Fall 2015 Deep Learning CMPSCI 697L

Deep Learning Lecture 1

Sridhar Mahadevan
Autonomous Learning Lab
UMass Amherst

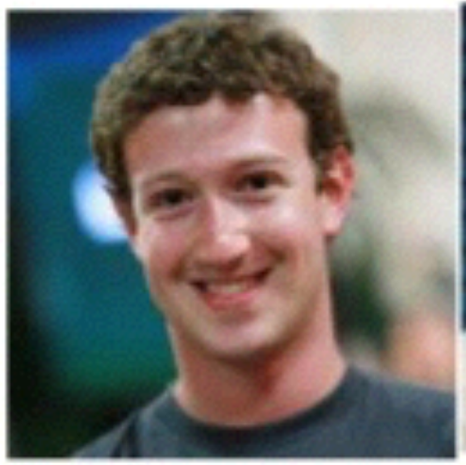
Outline of lecture

- Overview of the course
- What is deep learning?
- Group assignments
- Projects and software resources
- Paper readings

Deep Learning in the Press…

Hinton

Ng



Zuckerberg

LeCun

Kurzweil

Google Hires Brains that Helped Supercharge Machine Learning.
Wired 3/2013.

Facebook taps 'Deep Learning' Giant for New AI Lab.
Wired 12/2013.

Is "Deep Learning" A Revolutions in Artificial Intelligence?
New Yorker 11/2012.

The Man Behind the Google Brain: Andrew Ng and the Quest for the New AI.
Wired 5/2013.

New Techniques from Google and Ray Kurzweil Are Taking Artificial Intelligence to Another Level.
MIT Technology Review 5/2013.

Course specifics

- Meets 9:05-12, Room 142, Fridays
- Home page: http://www-edlab.cs.umass.edu/cs697l/
- Lecture notes, readings etc: http://www-edlab.cs.umass.edu/cs697l/2015-outline.htm

Course Structure

- Weekly readings
- Participate in class discussions
- Do suggested independent activities
- Mid-term common group project
- Final group project

CMPSCI 697L	Deep Learning	Fall 2015

Tentative Course Schedule

Lecture	Date	Topic	Suggested Reading	Further Readings	Independent activity	Public domain software
1	Fri Sept 11th	Historical overview of neural networks; Introduction to Deep Learning	Chapter 1, Learning Deep Architectures for AI	Hacker's guide to neural networks Three Classes of Deep Learning Architectures	Set up Theano on your machine, and run the logistic regression example using the MNIST dataset	Deep learning tutorial using Theano Caffe: Deep learning package in C++ Mocha: Deep learning package in Julia
2	Fri Sept 18th	Deep architectures: feedforward neural nets, convolutional networks, deep generative networks etc.	Chapter 4, Learning Deep Architectures for AI	Learning representations by back-propagating errors	Implement/download a feedforward neural net learner, and experiment with it on some datasets	Implementation of feedforward networks in Theano
3	Fri Sept 25th	Energy-based models; Boltzmann machines; deep belief networks	Chapter 5, Learning Deep Architectures for AI	Fast Learning Algorithm for Deep Belief Networks	Train an RBM or a deep belief network on the MNIST dataset (this requires a relativelu fast PC)	Deep belief networks implemented in Theano

Forum Discussions

Forum	Topic
Software	Find out new packages for deep learning and report on them
Papers	Keep track of new papers on deep learning on Arxiv, NIPS, ICML etc.
Press	What new articles are appearing about deep learning in the media?
Industry	New startups on deep learning, ideas for startups, etc.

Grading

Component	Weight
Final Project	40%
Mini Project	40%
Independent Activities	20%

GPUS MAKE DEEP LEARNING ACCESSIBLE

Deep learning with COTS HPC systems

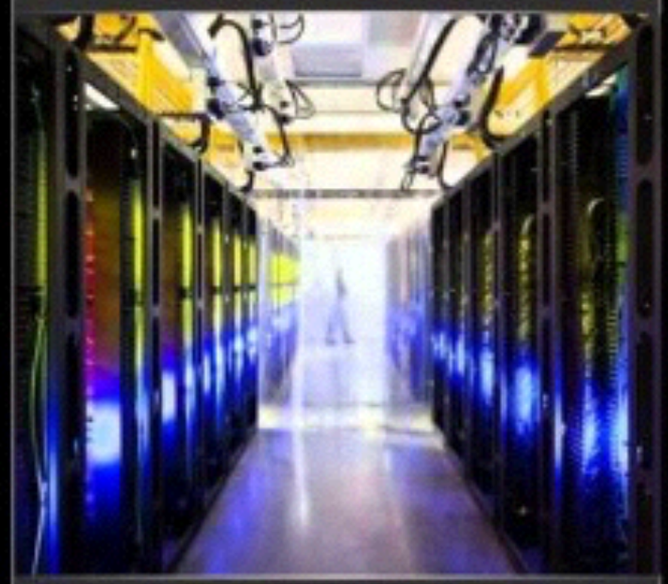
A. Coates, B. Huval, T. Wang, D. Wu, A. Ng, B. Catanzaro

ICML 2013

"Now You Can Build Google's $1M Artificial Brain on the Cheap"

WIRED

GOOGLE DATACENTER



1,000 CPU Servers
2,000 CPUs • 16,000 cores

600 kWatts
$5,000,000

STANFORD AI LAB

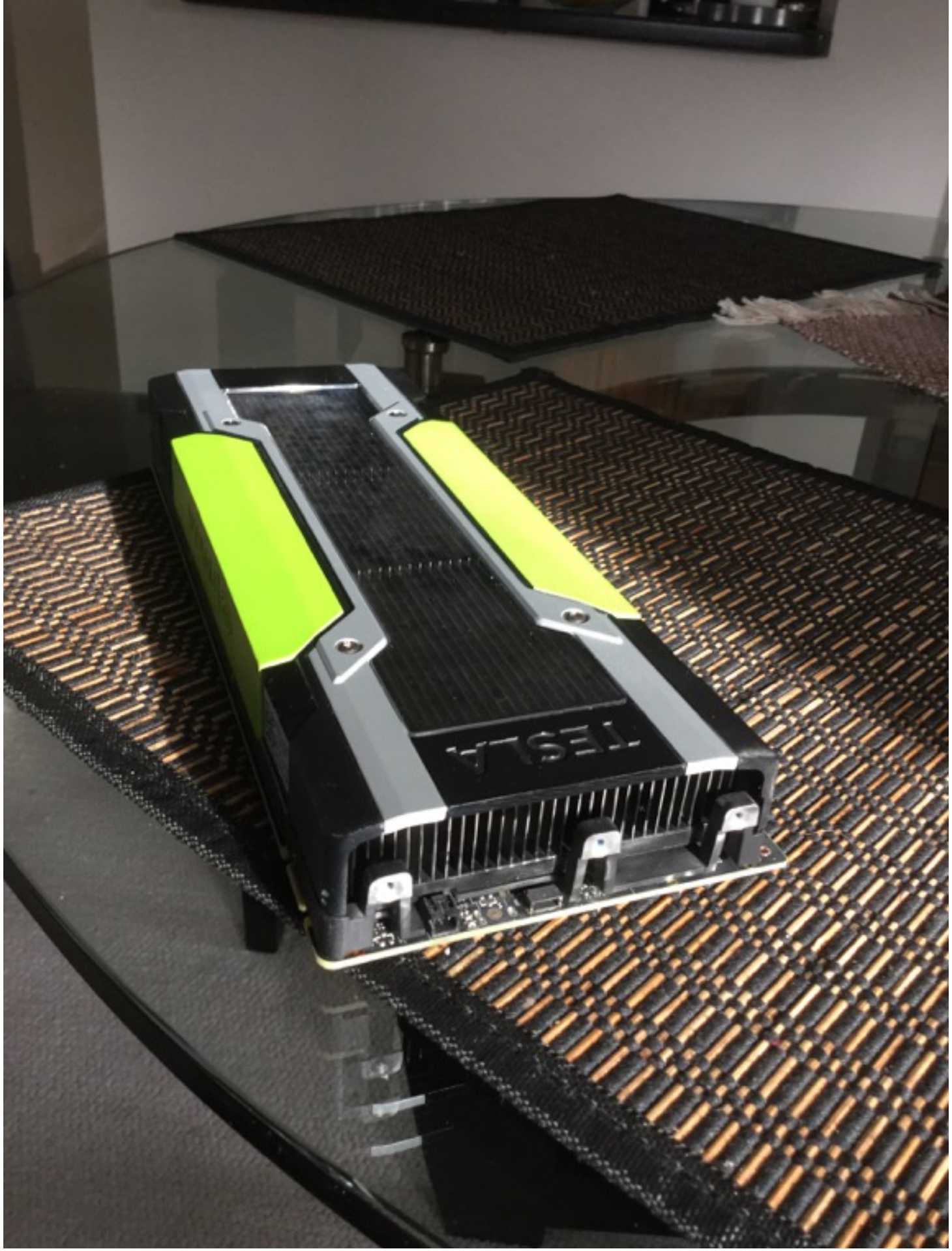
3 GPU-Accelerated Servers
12 GPUs • 18,432 cores

4 kWatts
$33,000

My GPU machine

nnVidia Tesla K80 GPU card

~6000 processors
~8 Teraflops



Dell Power Edge C4130

Supports 4 Tesla K80 cards

~20,000 GPU Cores

Deep Learning Software

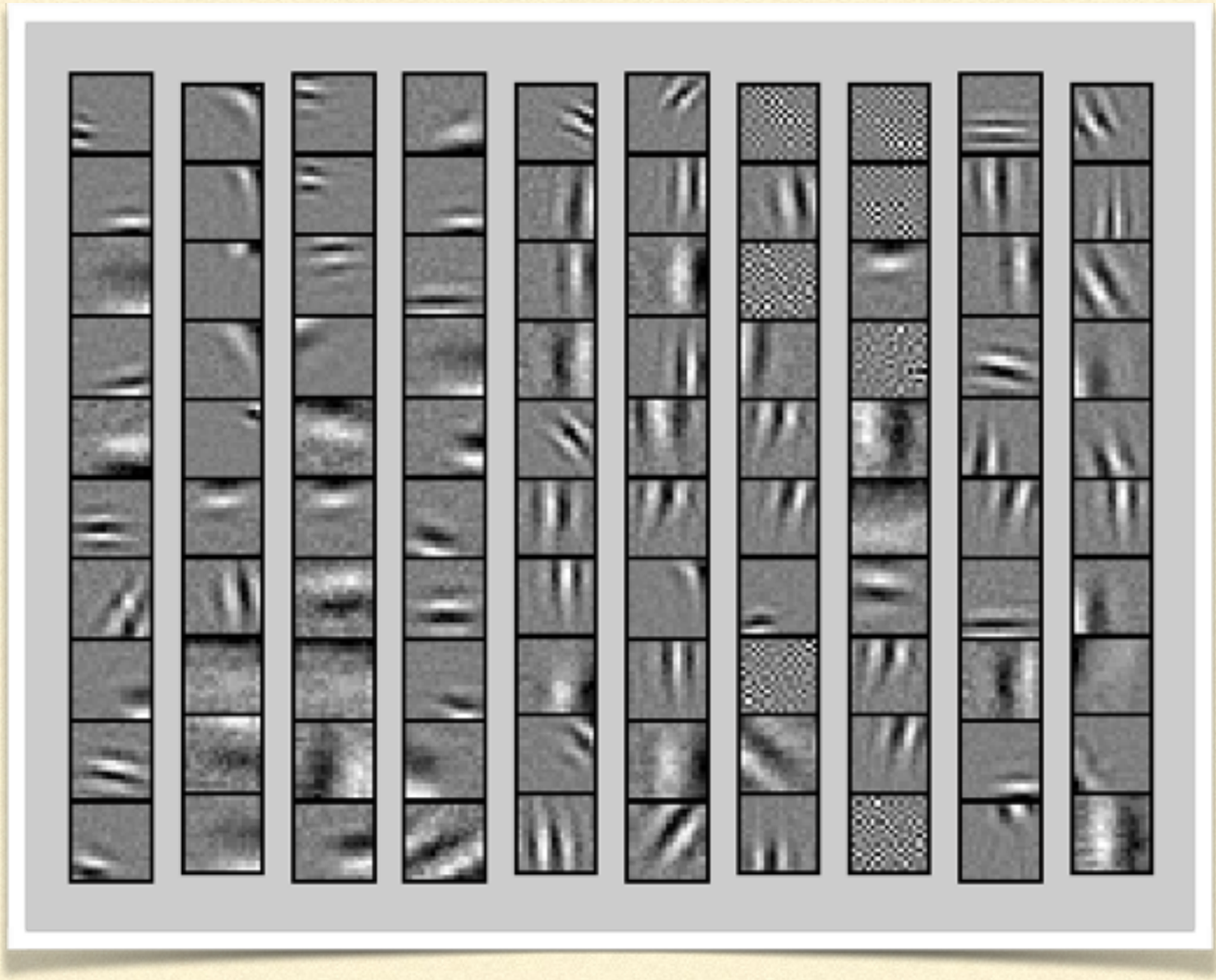
- Theano
- Caffe
- Mocha
- Minerva
- Torch/Lua

What is deep learning?

Deep Learning is an emerging area of machine learning

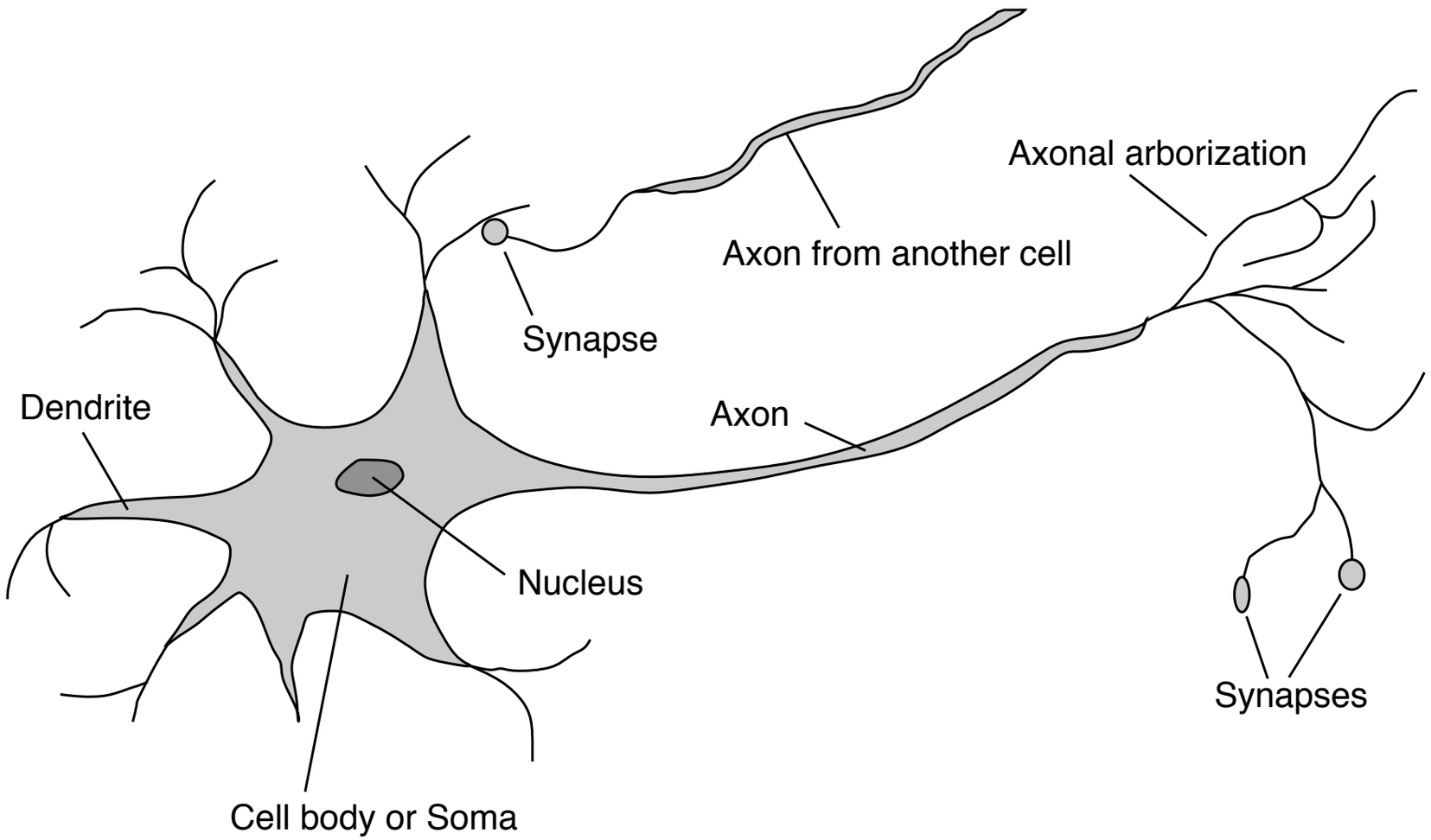
It uses a hierarchically organized model to learn multi-scale representations

It builds on several ideas: neural networks, graphical models, probabilistic inference, sampling



Human Brain

10^{11} neurons of > 20 types, 10^{14} synapses, 1ms–10ms cycle time
Signals are noisy "spike trains" of electrical potential



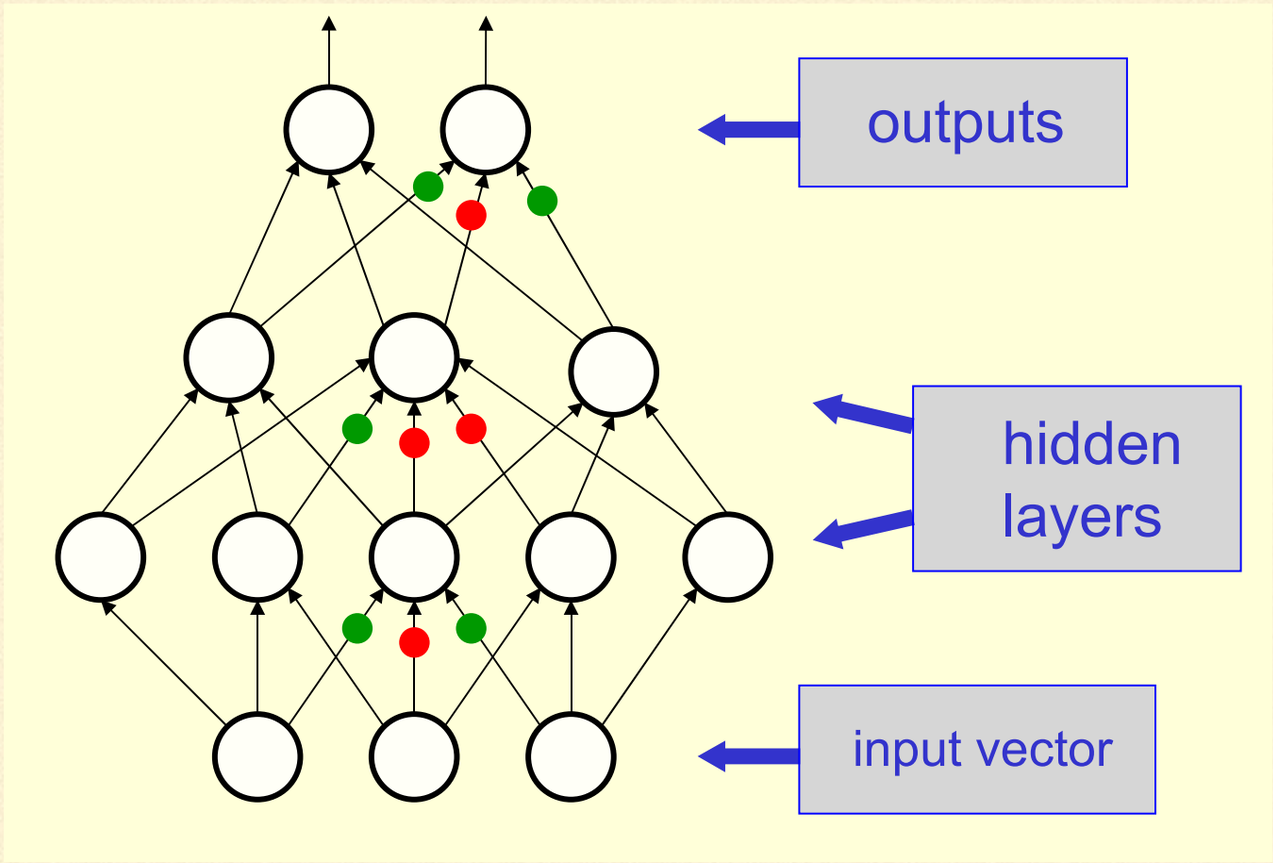
Brains vs. Computers

- Highly parallel distributed vs. serial processors
- Content addressable memory vs. random access
- Parallel distributed representation vs. localized representation
- Very slow computing units in the brain

Capabilities of the Brain

- Excels at perception, language, learning, and memory (of some types)
- Poor at arithmetic (can you multiply two 10 digit numbers?)
- Are these capabilities because of its unique architecture?

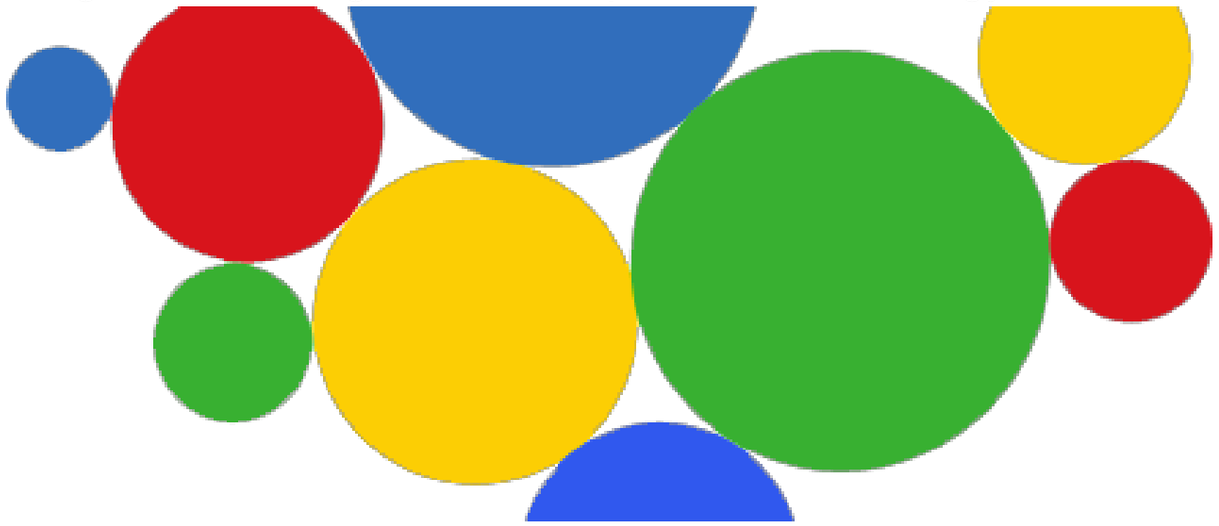
HISTORICAL BACKGROUND



- 1950s-1960s: Perceptrons, single-layer neural networks
- 1980s: Feedforward multi-layer networks, back propagation algorithm
- 1990s: Kernel methods, SVMs
- 2005-now: Deep learning

Google Official Blog

Insights from Googlers into our products, technology, and the Google culture



Using large-scale brain simulations for machine learning and A.I.

Posted: Tuesday, June 26, 2012

You probably use machine learning technology dozens of times a day without knowing it—it's a way of training computers on real-world data, and it enables high-quality speech recognition, practical computer vision, email spam blocking and even self-driving cars. But it's far from perfect—you've probably chuckled at poorly transcribed text, a bad translation or a misidentified image. We believe machine learning could be far more accurate, and that smarter computers could make everyday tasks much easier. So our research team has been working on some new approaches to large-scale machine learning.

Today's machine learning technology takes significant work to adapt to new uses. For example, say we're trying to build a system that can distinguish between pictures of cars and motorcycles. In the standard machine learning approach, we first have to collect tens of thousands of pictures that have already been labeled as "car" or "motorcycle"—what we call *labeled data*—to train the system. But labeling takes a lot of work, and there's comparatively little labeled data out there.

Fortunately, recent research on *self-taught learning* (PDF) and *deep learning* suggests we might be able to rely instead on *unlabeled data*—such as random images fetched off the web or out of YouTube videos. These algorithms work by building artificial neural networks, which loosely simulate neuronal (i.e., the brain's) learning processes.

We then ran experiments that asked, informally: If we think of our neural network as simulating a very small-scale "newborn brain," and show it YouTube video for a week, what will it learn? Our hypothesis was that it would learn to recognize common objects in those videos. Indeed, to our amusement, one of our artificial neurons learned to respond strongly to pictures of... cats. Remember that this network had never been told what a cat was, nor was it given even a single image labeled as a cat. Instead, it "discovered" what a cat looked like by itself from only unlabeled YouTube stills. That's what we mean by self-taught learning.

Google's cat detector trained on U-tube videos

MIT Technology Review

Facebook Launches Advanced AI Effort to Find Meaning in Your Posts

A technique called deep learning could help Facebook understand its users and their data better.

By Tom Simonite on September 20, 2013

Facebook is set to get an even better understanding of the 700 million people who use the social network to share details of their personal lives each day.

A new research group within the company is working on an emerging and powerful approach to artificial intelligence known as deep learning, which uses simulated networks of brain cells to process data. Applying this method to data shared on Facebook could allow for novel features and perhaps boost the company's ad targeting.

Deep learning has shown potential as the basis for software that could work out the emotions or events described in text even if they aren't explicitly referenced, recognize objects in photos, and make sophisticated predictions about people's likely future behavior.

ICLR 2014

Yahoo Acquires Startup LookFlow To Work On Flickr And ‘Deep Learning’

LookFlow, a startup that describes itself as “an entirely new way to explore images you love,” just announced that it has been acquired by Yahoo and will be joining the Flickr team[1,2,3]. The company is cofounded by Bobby Jaros and Simon Osindero. Their company was utilizing deep learning techniques for image recognition problems[1,2].

News sources:

[1] The next web, Emil Protalanski, http://thenextweb.com/insider/2013/10/23/yahoo-acquires-ai-startup-lookflow-improve-discovery-flickr-build-deep-learning-group/

[2] Techcrunch, Anthony Ha, http://techcrunch.com/2013/10/23/yahoo-acquires-startup-lookflow-to-work-on-flickr-and-deep-learning/

Lookflow’s web site:

[3] https://lookflow.com/

October 23rd, 2013 | Tags: acquisition, lookflow, startup, yahoo | Category: news

Deep Learning evolution

N Neural Network
P Probabilistic Model
Supervised learning
Unsupervised learning

N Perceptron [Rosenblatt 1958]
SVM [Vapnik 1979]
Boosting [Schapire 1990]
N RBM P [Hinton 1999]
Decision Tree(CLS) [Hunt 1966]
N AE [Hinton 1989]
GMM P [Reynolds 1992]

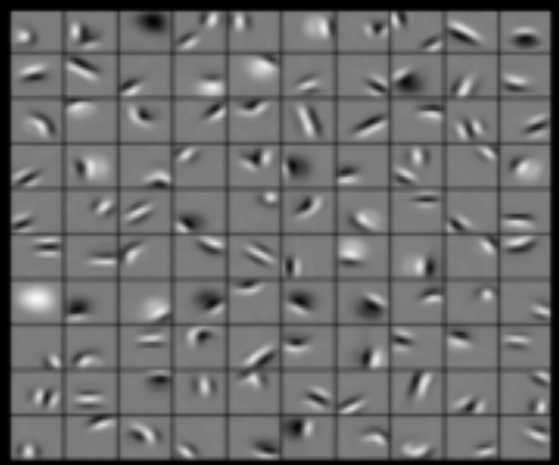
Traditional models

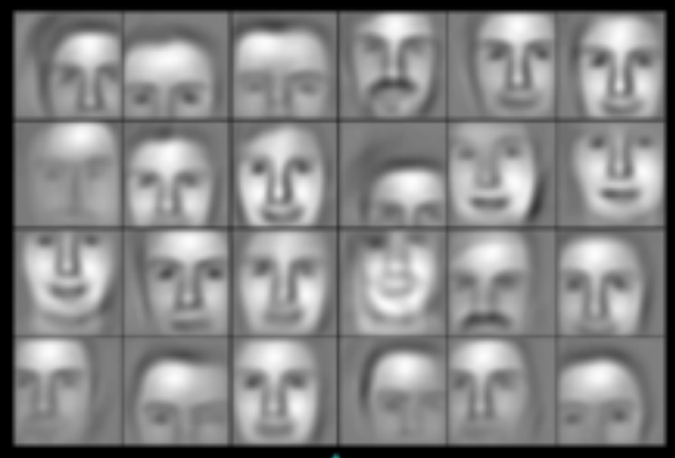
Deep models

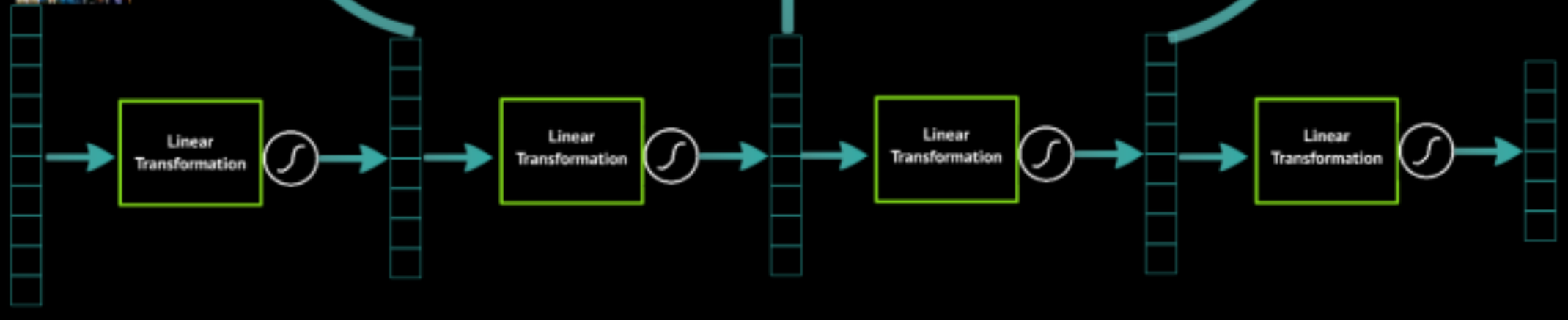
N DBN P [Hinton et al 2006]
N RNN [Grossberg 1973]
ΣΠ P [Hornik 1989]
N Sparse Coding P [Olshausen 1996]
N D-AE [Vincent 2008]
N Neural Nets [McCulloch & Pitt 1943]
N Conv. Net [Fukushima 1979]
BayesNP P [Teh & Jordan 2009]
N DBM P [Salakhutdinov & Hinton 2009]

Algorithms authors and dates often unclear. Oldest citations were assumed
Classifications based on Yann LeCun's Deep Learning class at NYU – spring 2014

Deep Learning learns layers of features

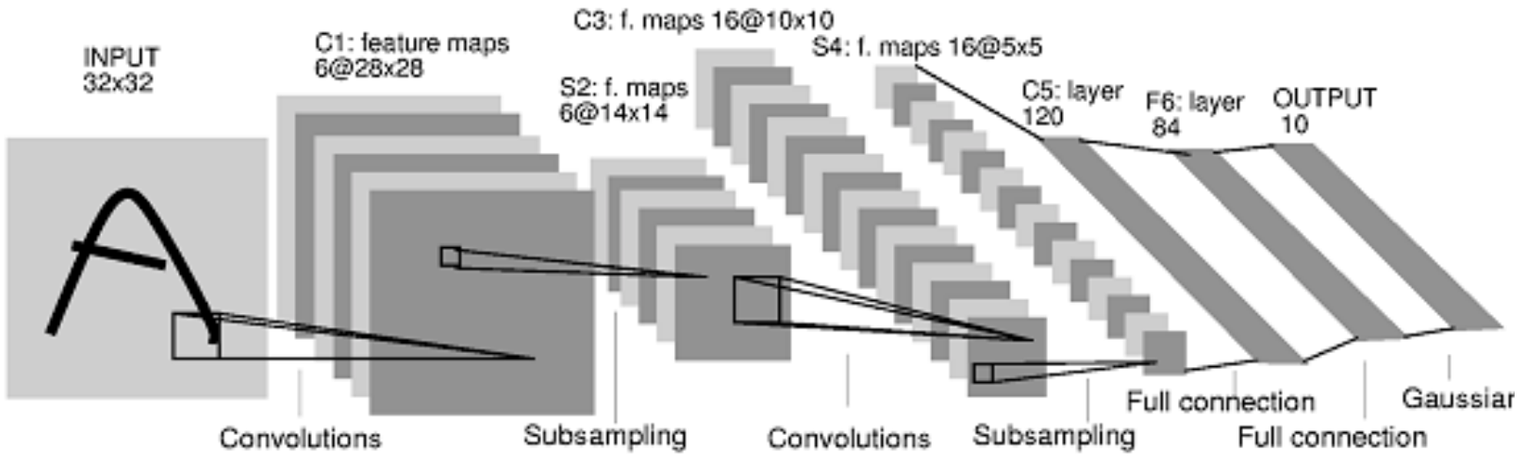






1998

LeCun et al.



# of transistors

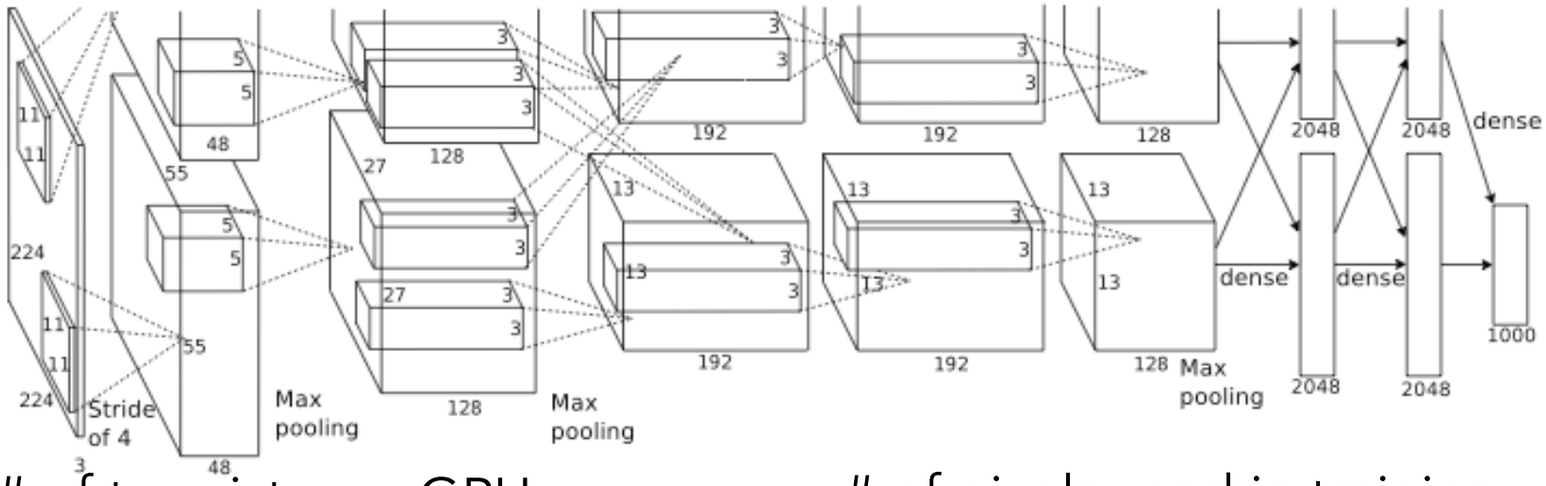
10^6

# 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

Large-scale Image recognition

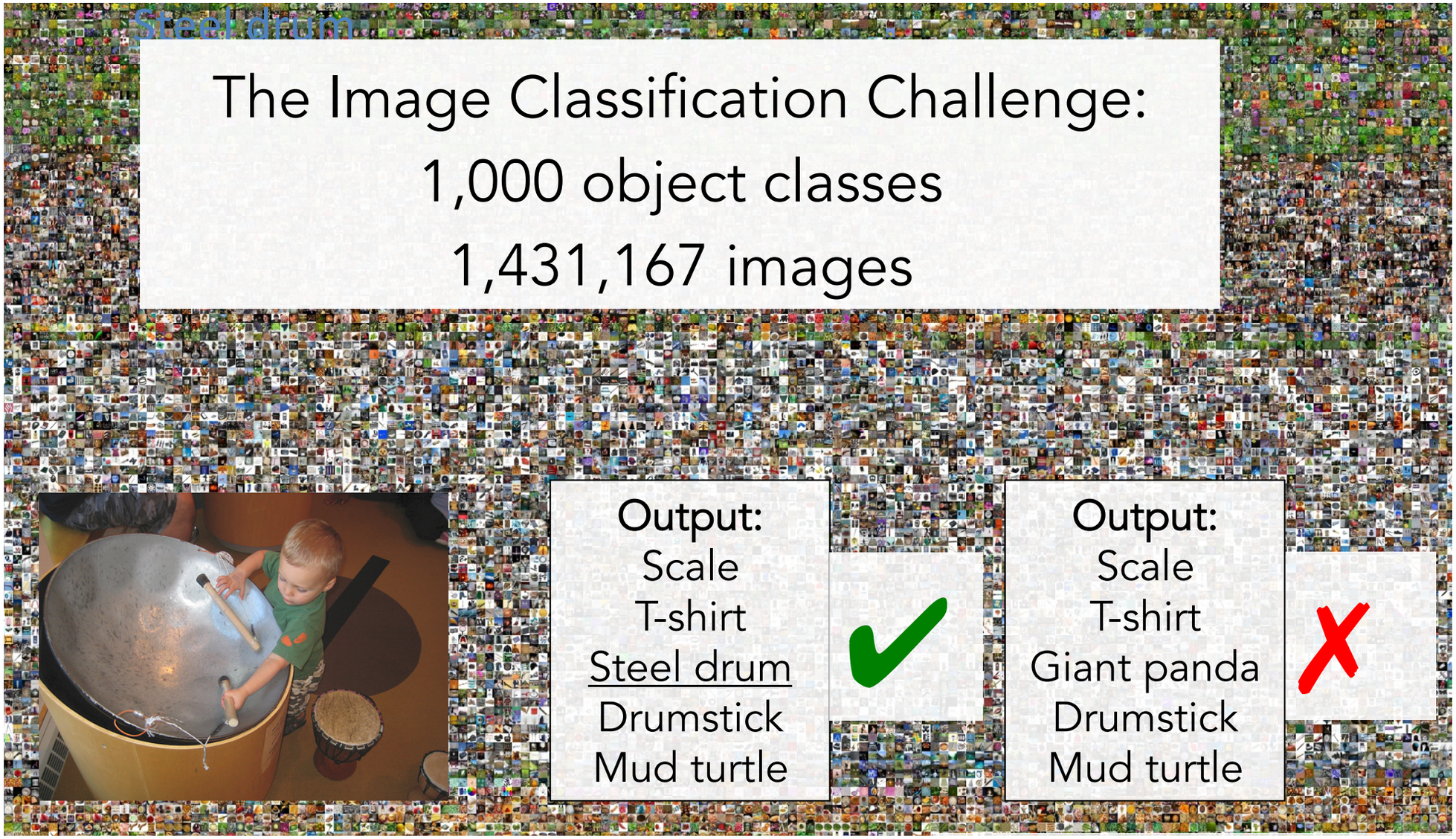
IMAGENET

www.image-net.org

22K categories and **14M** images

- Animals
  - Bird
  - Fish
  - Mammal
  - Invertebrate
- Plants
  - Tree
  - Flower
- Food
- Materials
- Structures
- Artifact
  - Tools
  - Appliances
  - Structures
- Person
- Scenes
  - Indoor
  - Geological Formations
- Sport Activities

Imagenet Challenge



IMAGENET Large Scale Visual Recognition Challenge

Year 2010

NEC-UIUC

Dense grid descriptor:
HOG, LBP

Coding: local coordinate, super-vector

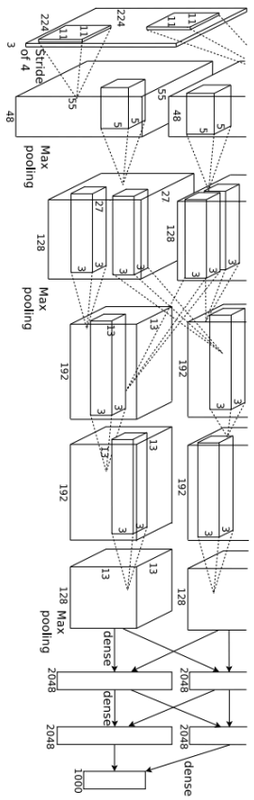
Pooling, SPM

Linear SVM

[Lin CVPR 2011]

Year 2012

SuperVision



[Krizhevsky NIPS 2012]

Year 2014

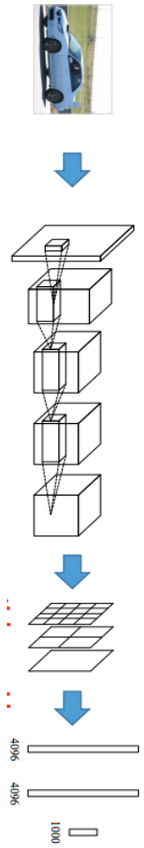
GoogLeNet

[Szegedy arxiv 2014]

VGG

[Simonyan arxiv 2014]

MSRA

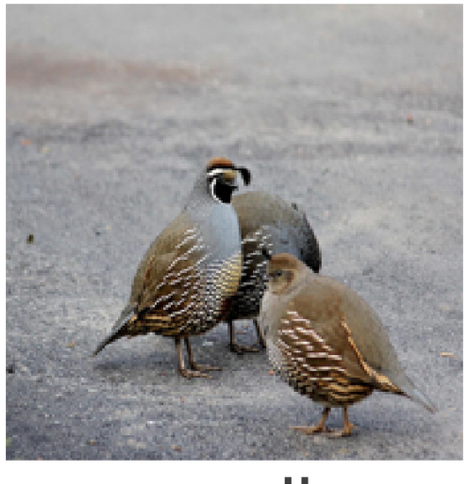


[He arxiv 2014]

flamingo

cock

ruffed grouse



quail

partridge

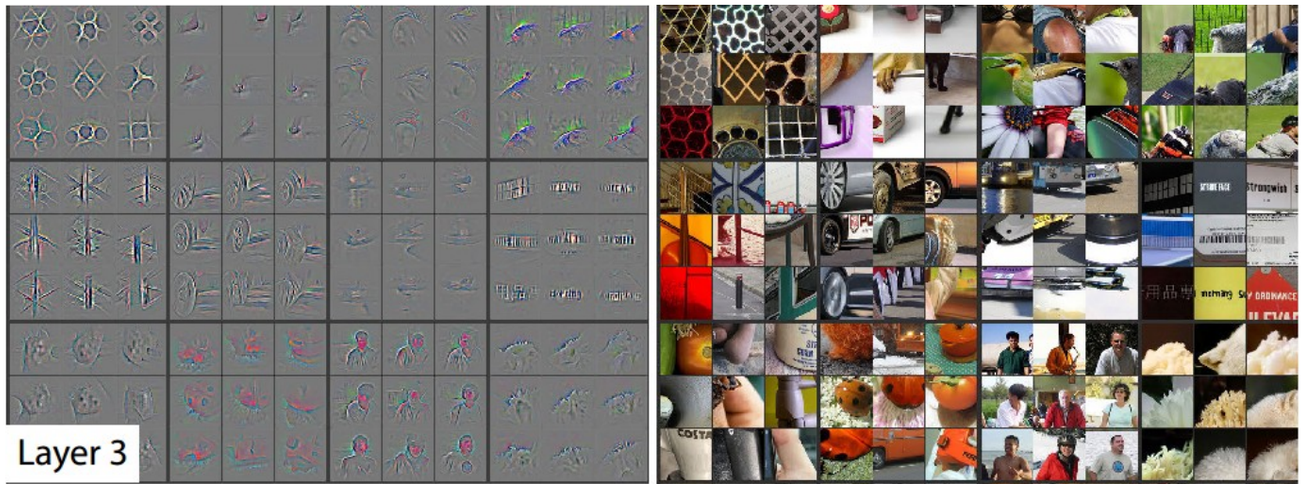
Egyptian cat

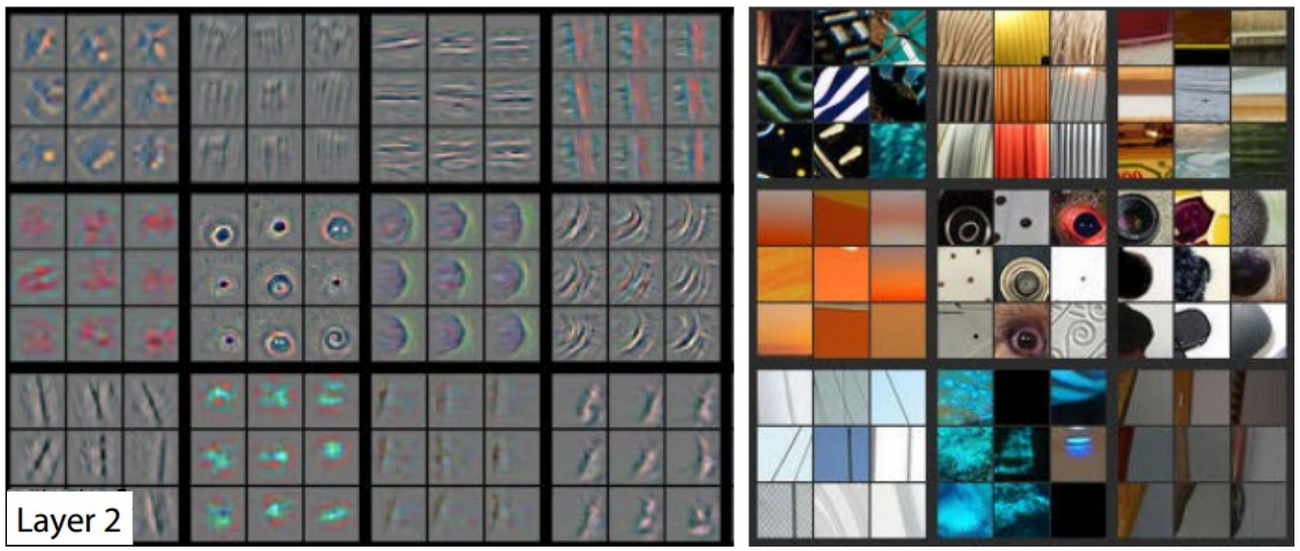
Persian cat

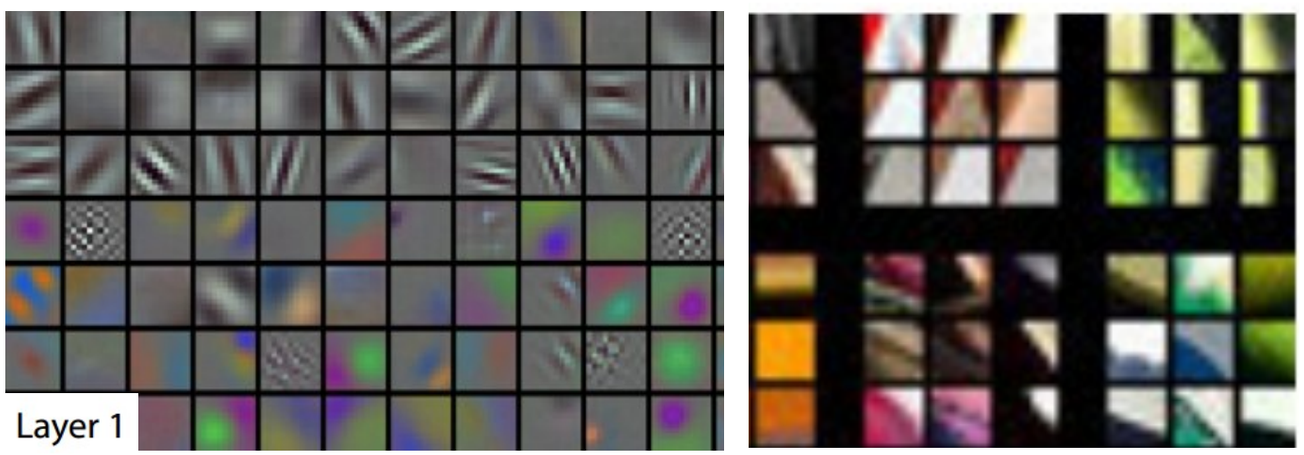
Siamese cat

tabby

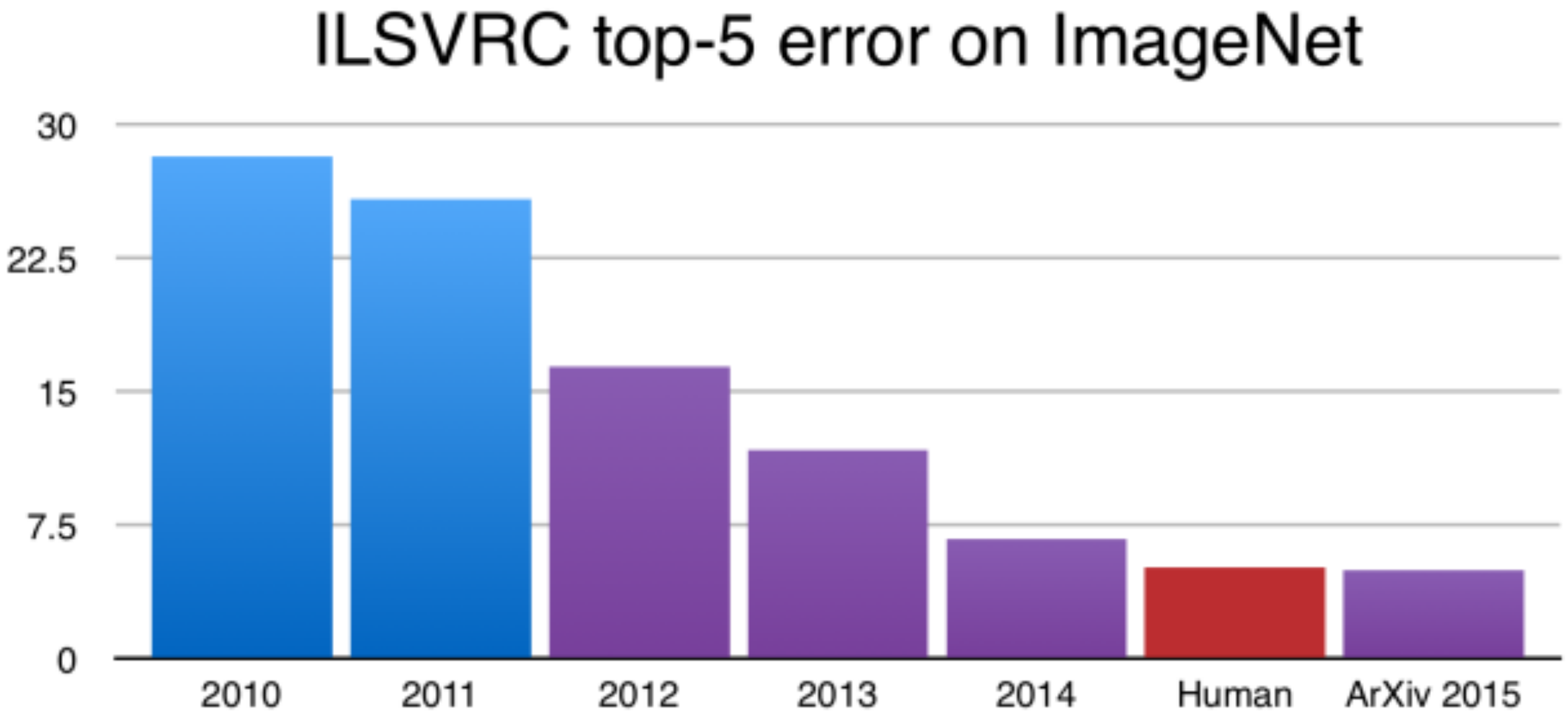
lynx







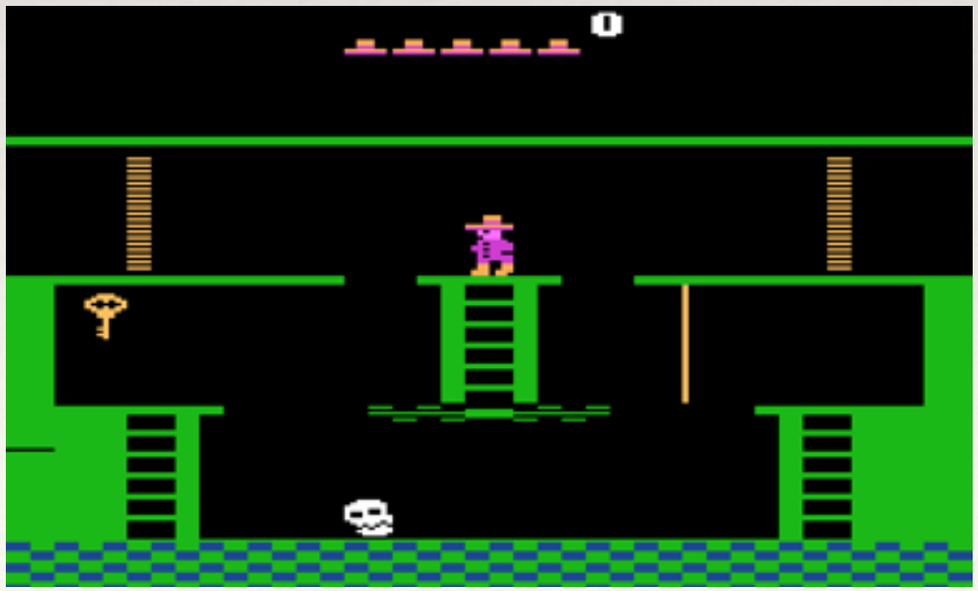
Zeiler and Fergus (2013)


ILSVRC top-5 error on ImageNet
30
22.5
15
7.5
0
2010
2011
2012
2013
2014
Human
ArXiv 2015

Reinforcement Learning in games

Samuel's Checker player, 1956, IBM 701

Tesauro, Backgammon, 1992



Deep Mind, Atari, 2015, GPU

TD-Gammon

(Tesauro, 1992)

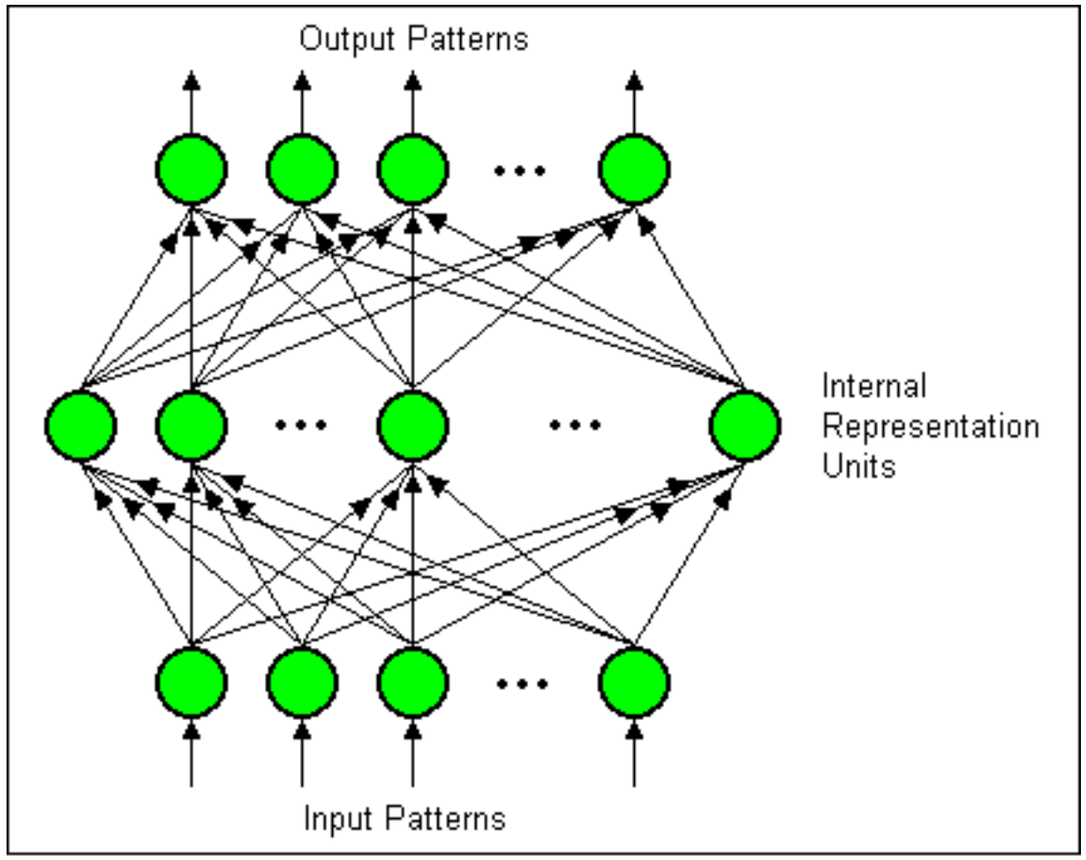


Figure 1. An illustration of the multilayer perception architecture used in TD-Gammon's neural network. This architecture is also used in the popular backpropagation learning procedure. Figure reproduced from [9].

Program	Training Games	Opponents	Results
TDG 1.0	300,000	Robertie, Davis, Magriel	-13 pts/51 games (-0.25 ppg)
TDG 2.0	800,000	Goulding, Woolsey, Snellings, Russell, Sylvester	-7 pts/38 games (-0.18 ppg)
TDG 2.1	1,500,000	Robertie	-1 pt/40 games (-0.02 ppg)

Table 1. Results of testing TD-Gammon in play against world-class human opponents. Version 1.0 used 1-play search for move selection; versions 2.0 and 2.1 used 2-ply search. Version 2.0 had 40 hidden units; versions 1.0 and 2.1 had 80 hidden units.

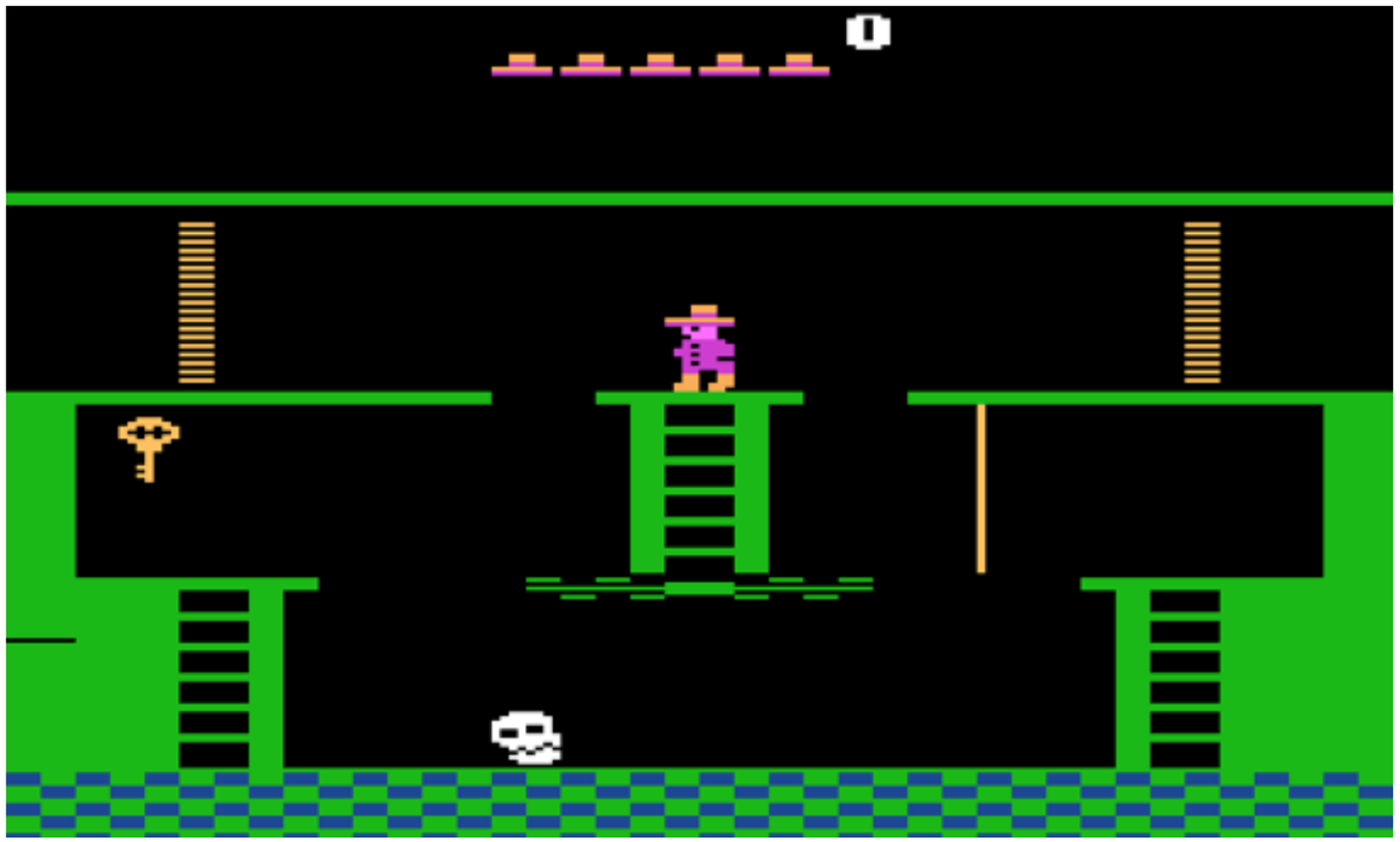
$$w_{t+1} - w_t = \alpha(Y_{t+1} - Y_t) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w Y_k$$

TD(λ)

Deep Reinforcement Learning on the Atari 2600

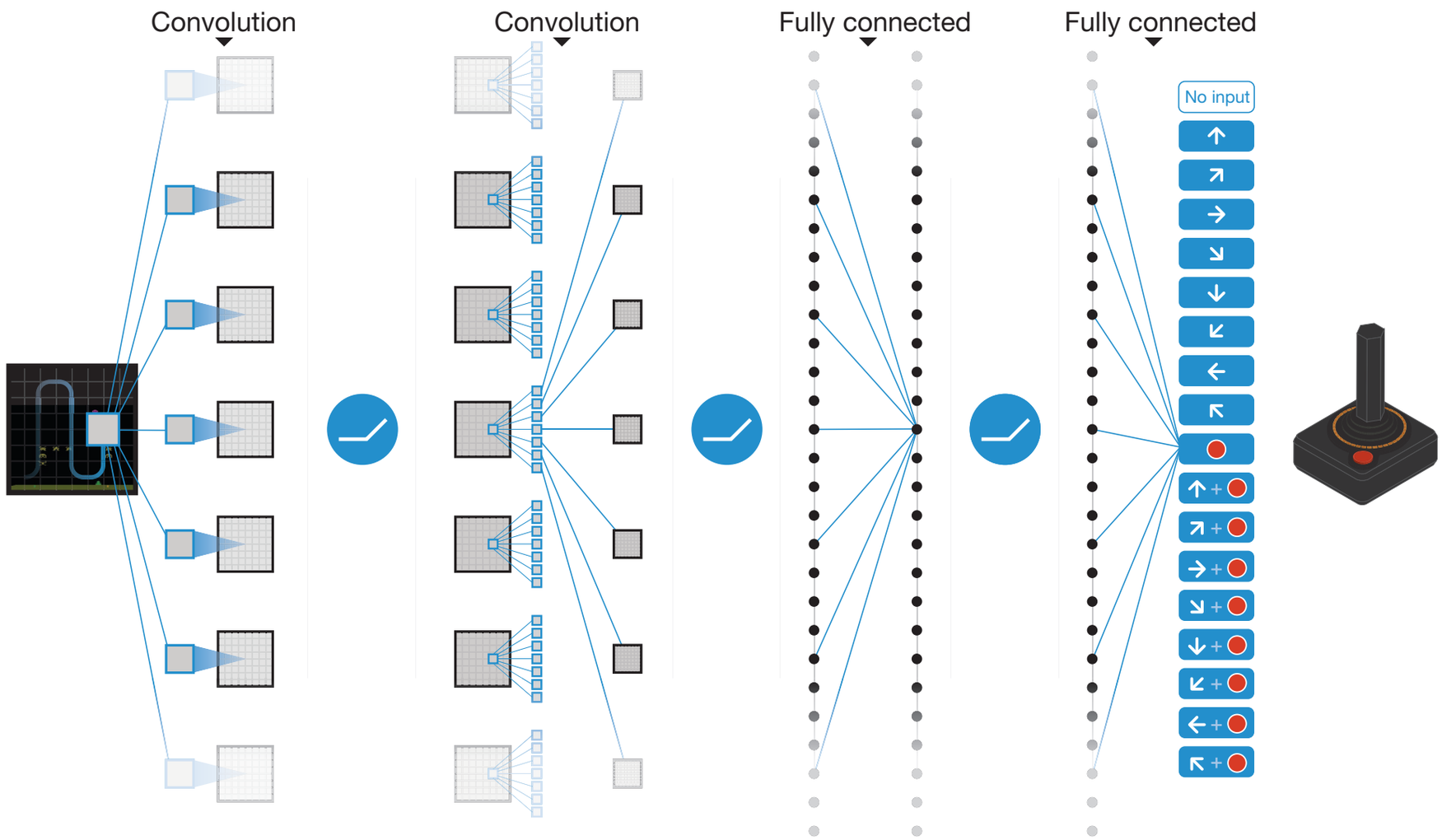
(Nature, 26 Feb 2015)

Uses deep learning to play 49 games in Atari 2600 series



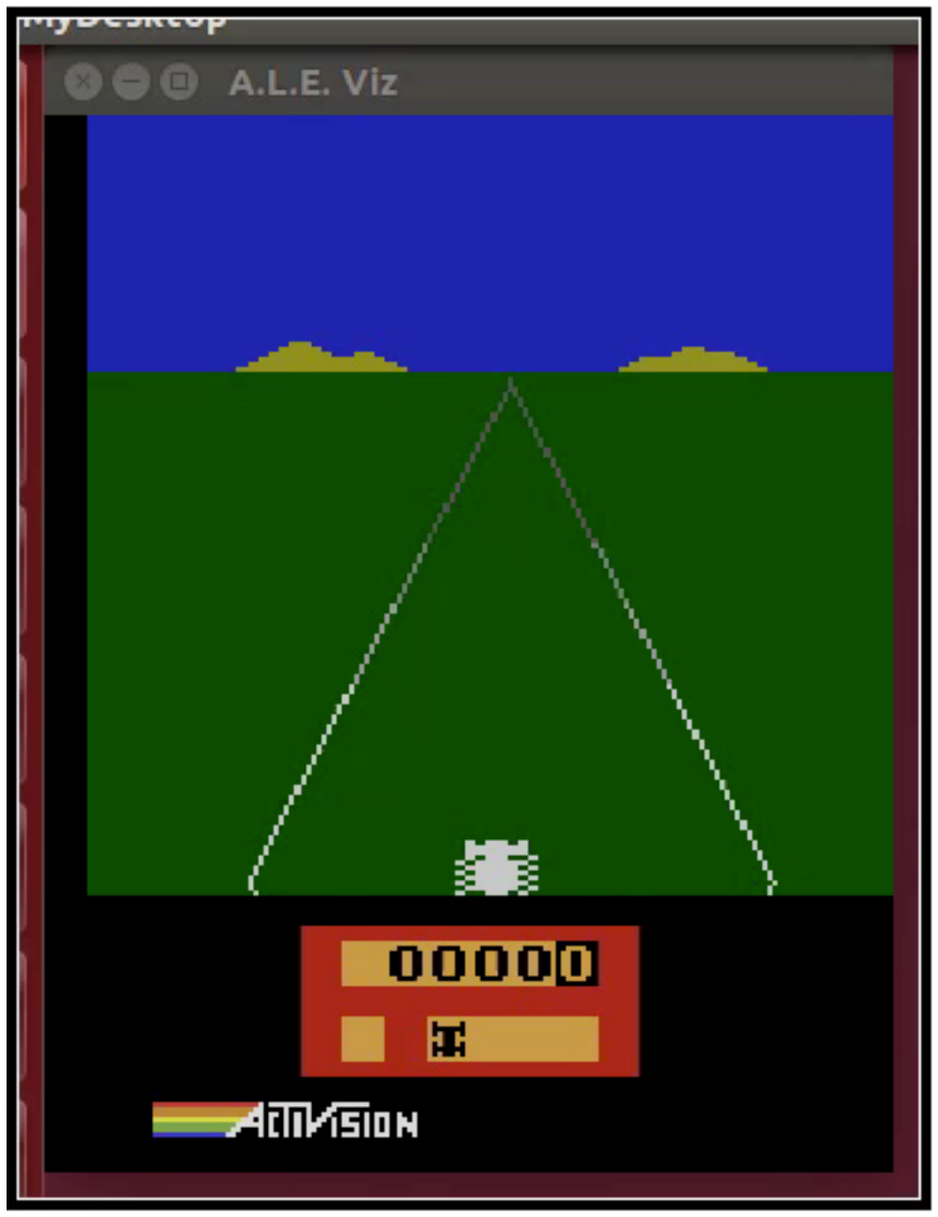
Montezuma’s revenge

Atari Deep Learning Architecture



“Let's see some demos!”

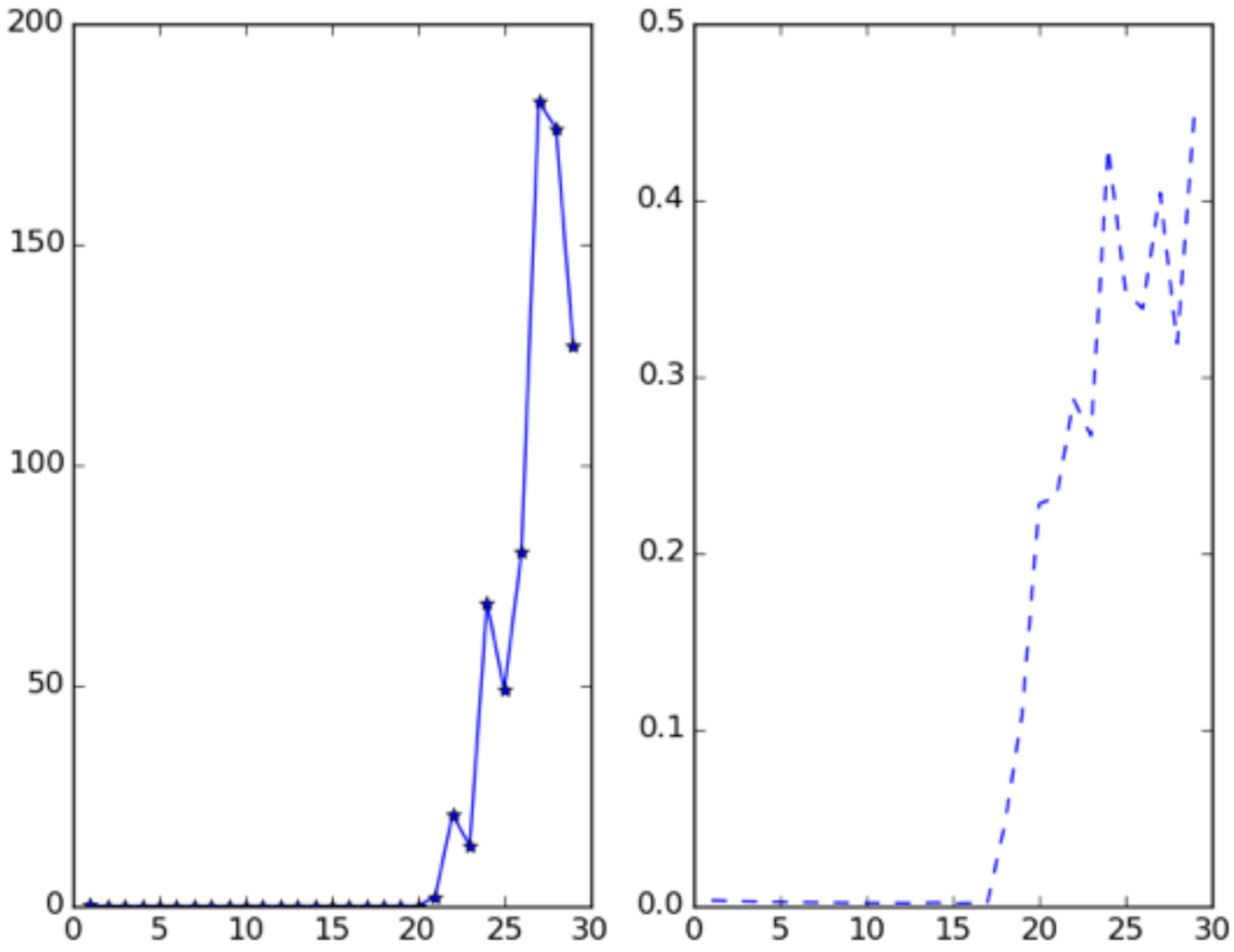
Enduro (1 episode, 50,000 steps, ~20 minutes)



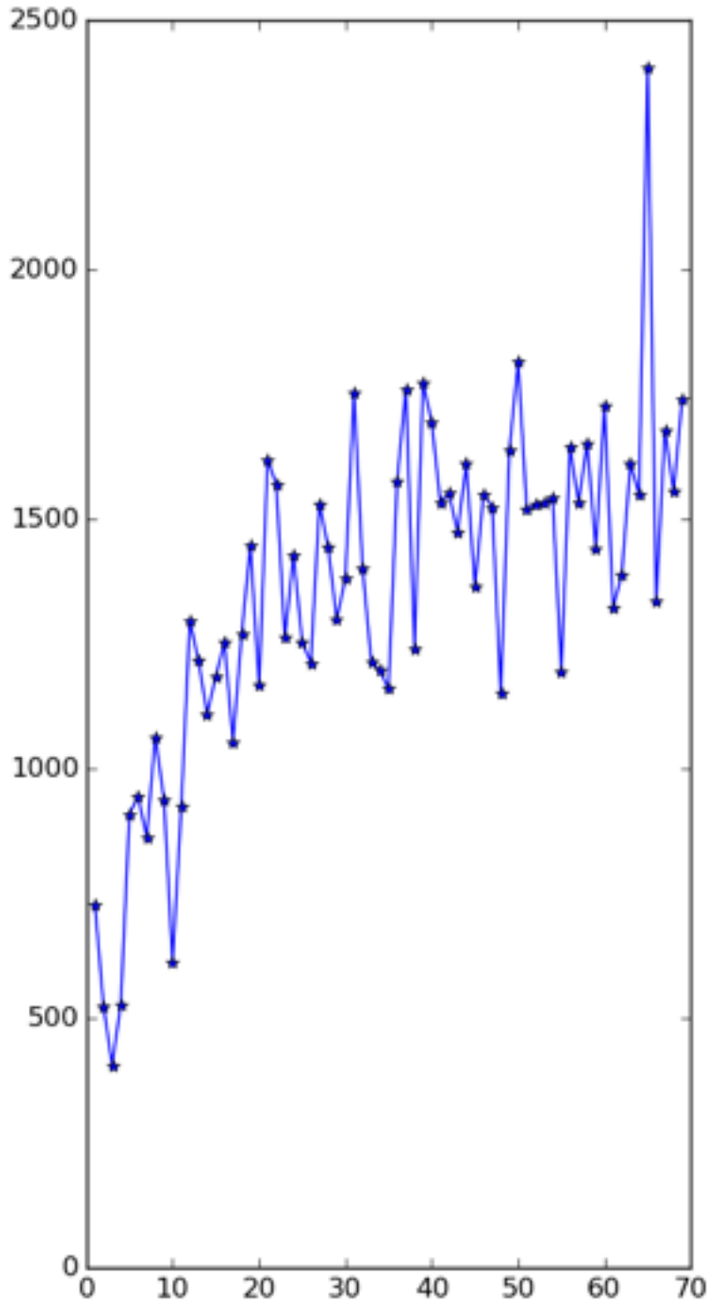
Enduro (20 episodes, 1,000,000 steps, ~7 hours)

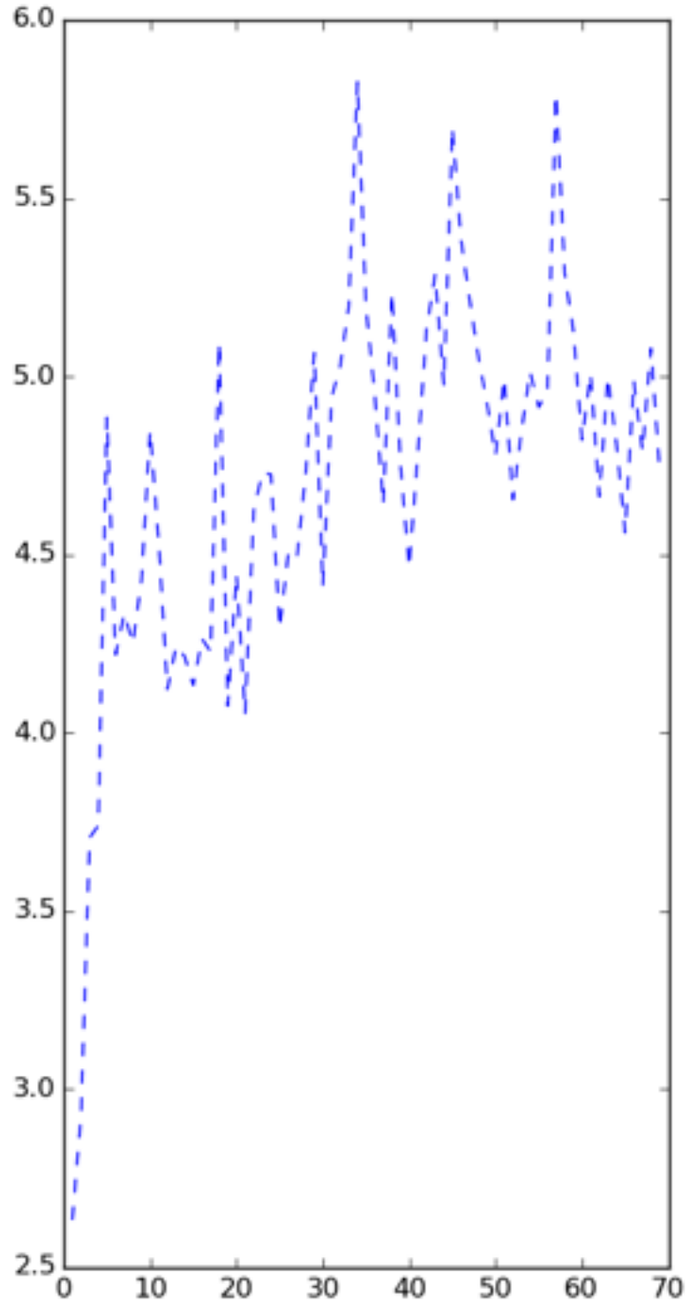
Enduro (30 episodes, 1,500,000 steps, ~10 hours)

Results in Enduro

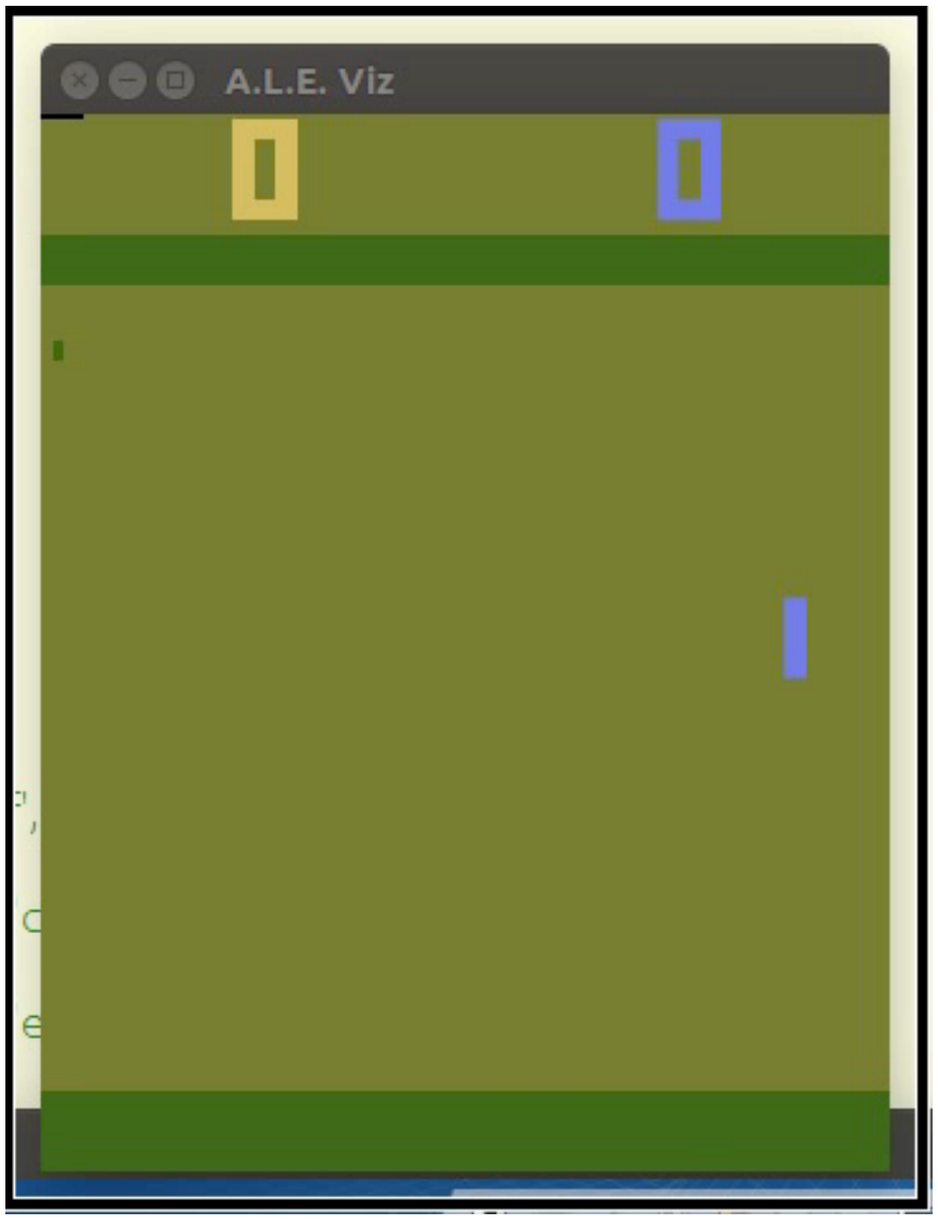


Results: Pong

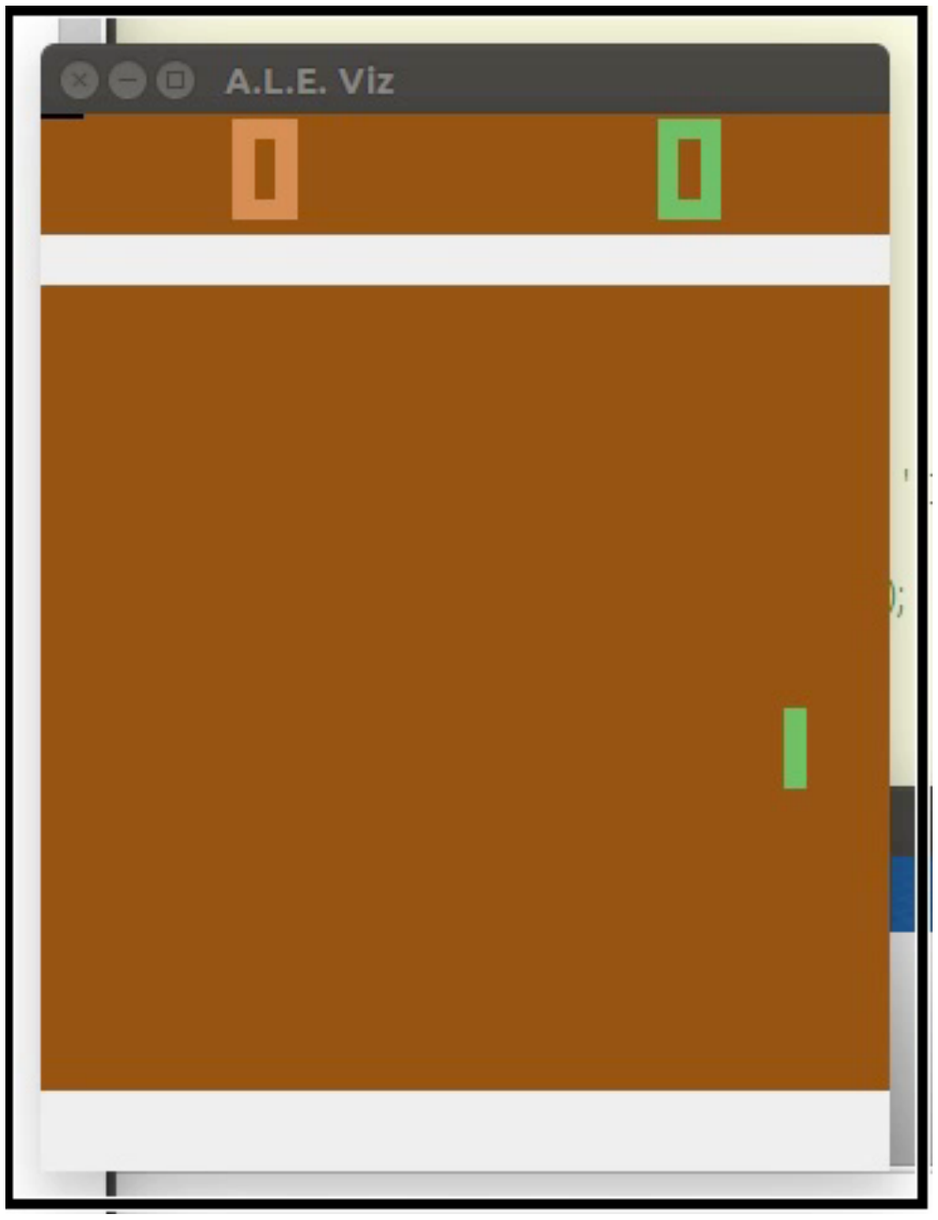




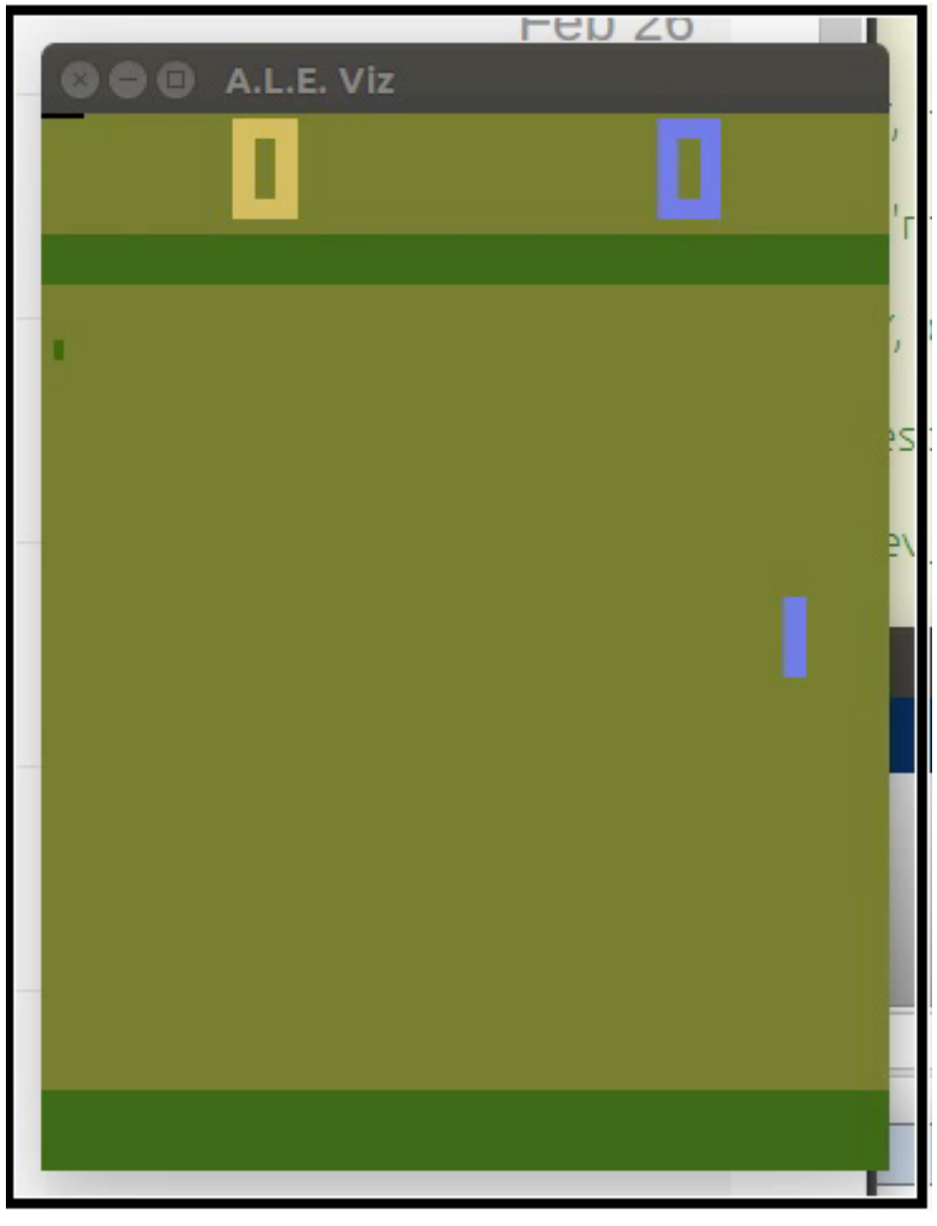
Pong: after 50,000 steps (~20 minutes)



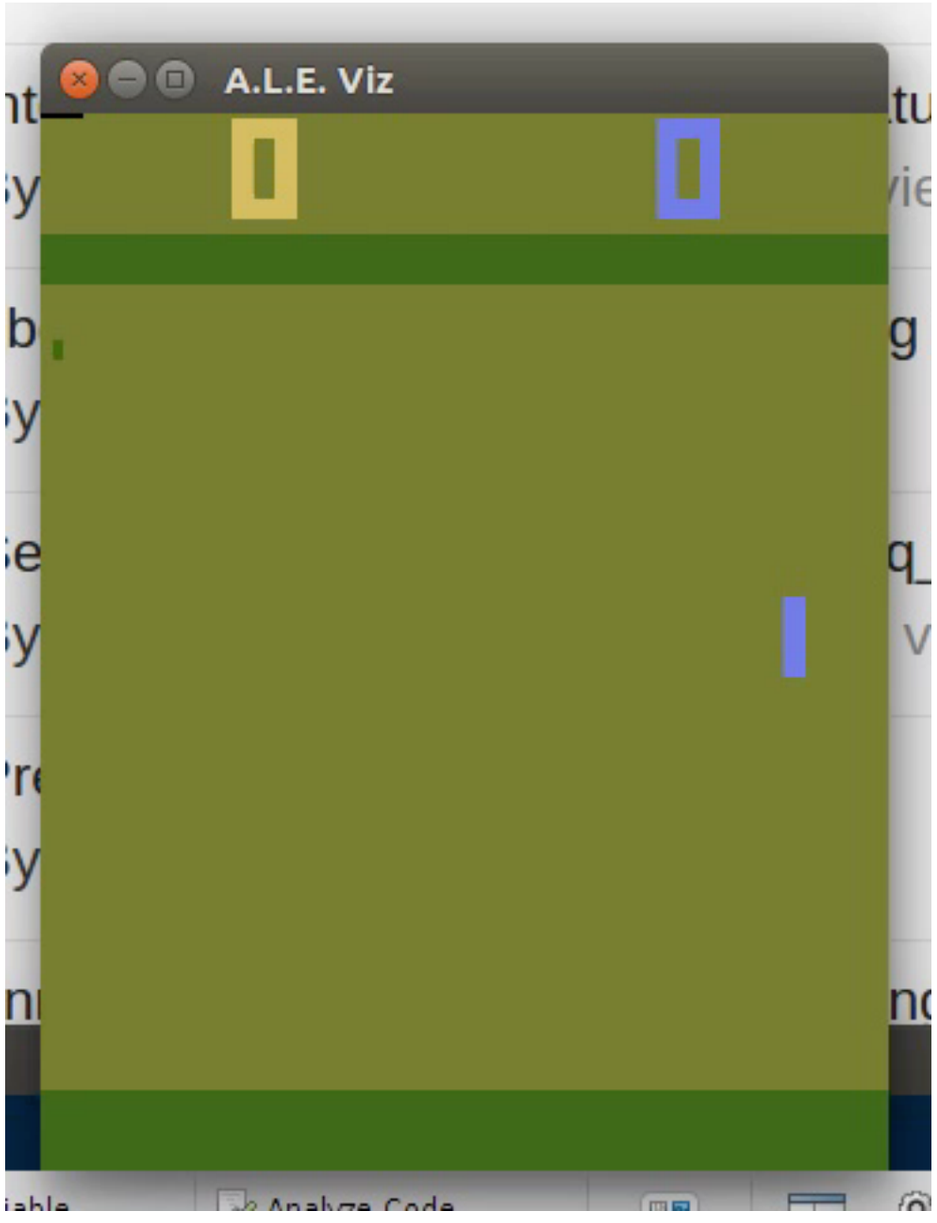
Pong: after 500,000 steps (~3 hours)



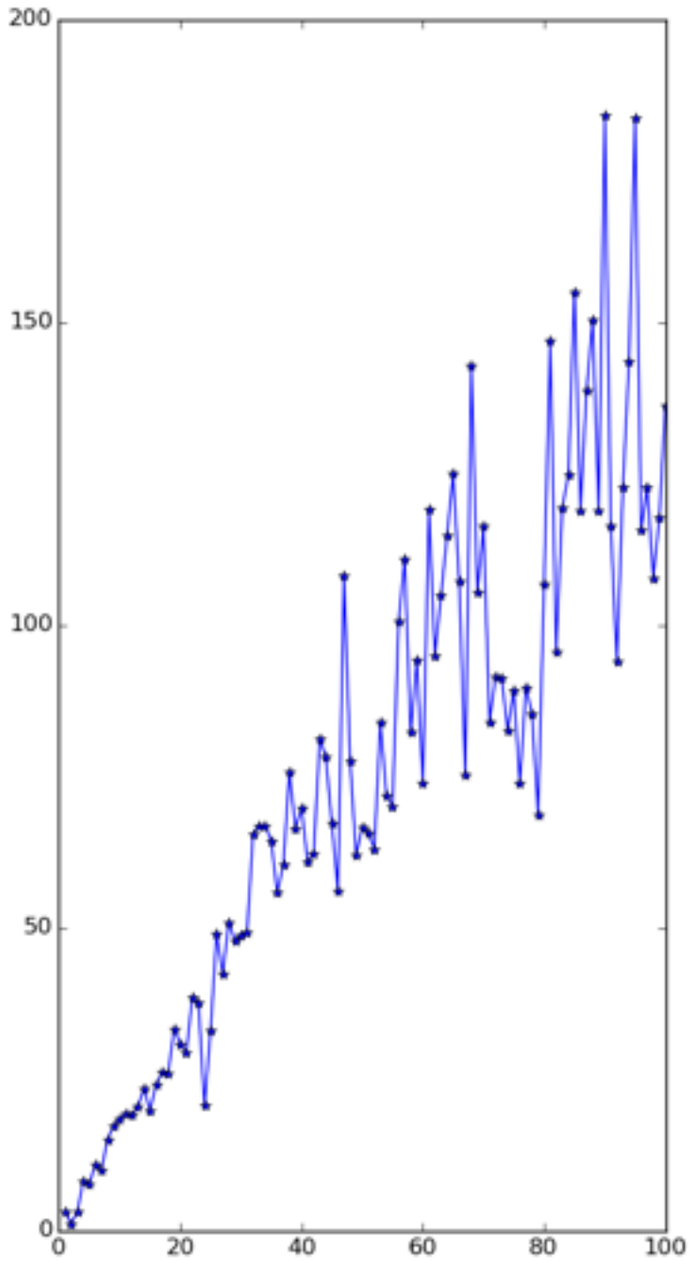
Pong: after 1,500,000 steps (~10 hours)

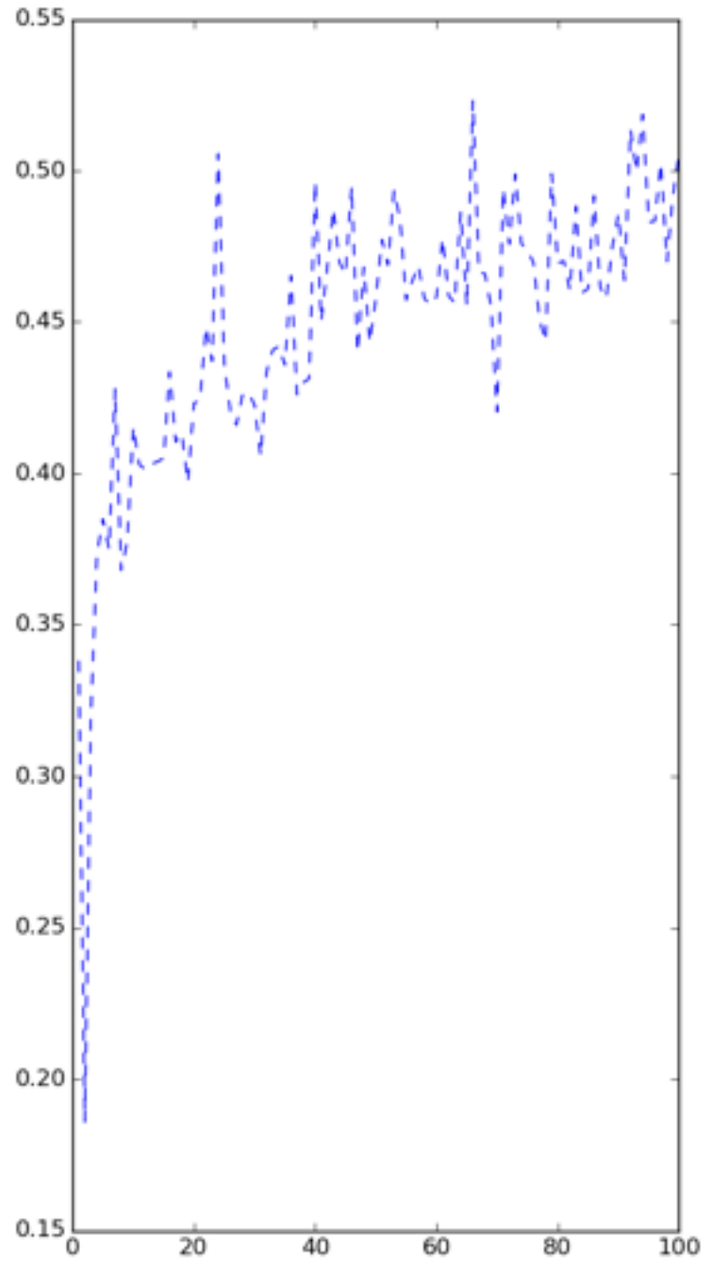


Pong: after 2,500,000 steps (~20 hours)



Results: Breakout



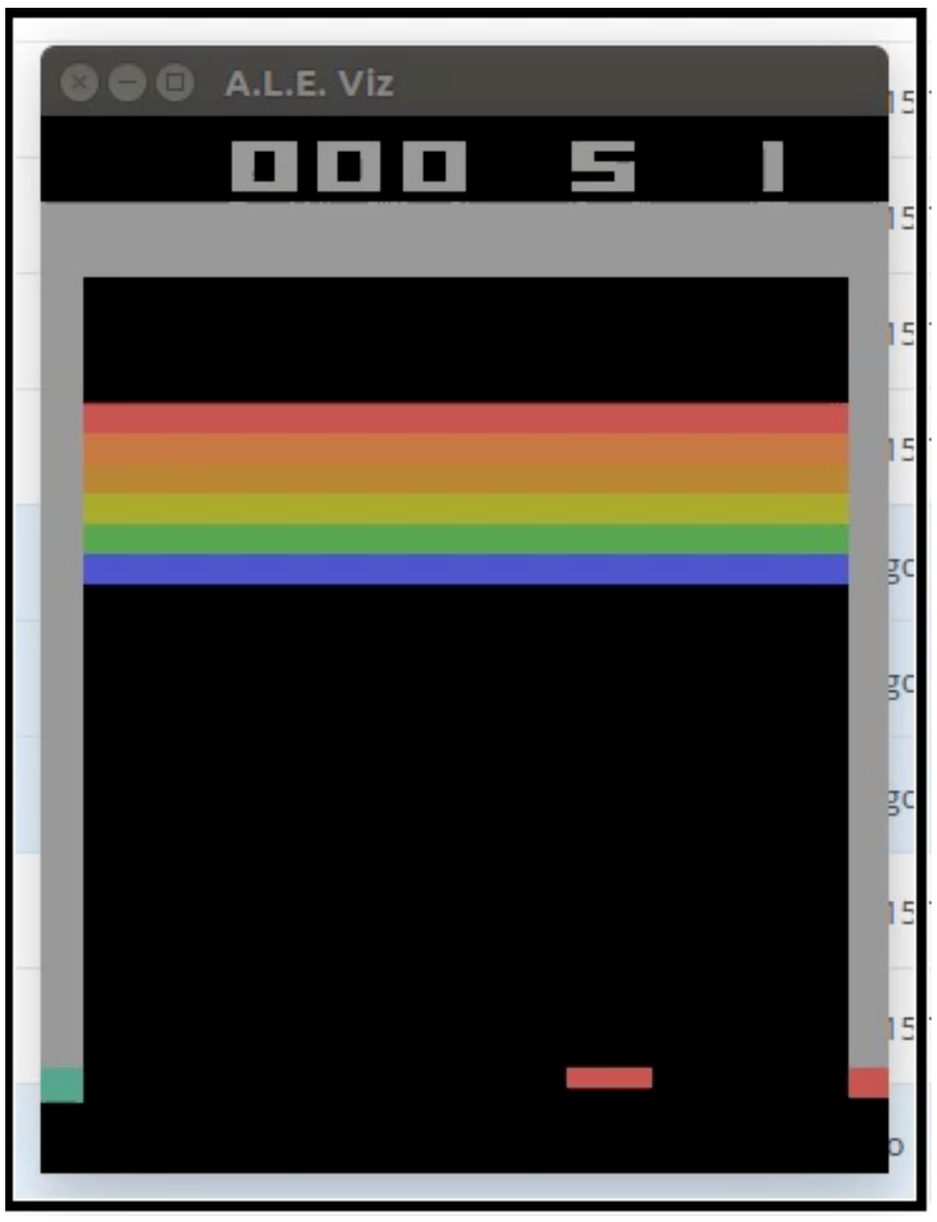


Breakout (5,000 steps, ~20 minutes)

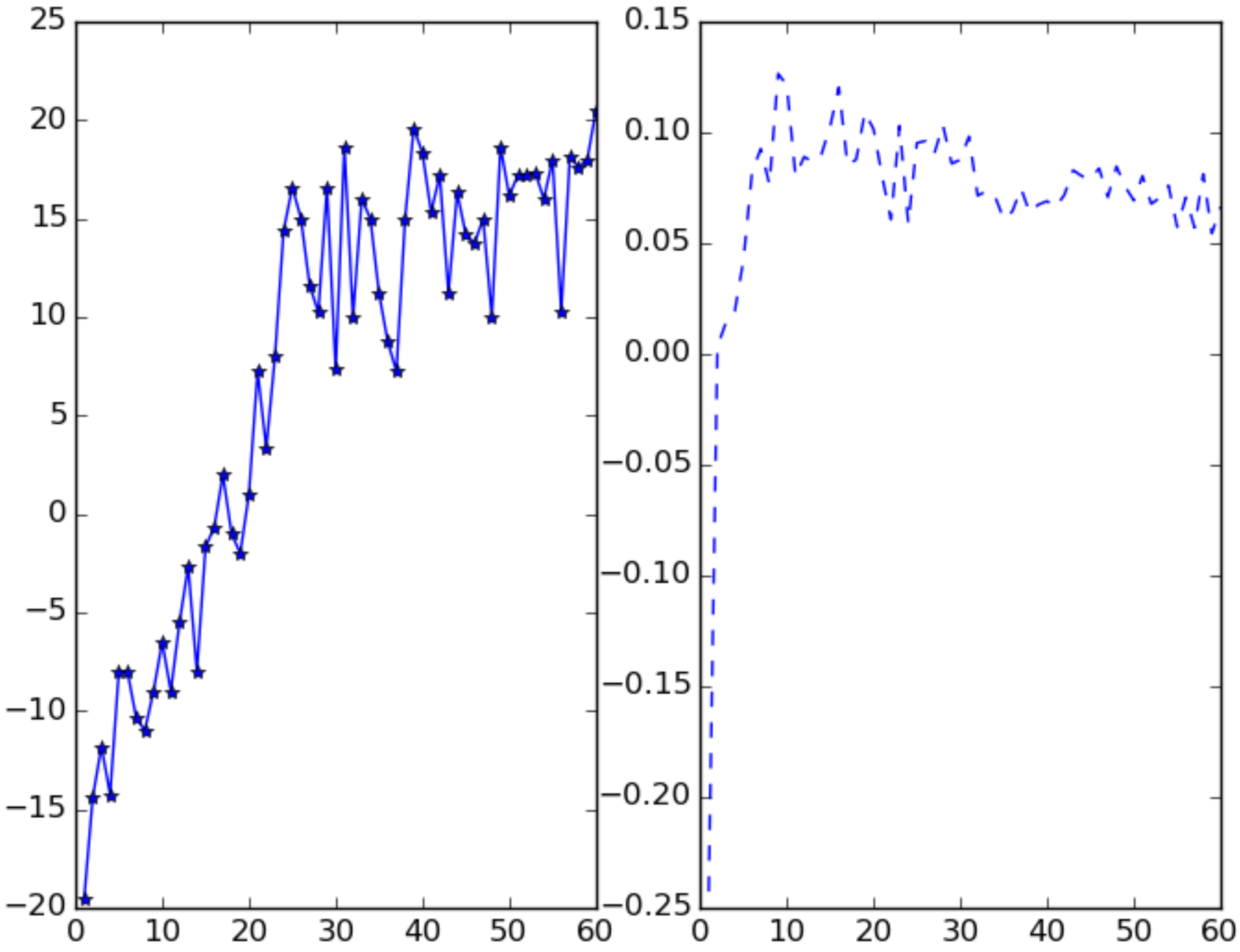
Breakout (500,000 steps, 3 hours)

Breakout (2,500,000 steps, ~16 hours)

Breakout (5,000,000 steps, ~30 hours)

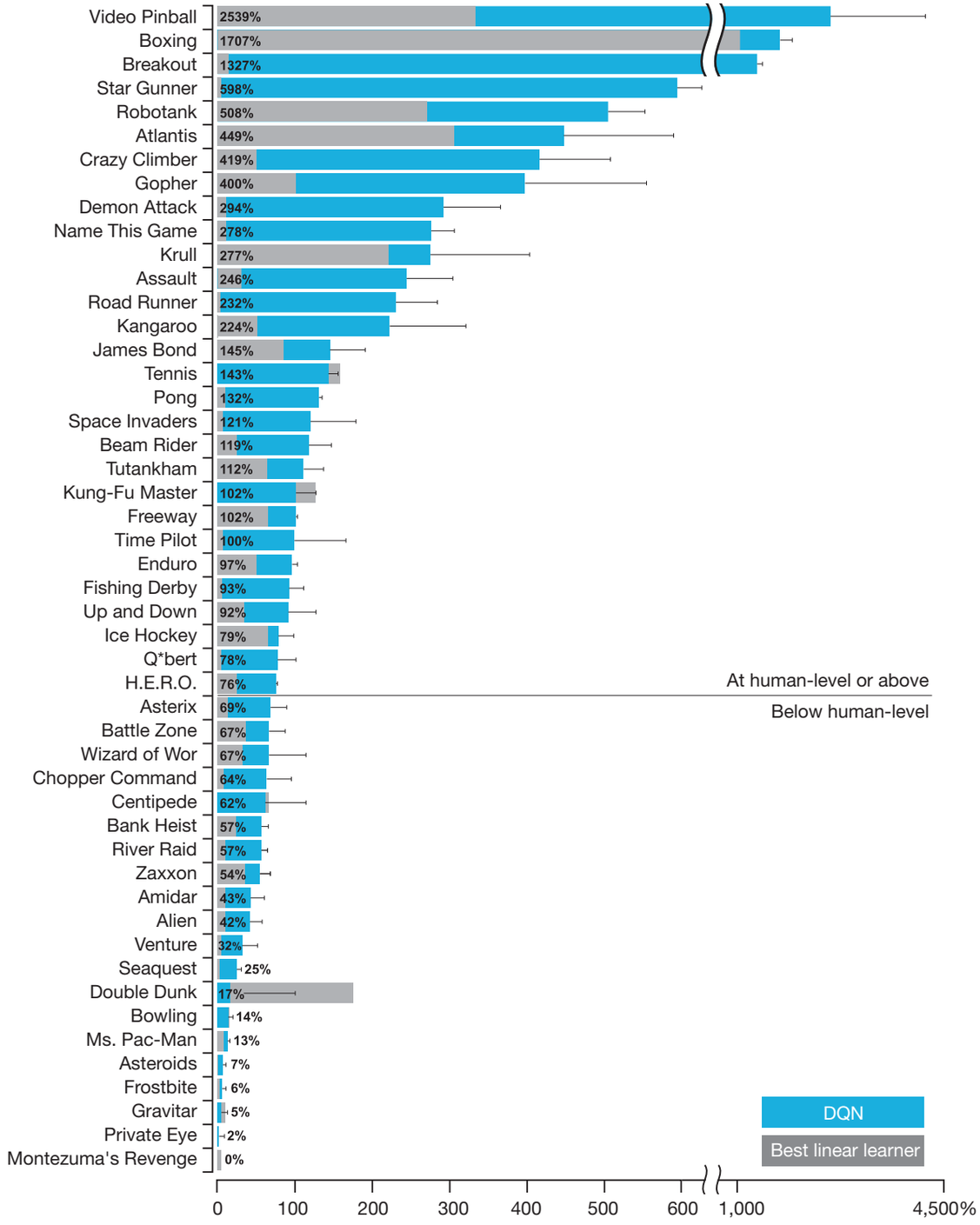


Results: Pacman



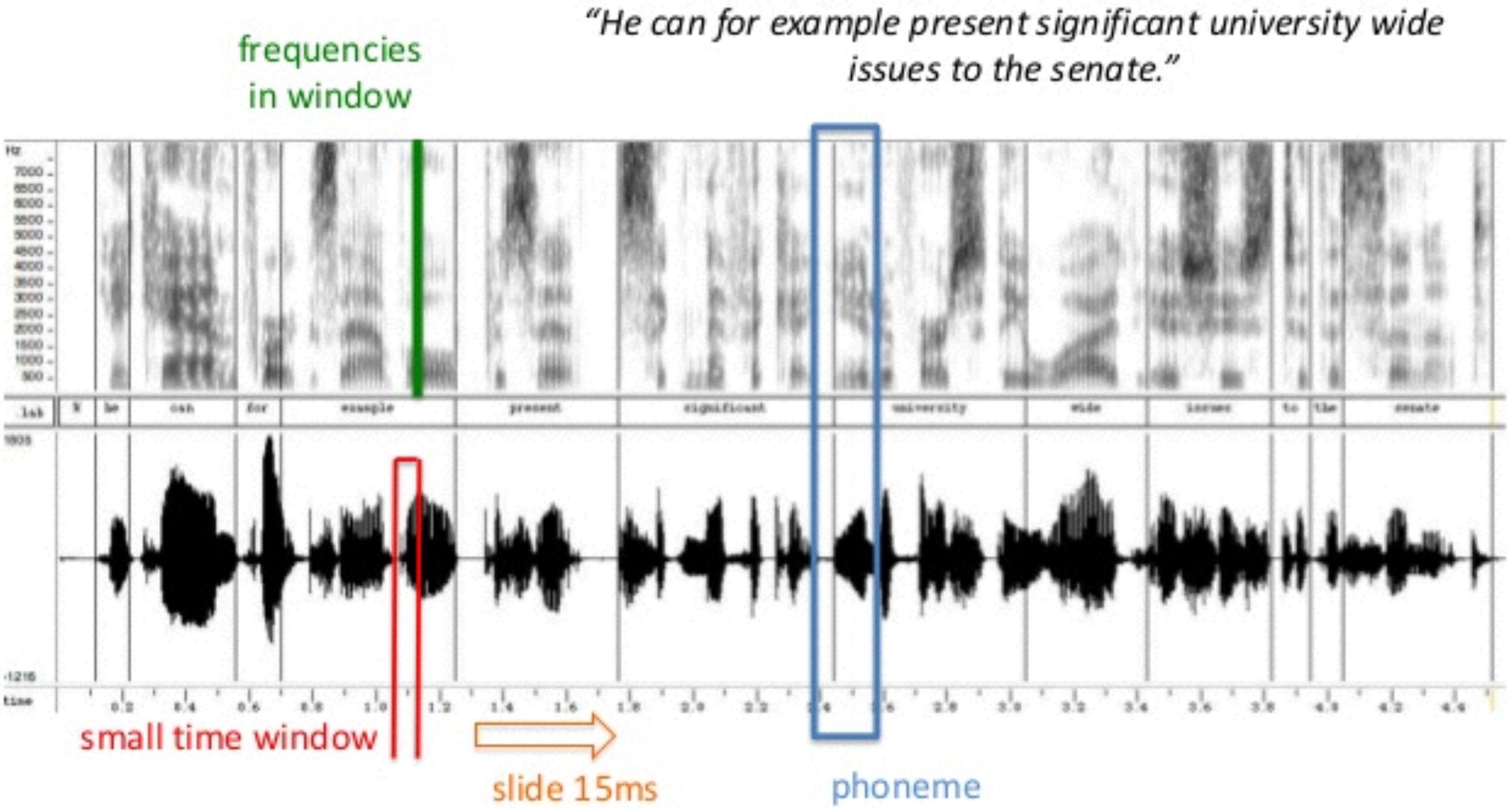
Pacman

Results



Human-Level play

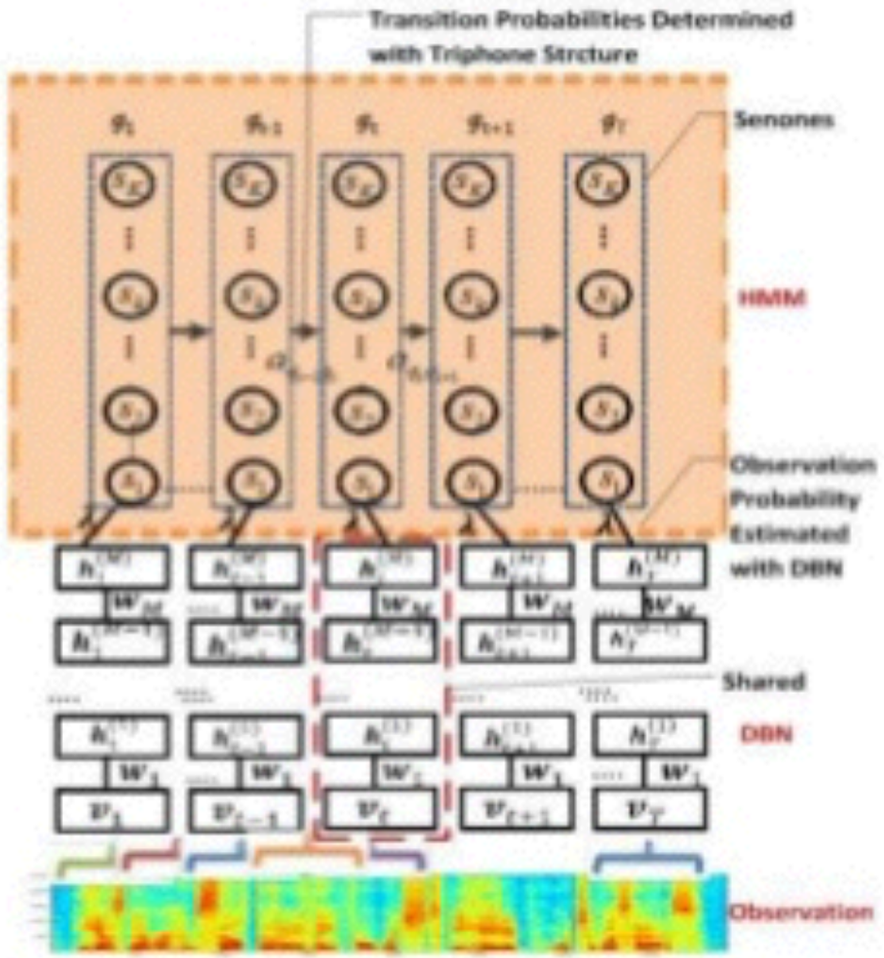
Application: Speech



Spectrogram: window in time -> vector of frequences; slide; repeat

Speech Recognition

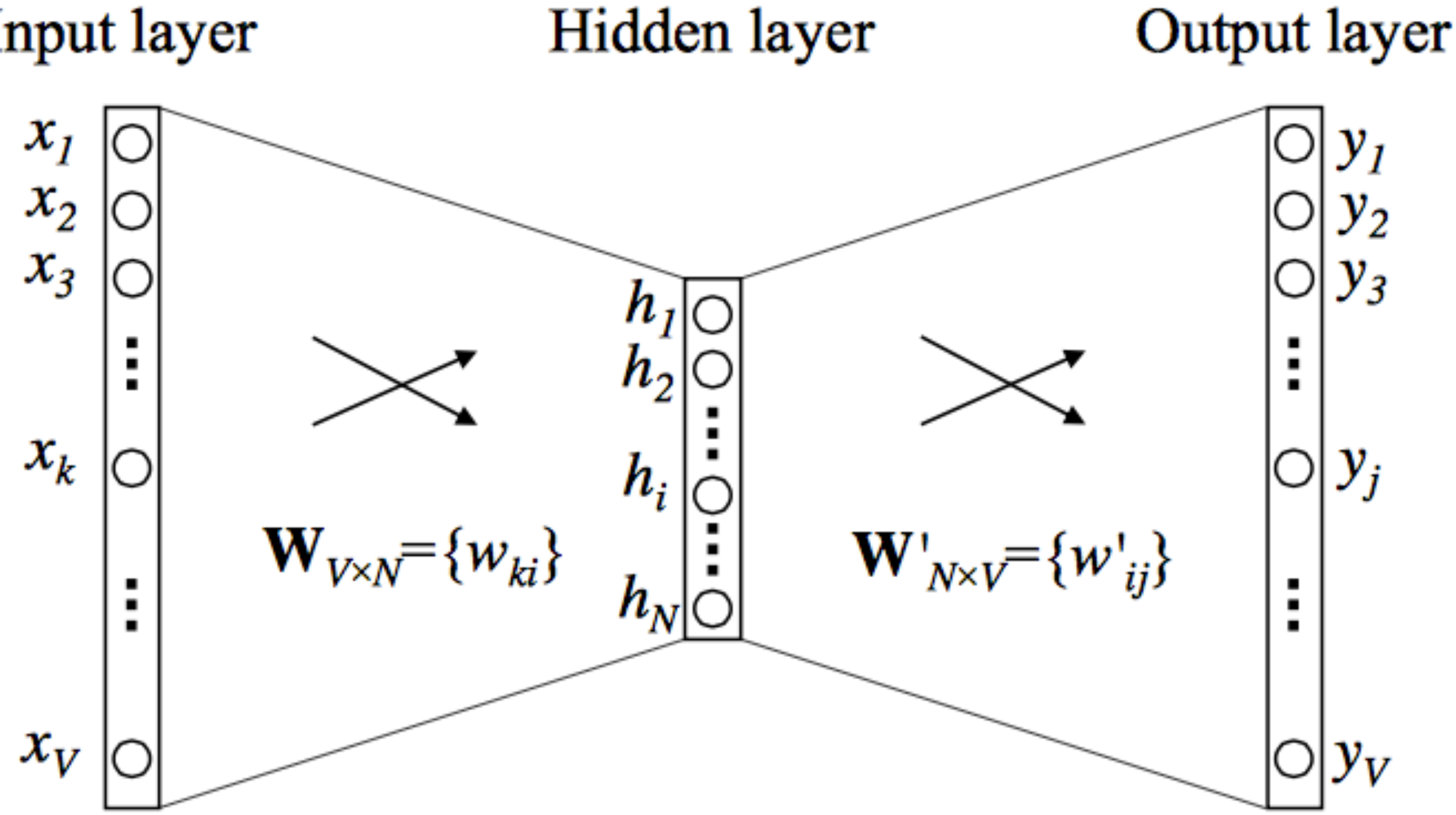
- DNN is used to replace GMM to learn state output probability in HMM.
- FF and DBN have been used for ASR
- CNN starts being used to further improve WER
- Rectified Linear Activation seems better than Sigmoid
- Models are relatively small (e.g. 5 layers, 2560 neurons/hidden layer)



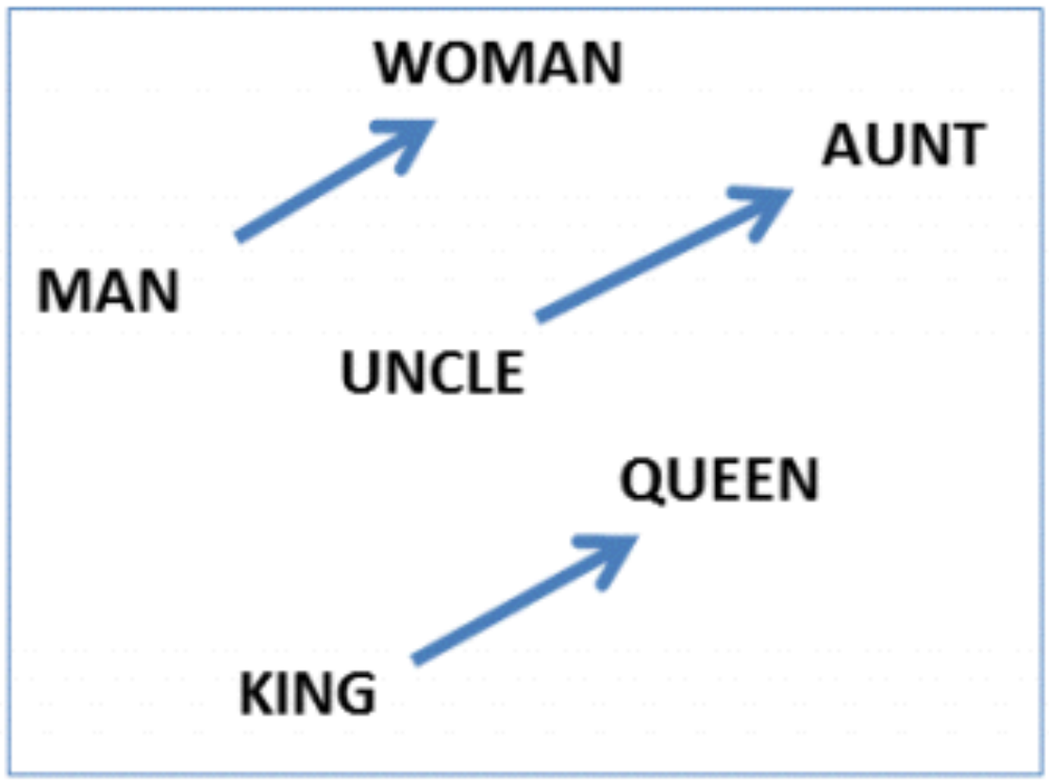
Li Deng, A Tutorial Survey of Architectures, Algorithms, and Applications for Deep Learning

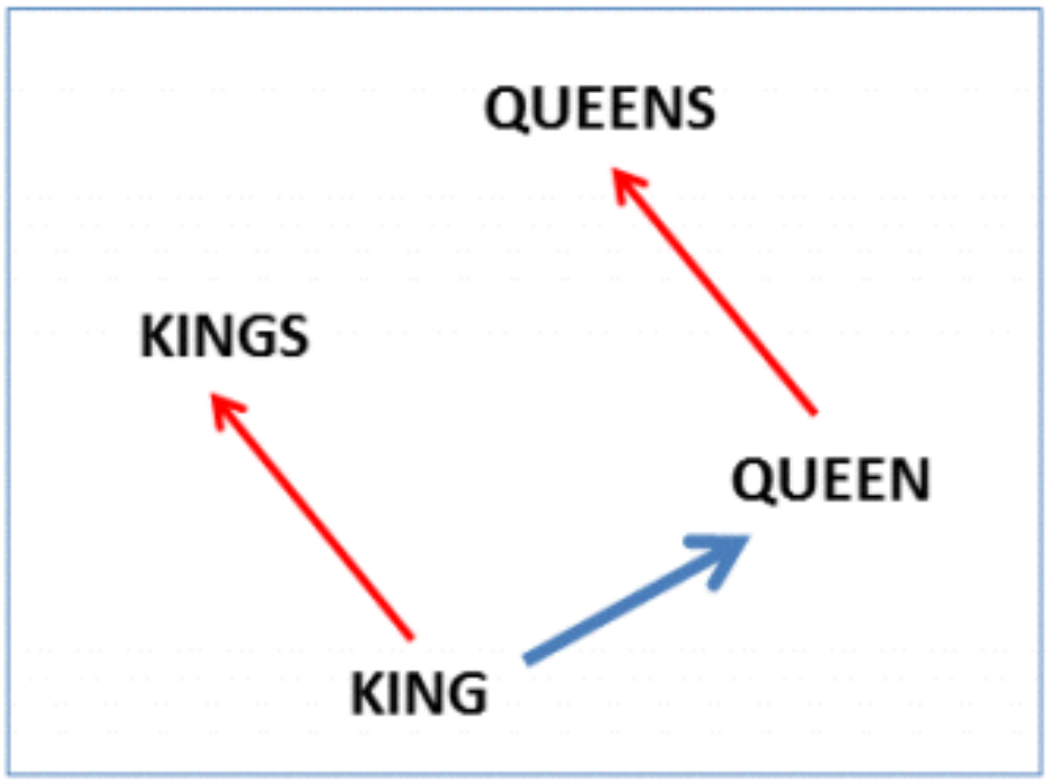
Task	GMM	DNN	Research Group
	WER (%)		
Switchboard	27.4	18.5	Microsoft
YouTube	52.3	47.6	Google
Broadcast News	17.2	14.9	IBM

Deep Learning in NLP


Input layer
Hidden layer
Output layer
x_1
x_2
x_3
x_k
x_V
h_1
h_2
h_i
h_N
y_1
y_2
y_3
y_j
y_V
$\mathbf{W}_{V\times N}=\{w_{ki}\}$
$\mathbf{W}'_{N\times V}=\{w'_{ij}\}$

“obama” —> R^{100}





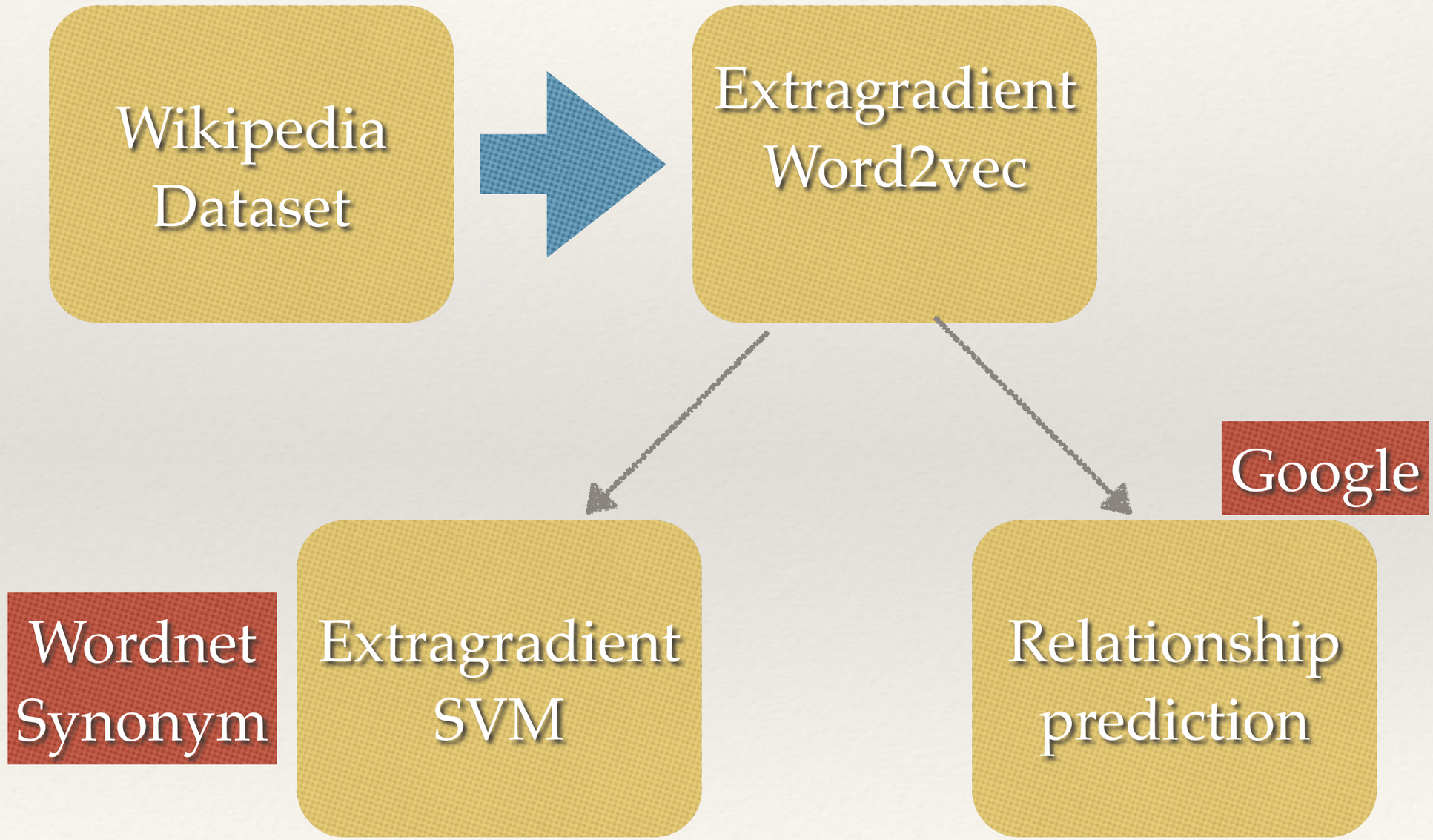
(Mikolov et al., NAACL HLT, 2013)

Problem Statement

a is to a*
as
b is to b*

Relation
capital-common-countries
capital-world
city-in-state
currency
family (gender inflections)
gram1-adjective-to-adverb
gram2-opposite
gram3-comparative
gram4-superlative
gram5-present-participle
gram6-nationality-adjective
gram7-past-tense
gram8-plural (nouns)
gram9-plural-verbs

IBM Sabbatical Project



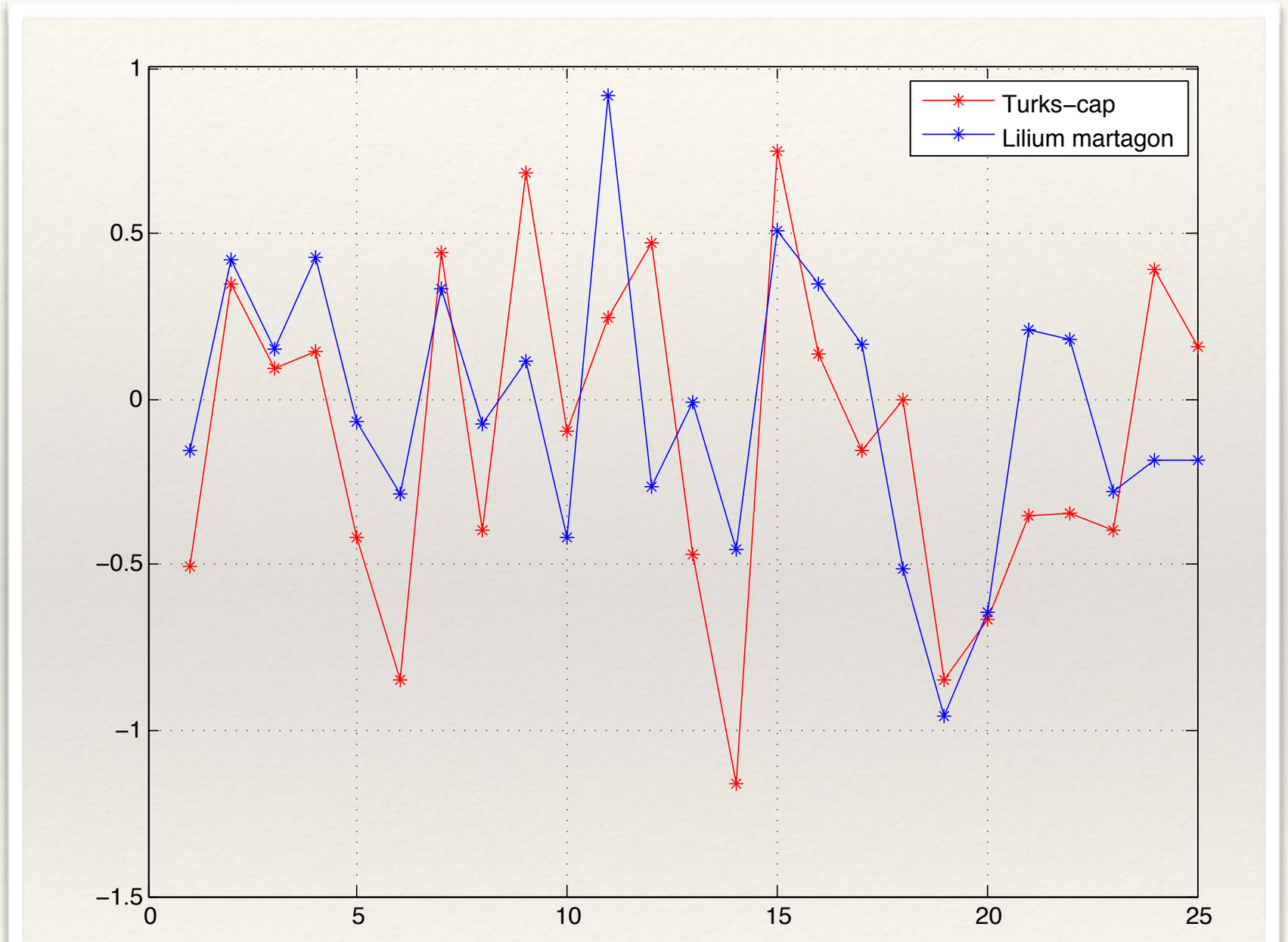
Synonym-Antonym Task

- Training data set: ~87,000 synonym antonym pairs from Wordnet
- Equal distribution of synonyms and antonyms
- Binary classification problem

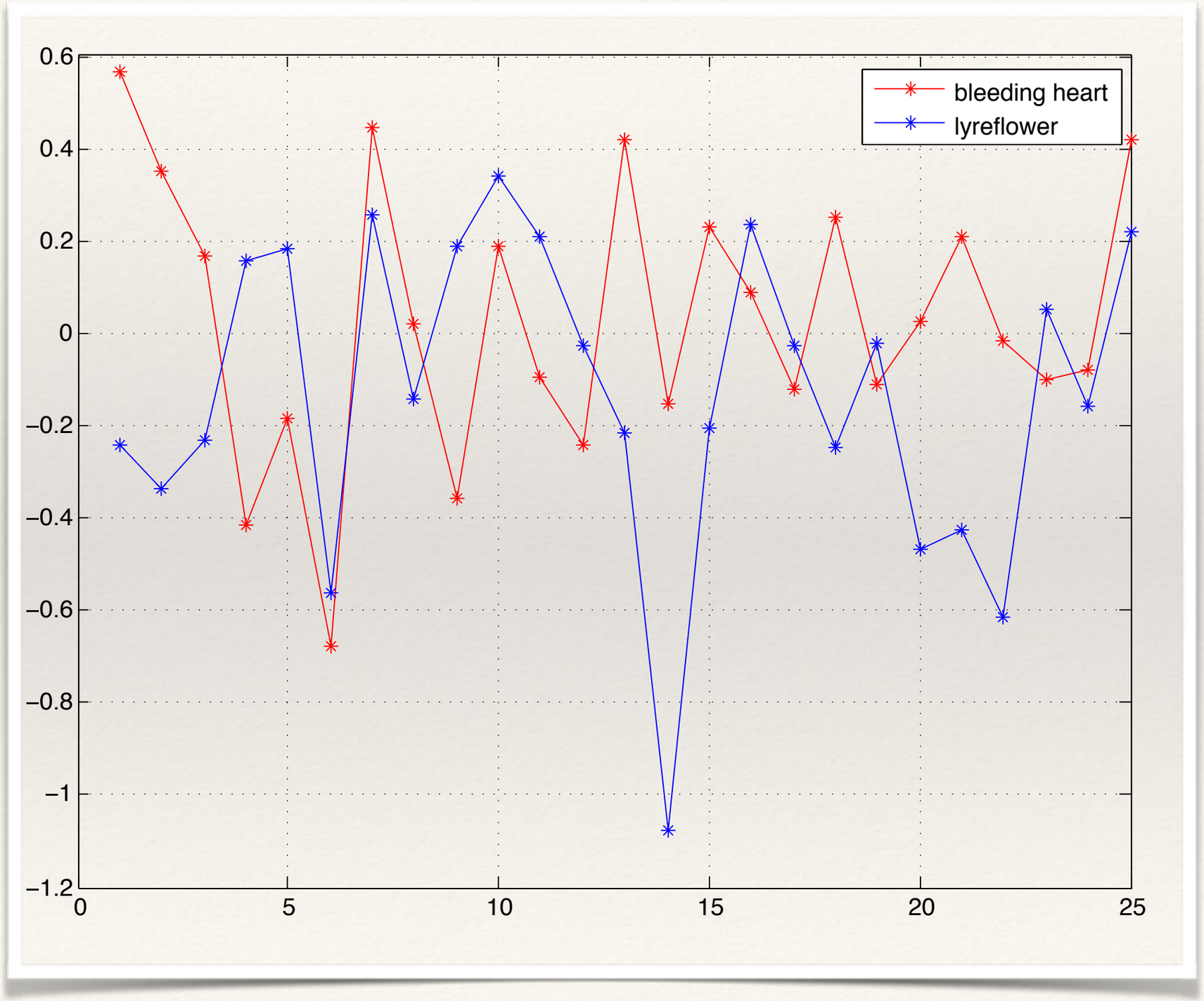
Synonym-Antonym Task

- Modified word2vec to use extragradient update
- Trained on 3 billion Wikipedia corpus (4 million words)
- Generated feature vectors of varying dimensionality
- Number of synonyms and antonyms split equally among 87,000 word pairs

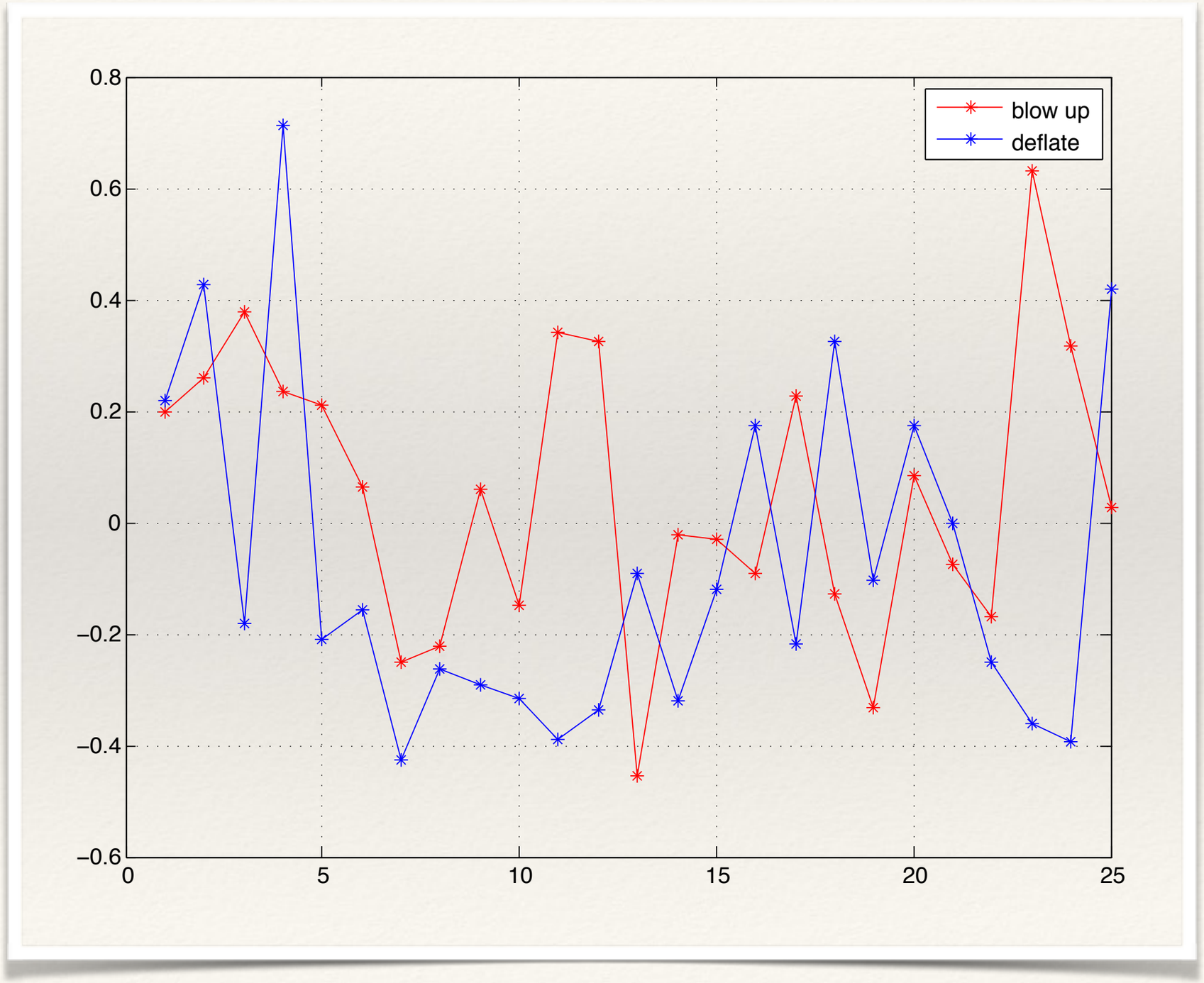
Synonym or Antonym?



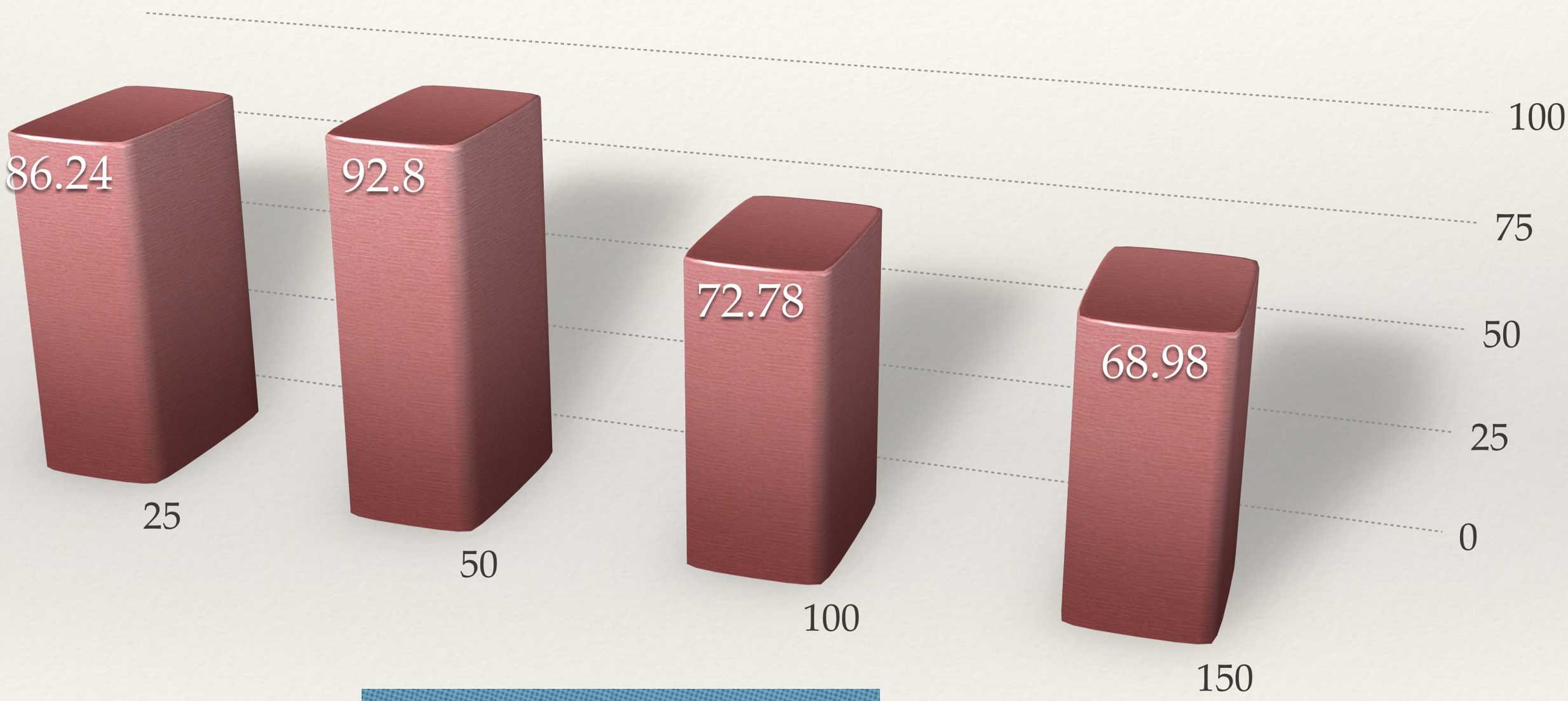
Synonym or Antonym?



Synonym or Antonym?



Classification Results



Results using liblinear

Sample False Negatives

Moellers glossitis	glossodynia exfoliativa
major-domo	seneschal
coral-wood	peacock flower fence
binge-vomit syndrome	bulima nervosa
taximan	livery driver

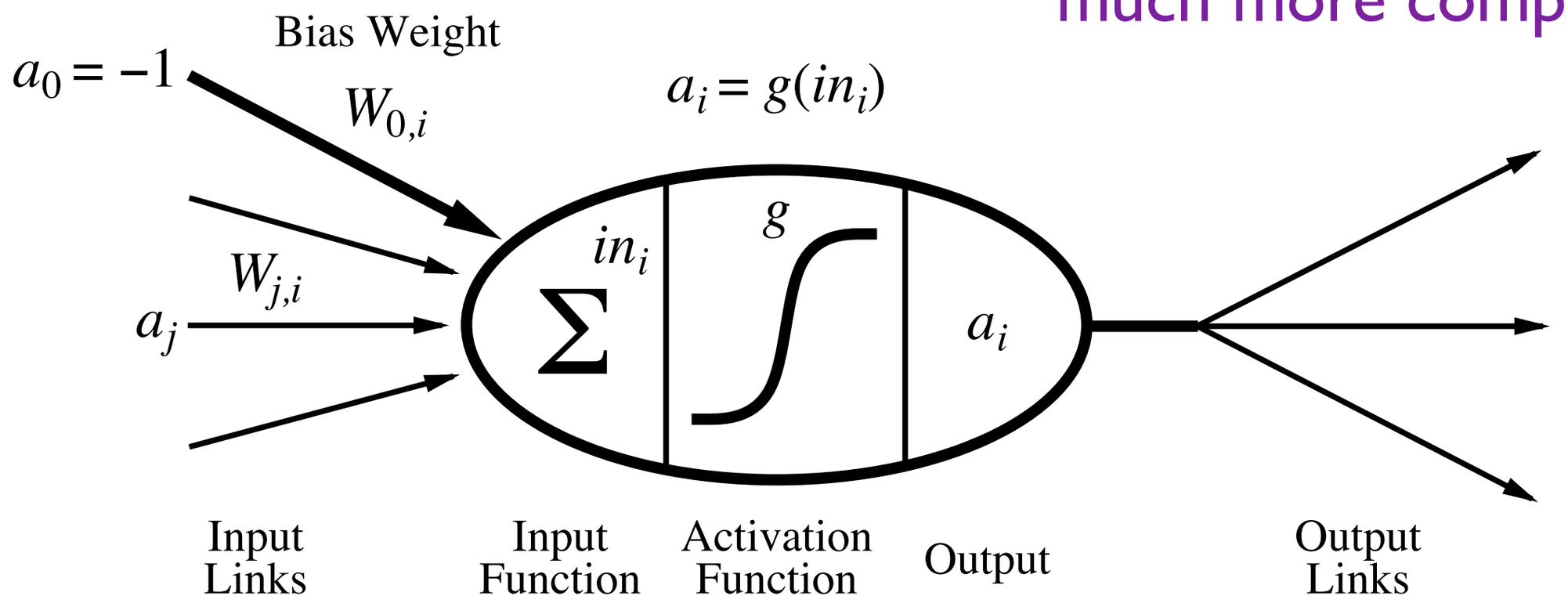
Quick Overview of Neural Networks

Simple Model of Neuron

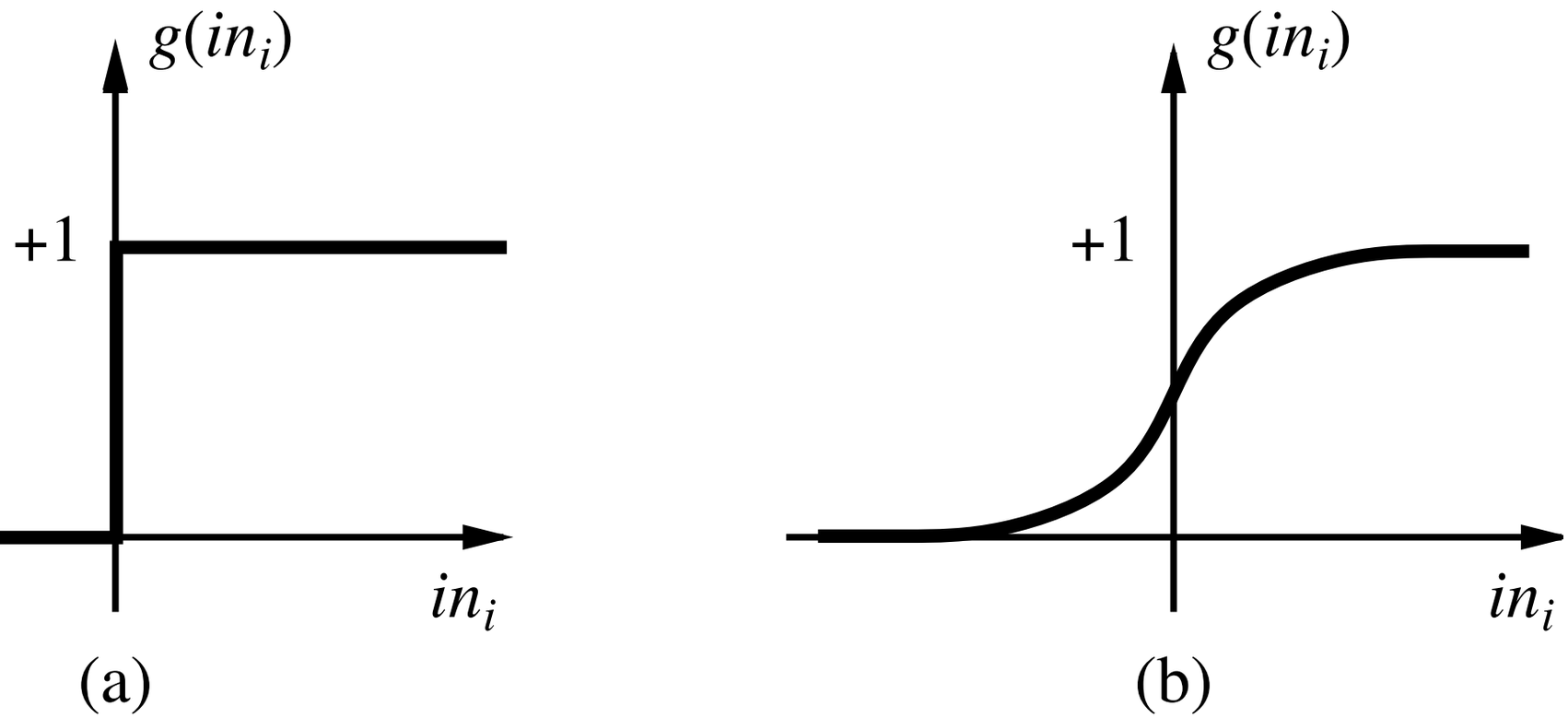
Output is a "squashed" linear function of the inputs:

$$a_i \leftarrow g(in_i) = g\left(\Sigma_j W_{j,i} a_j\right)$$

Real neurons are much more complex!



Activation Function

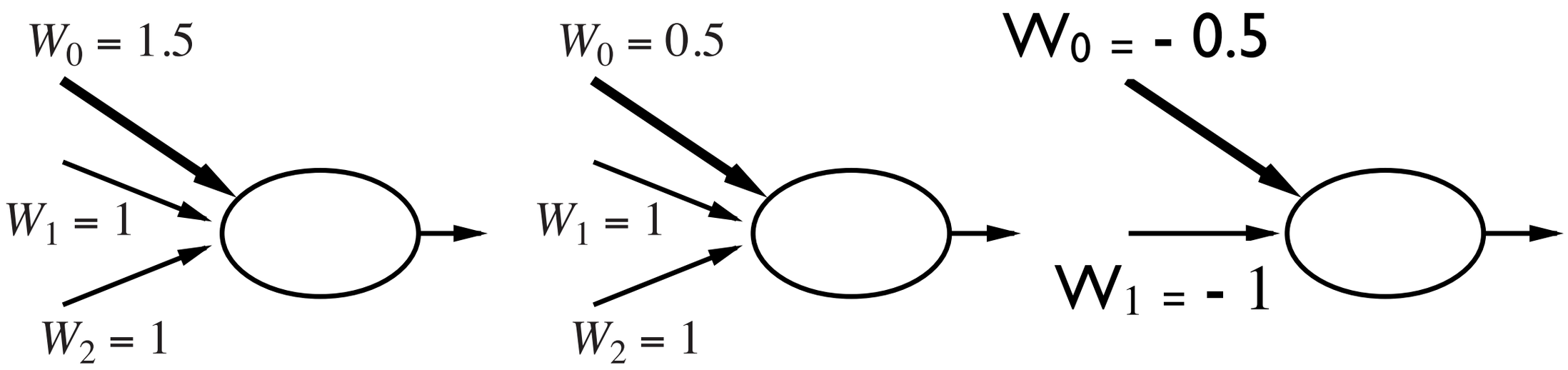


(a) is a step function or threshold function

(b) is a sigmoid function $1/(1+e^{-x})$

Changing the bias weight $W_{0,i}$ moves the threshold location

Boolean Functions



AND **OR** **NOT**

McCulloch and Pitts: every Boolean function can be implemented

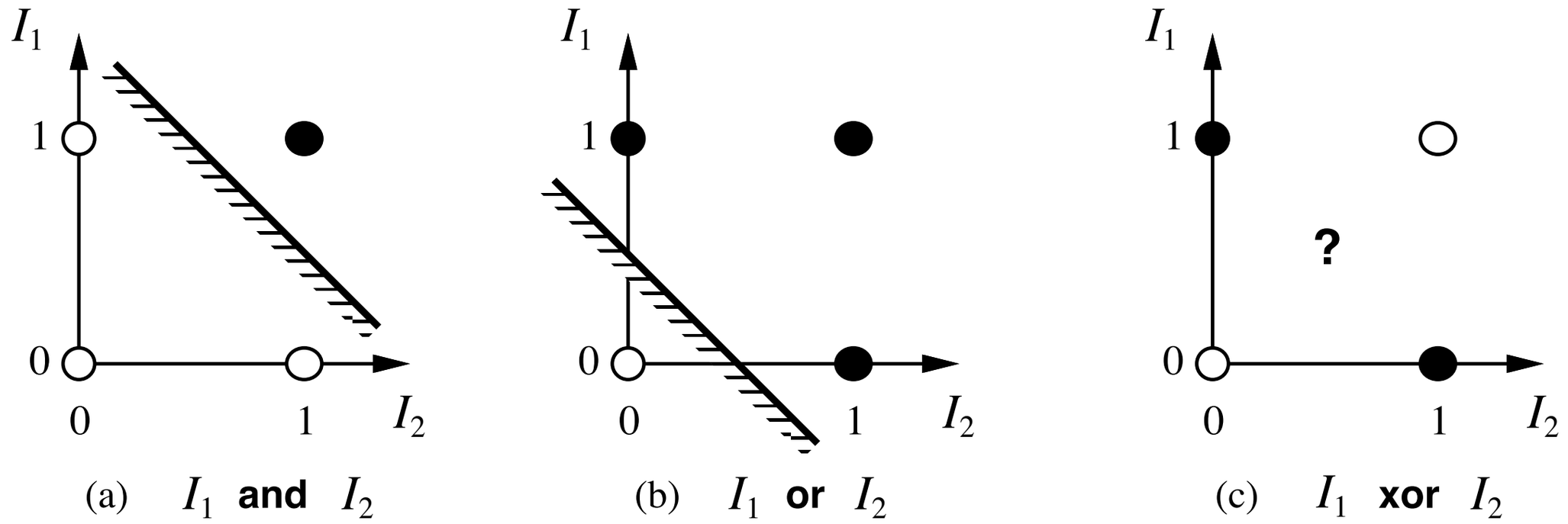
Perceptrons are limited

Consider a perceptron with $g =$ step function (Rosenblatt, 1957, 1960)

Can represent AND, OR, NOT, majority, etc.

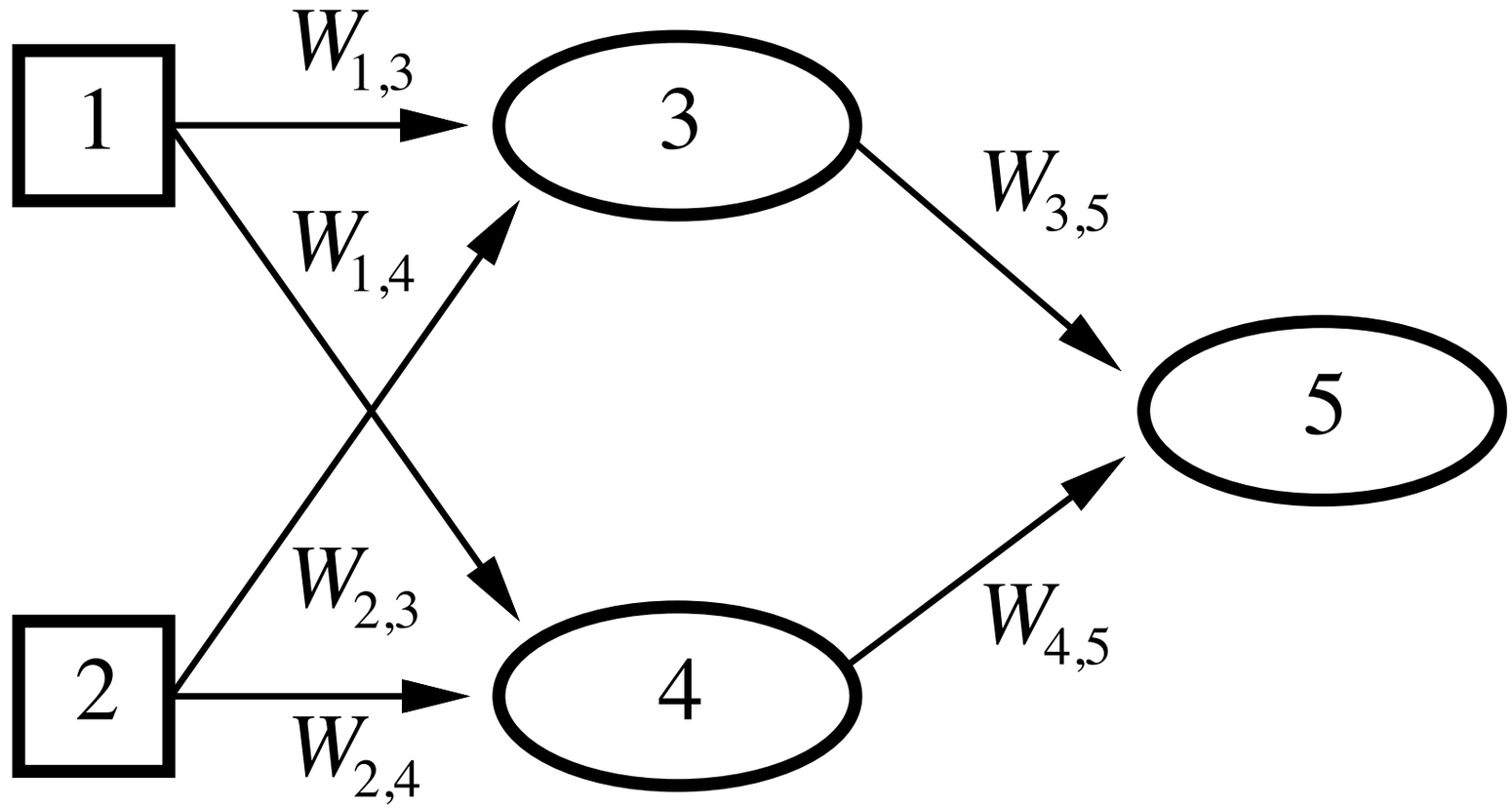
Represents a linear separator in input space:

$$\Sigma_j W_j x_j > 0 \quad \text{or} \quad \mathbf{W} \cdot \mathbf{x} > 0$$



(a) I_1 **and** I_2 (b) I_1 **or** I_2 (c) I_1 **xor** I_2

Feedforward Networks



Feed-forward network = a parameterized family of nonlinear functions:

$$\begin{aligned} a_5 &= g(W_{3,5} \cdot a_3 + W_{4,5} \cdot a_4) \\ &= g(W_{3,5} \cdot g(W_{1,3} \cdot a_1 + W_{2,3} \cdot a_2) + W_{4,5} \cdot g(W_{1,4} \cdot a_1 + W_{2,4} \cdot a_2)) \end{aligned}$$

Gradient Learning Rule

Learn by adjusting weights to reduce error on training set

The squared error for an example with input $\mathbf{x}$ and true output y is

$$E = \frac{1}{2}Err^2 \equiv \frac{1}{2}(y - h_{\mathbf{W}}(\mathbf{x}))^2 \ ,$$

Perform optimization search by gradient descent:

$$\begin{aligned}\frac{\partial E}{\partial W_j} &= Err \times \frac{\partial Err}{\partial W_j} = Err \times \frac{\partial}{\partial W_j}\left(y - g(\Sigma_{j=0}^{n} W_j x_j)\right) \\ &= -Err \times g'(in) \times x_j\end{aligned}$$

Simple weight update rule:

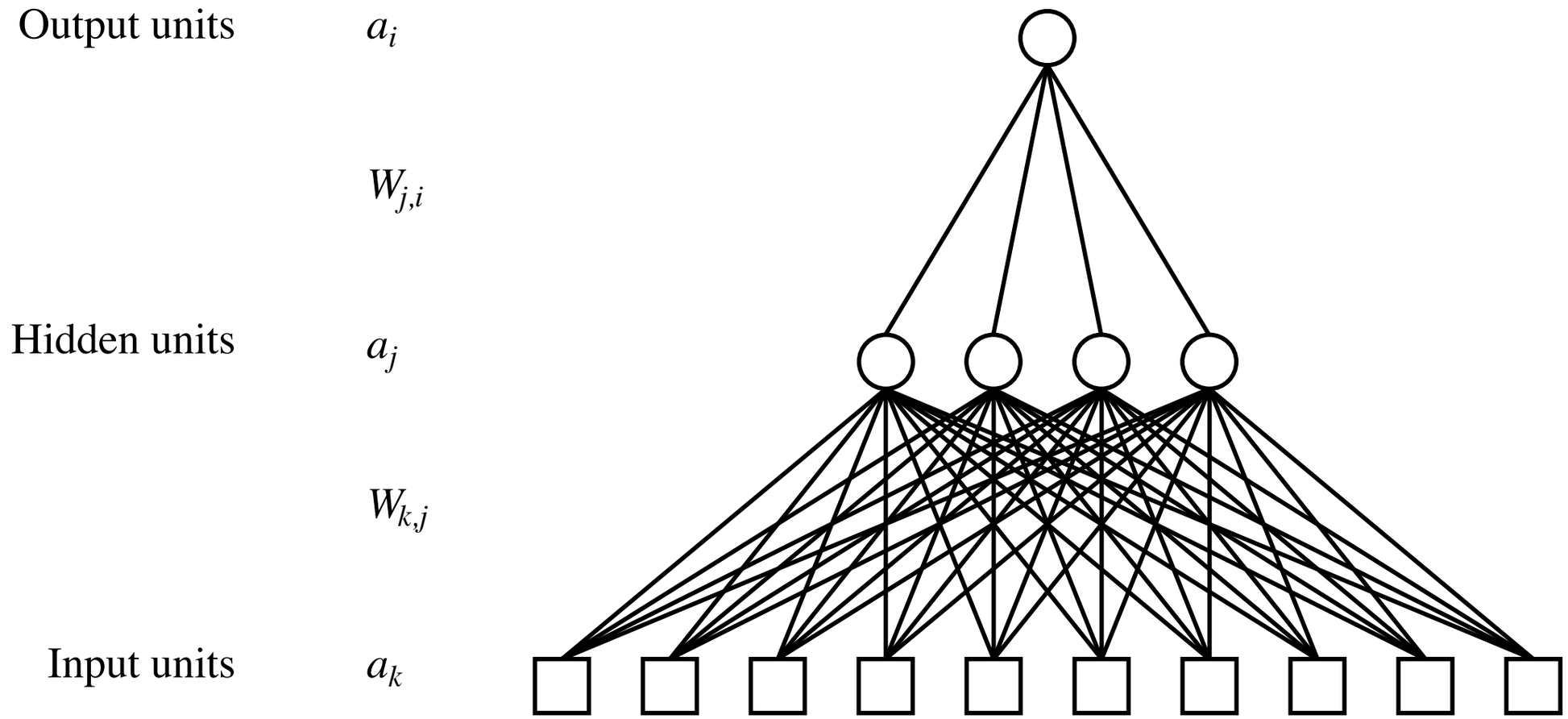
$$W_j \leftarrow W_j + \alpha \times Err \times g'(in) \times x_j$$

E.g., +ve error $\Rightarrow$ increase network output

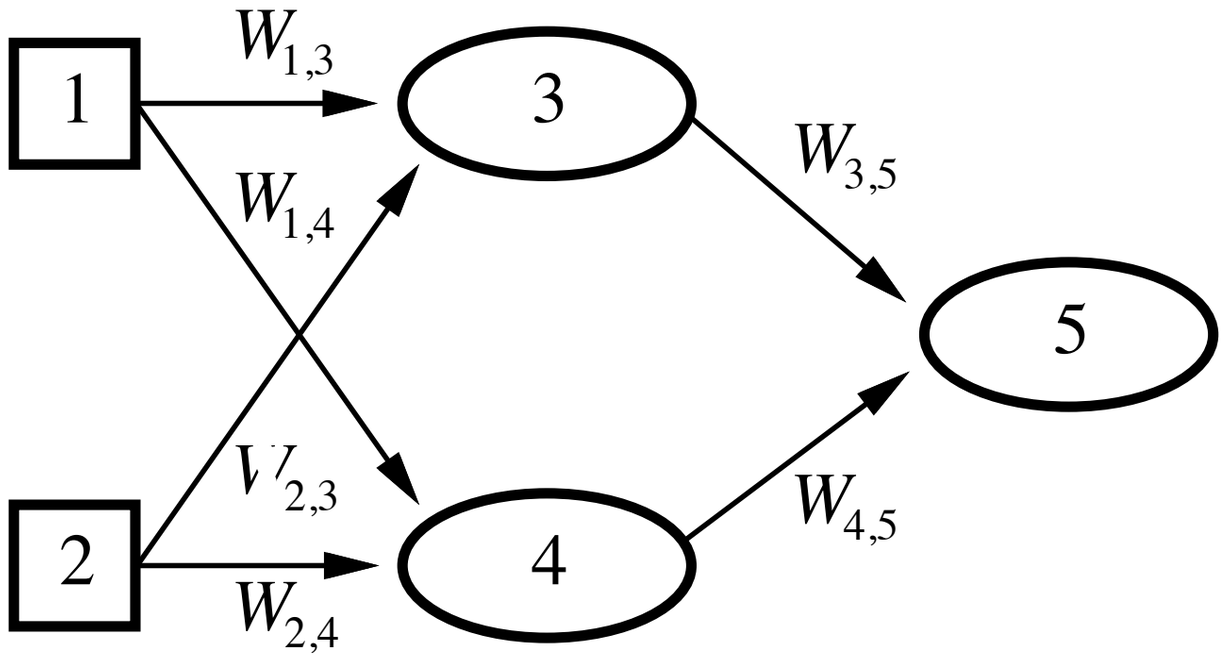
$\Rightarrow$ increase weights on +ve inputs, decrease on -ve inputs

Multilayer Perceptrons

Layers are usually fully connected;
numbers of hidden units typically chosen by hand

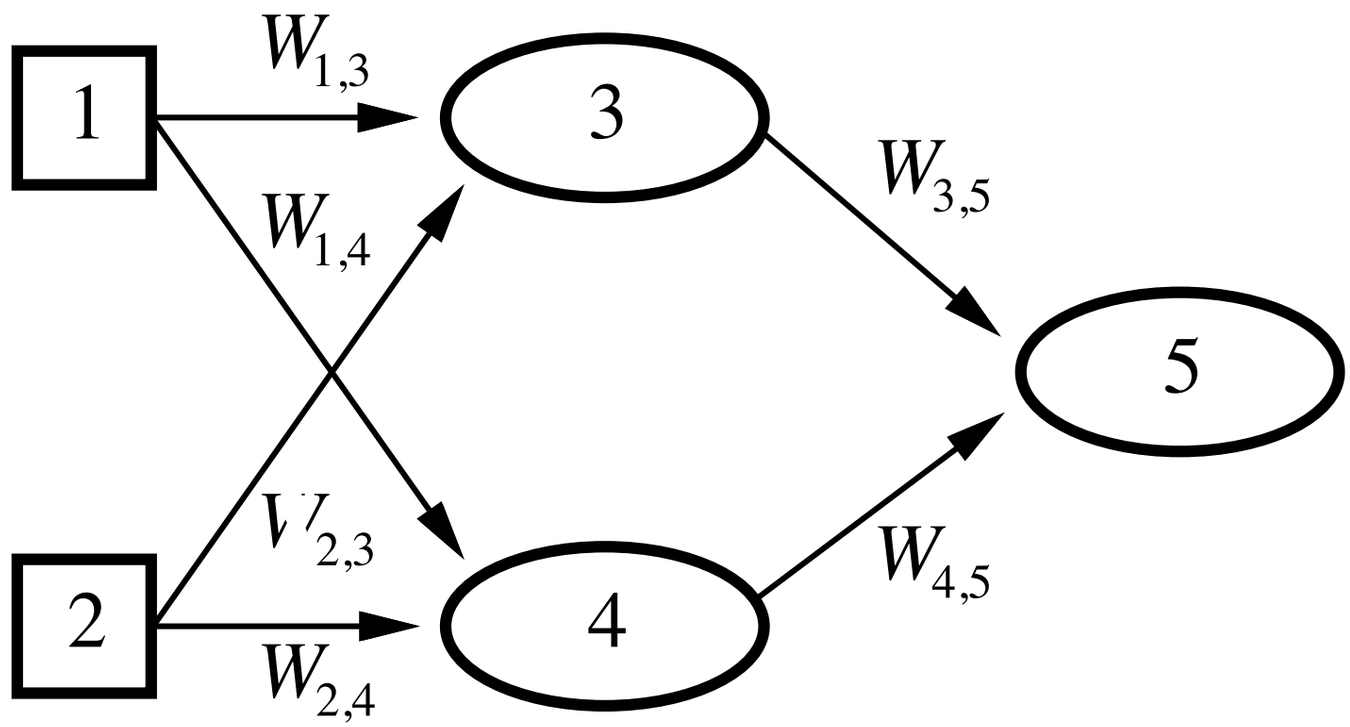


What's hard about training feedforward networks?



There are training signals for the output and input layers. But, what are the hidden nodes supposed to compute?

Backpropagation



Forward propagation: compute activation levels of each unit on a particular input

Backpropagation: compute errors

Gradient Training Rule

The squared error on a single example is defined as

$$E = \frac{1}{2}\sum_i (y_i - a_i)^2 \ ,$$

where the sum is over the nodes in the output layer.

$$\begin{aligned}
\frac{\partial E}{\partial W_{j,i}} &= -(y_i - a_i)\frac{\partial a_i}{\partial W_{j,i}} = -(y_i - a_i)\frac{\partial g(in_i)}{\partial W_{j,i}} \\
&= -(y_i - a_i)g'(in_i)\frac{\partial in_i}{\partial W_{j,i}} = -(y_i - a_i)g'(in_i)\frac{\partial}{\partial W_{j,i}}\left(\sum_j W_{j,i}a_j\right) \\
&= -(y_i - a_i)g'(in_i)a_j = -a_j\Delta_i
\end{aligned}$$

Hidden Units

$$
\begin{aligned}
\frac{\partial E}{\partial W_{k,j}} &= -\sum_i (y_i - a_i)\frac{\partial a_i}{\partial W_{k,j}} = -\sum_i (y_i - a_i)\frac{\partial g(in_i)}{\partial W_{k,j}} \\
&= -\sum_i (y_i - a_i) g'(in_i)\frac{\partial in_i}{\partial W_{k,j}} = -\sum_i \Delta_i \frac{\partial}{\partial W_{k,j}}\left(\sum_j W_{j,i} a_j\right) \\
&= -\sum_i \Delta_i W_{j,i}\frac{\partial a_j}{\partial W_{k,j}} = -\sum_i \Delta_i W_{j,i}\frac{\partial g(in_j)}{\partial W_{k,j}} \\
&= -\sum_i \Delta_i W_{j,i} g'(in_j)\frac{\partial in_j}{\partial W_{k,j}} \\
&= -\sum_i \Delta_i W_{j,i} g'(in_j)\frac{\partial}{\partial W_{k,j}}\left(\sum_k W_{k,j} a_k\right) \\
&= -\sum_i \Delta_i W_{j,i} g'(in_j) a_k = -a_k \Delta_j
\end{aligned}
$$

Backpropagation Algorithm

- Given: training examples $\{(x_i,y_i)\}$, network
- Randomly set initial weights of network
- Repeat
  - For each training example
    - Compute error beginning with output units, and then for each hidden layer of units
    - Adjust weights in direction of lower error
- Until error is acceptable

Backpropagation Algorithm

- Initialize weights to small random values
- REPEAT
  - For each training example:
    - FORWARD PROPAGATION: Fix network inputs using training example and compute network outputs
    - BACKPROPAGATION:
- For output unit k, compute delta value $\Delta_k = a_k (1-a_k)(t_k - a_k)$
- Compute delta values of hidden units

$$\Delta_h = a_h (1 - a_h) \Sigma_k W_{hk} \Delta_k$$

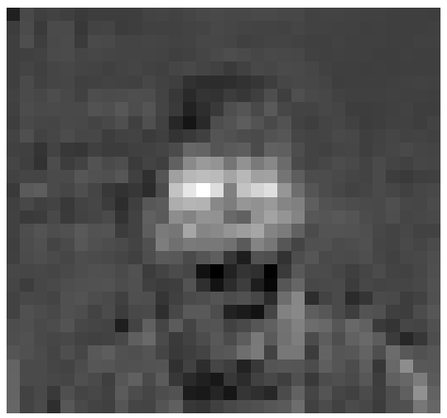
- Update each network weight

$$W_{ij} = W_{ij} + \eta a_i \Delta_j$$

Facial Pose Detection

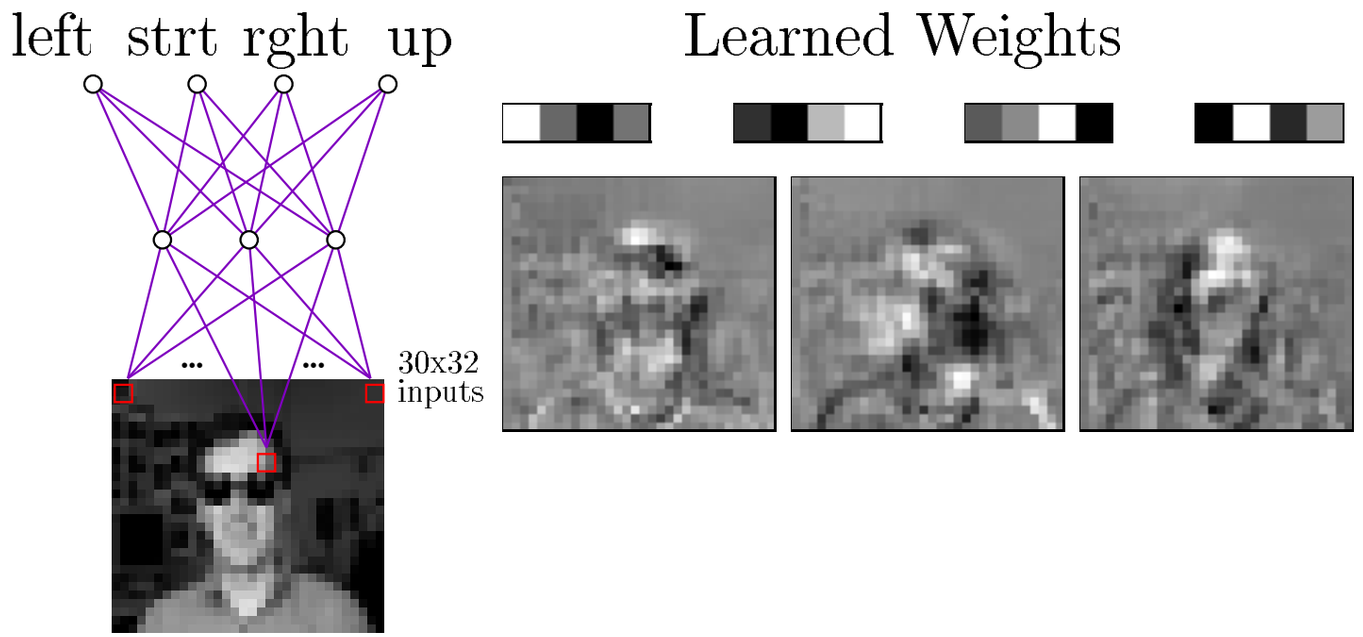
Tom Mitchell (CMU)

"Hinton" diagram (showing activation of hidden units)



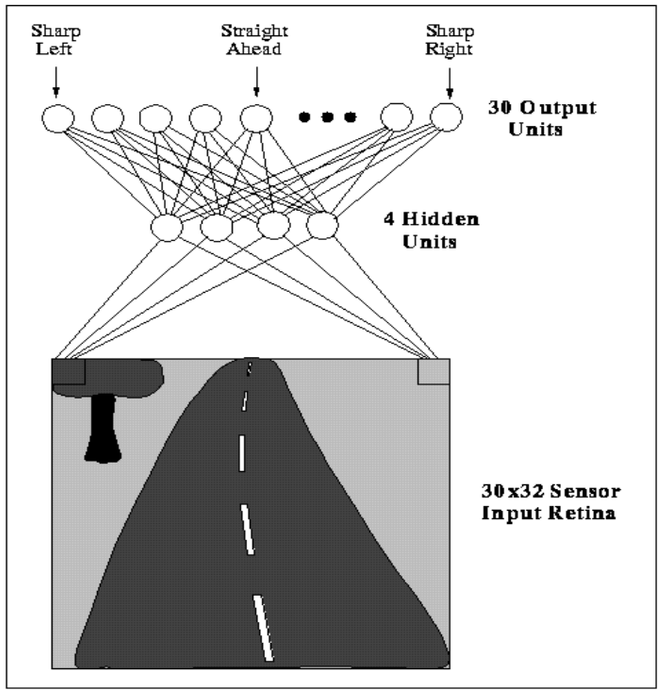
"Sunglass detector"

Hidden Unit Detectors



ALVINN

ALVINN learns
from a human driver



Neural
Network

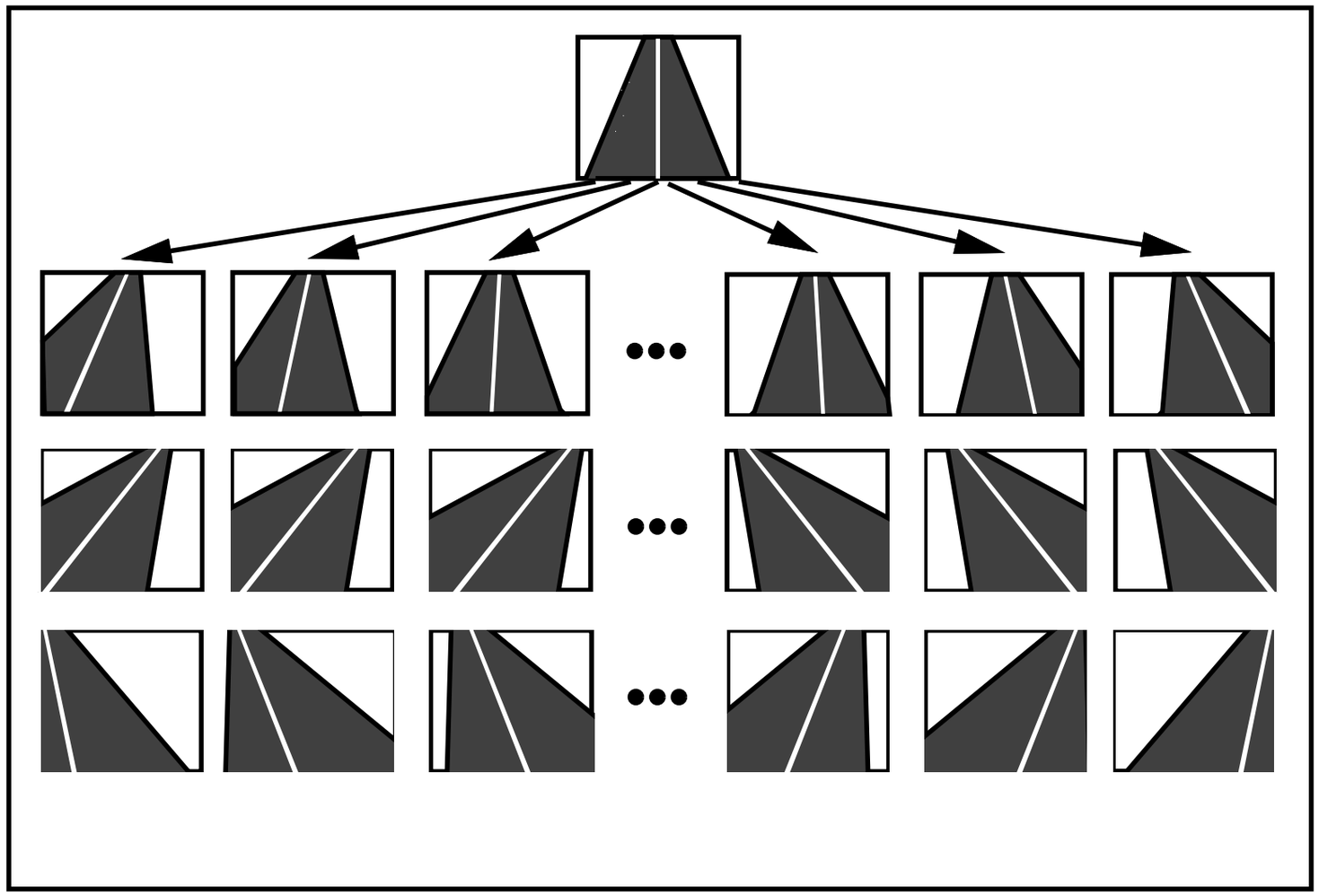
Can drive on actual highways at 70 miles per hour!

ALVINN training

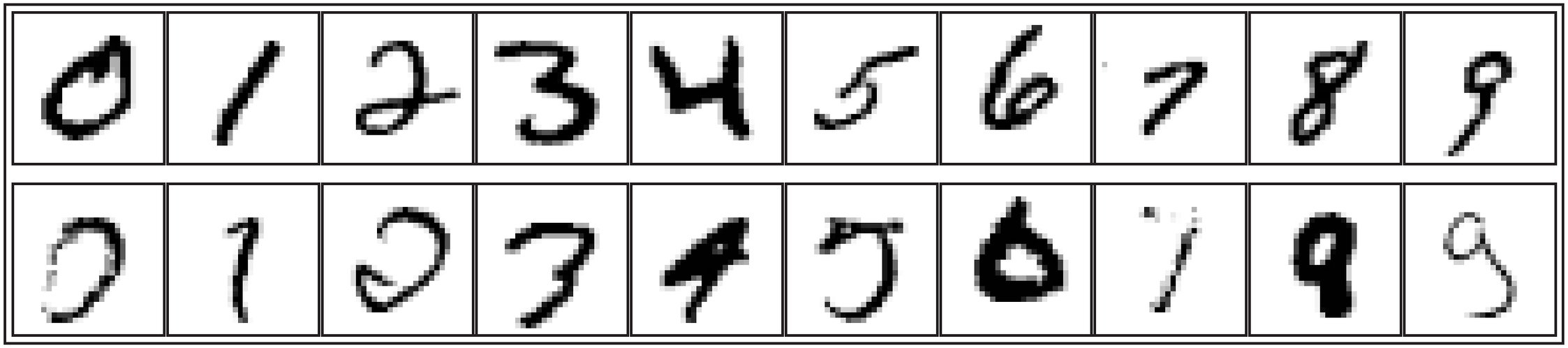
Examples of roads traversed by ALVINN

ALVINN training

Synthetic training data created from actual data



Digit Recognition



3-nearest-neighbor = 2.4% error

400–300–10 unit MLP = 1.6% error

LeNet: 768–192–30–10 unit MLP = 0.9%