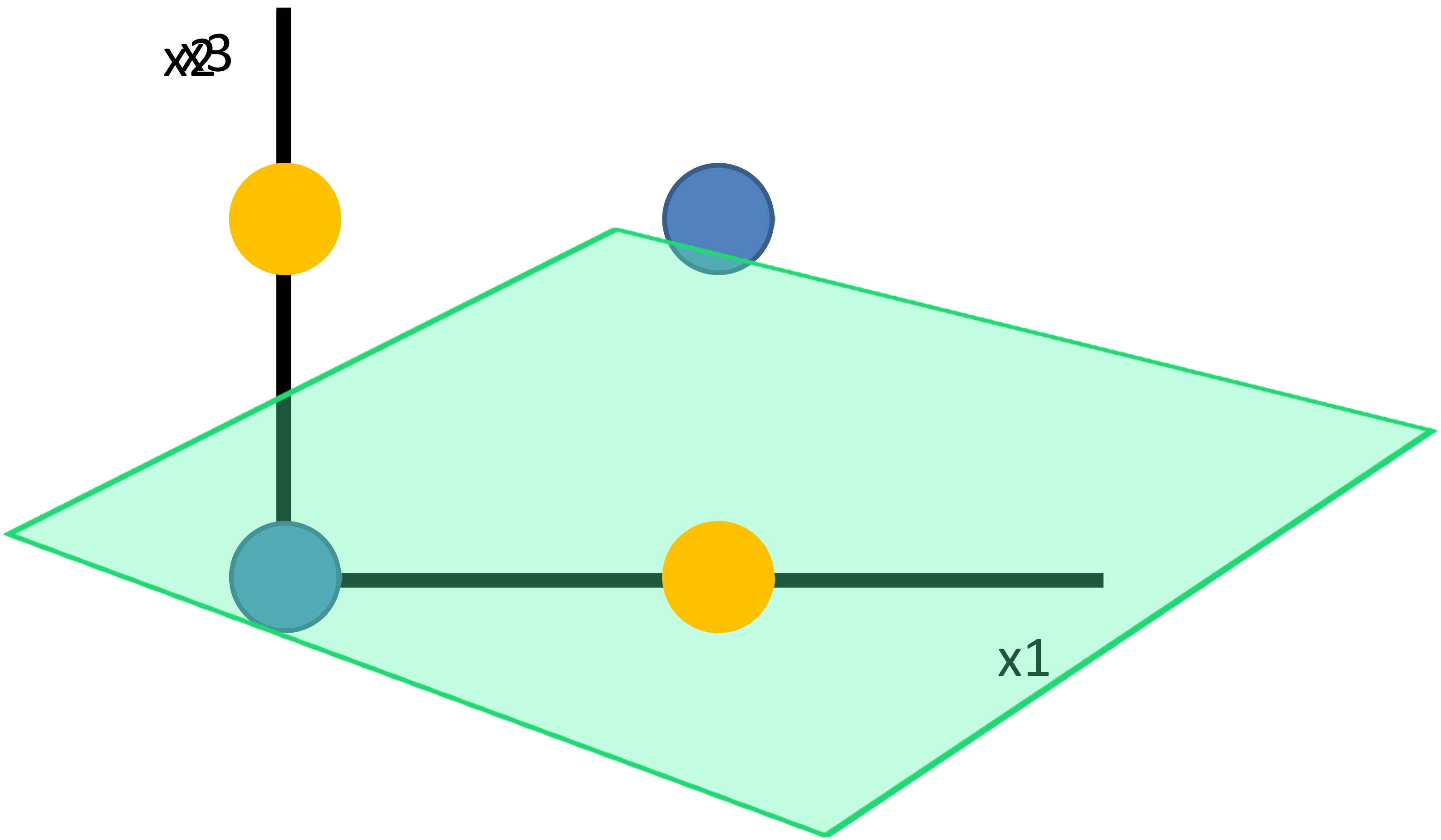


$$s(x) = \frac{1}{1 + e^{-cx}}$$

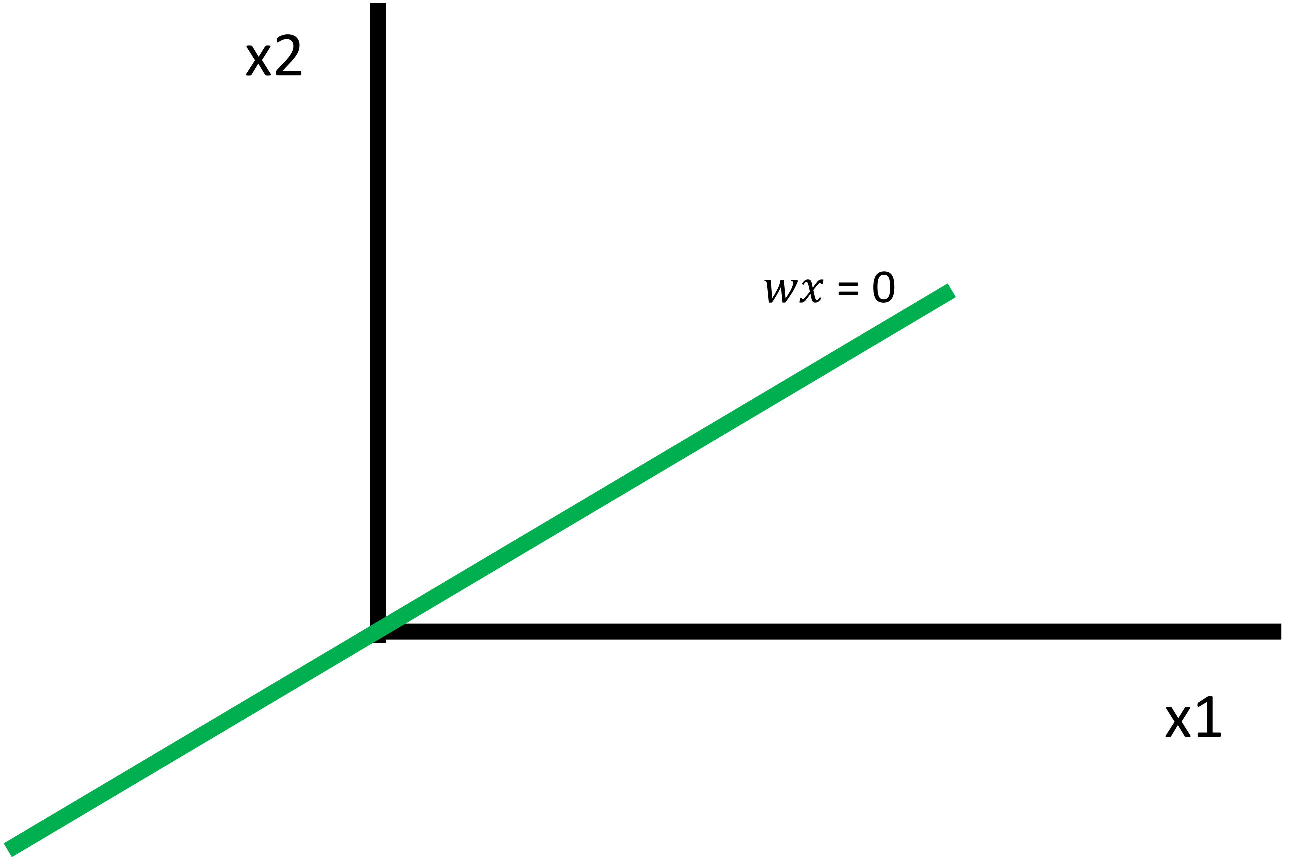
The XOR Problem



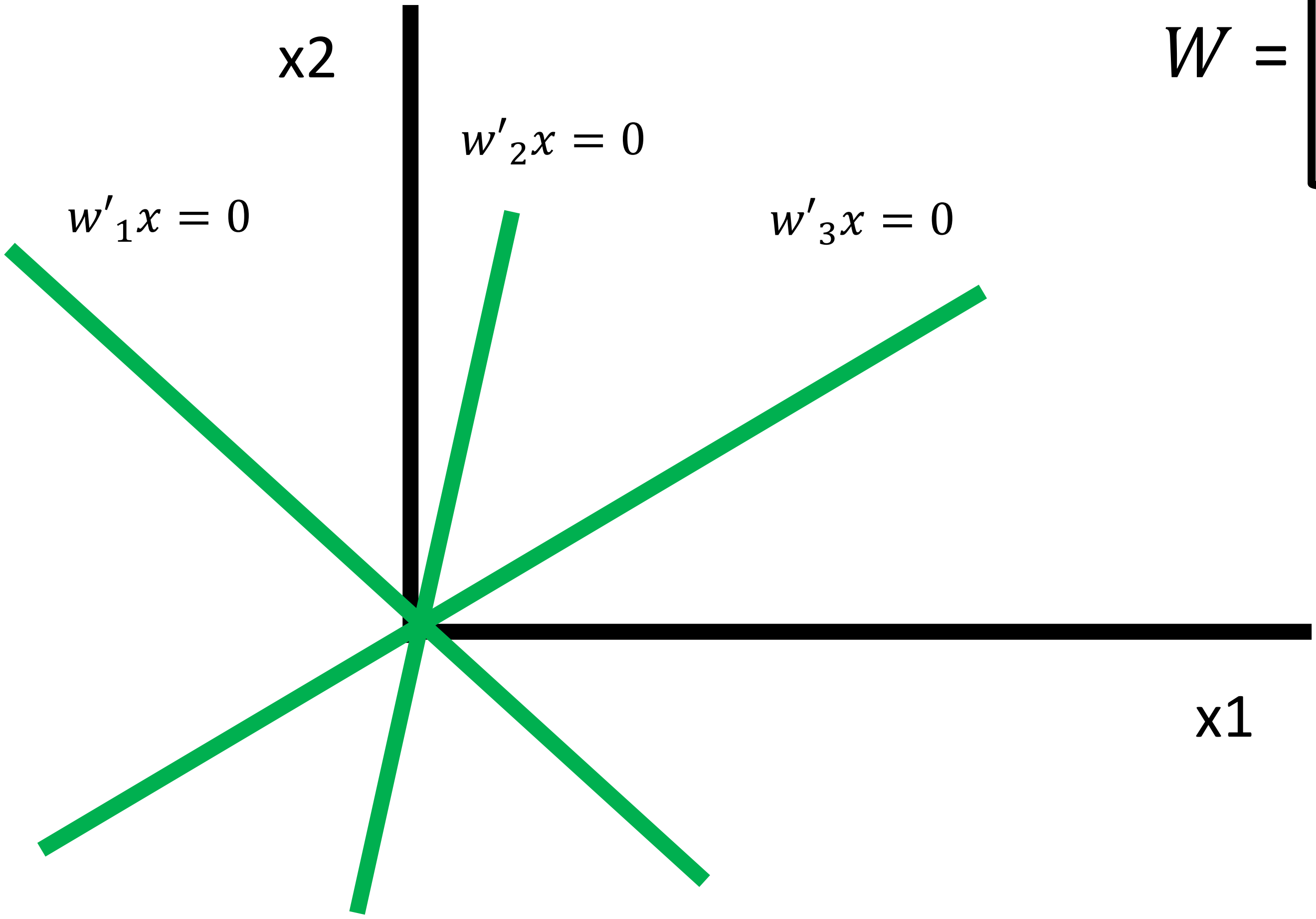
The XOR Problem



Perceptron



Multi-Perceptron

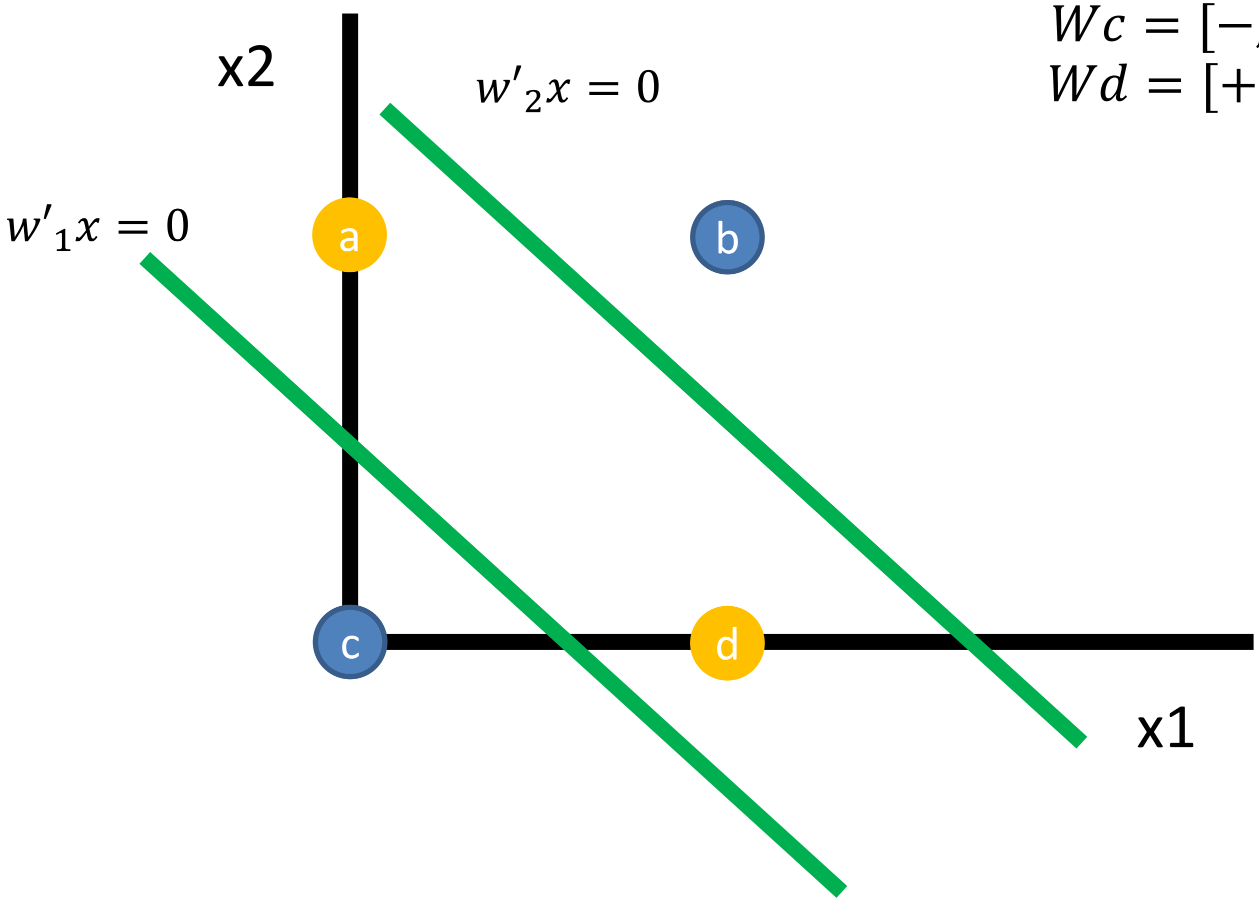


$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix}$$

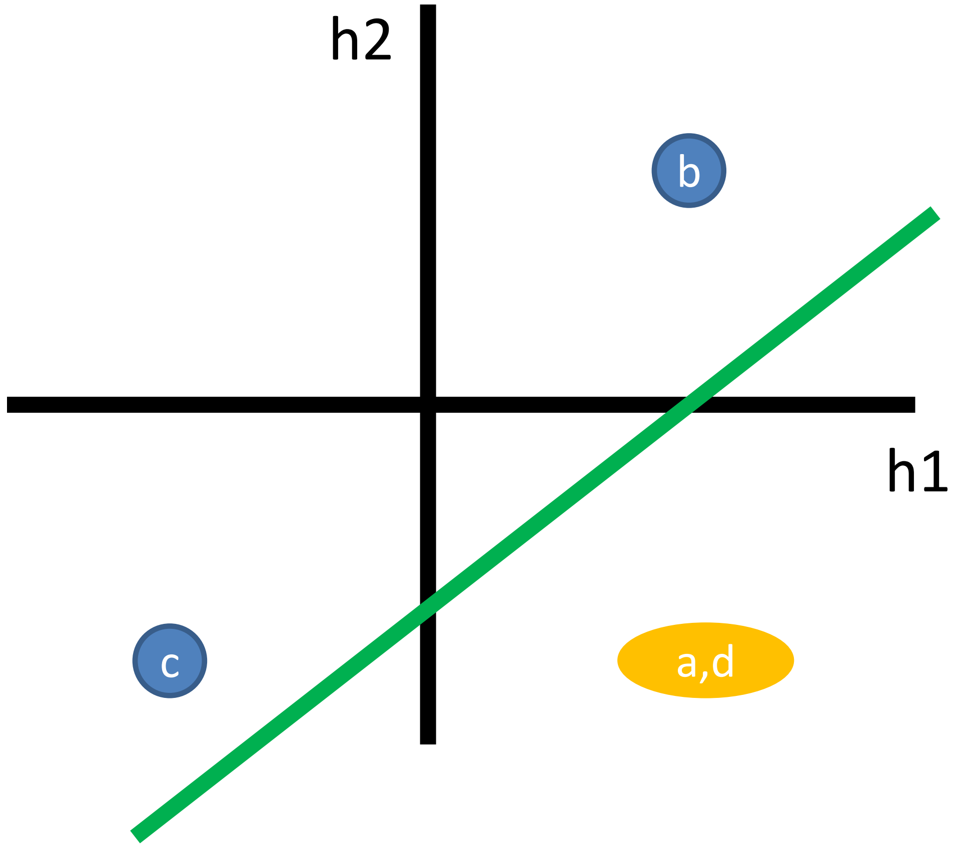
$$Wx = ?$$

Xor Problem

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

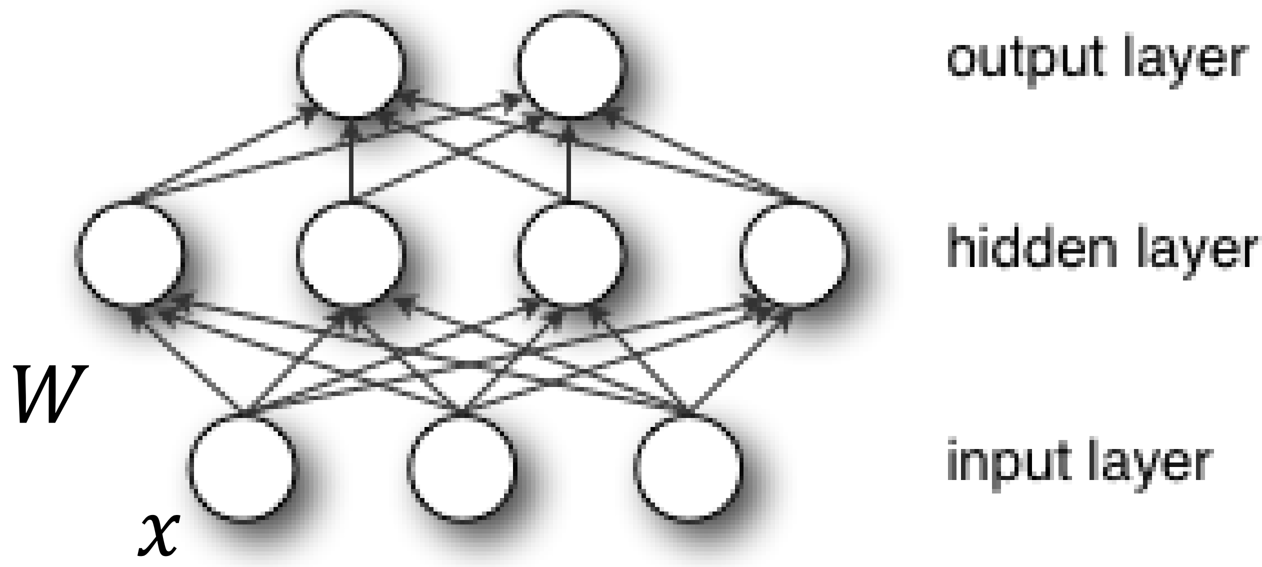


Xor Problem



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

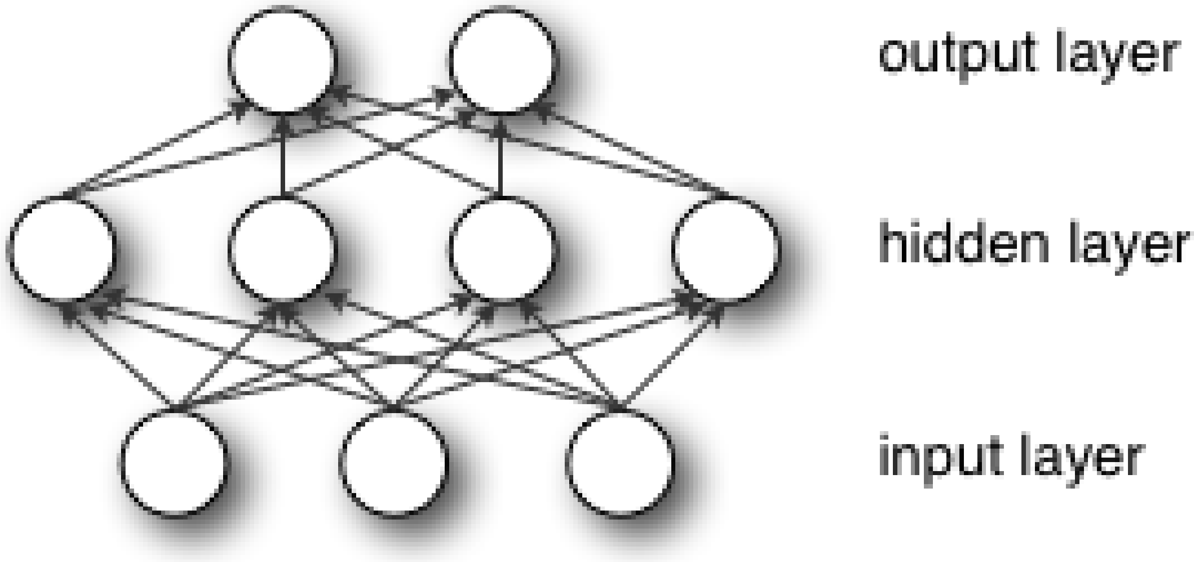
Multi-Layer Perceptron



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

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

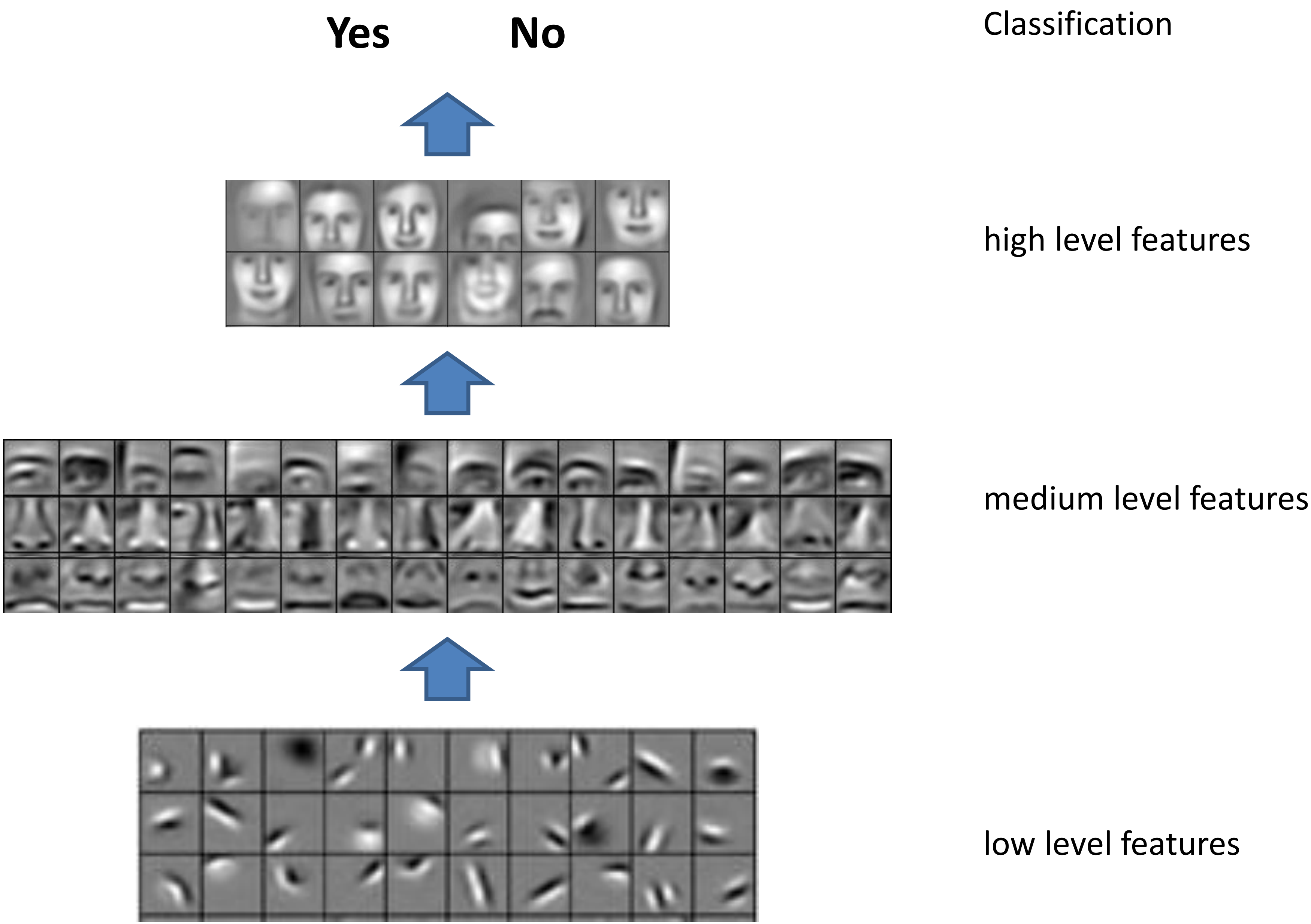
Multi-Layer Perceptron



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

G: logistic function, softmax for multiclass

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

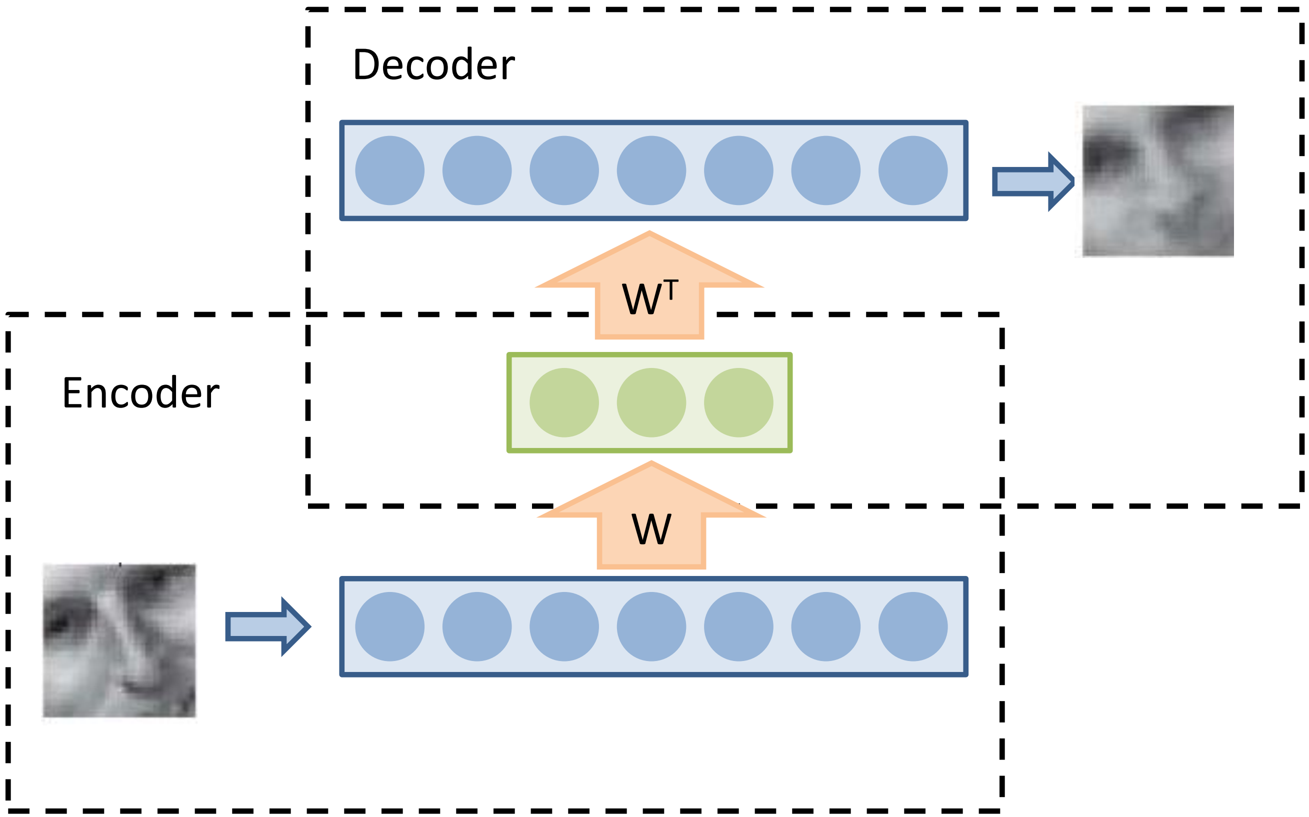

Yes
No
Classification
high level features
medium level features
low level features

Autoencoder

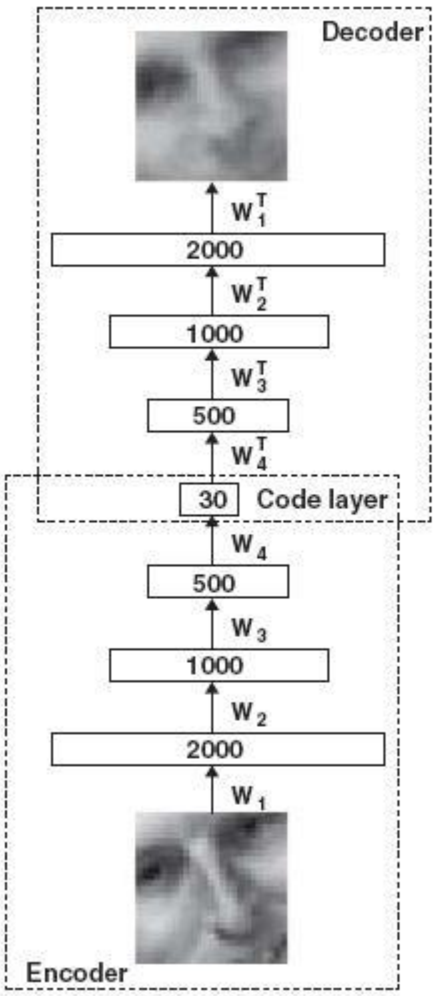
- This is what Google used for their Google brain
- Basically just a MLP
- Output size is equal to input size

- Popular for pre-training a network on unlabeled data

Autoencoder



Deep Autoencoder



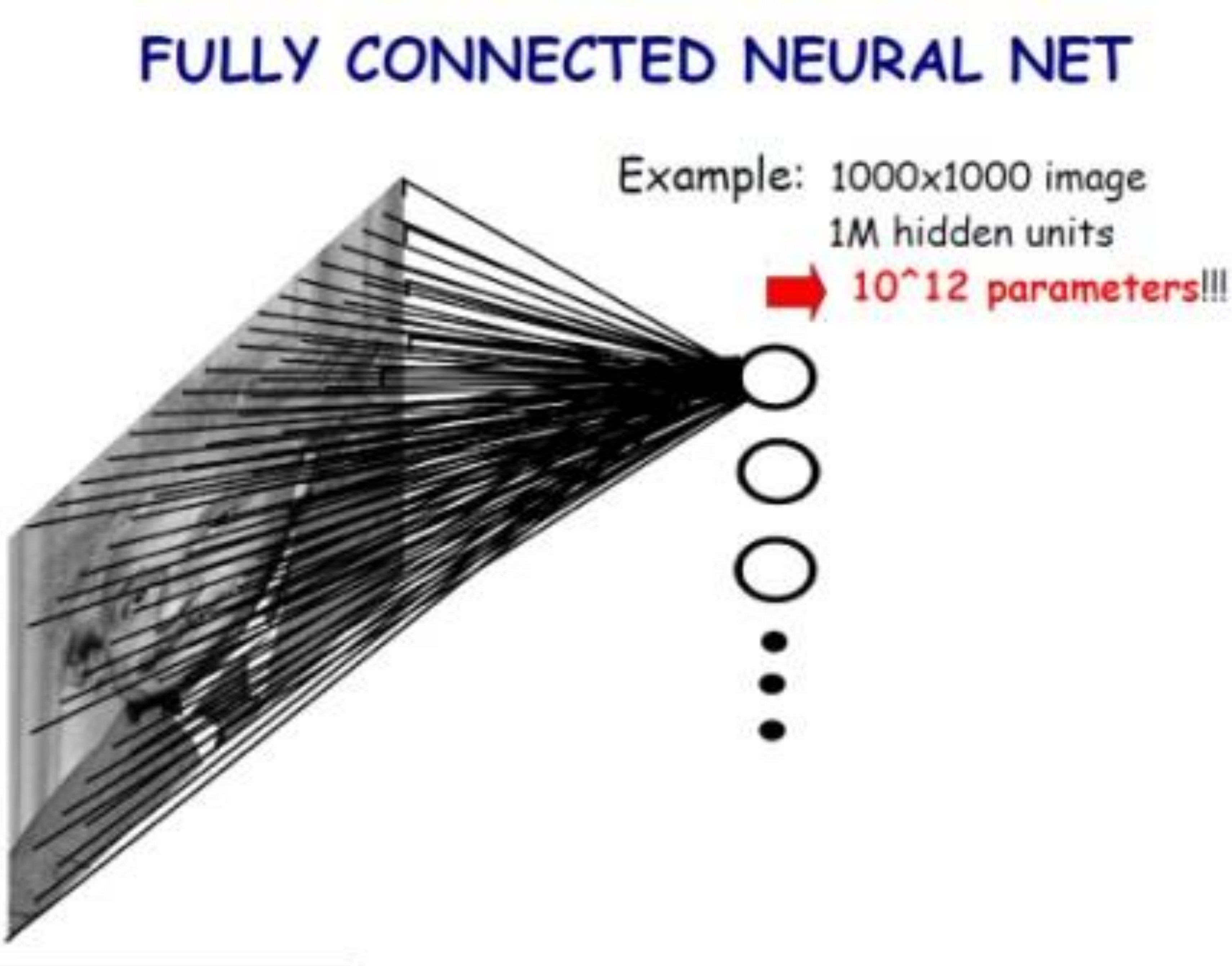
- Reconstruct image from learned low dimensional code
- Weights are tied
- Learned features are often useful for classification
- Can add noise to input image to prevent overfitting

Salakhutdinov & Hinton, NIPS 2007

From MLP to CNN

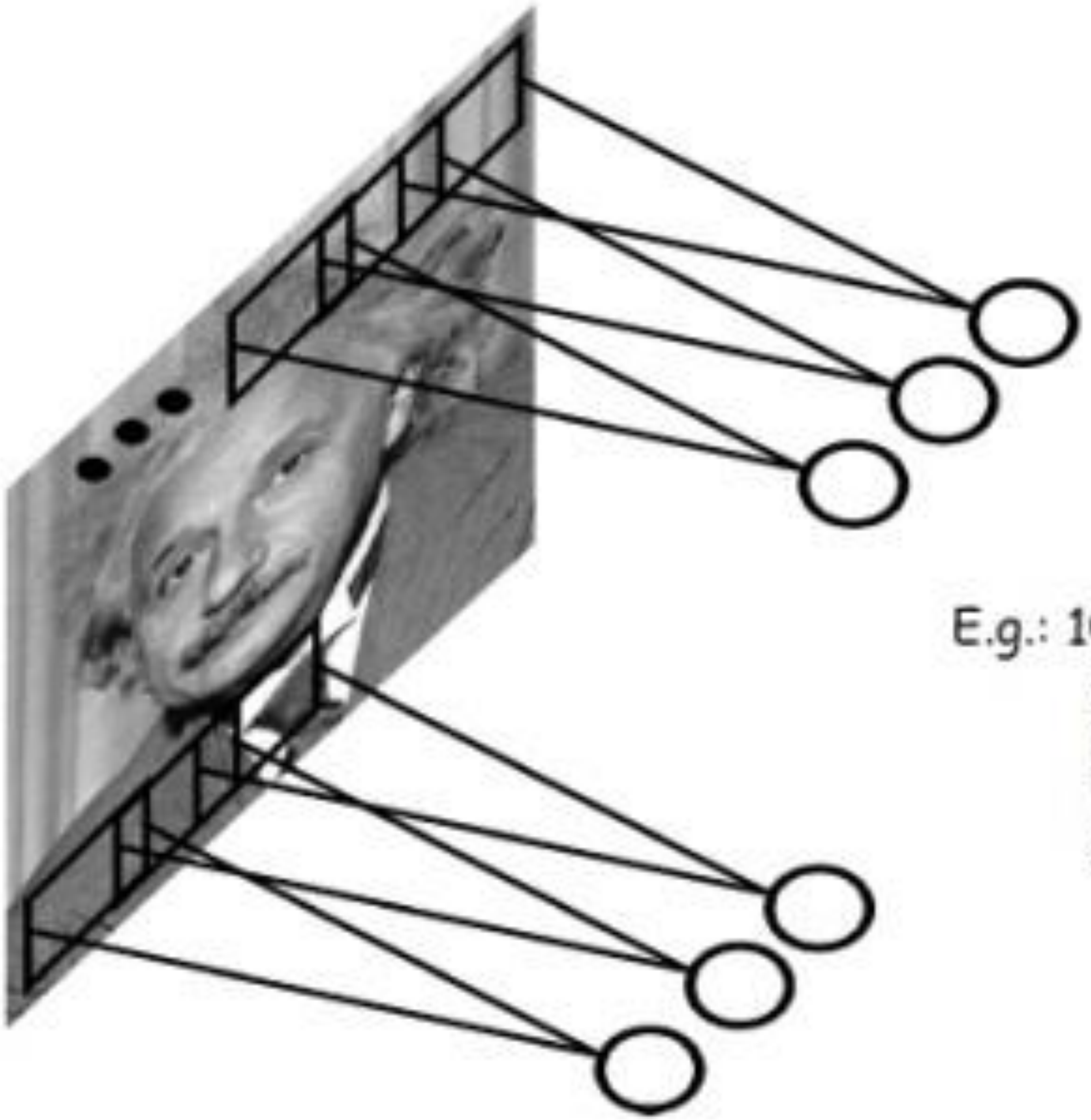
- So far no notion of neighborhood
- Invariant to permutation of input
- A lot of data is structured:
  - Images
  - Speech
  - …

- Convolutional neural networks preserve neighborhood



http://www.amolgmanurkar.com/classifySTLusingCNN.html

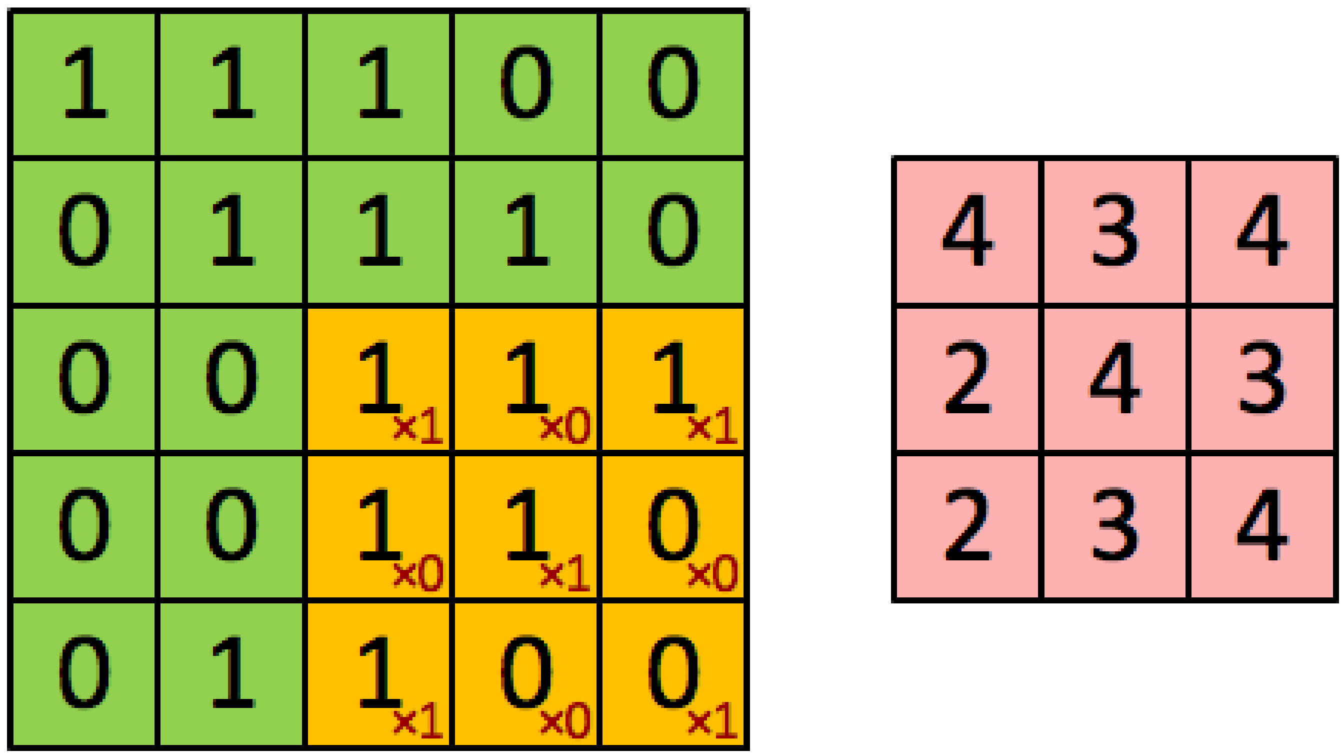
CONVOLUTIONAL NET



E.g.: 1000x1000 image
100 Filters
Filter size: 10x10
10K parameters

http://www.amolgmahurkar.com/classifySTLusingCNN.html

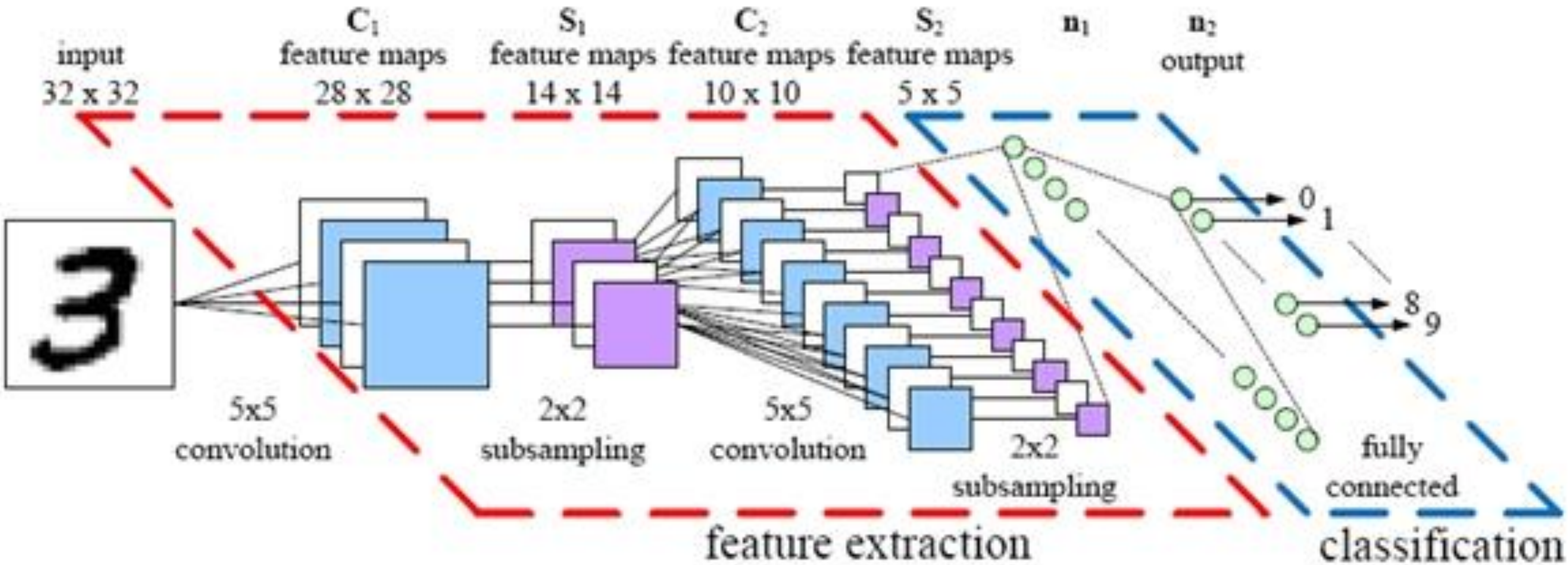
Convolution



Image

Convolved Feature

Convolutional Network



http://parse.ele.tue.nl/cluster/2/CNNArchitecture.jpg

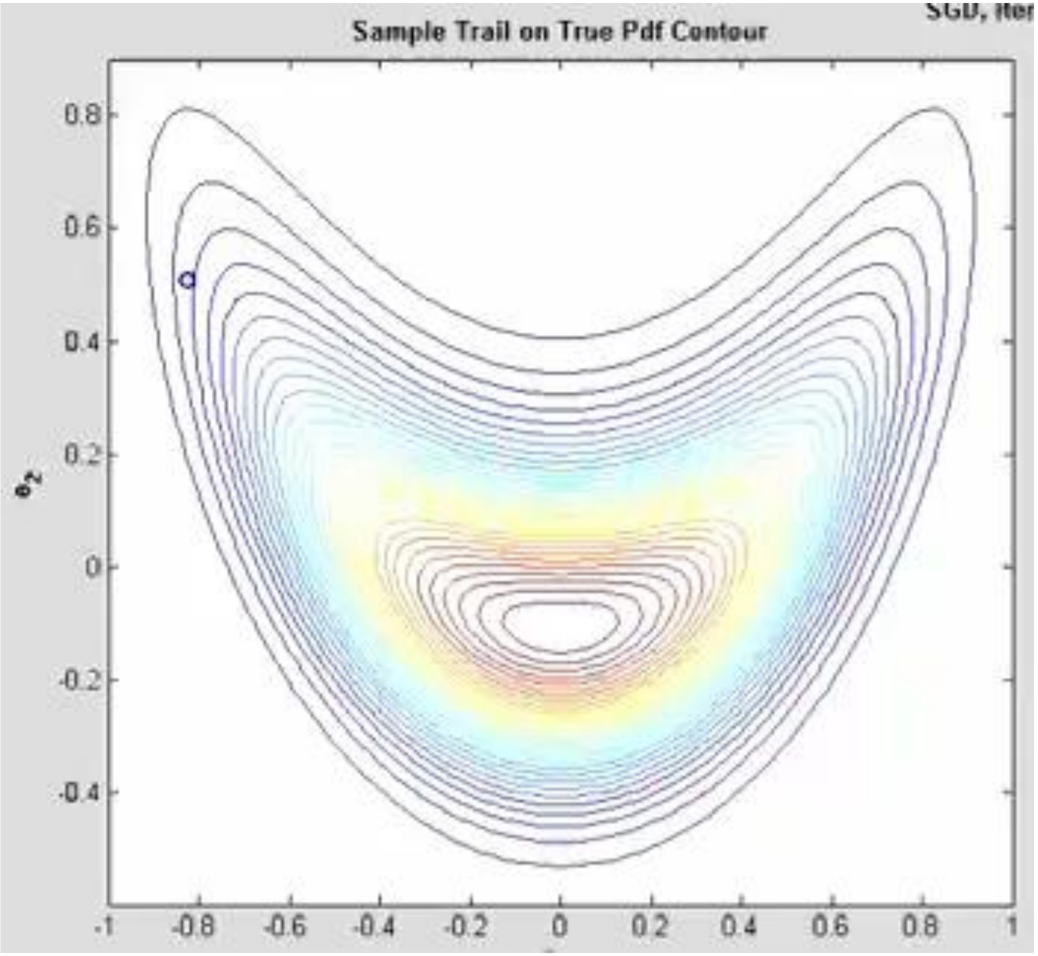
CNN Advantages

- neighborhood preserved
- translation invariant
- tied weights

DNNs are hard to train

- backpropagation – gradient descent
- many local minima
- prone to overfitting
- many parameters to tune
- SLOW

Stochastic Gradient Decent



https://www.youtube.com/watch?v=HvLJUsEc6dw

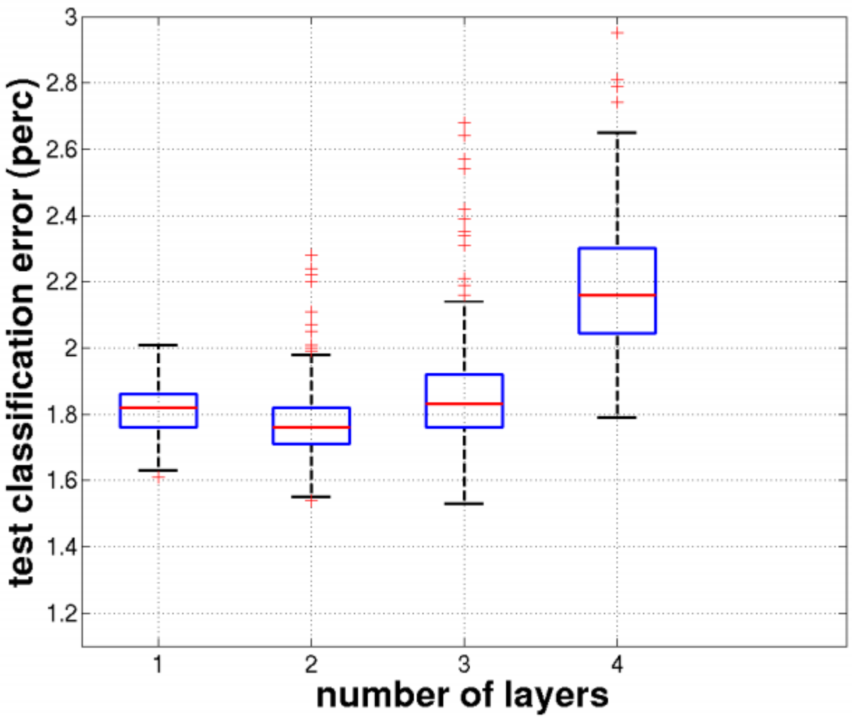
Development

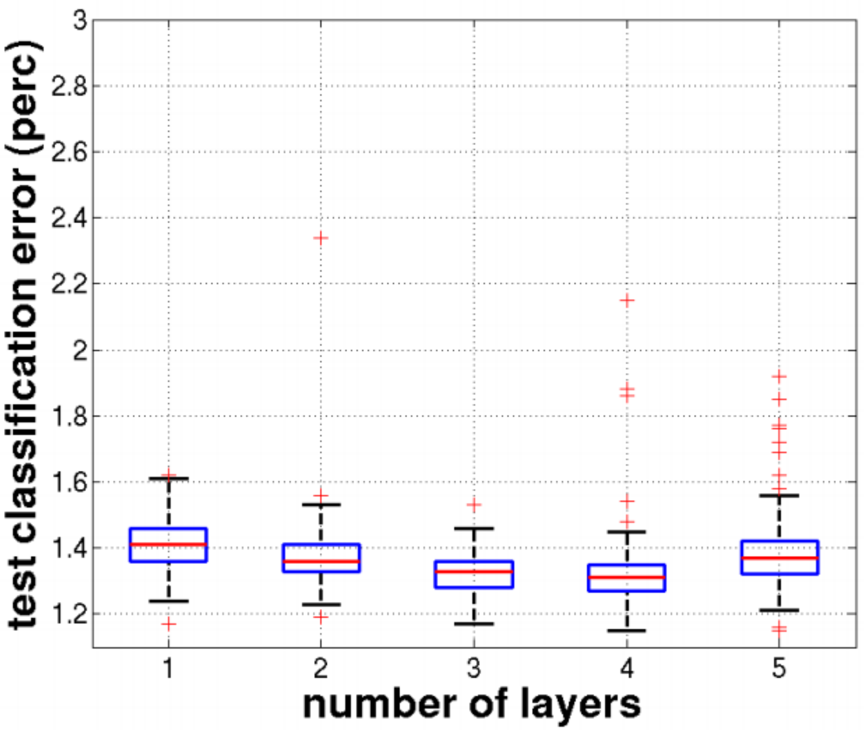
- Computers got faster!
- Data got bigger.
- Initialization got better.

2006 Breakthrough

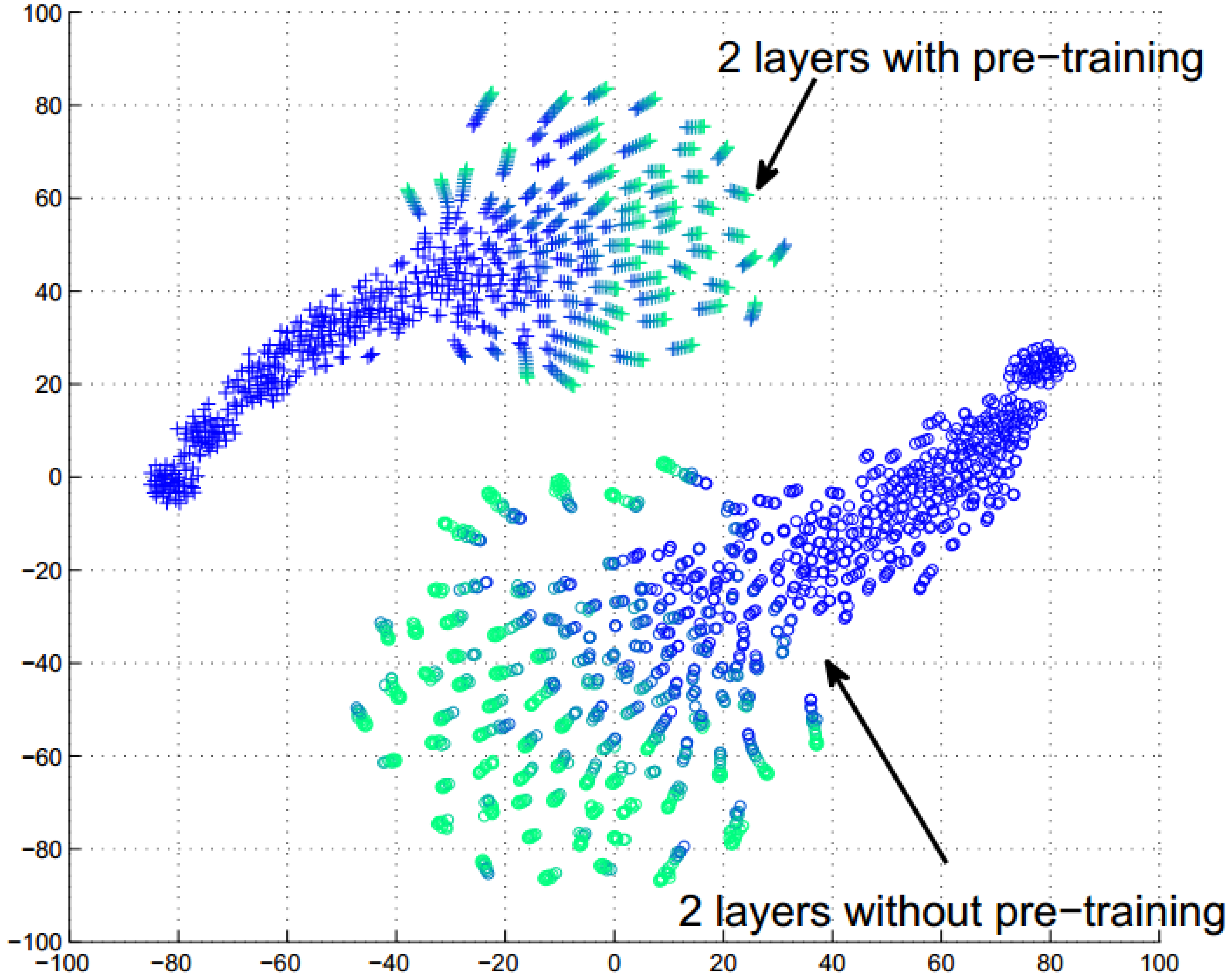
- Ability to train deep architectures by using layer-wise unsupervised learning, whereas previous purely supervised attempts had failed
- Unsupervised feature learners:
  - RBMs
  - Auto-encoder variants
  - Sparse coding variants

Unsupervised Pretraining





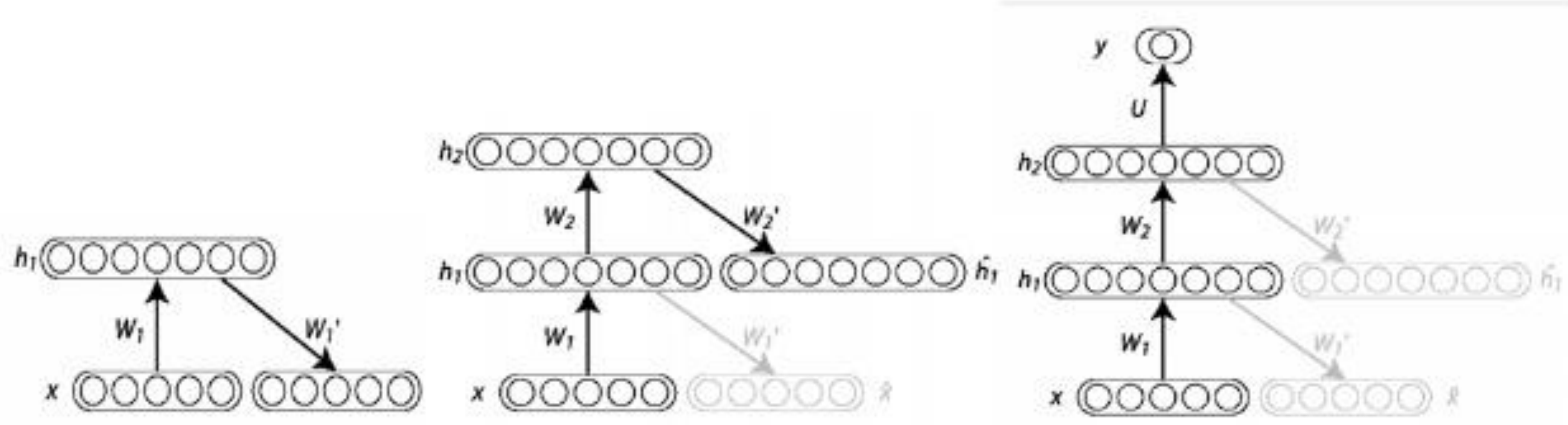
http://jmlr.org/papers/volume11/erhan10a/erhan10a.pdf



http://jmlr.org/papers/volume11/erhan10a/erhan10a.pdf

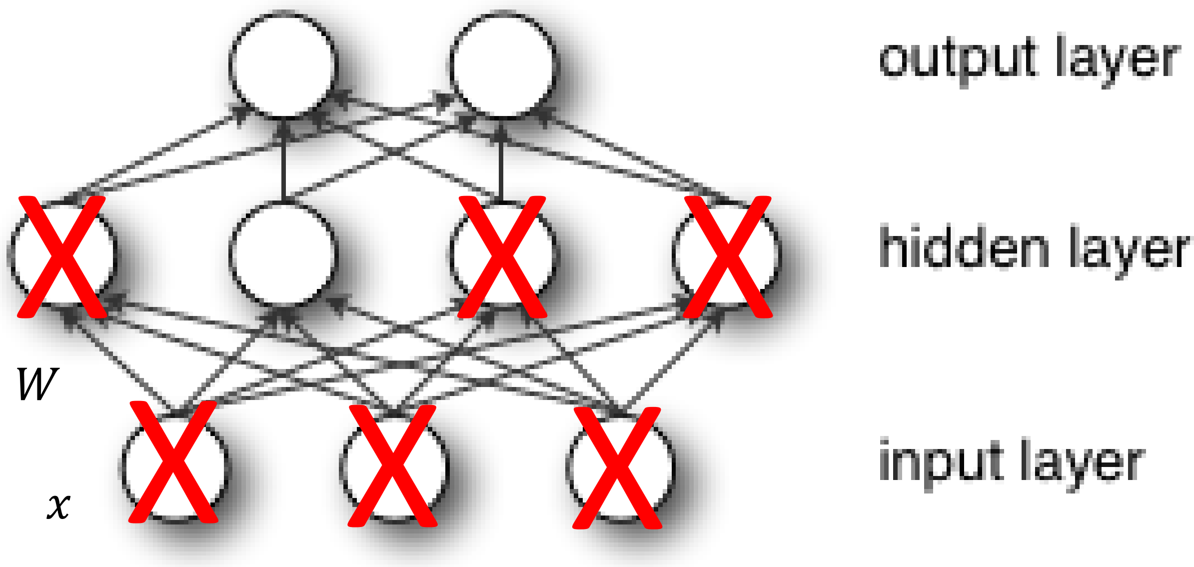
Pretraining: Stacked Denoising Auto-encoder

- Stacking Auto-Encoders



from: Bengio ICML 2009

Dropout



- Helps with overfitting
- Typically used with random initialization
- Training is slower than without dropout

Deep Learning for Sequences

- MLPs and CNNs have fixed input size
- How would you handle sequences?

- Example: Complete a sentence
  - …
  - are …
  - How are …

Training a recurrent net to predict the next character

- Ilya Sutskever used 5 million strings of 100 characters each, taken from Wikipedia. For each string he starts predicting at the 11th character.
- It takes a month on a GPU board to get a really good model. It needs very big mini-batches.
- Ilya's best model is about equal to the state of the art for character prediction, but works in a very different way from the best other models.
  - It can balance quotes and brackets over long distances.

Slide from G. Hinton

Meaning of Life

meaning_of_life.mp4

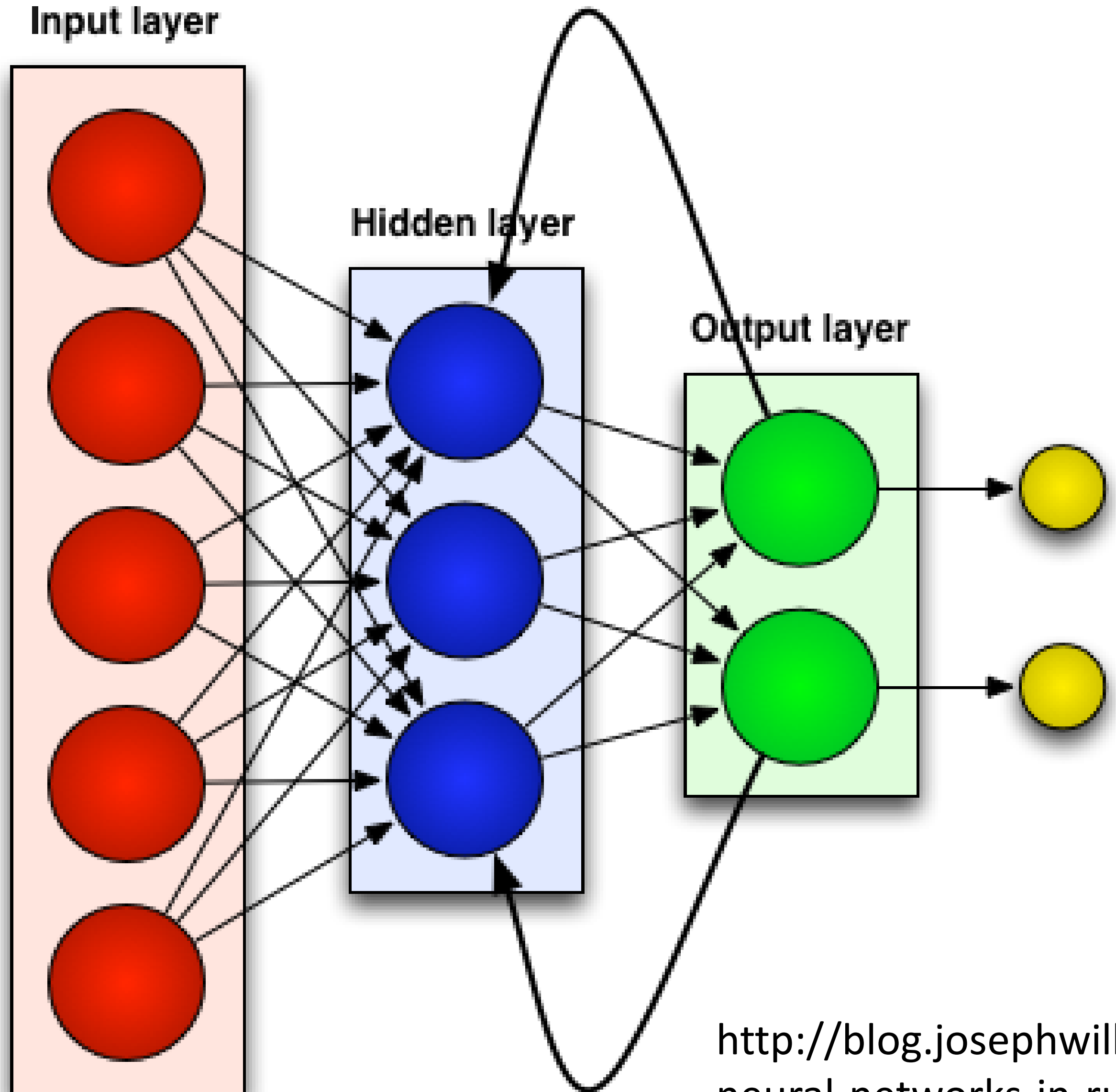
https://www.youtube.com/watch?v=vShMxxqtDDs

Completing a sentence using the neural network

The meaning of life is the tradition of the ancient human reproduction: it is less favorable to the good boy for when to remove her bigger.

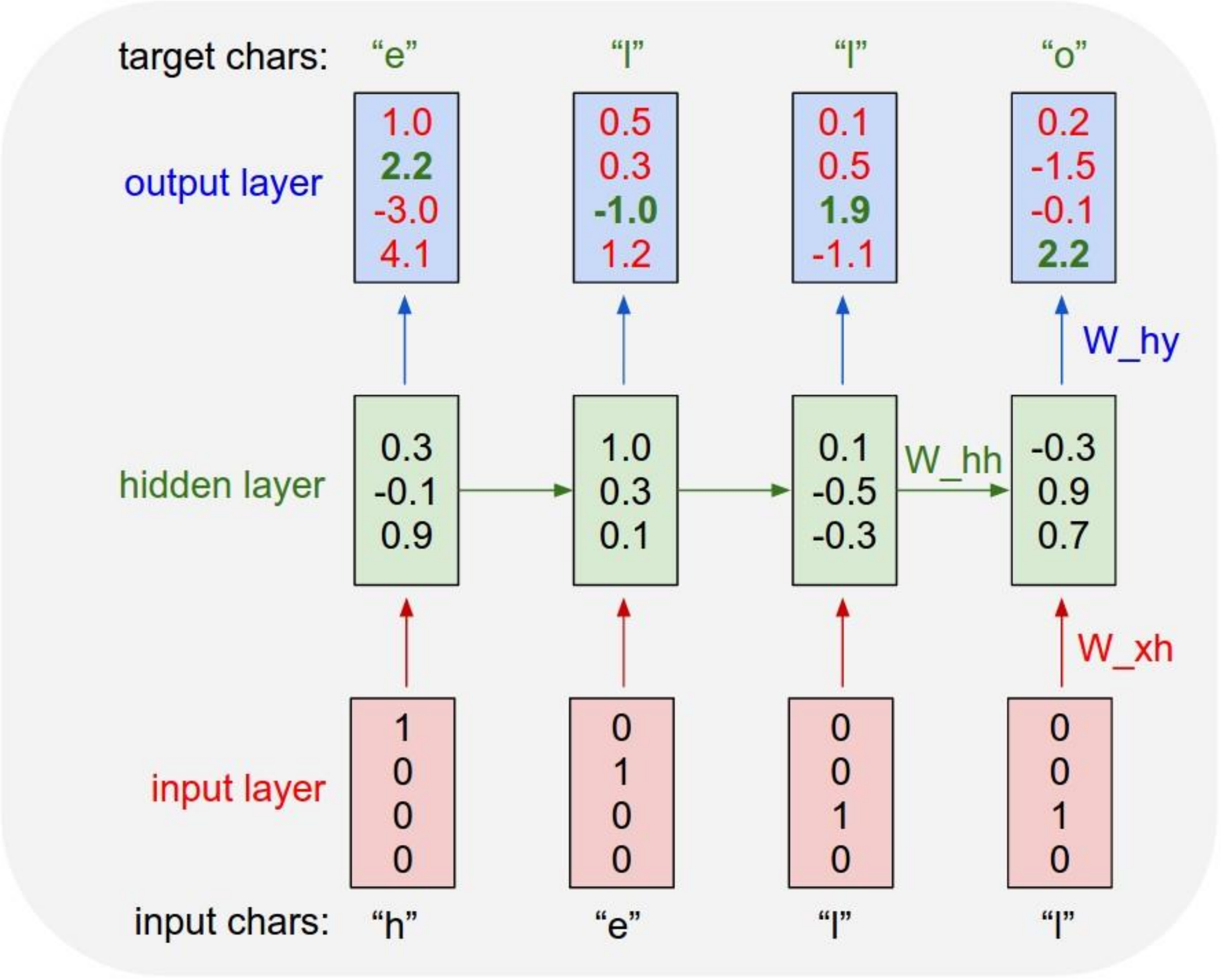
Slide from G. Hinton

Recurrent Neural Network



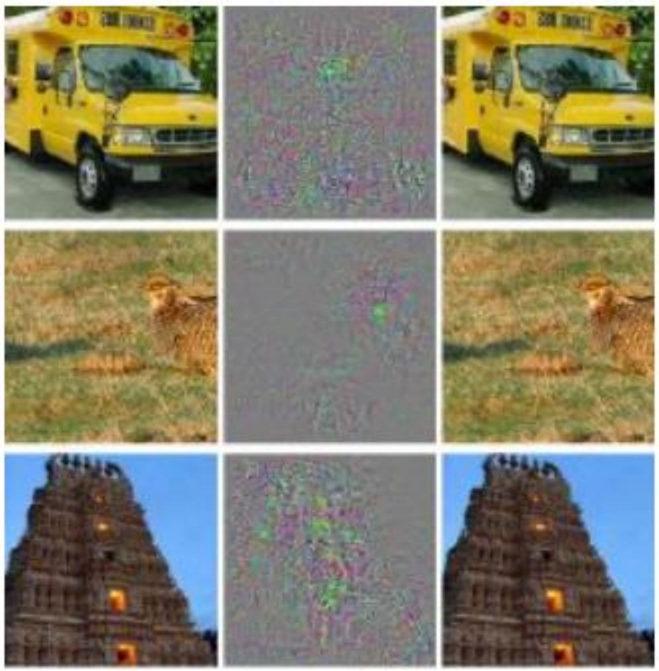
http://blog.josephwilk.net/ruby/recurrent-neural-networks-in-ruby.html

Recurrent Neural Network



http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Intriguing properties of neural networks



Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan,Ian Goodfellow, Rob Fergus
International Conference on Learning Representations (2014)

http://www.datascienceassn.org/sites/default/files/Intriguing%20Properties%20of%20Neural%20Networks_0.pdf

Libraries

- Theano
- Torch
- Caffe

- TensorFlow
- …

Theano

- Full disclosure: My favorite
- Python
- Transparent GPU integration
- Symbolical Graphs
- Auto-gradient
- Low level – in a good way!
- If you want high-level on top:
  - Pylearn2
  - Keras, Lasagne, Blocks
  - …

Torch

- Lua (and no Python interface)
- Very fast convolutions
- Used by Google Deep Mind, Facebook AI, IBM
- Layer instead of graph based

https://en.wikipedia.org/wiki/Torch_(machine_learning)

Caffe

- C++ based
- Higher abstraction than Theano or Torch
- Good for training standard models
- Model zoo for pre-trained models

Tensorflow

- Symbolic graph and auto-gradient
- Python interface
- Visualization tools
- Some performance issues regarding speed and memory

https://github.com/soumith/convnet-benchmarks/issues/66

TensorFlow

Tips and Tricks

Number of Layers / Size of Layers

- If data is unlimited larger and deeper should be better
- Larger networks can overfit more easily
- Take computational cost into account

Learning Rate

- One of the most important parameters
- If network diverges most probably learning rate is too large
- Smaller works better
- Can slowly decay over time
- Can have one learning rate per layer

Other tips for SGD:
http://leon.bottou.org/publications/pdf/tricks-2012.pdf

Momentum

- Helps to escape local minima
- Crucial to achieve high performance

$$\begin{aligned} v_{t+1} &= \mu v_t - \varepsilon \nabla f(\theta_t) \\ \theta_{t+1} &= \theta_t + v_{t+1} \end{aligned}$$

More about Momentum:
http://www.jmlr.org/proceedings/papers/v28/sutskever13.pdf

Convergence

- Monitor validation error
- Stop when it doesn't improve within n iterations

- If learning rate decays you might want to adjust number of iterations

Initialization of W

- Need randomization to break symmetry
- Bad initializations are untrainable
- Most heuristics depend on the number of input (and output) units
- Sometimes W is rescaled during training
  - Weight decay (L2 regularization)
  - Normalization

Data Augmentation

- Exploit invariances of the data
- Rotation, translation
- Nonlinear transformation

Type	Classifier
Neural network	6-layer NN 784-2500-2000-1500-1000-500-10 (on GPU), with elastic distortions
Convolutional neural network	Committee of 35 conv. net, 1-20-P-40-P-150-10, with elastic distortions

Preprocessing	Error rate (%)
None	0.35[17]
Width normalizations	0.23[8]

http://en.wikipedia.org/wiki/MNIST_database

Data Normalization

- We have seen std and mean normalization
- Whitening
  - Neighbored pixels often are redundant
  - Remove correlation between features

More about preprocessing:
http://deeplearning.stanford.edu/wiki/index.php/Data_Preprocessing

Non-Linear Activation Function

- Sigmoid
  - Traditional choice
- Tanh
  - Symmetric around the origin
  - Better gradient propagation than Sigmoid
- Rectified Linear
  - max(x,0)
  - State of the art
  - Good gradient propagation
  - Can “die”

L1 and L2 Regularization

- Most pictures of nice filters involve some regularization

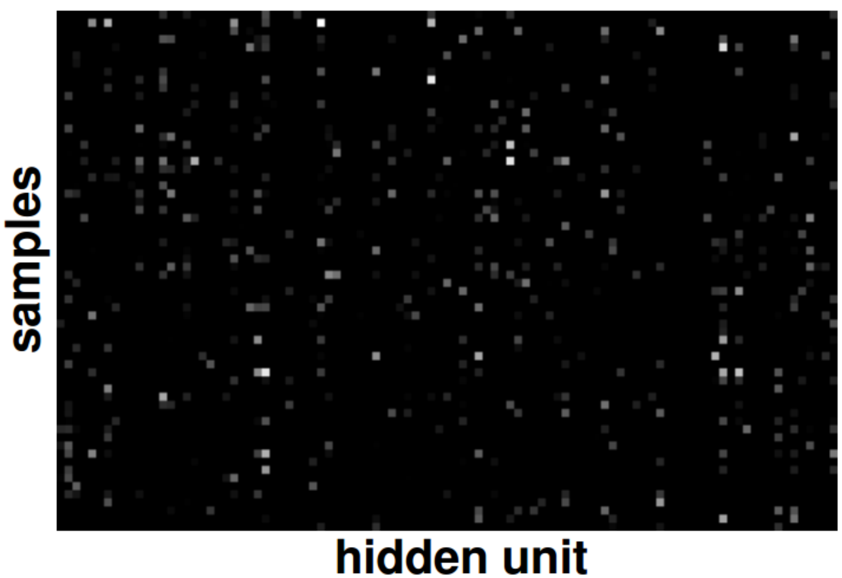
- L2 regularization corresponds to weight decay
- L2 and early stopping have similar effects

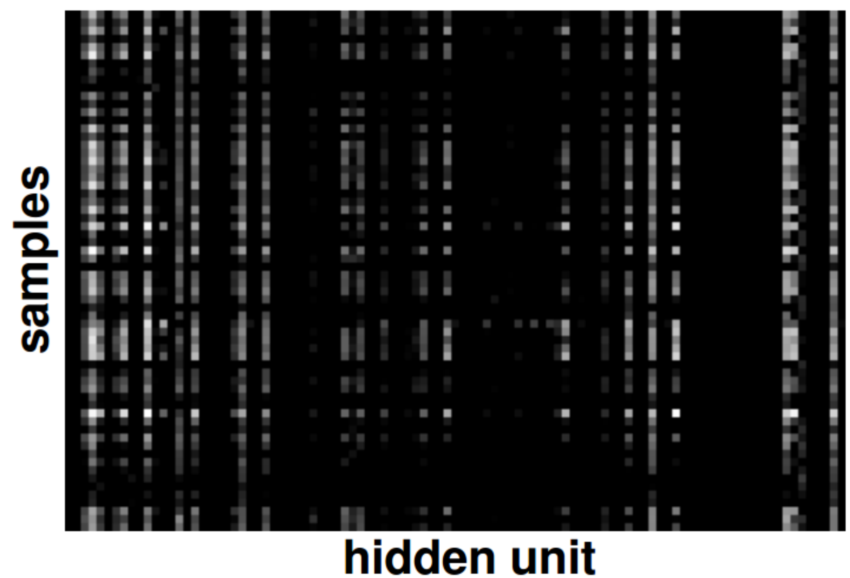
- L1 leads to sparsity

- Might not be needed anymore (more data, dropout)

Monitoring Training

- Monitor training and validation performance
- Can monitor hidden units
- Good: Uncorrelated and high variance





Further Resources

- More about theory:
  - Yoshua Bengio's book:http://www.iro.umontreal.ca/~bengioy/dlbook/
  - Deep learning reading list: http://deeplearning.net/reading-list/
- More about Theano:
  - http://deeplearning.net/software/theano/
  - http://deeplearning.net/tutorial/