

Deep Learning CS583

Jia Xu

Stevens Institute of Technology

2023 Fall

about CS583: contacts

Lecture time:

Section A: Gateway North 204 Mondays 3:00PM-5:30PM

Section B: Gateway South 216 Thursdays 3:30PM-6:00PM

Office hours:

Instructor: Tuesdays 11:00AM-1:00PM

Recitations:

Ahmed Kowsher Tuesdays 1:00PM-3:00PM

Email:

Instructor: jxu70@stevens.edu

TAs:

Ahmed Kosher ga.kowsher@gmail.com Mondays 1:00PM-3:00PM

Preet Jhangiani [pjhangl1@stevens.edu](mailto:pjhagnl1@stevens.edu) Wednesdays 3:00PM-5:00PM

Abhijeet Gusain agusain1@stevens.edu Fridays 3:00PM-5:00PM

about CS583: grading

Grading:

- A: excellent empirical ability and deep understanding
- B: good empirical ability and good understanding
- C: meet basic requirement
- D: fail in following up the course content

Grading proportional to
the ranking of your total score in the class

about CS583: how to score

Midterm exam: 15%

Final exam: 20%

Q&A participation: 10%

Quizz (2): 20%

Presentation: 15%

Project assignment (1): 20%

about CS583: textbook

Textbook:

Deep Learning. Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
<http://www.deeplearningbook.org>

Other Readings:

S. Boyd and L.Vandenberghe. Introduction to Applied Linear Algebra. Cambridge University Press, 2018. (Available online.)

Y. Nesterov. Introductory Lectures on Convex Optimization Book. Springer, 2013. (Available online.)

D. S. Watkins. Fundamentals of Matrix Computations. John Wiley & Sons, 2004.

Francois Chollet. Deep learning with Python. Manning Publications Co., 2017. (Available online.)

M. Mohri, A. Rostamizadeh, and A. Talwalkar. Foundations of machine learning. MIT press, 2012.

J. Friedman, T. Hastie, and R. Tibshirani. The elements of statistical learning. Springer series in statistics, 2001. (Available online.)

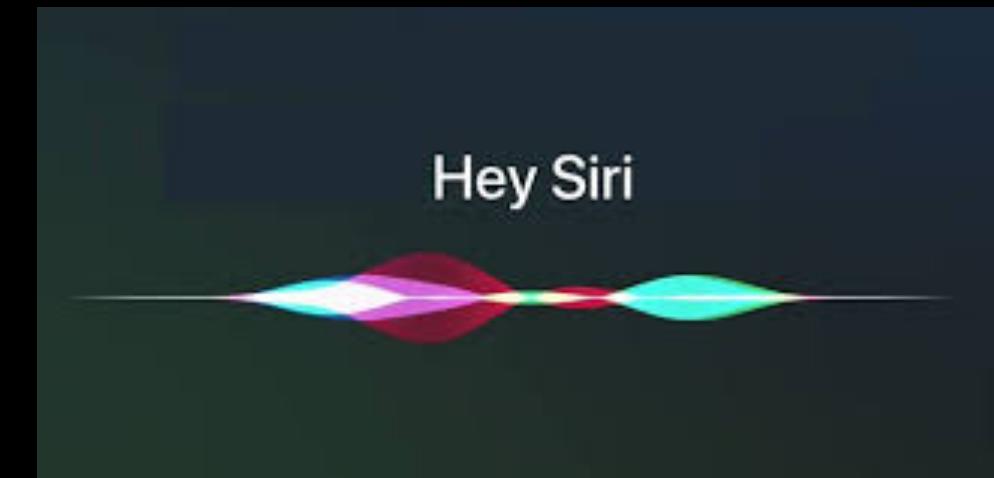
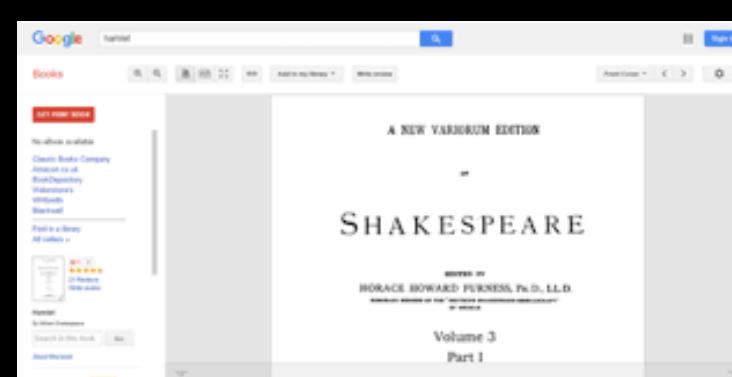
about CS583: tentative topics

Lecture slides	Main topics	Sections in Textbook “Deep Learning” by Goodfellow, Bengio, Courville
Week 1	Introduction, ML basics	Section 5
Week 2	Math Background	Section 2-4
Week 3	MLP	Section 6
Week 4	Stochastic Gradient Descent	Section 6
Week 5	Deep Feedforward Networks	Section 6
Week 6	Regularization, Optimization	Section 7, 8
Week 7	Midterm Exam	All above
Week 8	RNN, LSTM	Section 10
Week 9	CNN	Section 9
Week 10	Attention	beyond textbook
Week 11	GAN	Section 20
Week 12	Auto-encoders	Section 14
Week 13	Thanksgiving	
Week 14	Reinforcement learning	beyond textbook
Week 15	Final exam	All above

Question?

Deep Learning: introduction

Now, deep learning is everywhere



nowadays deep learning
is applied in society, science,
arts, commerce and finance,
literature, military, ...



Back in time,

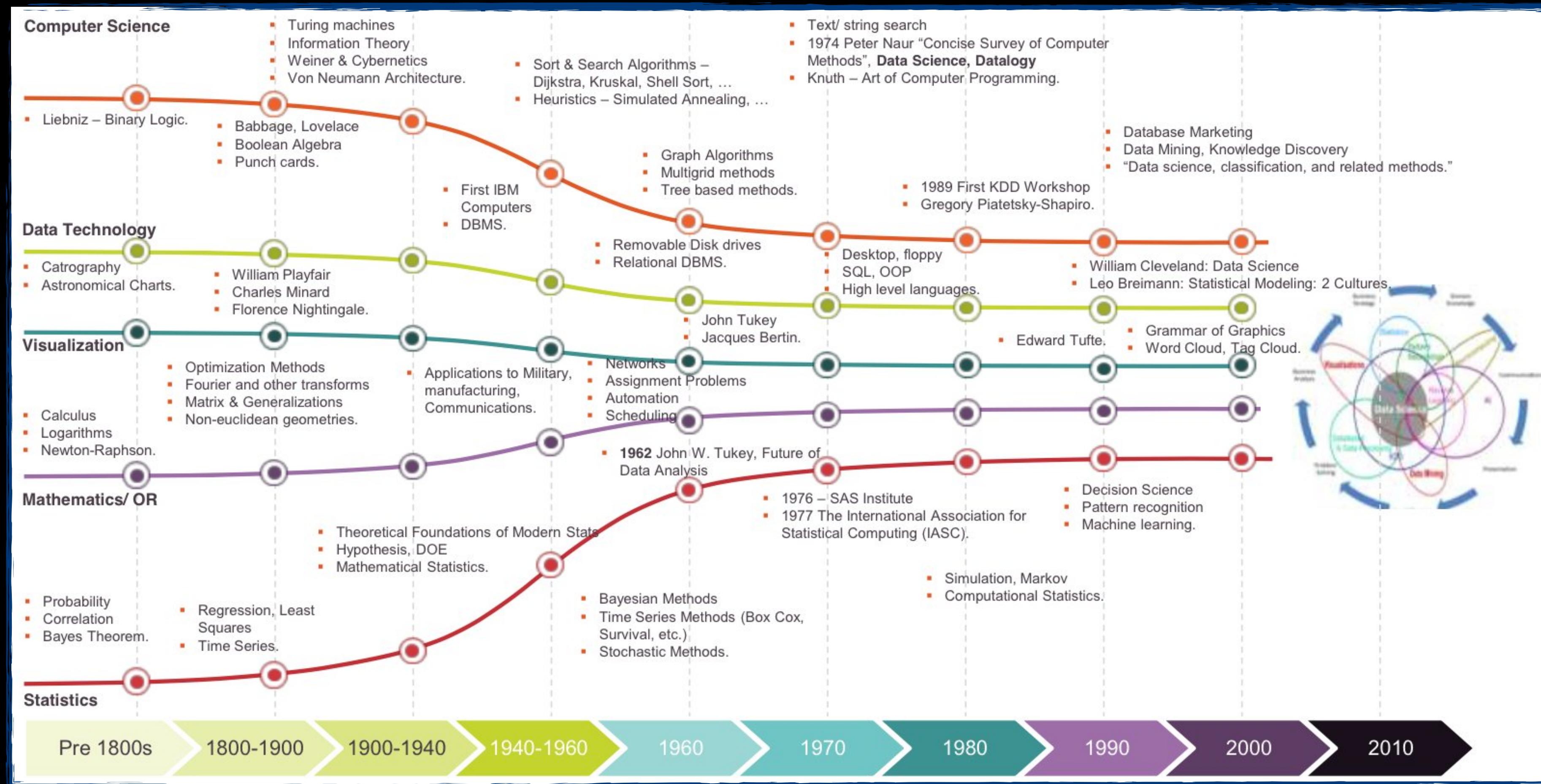
...

How was deep learning introduced and
developed?

...

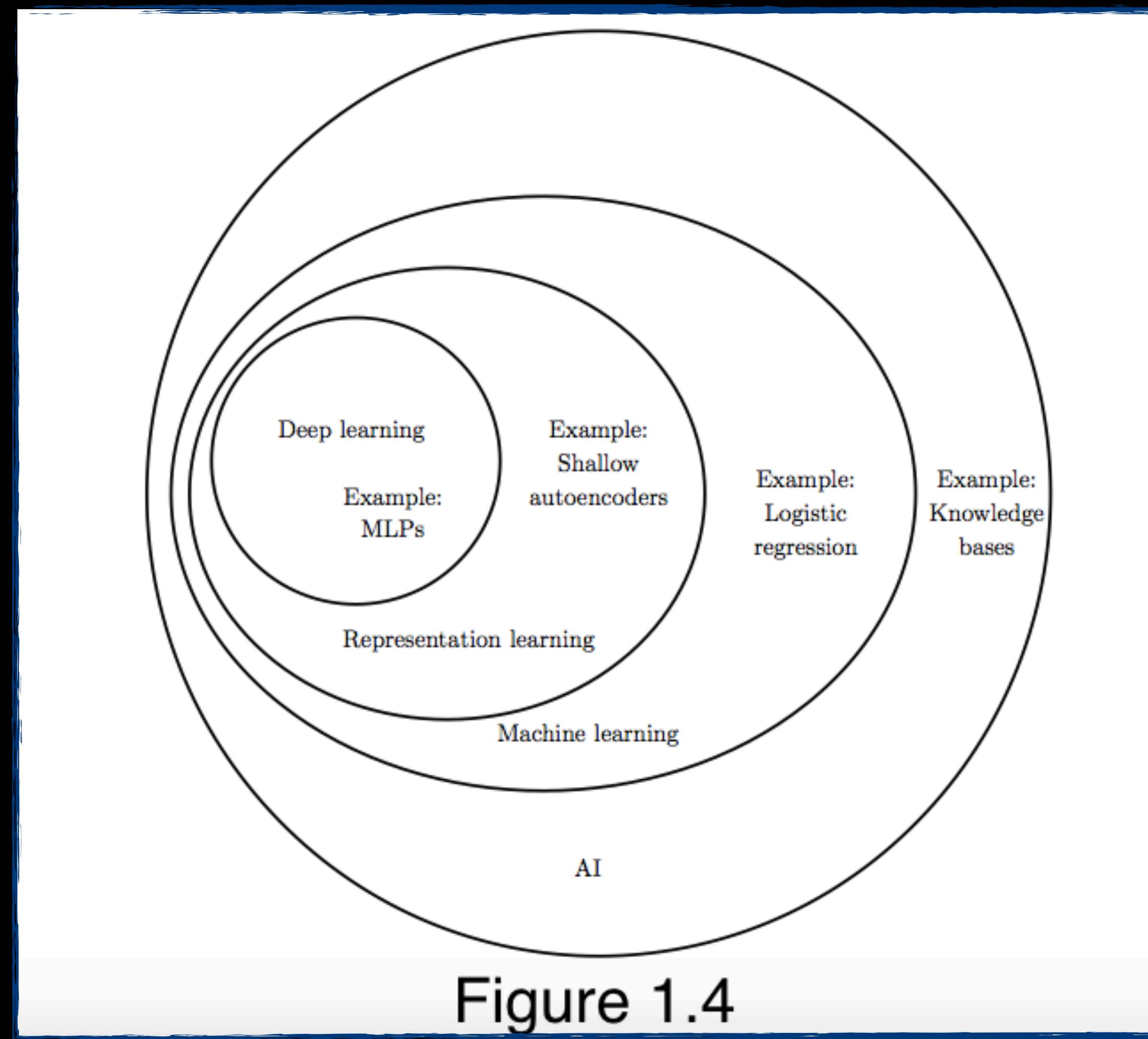
What is the connection of deep learning
with other fields?

from CS, statistics to machine learning

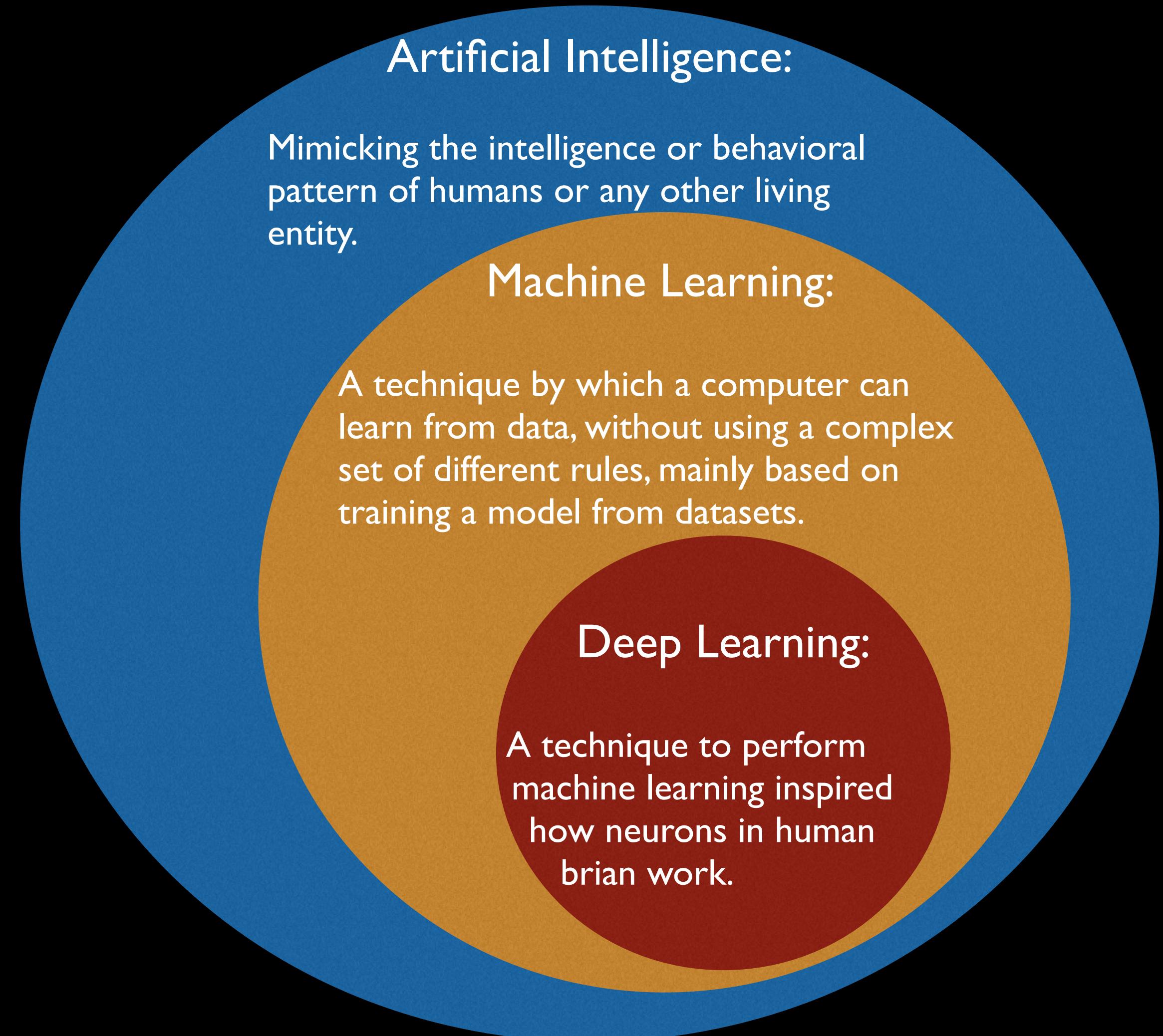


by Kirk Borne

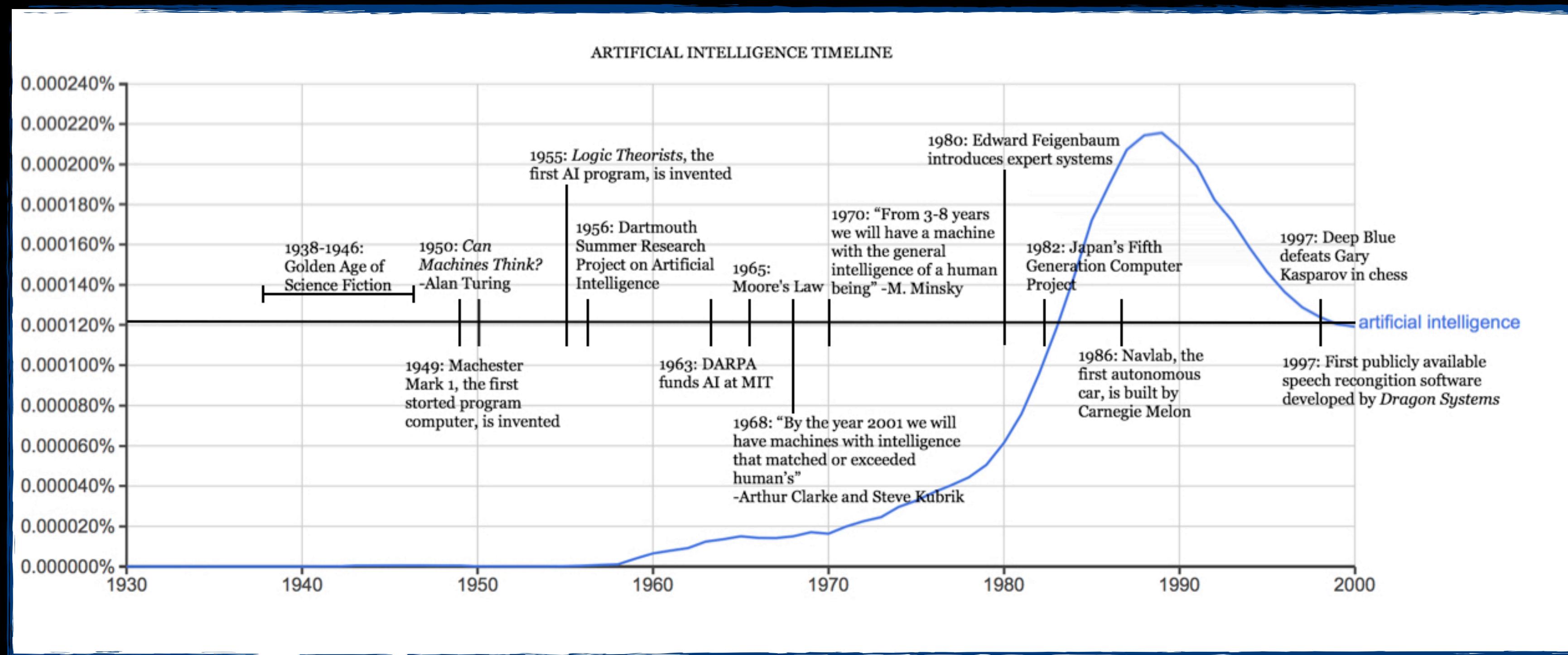
examples of AI, machine learning and deep learning



AI, machine learning, deep learning

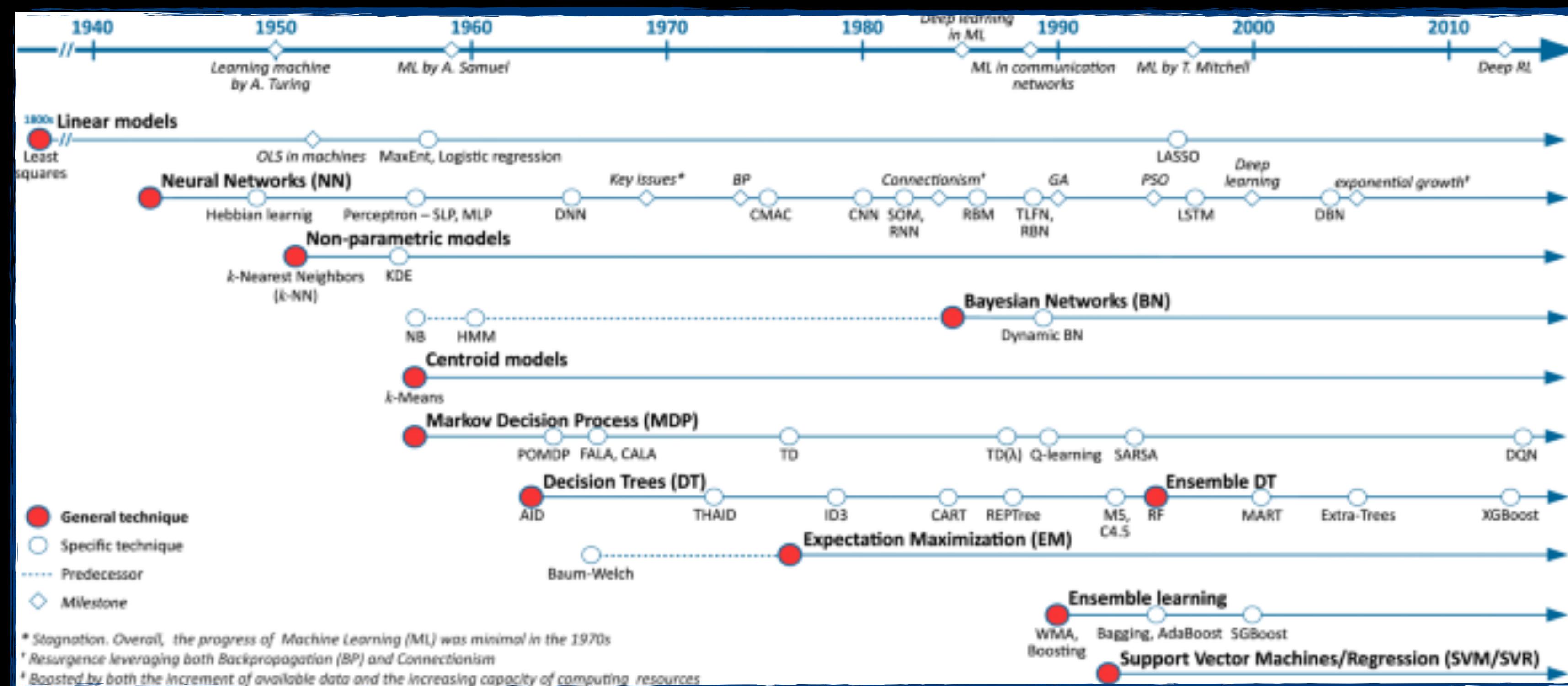


AI history

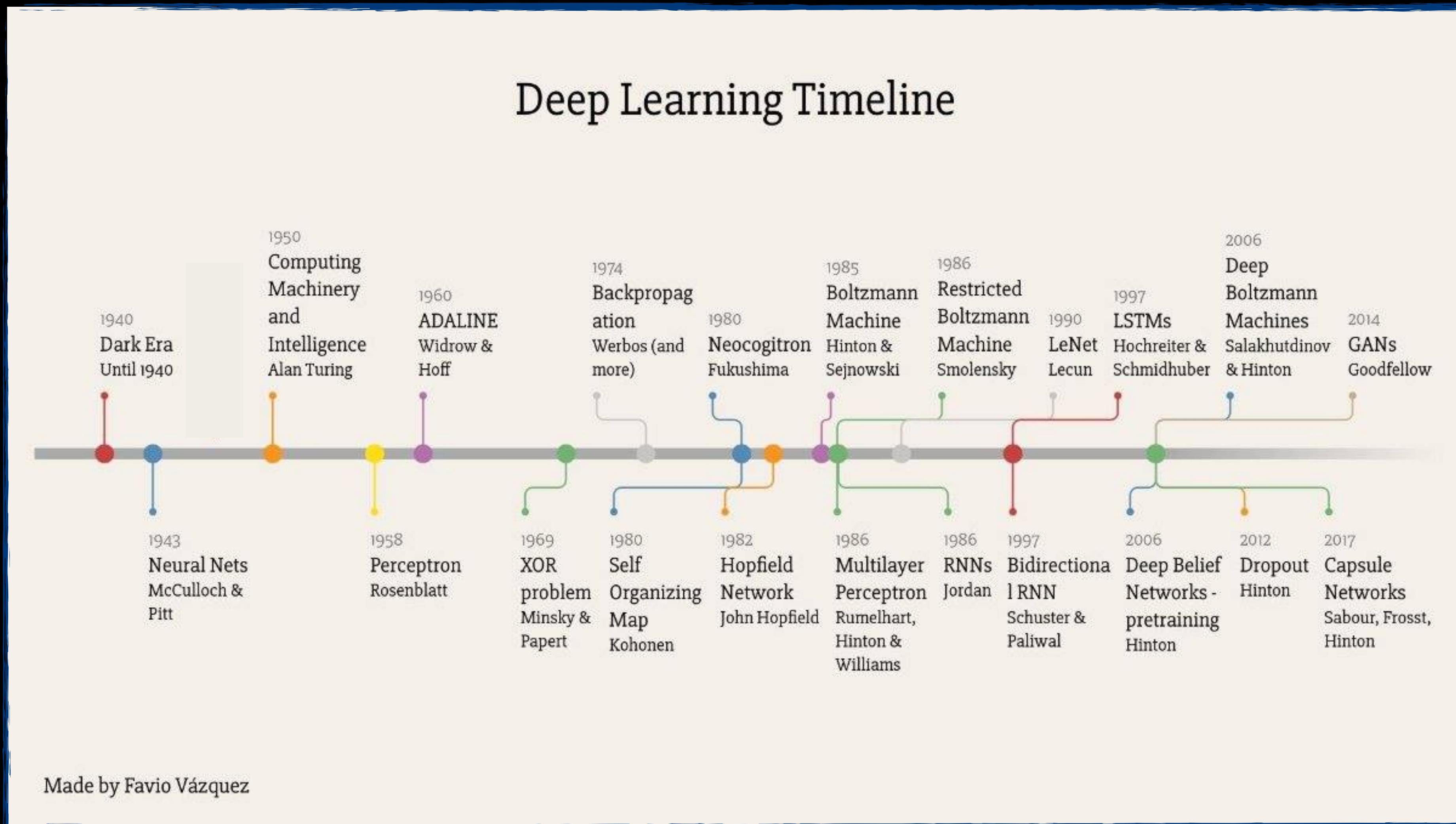


<http://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/>

machine learning history



deep learning history



deep learning history

1943: McCulloch and Pitts, a computational model for neural networks based on threshold logic.

1958: Frank Rosenblatt, the perceptron, a two-layer computer neural network using simple addition and subtraction.

1980: Kunihiko Fukushima proposes the Neoconitron, a hierarchical, multilayered artificial neural network for handwriting recognition.

Mid-2000s: Geoffrey Hinton and Ruslan Salakhutdinov, “deep learning” — a many-layered neural network could be pre-trained one layer at a time.

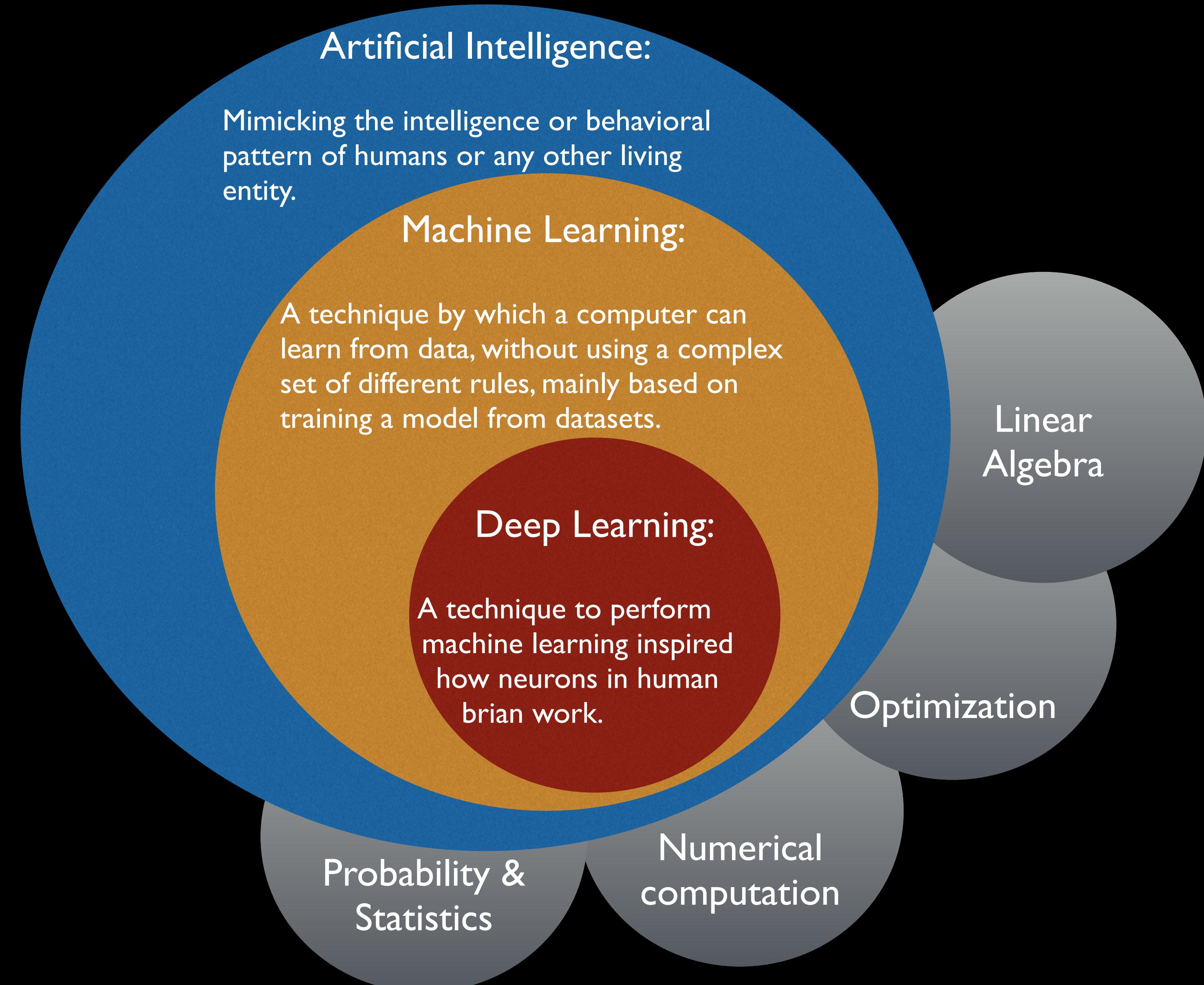
2009: NIPS Workshop on Deep Learning for Speech Recognition

2015: Facebook DeepFace

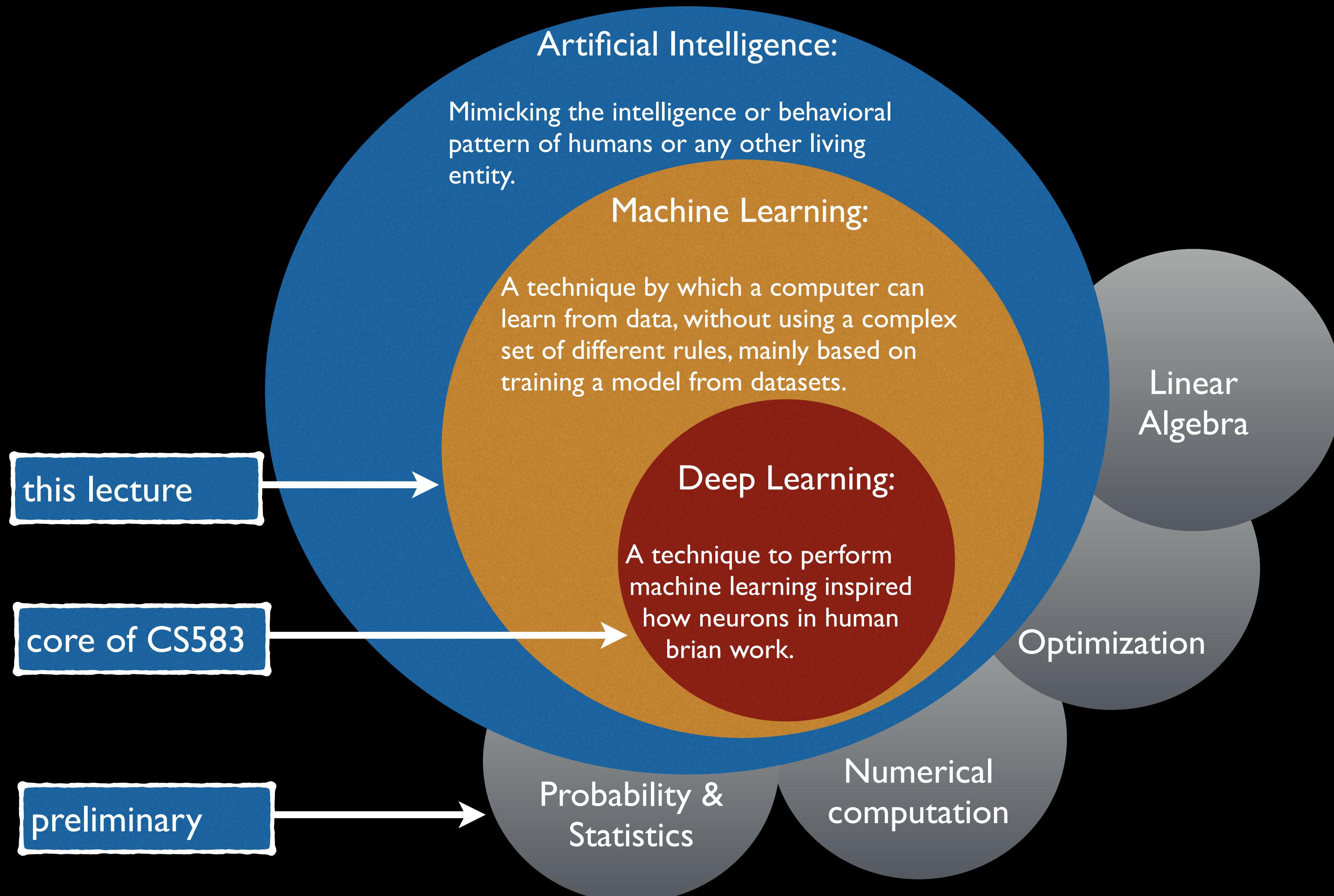
2016: Google DeepMind’s algorithm AlphaGo

...

AI, machine learning, deep Learning



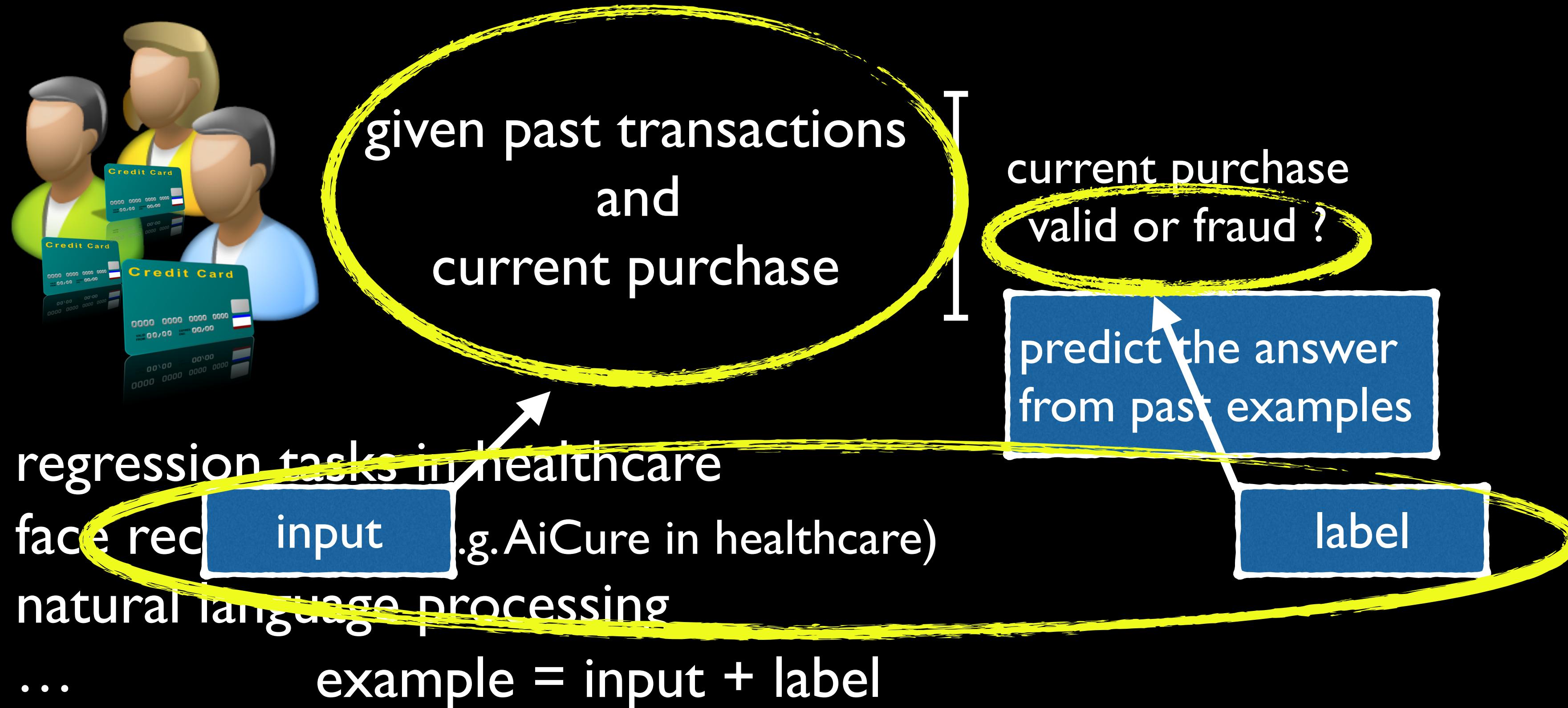
AI, machine learning, deep Learning



Machine Learning

learning from examples

- detect credit card fraud



can big data help to **generically**
improve the prediction accuracy ?

machine learning example

binary classification

label (classify) emails
as
spam or ham

From: howtobeabillioner.com
Subject: lottery

You've been selected
among millions to receive....

...viagra...

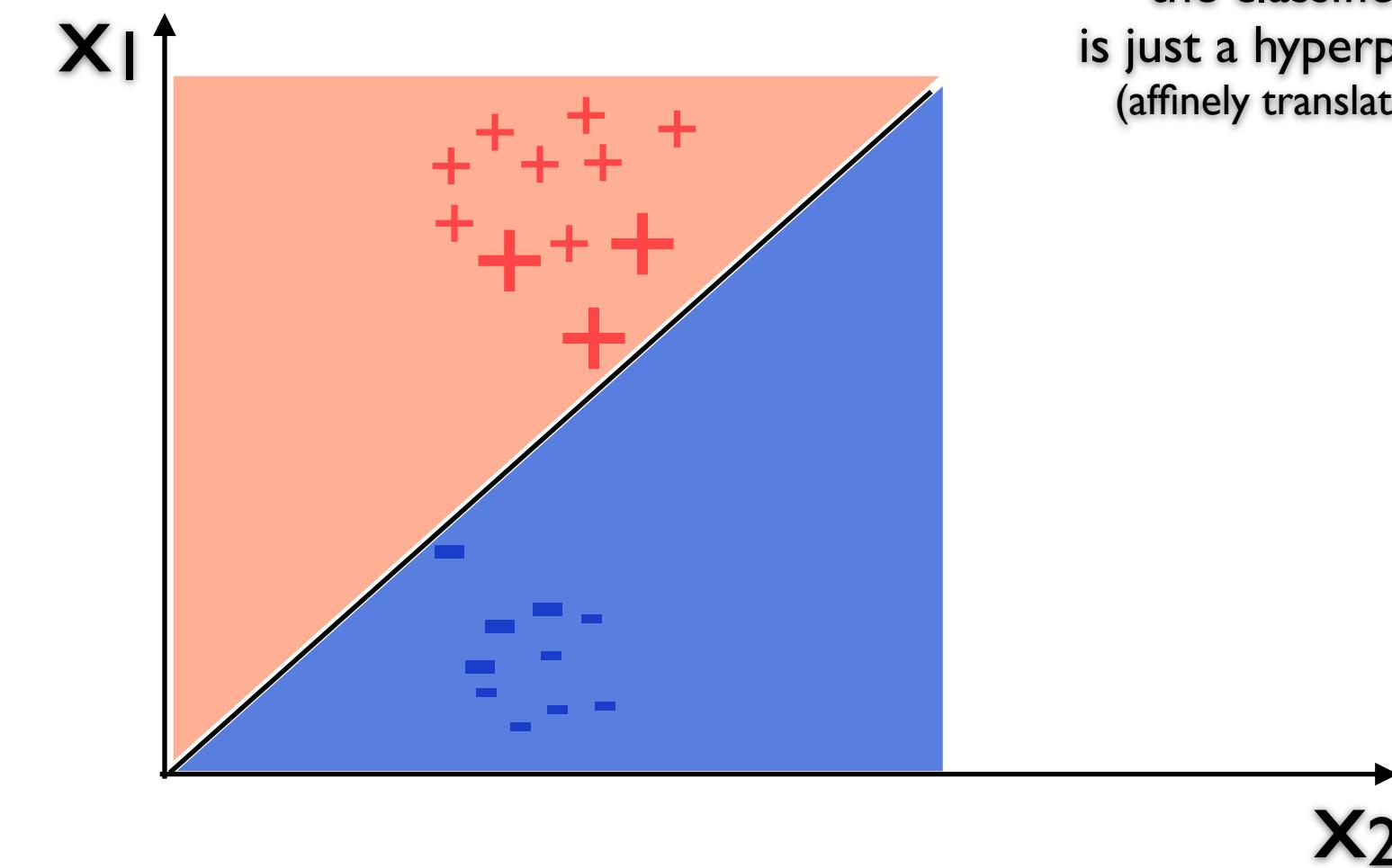
...Rolex...

From: mike@tsinghua.edu.cn
Subject: manuscript

hi Nick,
Please find attached the
corrections to our
manuscript.
...

given an email
extract two features:
 x_1 and x_2 (real numbers)

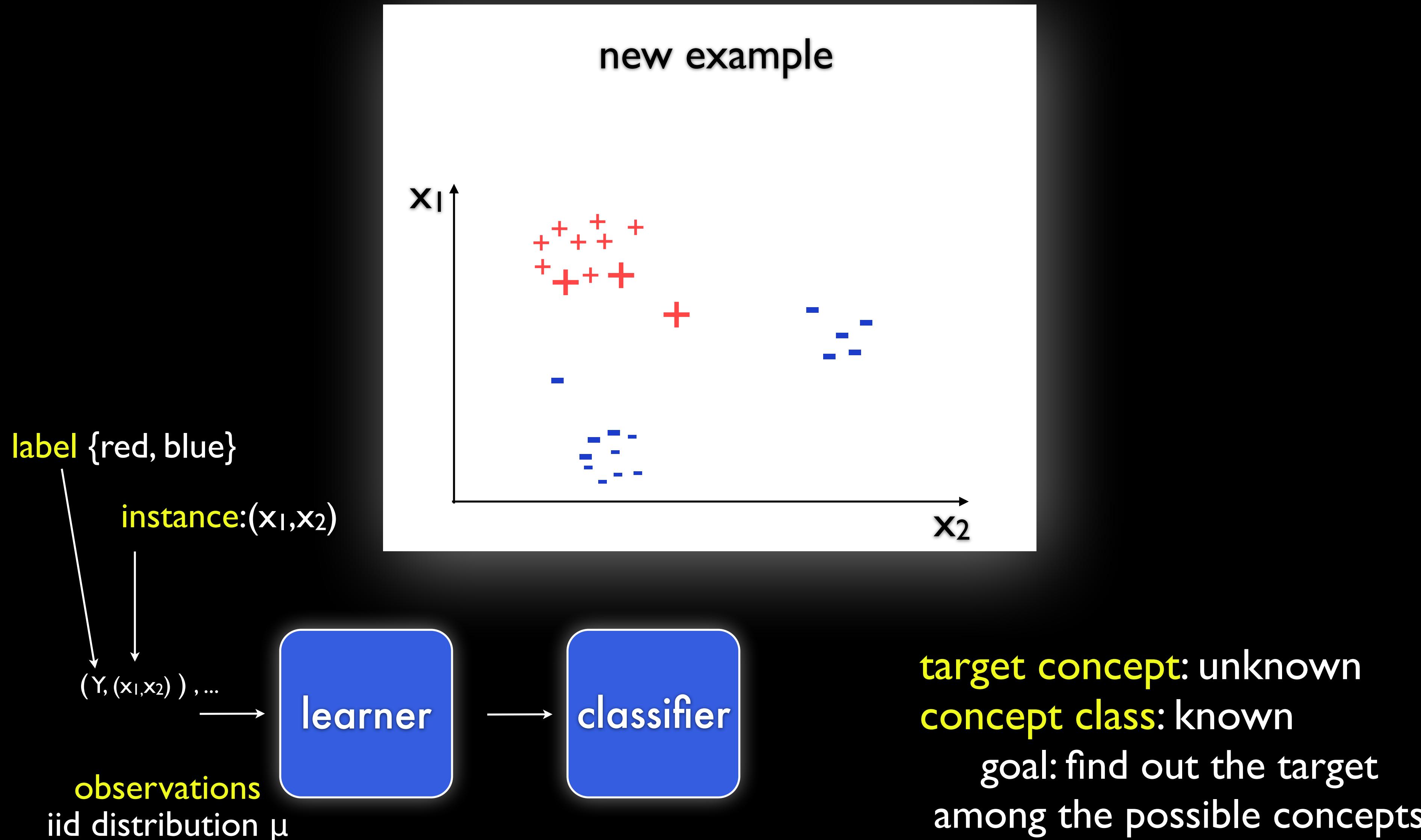
observation (training set)



this example: no generalization error

machine learning example

who constructs the classifier?



machine learning example

who constructs the classifier?

an ensemble can be better than any of the trained classifiers alone

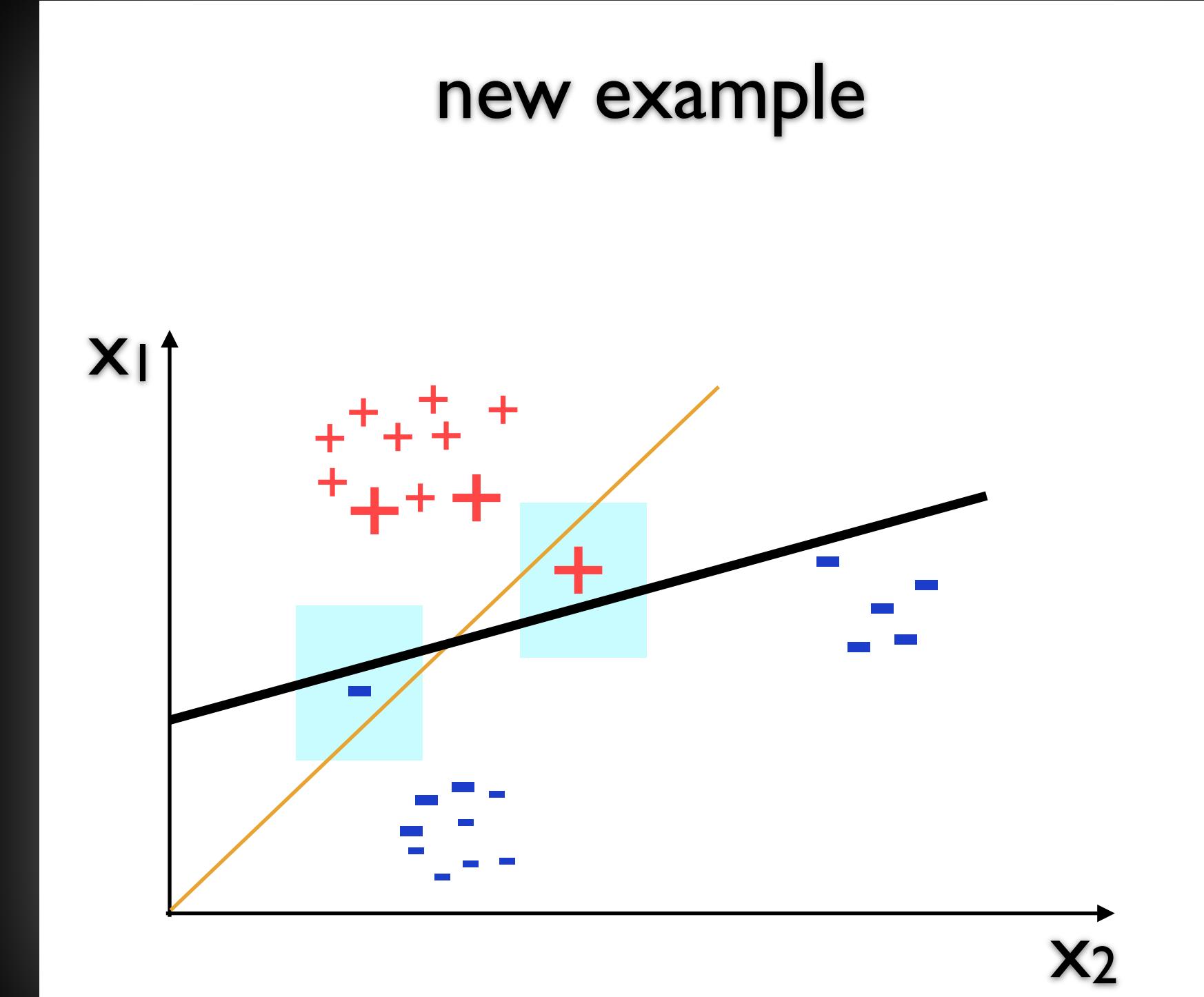
label {red, blue}

instance: (x_1, x_2)

$(Y, (x_1, x_2)) , \dots$

observations

iid distribution μ



the black line has the form

$$0.9x + 1.03$$

but...

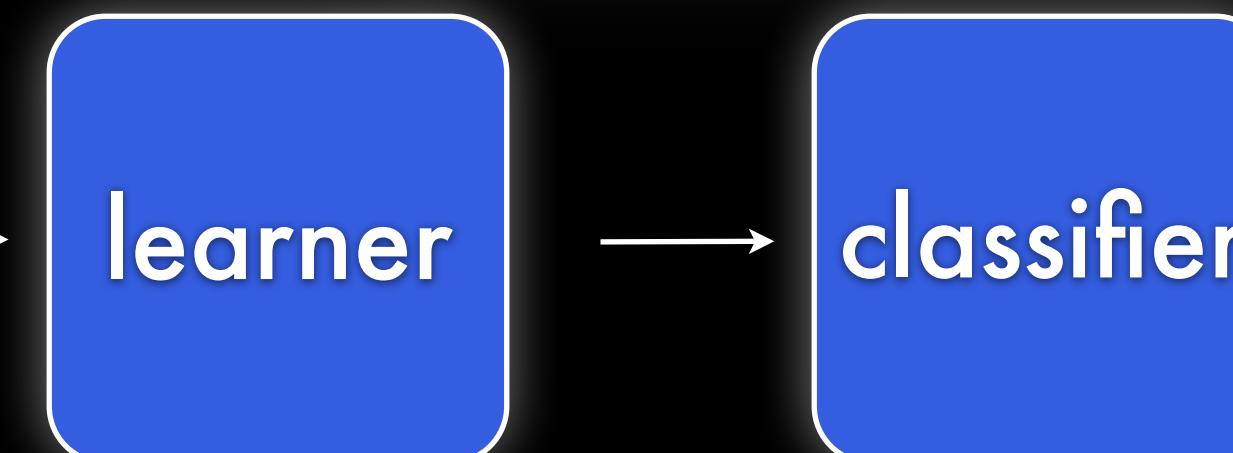
$$0.2(x+2)$$

$$+ 0.2(x+1.2)$$

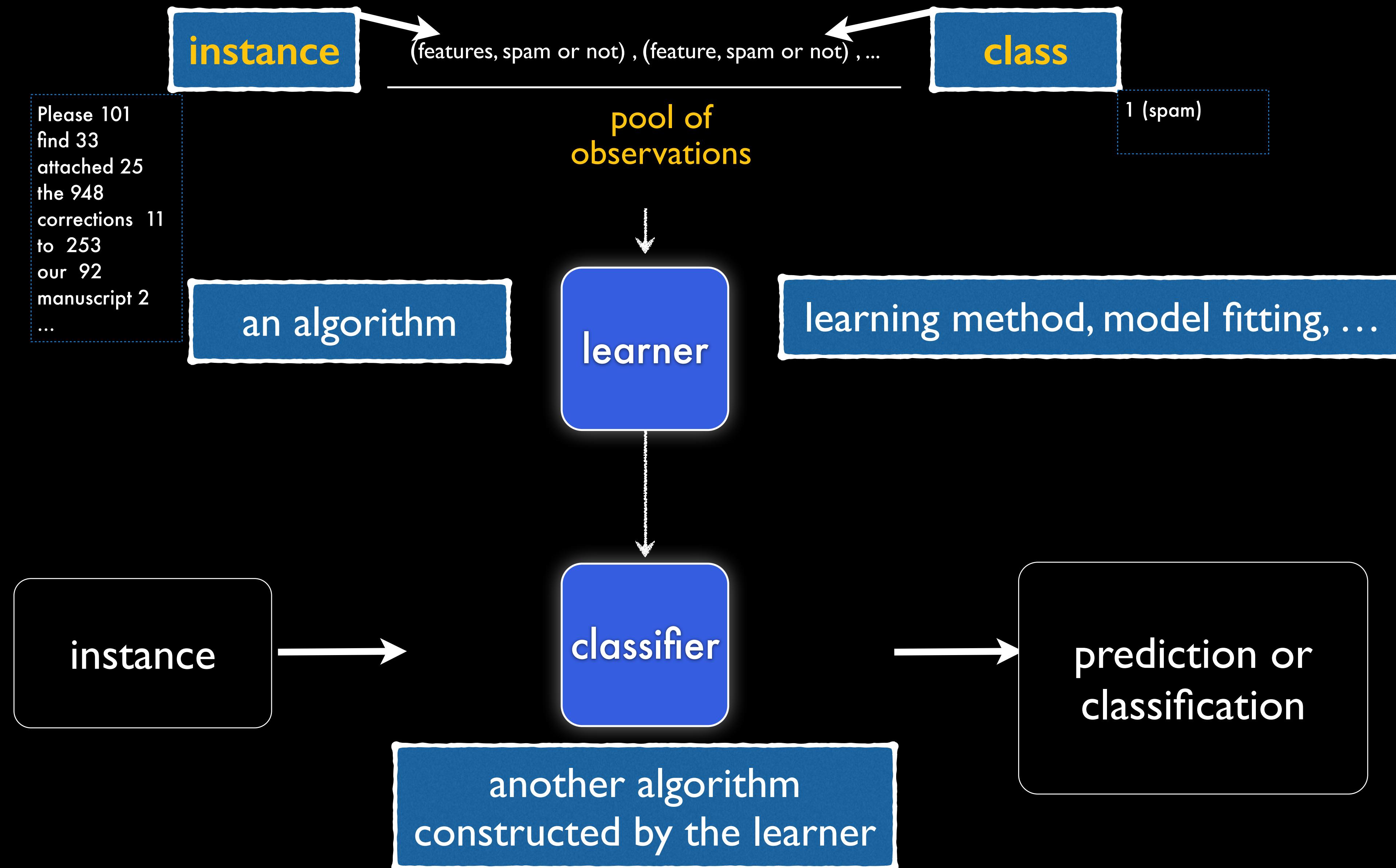
$$+ 0.3(x+1.3)$$

$$0.9x + 1.03$$

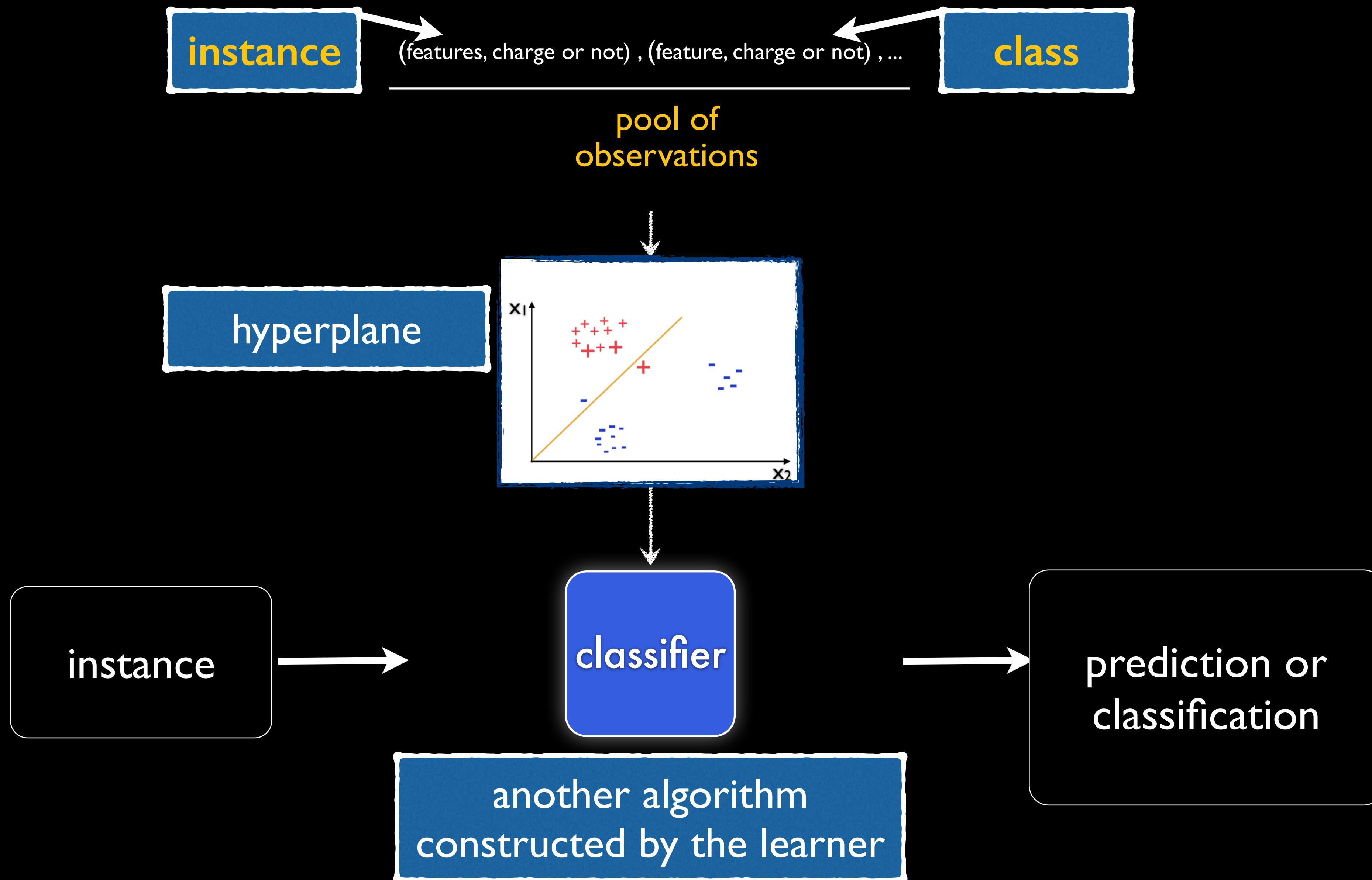
but...our learner can only learn affine lines
 $x+\beta$, where $|\beta| \leq 2$



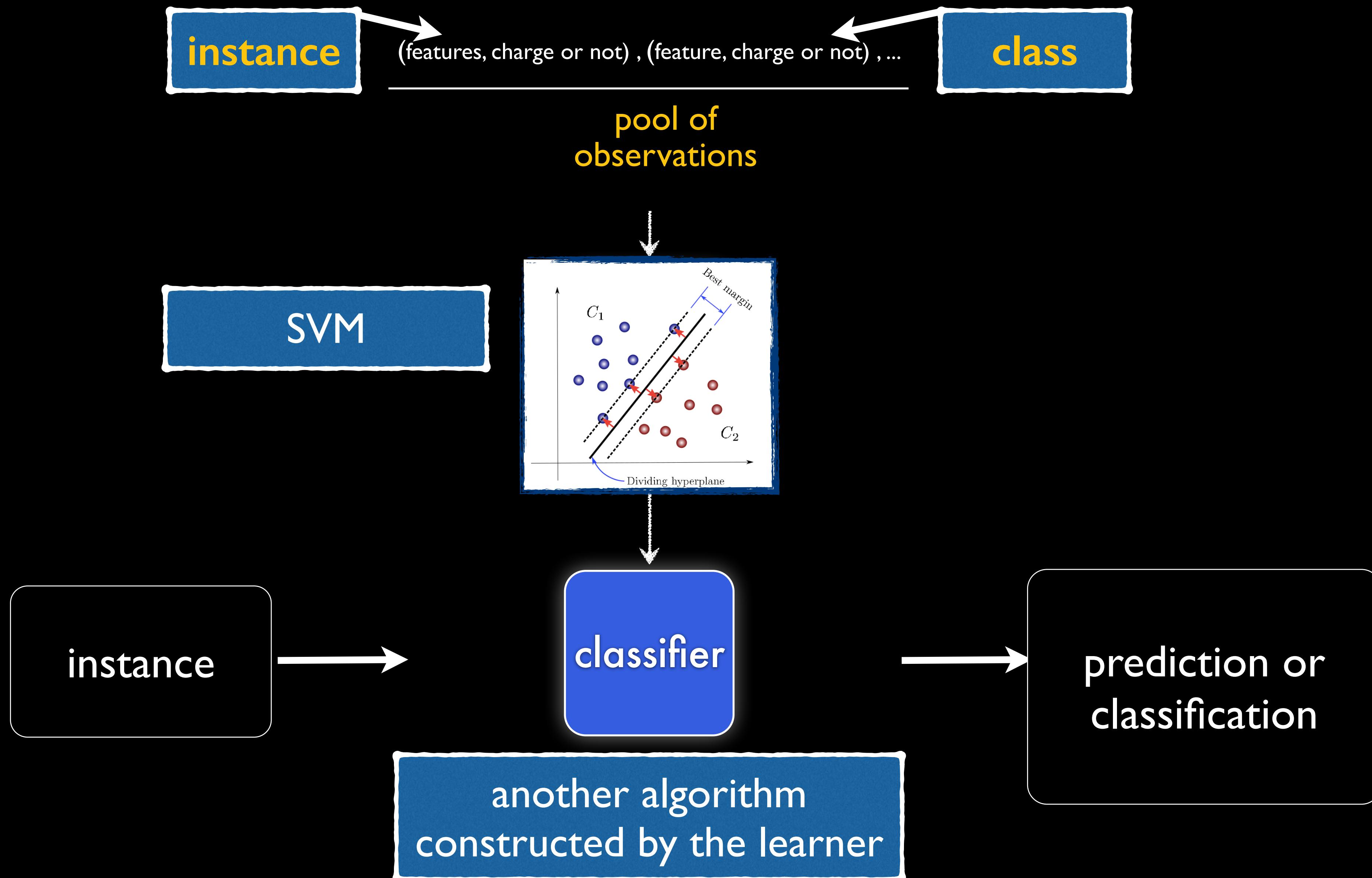
workflow of machine learning



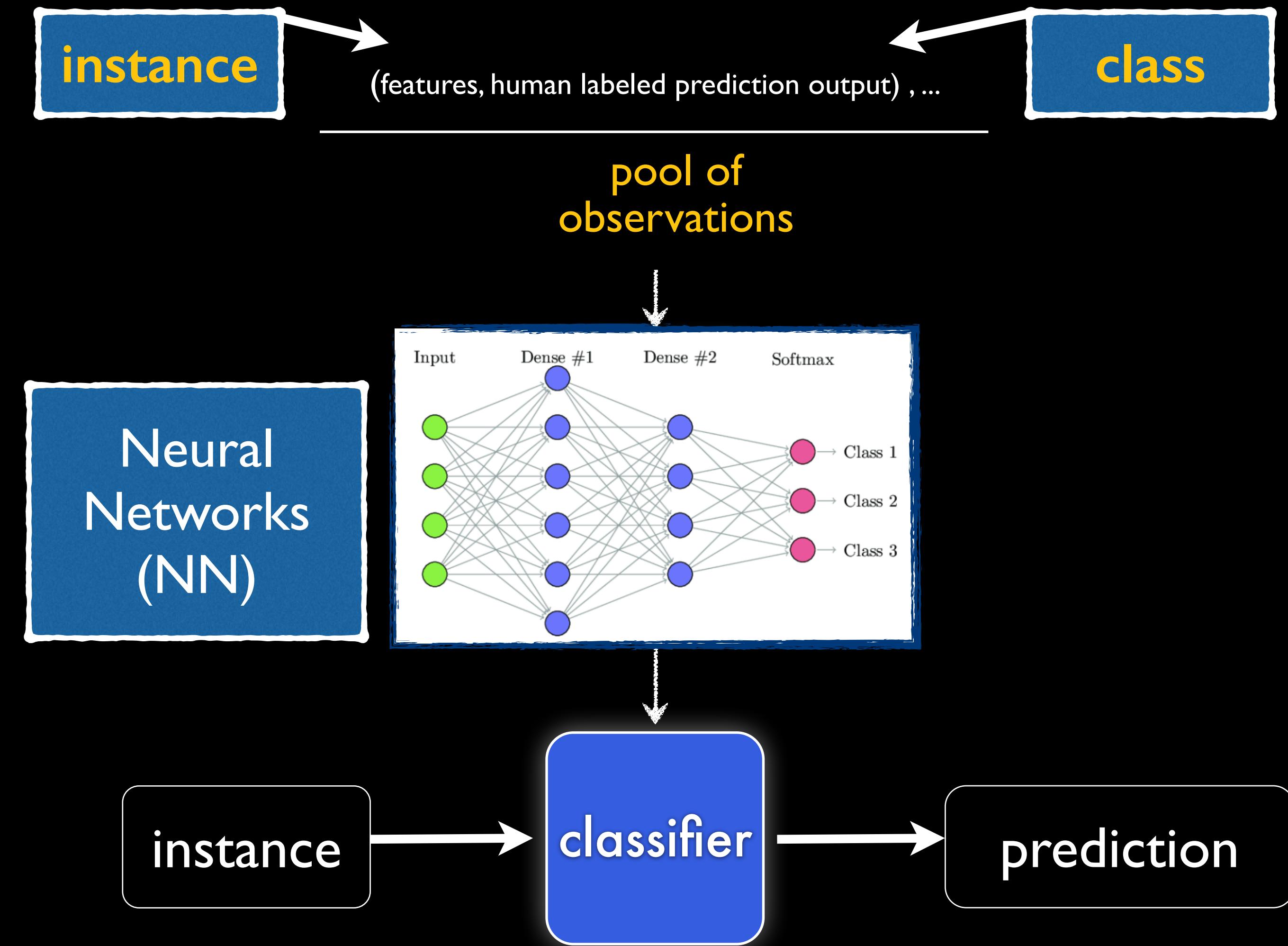
workflow of machine learning



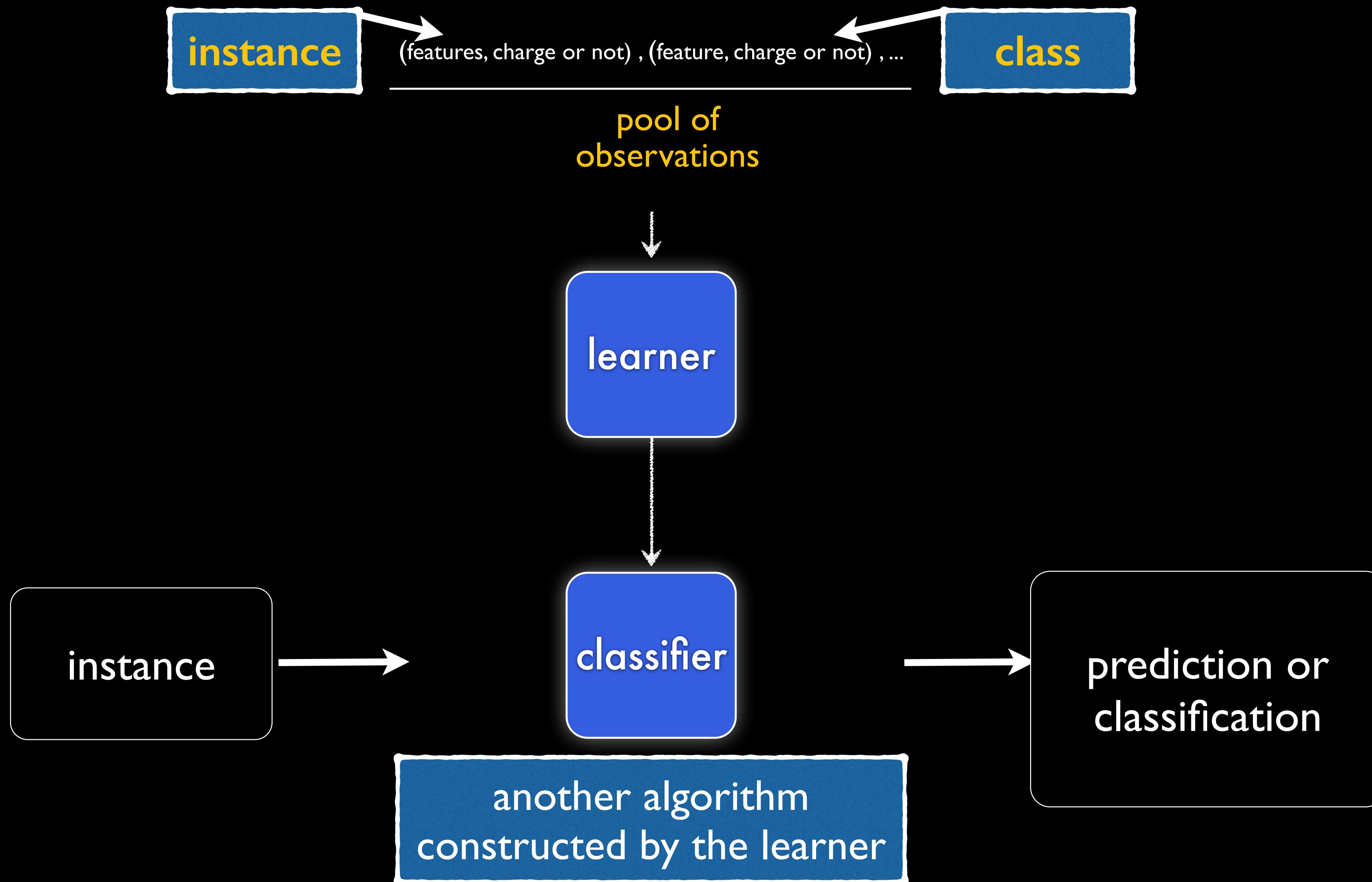
workflow of machine learning



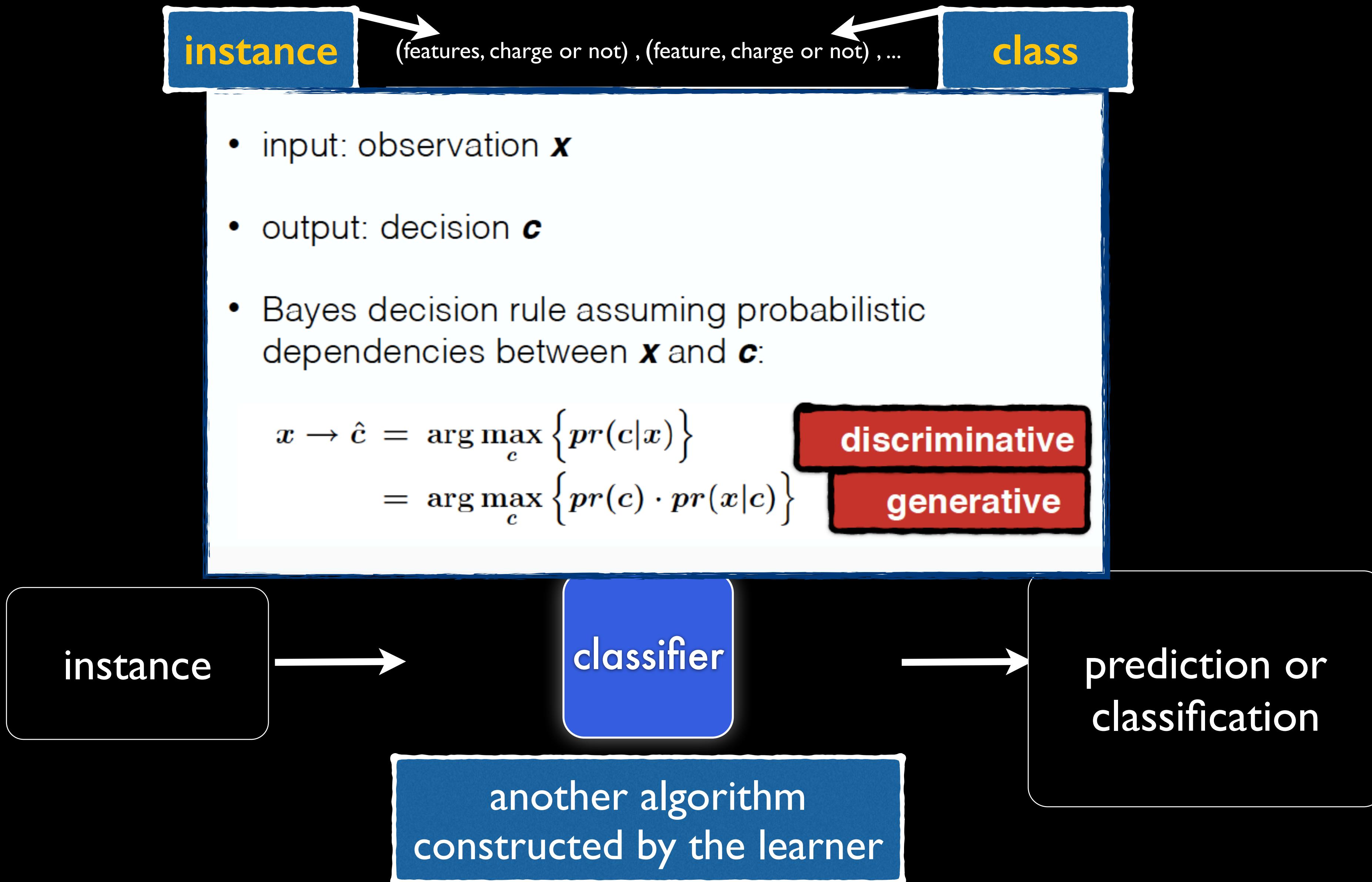
workflow of machine learning



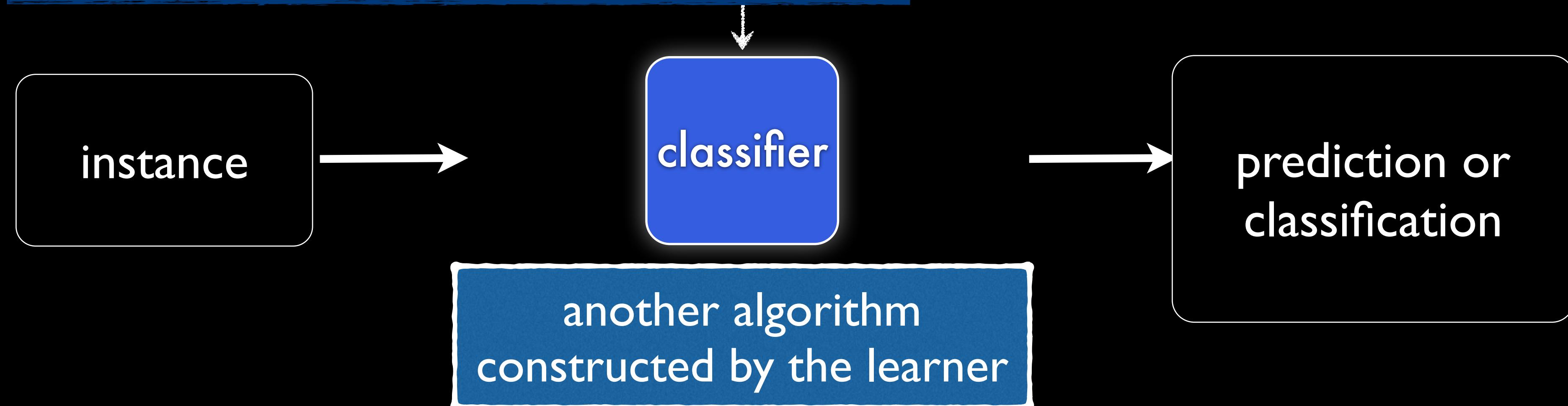
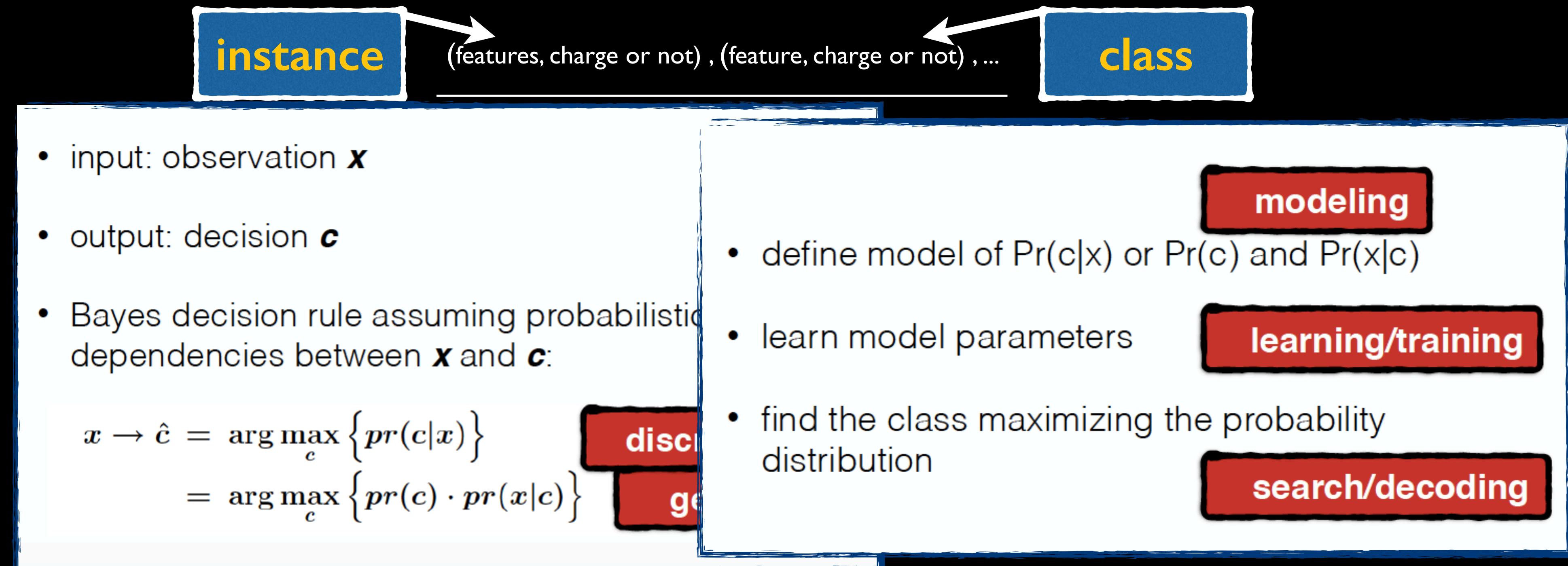
workflow of machine learning



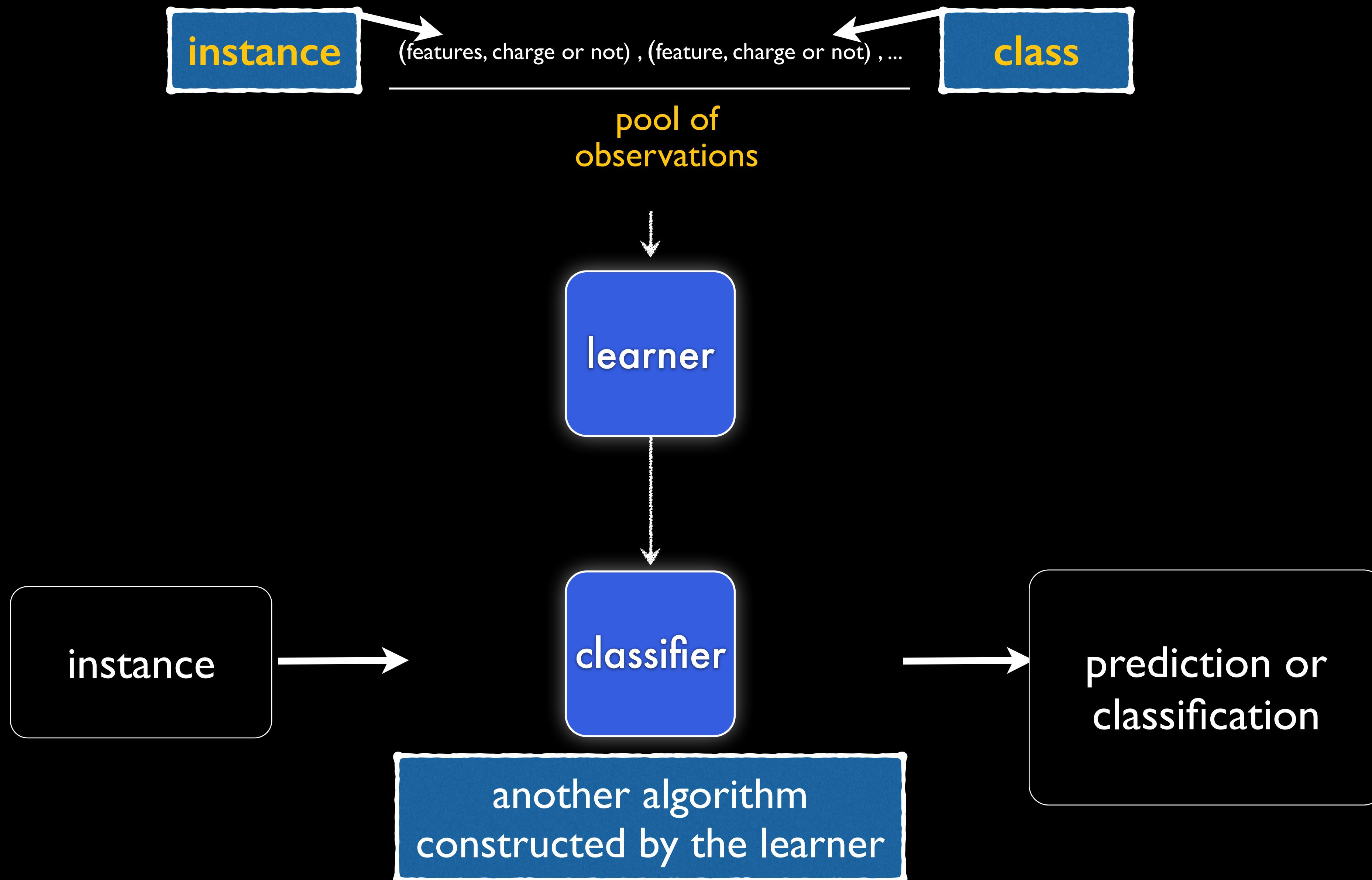
workflow of machine learning



workflow of machine learning



workflow of machine learning



Machine Learning tasks

Classification: spam filtering, face recognition

Osman Khan to Carlos show details Jan 7 (6 days ago) Reply ▾
sounds good +ok
Carlos Guestrin wrote:
Let's try to chat on Friday a little to coordinate and more on Sunday in person?
Carlos

Welcome to New Media Installation: Art that Learns
Carlos Guestrin to 10615-announce, Osman, Michele show details 3:15 PM (8 hours ago) Reply ▾
Hi everyone,
Welcome to New Media Installation:Art that Learns
The class will start tomorrow.
Make sure you attend the first class, even if you are on the Wait List.
The classes are held in Doherty Hall C316, and will be Tue, Thu 01:30-4:20 PM.
By now, you should be subscribed to our course mailing list: 10615-announce@cs.cmu.edu.
You can contact the instructors by emailing: 10615-instructors@cs.cmu.edu

Natural _LoseWeight SuperFood Endorsed by Oprah Winfrey, Free Trial 1 bottle, pay only \$5.95 for shipping mfw rlk Spam | X
Jaquelyn Halley to nherrlein, bcc: thehorney, bcc: ang show details 9:52 PM (1 hour ago) Reply ▾
== Natural WeightLOSS Solution ==
Vital Acai is a natural WeightLOSS product that Enables people to lose wieght and cleansing their bodies faster than most other products on the market.
Here are some of the benefits of Vital Acai that You might not be aware of. These benefits have helped people who have been using Vital Acai daily to Achieve goals and reach new heights in there dieting that they never thought they could.
* Rapid WeightLOSS
* Increased metabolism - BurnFat & calories easily!
* Better Mood and Attitude
* More Self Confidence
* Cleanse and Detoxify Your Body
* Much More Energy
* BetterSexLife
* A Natural Colon Cleanse



Regression: weather, stock trend prediction



Ranking: web search, find similar images

Google

learning to rank

learning to rank

learning to rank **for information retrieval** I'm Feeling Lucky »

learning to rank **using gradient descent**

learning to rank **tutorial**

Search

Web [Learning to rank - Wikipedia, the free encyclopedia](#)
en.wikipedia.org/wiki/Learning_to_rank
Learning to rank or machine-learned ranking (MLR) is a type of supervised or semi-supervised machine learning problem in which the goal is to automatically ...

Images Applications Feature vectors Evaluation measures Approaches

Maps

Videos

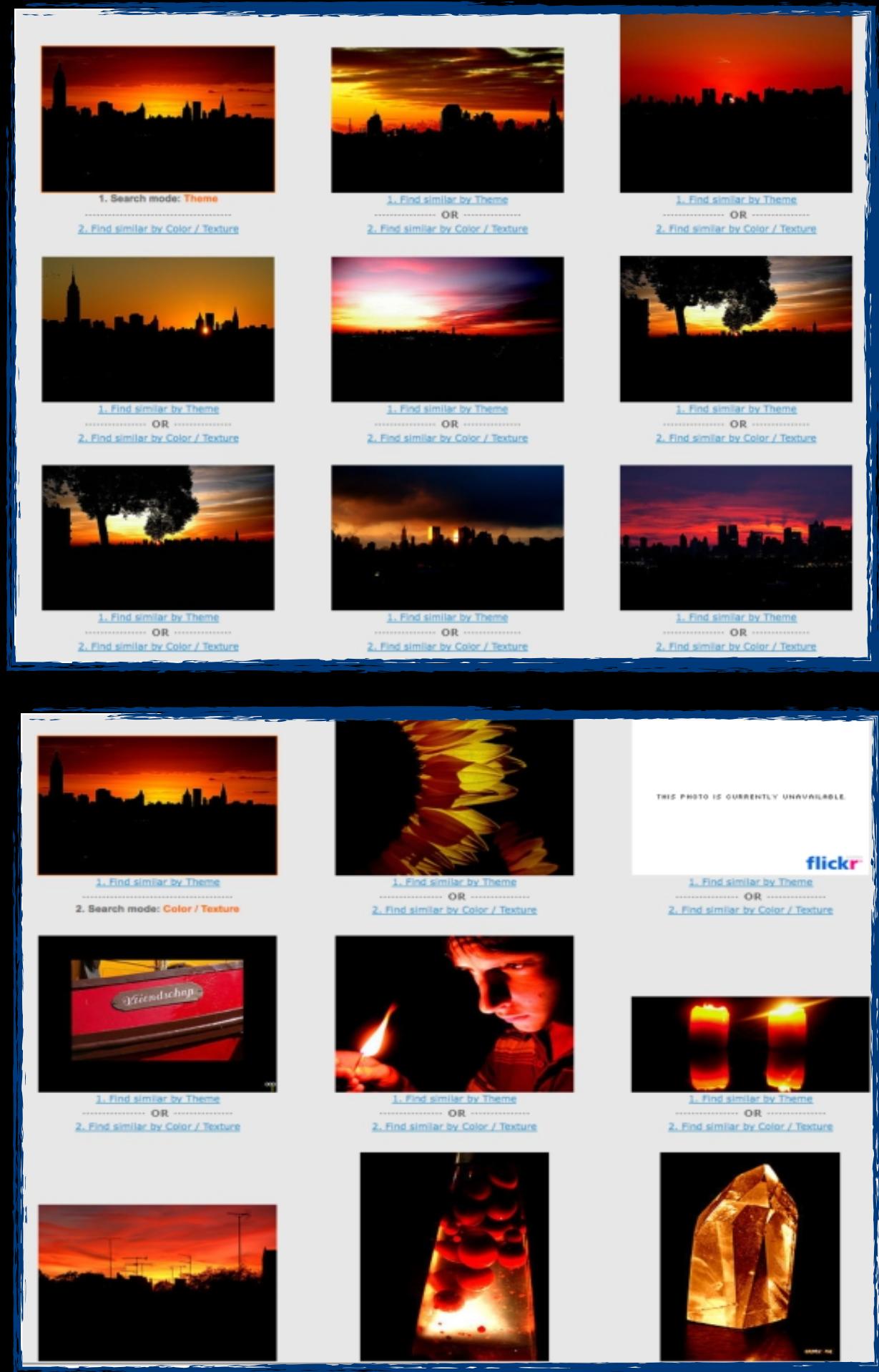
News [Yahoo! Learning to Rank Challenge](#)
learningtorankchallenge.yahoo.com/
Learning to Rank Challenge is closed! Close competition, innovative ideas, and fierce determination were some of the highlights of the first ever Yahoo!

Shopping

More

Manhattan, NY 10012 [Change location](#)

Show search tools



Collaborative filtering: recommendation

David's Amazon.com | Today's Deals | Gift Cards | Sell | Help

Shop by Department | Search Books | Go | Hello, David Your Account | Try Prime | Cart | Wish List

Your Amazon.com | Your Browsing History | Recommended For You | Amazon Betterizer | Improve Your Recommendations | Your Profile | Learn More

Your Amazon.com > Recommended for You > Books > Subjects > Science & Math > History & Philosophy

These recommendations are based on items you own and more.

view: All | New Releases | Coming Soon

Just For Today

[Browse Recommended](#)

Recommendations History & Philosophy

[History of Science](#)
[Philosophy of Biology](#)
[Philosophy of Medicine](#)

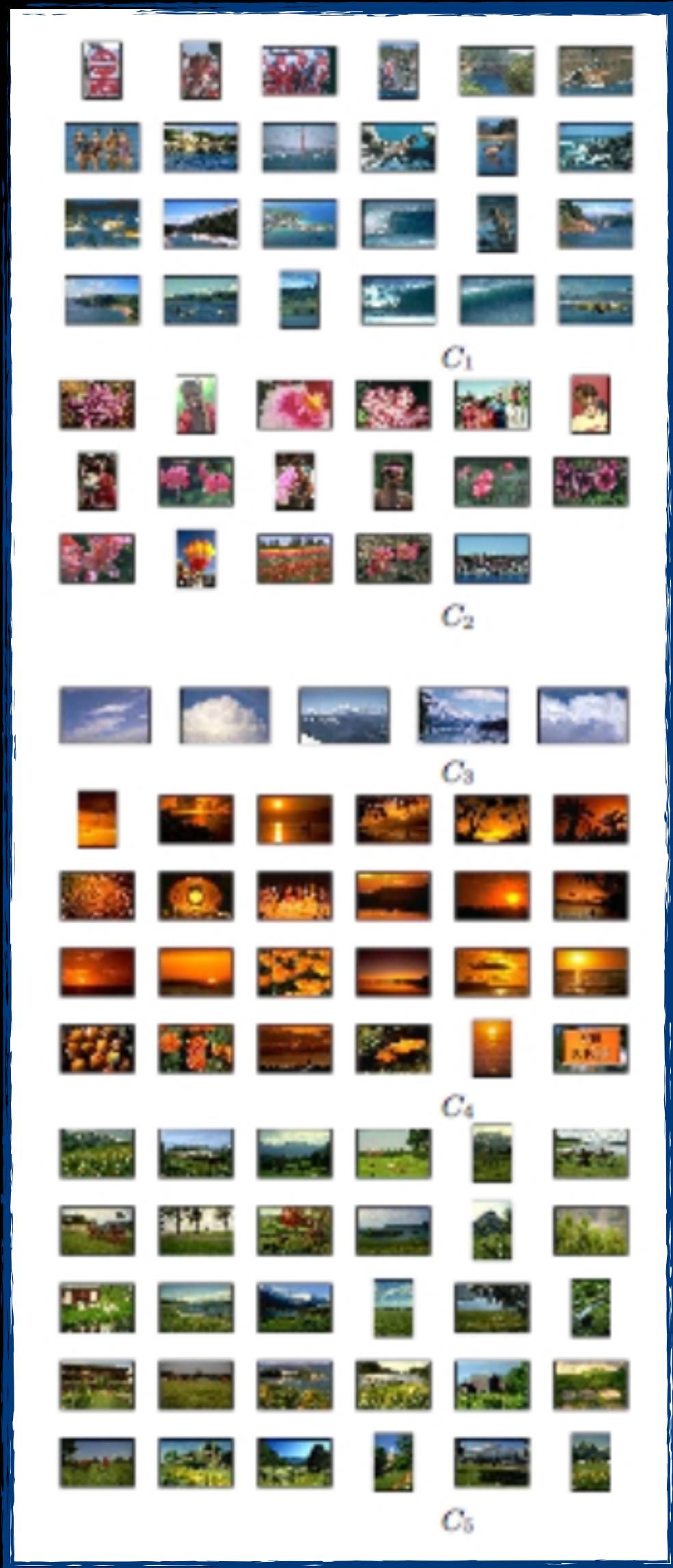
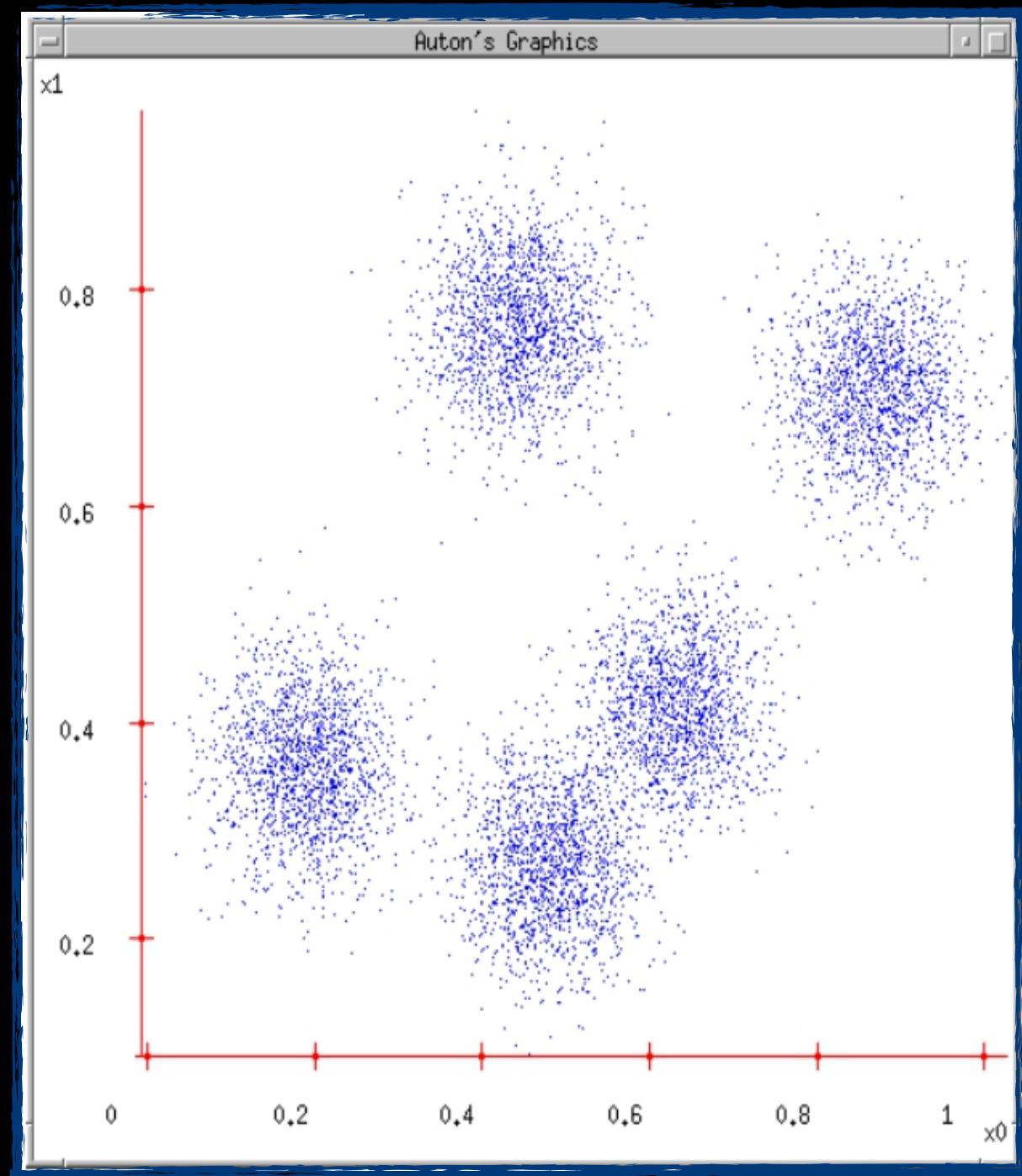
1. Causality: Models, Reasoning and Inference
by Judea Pearl (September 14, 2009)
Average Customer Review: ★★★★★ (10)
In Stock
List Price: \$50.00
Price: \$32.49 [Add to Cart](#) [Add to Wish List](#)
61 used & new from \$28.00
 I own it Not interested Rate this item
Recommended because you purchased Probabilistic Graphical Models and more (Fix this)

2. The Lady Tasting Tea: How Statistics Revolutionized Science in the Twentieth Century
by David Salsburg (May 1, 2002)
Average Customer Review: ★★★★★ (26)
In Stock
List Price: \$18.99
Price: \$13.88 [Add to Cart](#) [Add to Wish List](#)
81 used & new from \$9.00
 I own it Not interested Rate this item
Recommended because you added The Theory That Would Not Die to your Wish List (Fix this)

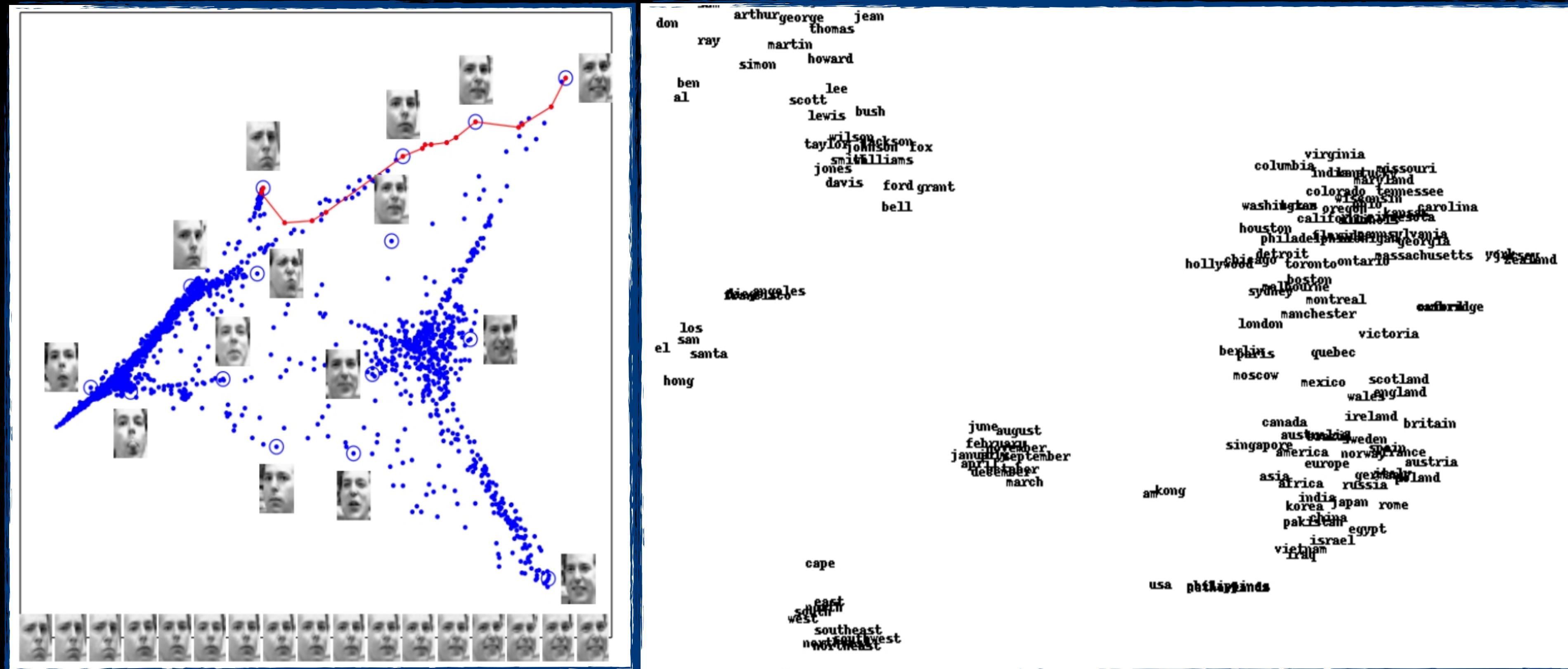
3. The Eighth Day of Creation: Makers of the Revolution in Biology, 25th Anniversary Edition
by Horace Freeland Judson (November 1, 1996)
Average Customer Review: ★★★★★ (10)
In stock on September 4, 2013
List Price: \$56.00
Price: \$36.09 [Add to Cart](#) [Add to Wish List](#)
59 used & new from \$26.95
 I own it Not interested Rate this item
Recommended because you purchased Molecular Biology of the Cell (Fix this)

4. The Machinery of Life
by David S. Goodsell (April 28, 2009)
Average Customer Review: ★★★★★ (41)
In Stock
List Price: \$25.00
Price: \$17.49 [Add to Cart](#) [Add to Wish List](#)
92 used & new from \$12.00

Clustering: group similar things



embedding: visualize data



Machine learning applications

Machine learning is preferred approach to

- Speech recognition
- Natural language processing
- Computer vision
- Medical outcomes analysis
- Robot control
- Computational biology
- Sensor networks
- ...

This trend is accelerating

- Big data
- Improved machine learning algorithms
- Faster computers
- Good open-source software

Machine Learning problem

Supervised Learning

- Given: Training set $\{(x_i, y_i) \mid i = 1 \dots N\}$
- Find: A good approximation to $f : X \rightarrow Y$

Examples: what are X and Y ?

- Spam Detection
 - Map email to {Spam, Not Spam}
- Digit recognition
 - Map pixels to {0,1,2,3,4,5,6,7,8,9}
- Stock Prediction

A Supervised Learning Problem

Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

- Our goal is to find a function $f : X \rightarrow Y$
 - $X = \{0,1\}^4$
 - $Y = \{0,1\}$
- Question 1: How should we pick the hypothesis space, the set of possible functions f ?
- Question 2: How do we find the best f in the hypothesis space?

Most General Hypothesis Space

Consider all possible boolean functions over four

- 2^{16} possible hypotheses
- 2^9 are consistent with our dataset
- How do we choose the best one?

x_1	x_2	x_3	x_4	y
0	0	0	0	?
0	0	0	1	?
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	?
1	0	0	0	?
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	0
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

A Restricted Hypothesis Space

Consider all conjunctive boolean functions.

- 16 possible hypotheses
- None are consistent with our dataset
- How do we choose the best one?

Rule	Counterexample	Dataset:					
		Example	x_1	x_2	x_3	x_4	y
$\Rightarrow y$	1	1	0	0	1	0	0
$x_1 \Rightarrow y$	3	2	0	1	0	0	0
$x_2 \Rightarrow y$	2	3	0	0	1	1	1
$x_3 \Rightarrow y$	1	4	1	0	0	1	1
$x_4 \Rightarrow y$	7	5	0	1	1	0	0
$x_1 \wedge x_2 \Rightarrow y$	3	6	1	1	0	0	0
$x_1 \wedge x_3 \Rightarrow y$	3	7	0	1	0	1	0
$x_1 \wedge x_4 \Rightarrow y$	3						
$x_2 \wedge x_3 \Rightarrow y$	3						
$x_2 \wedge x_4 \Rightarrow y$	3						
$x_3 \wedge x_4 \Rightarrow y$	4						
$x_1 \wedge x_2 \wedge x_3 \Rightarrow y$	3						
$x_1 \wedge x_2 \wedge x_4 \Rightarrow y$	3						
$x_1 \wedge x_3 \wedge x_4 \Rightarrow y$	3						
$x_2 \wedge x_3 \wedge x_4 \Rightarrow y$	3						
$x_1 \wedge x_2 \wedge x_3 \wedge x_4 \Rightarrow y$	3						

Occam's Razor Principle

- William of Occam: Monk living in the 14th century
- Principle of parsimony:

“One should not increase, beyond what is necessary, the number of entities required to explain anything”

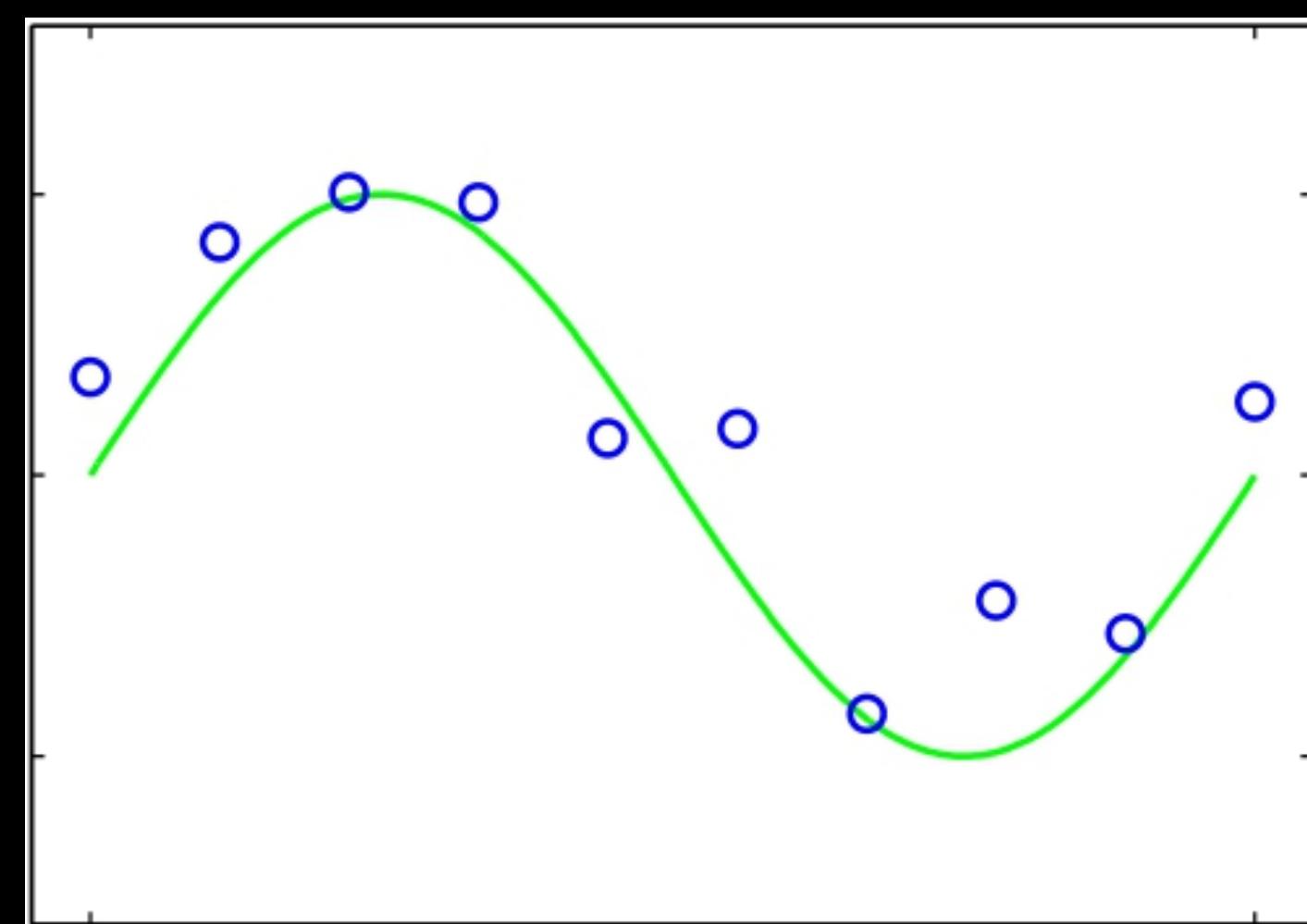
- When many solutions are available for a given problem, we should select the simplest one
- But what do we mean by simple?
- We will use prior knowledge of the problem to define what is a simple solution

Example of a prior: smoothness

[Samy Bengio]

Another Example: Regression

Dataset: 10 (X,Y) points generated
from a sin function, with noise

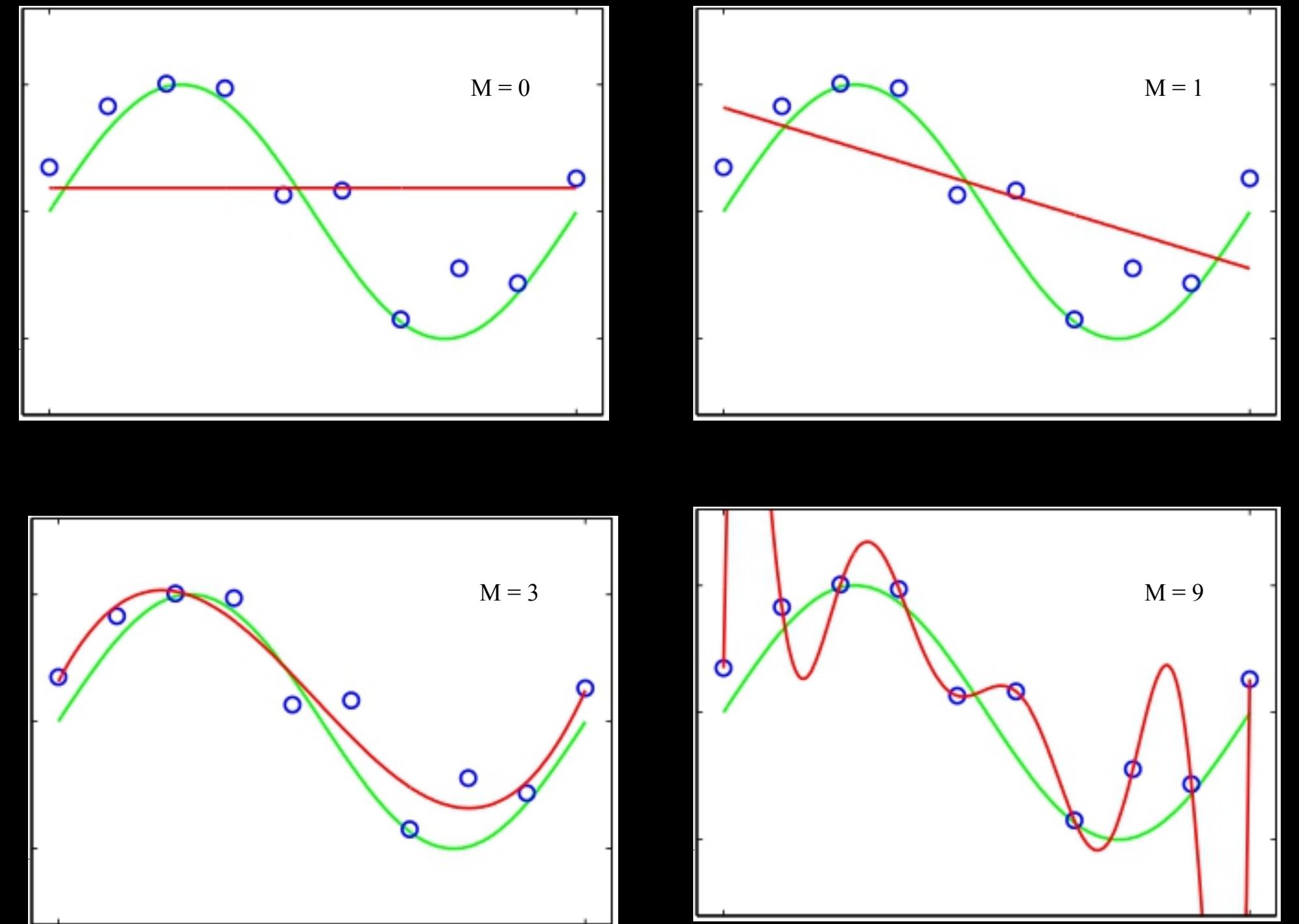


- Regression:
 - $f : X \rightarrow Y$
 - $X = \mathbb{R}$
 - $Y = \mathbb{R}$

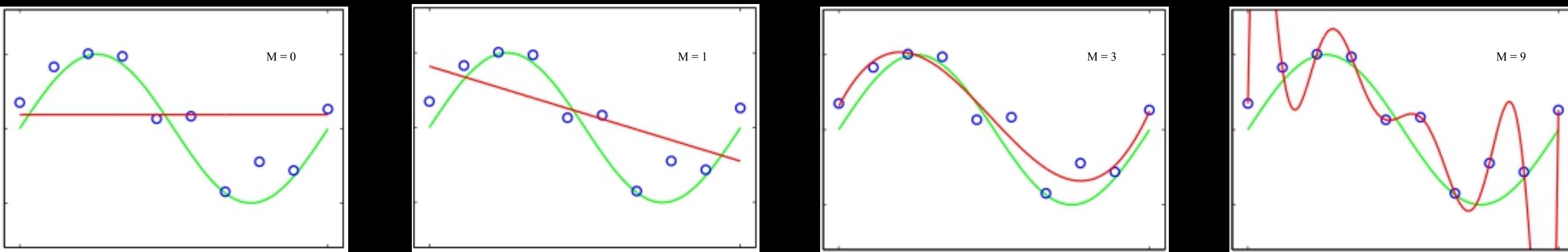
Degree-M Polynomials

How about letting f be a degree M polynomial?

- Which one is **best** ?



Hypo. Space: Degree-N Polynomials



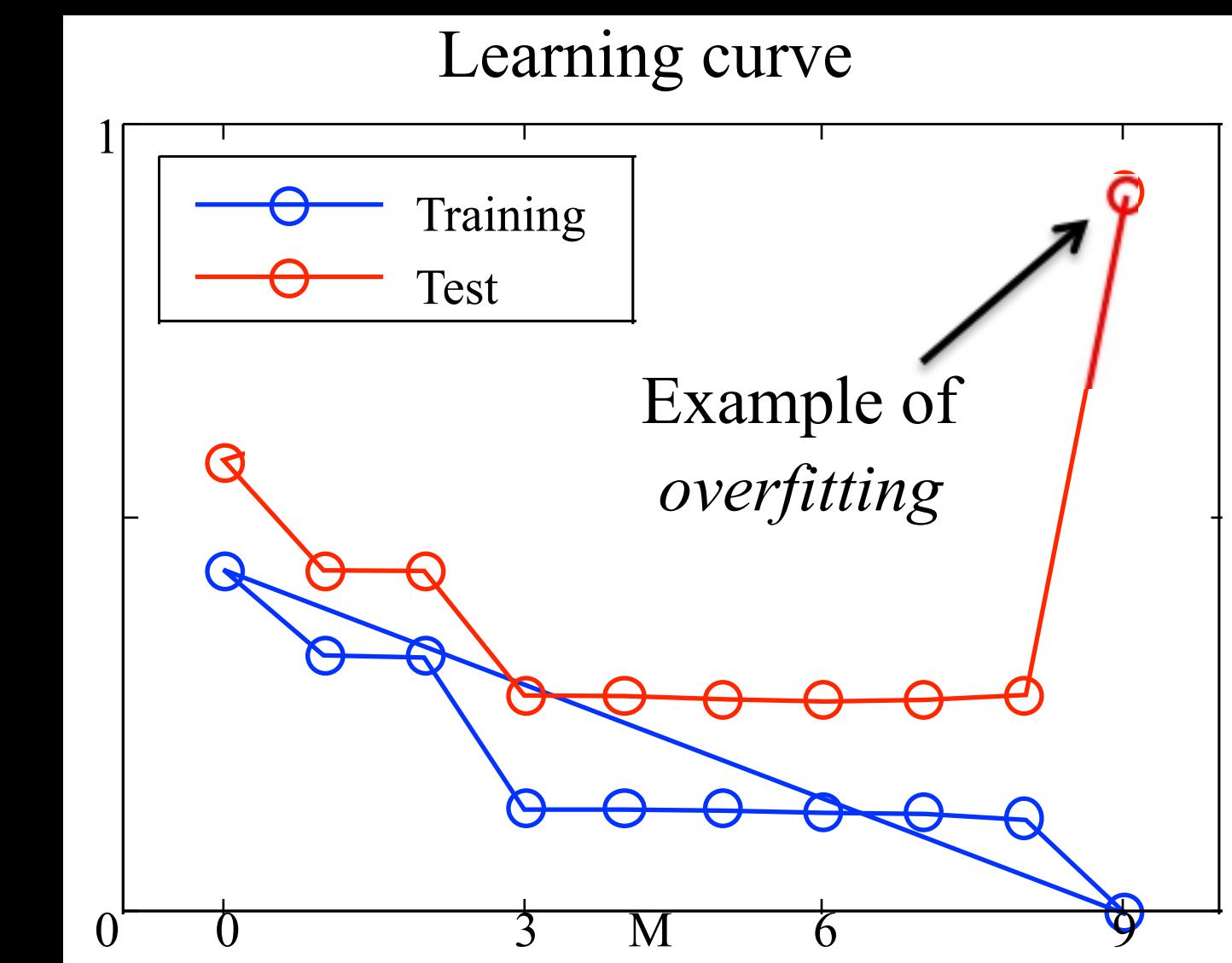
We measure error using a *loss function*

For regression, a common choice is squared loss:

$$L(y_i, f(x_i)) = (y_i - f(x_i))^2$$

The empirical loss of the function f applied to the training data is then:

$$\frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2$$



Key Issues in Machine Learning

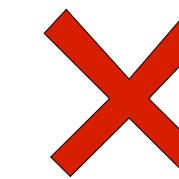
- How do we choose a hypothesis space?
 - Often we use **prior knowledge** to guide this choice
- How can we gauge the accuracy of a hypothesis on unseen data?
 - **Occam's razor:** use the *simplest* hypothesis consistent with data!
This will help us avoid overfitting.
 - **Learning theory** will help us quantify our ability to **generalize** as a function of the amount of training data and the hypothesis space
- How do we find the best hypothesis?
 - This is an **algorithmic** question, the main topic of computer science
- How to model applications as machine learning problems?
(engineering challenge)

Neural Network Basics

Perceptron algorithm

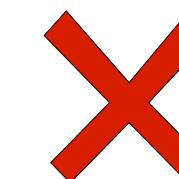
Binary classification

- Input: email
- Output: spam/ham
- Setup:
 - Get a large collection of example emails, each labeled “spam” or “ham”
 - Note: someone has to hand label all this data!
 - Want to learn to predict labels of new, future emails
- Features: The attributes used to make the ham / spam decision
 - Words: FREE!
 - Text Patterns: \$dd, CAPS
 - Non-text: SenderInContacts
 - ...



Dear Sir.

First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidential and top secret. ...



TO BE REMOVED FROM FUTURE
MAILINGS, SIMPLY REPLY TO THIS
MESSAGE AND PUT "REMOVE" IN THE
SUBJECT.

99 MILLION EMAIL ADDRESSES
FOR ONLY \$99



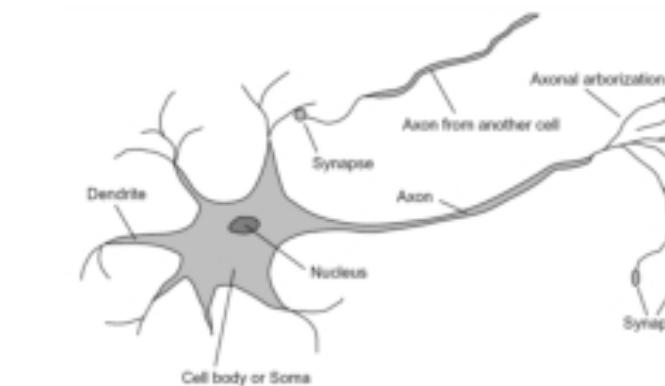
Ok, I know this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

The perceptron algorithm

- 1957: Perceptron algorithm invented by Rosenblatt

Wikipedia: “A handsome bachelor, he drove a classic MGA sports... for several years taught an interdisciplinary undergraduate honors course entitled "Theory of Brain Mechanisms" that drew students equally from Cornell's Engineering and Liberal Arts colleges...this course was a melange of ideas .. experimental brain surgery on epileptic patients while conscious, experiments on .. the visual cortex of cats, ... analog and digital electronic circuits that modeled various details of neuronal behavior (i.e. the perceptron itself, as a machine).”

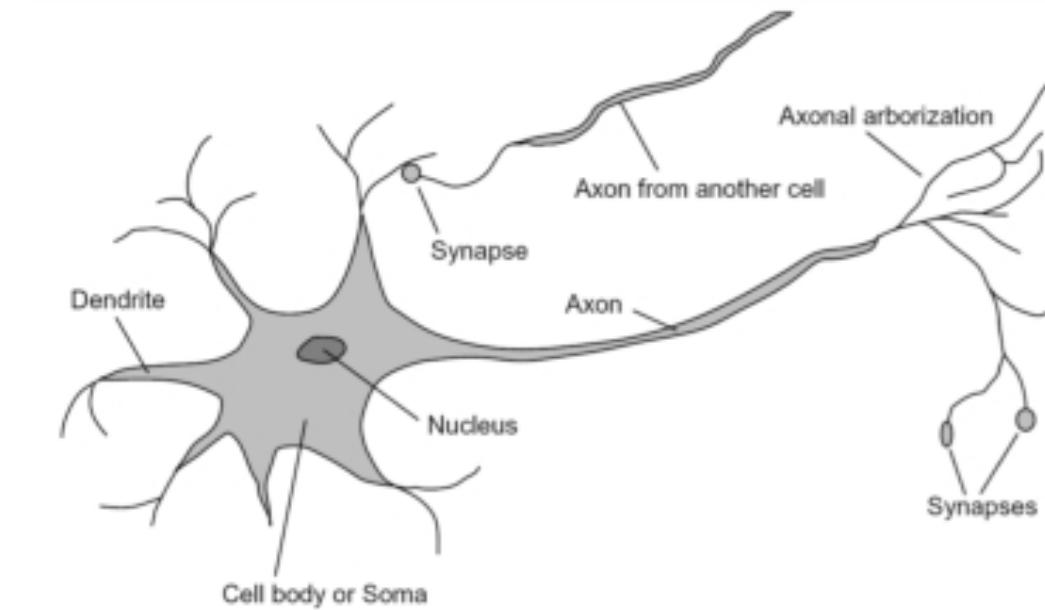
- Built on work of Hebb (1949); also developed by Widrow-Hoff (1960)
- 1960: Perceptron Mark 1 Computer – hardware implementation
- 1969: Minsky & Papert book shows perceptrons limited to *linearly separable* data
- 1970’s: Learning methods for two-layer neural networks



[William Cohen]

Linear Classifiers

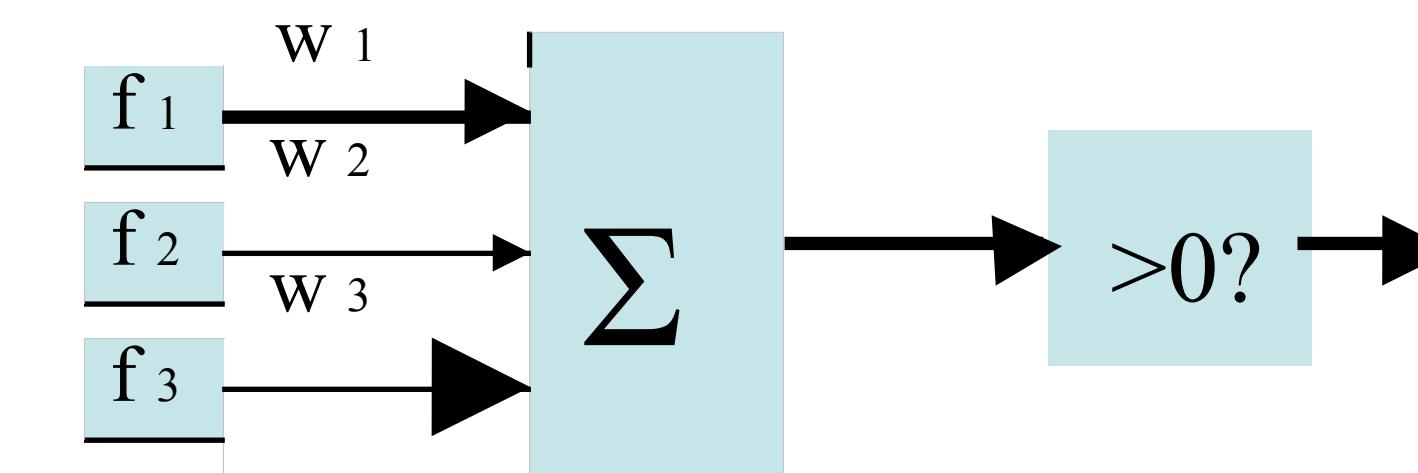
- Inputs are **feature values**
- Each feature has a **weight**
- Sum is the **activation**



Important note: changing notation!

$$\text{activation}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- If the activation is:
 - Positive, output *class 1*
 - Negative, output *class 2*



Example: Spam

- Imagine 3 features (spam is “positive” class):

- free (number of occurrences of “free”)
- money (occurrences of “money”)
- BIAS (intercept, always has value 1)

x
“ free money”

$f(x)$	
BIAS	: 1
free	: 1
money	: 1
...	

w

BIAS	: -3
free	: 4
money	: 2
...	

$$\begin{aligned} & w \cdot f(x) \\ & \sum_i w_i \cdot f_i(x) \\ & (1)(-3) + \\ & (1)(4) + \\ & (1)(2) + \\ & \dots \\ & = 3 \end{aligned}$$

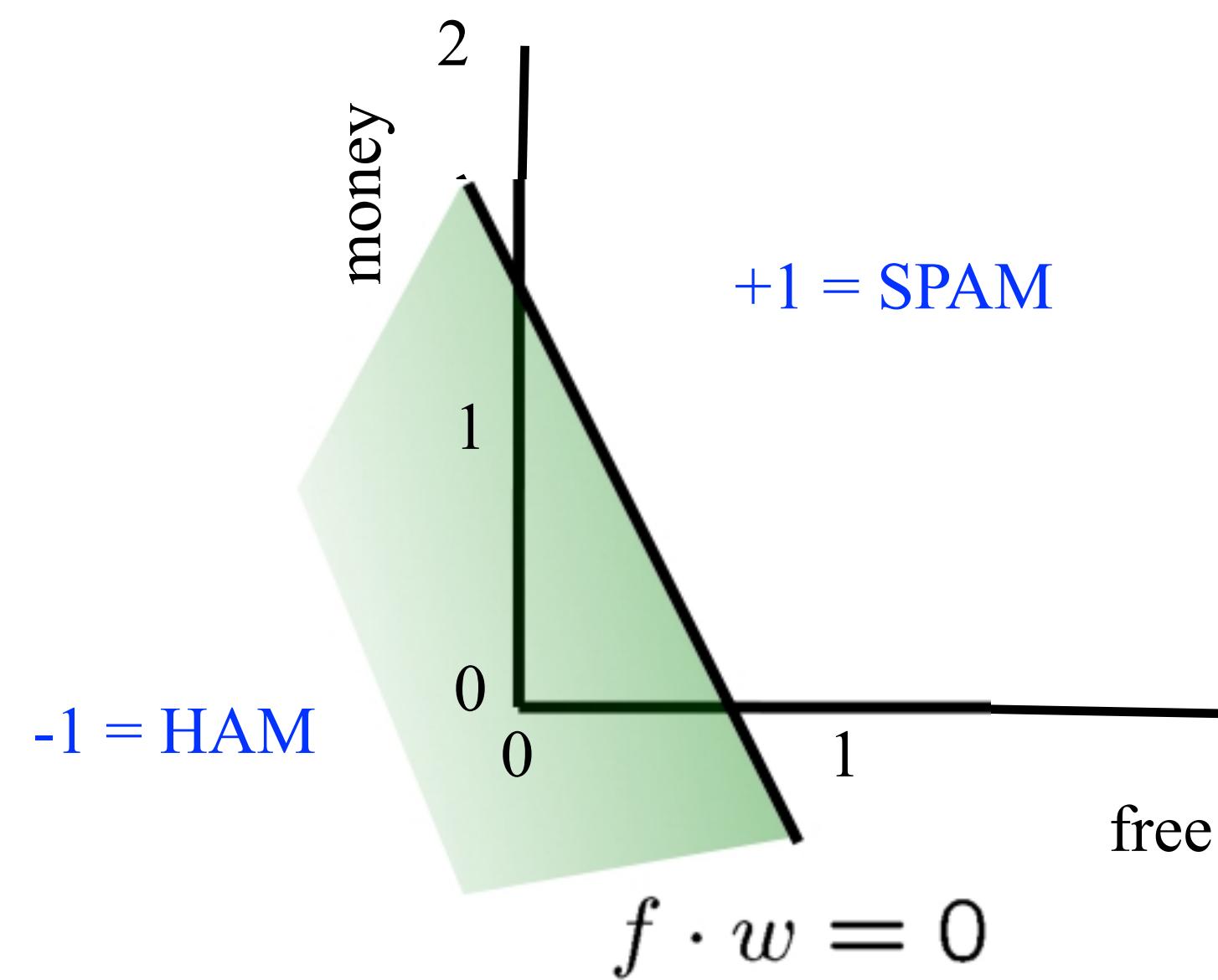
$w.f(x) > 0$ SPAM!!!

Binary Decision Rule

- In the space of feature vectors
 - Examples are points
 - Any weight vector is a hyperplane
 - One side corresponds to $Y=+1$
 - Other corresponds to $Y=-1$

w

BIAS : -3
free : 4
money : 2
...



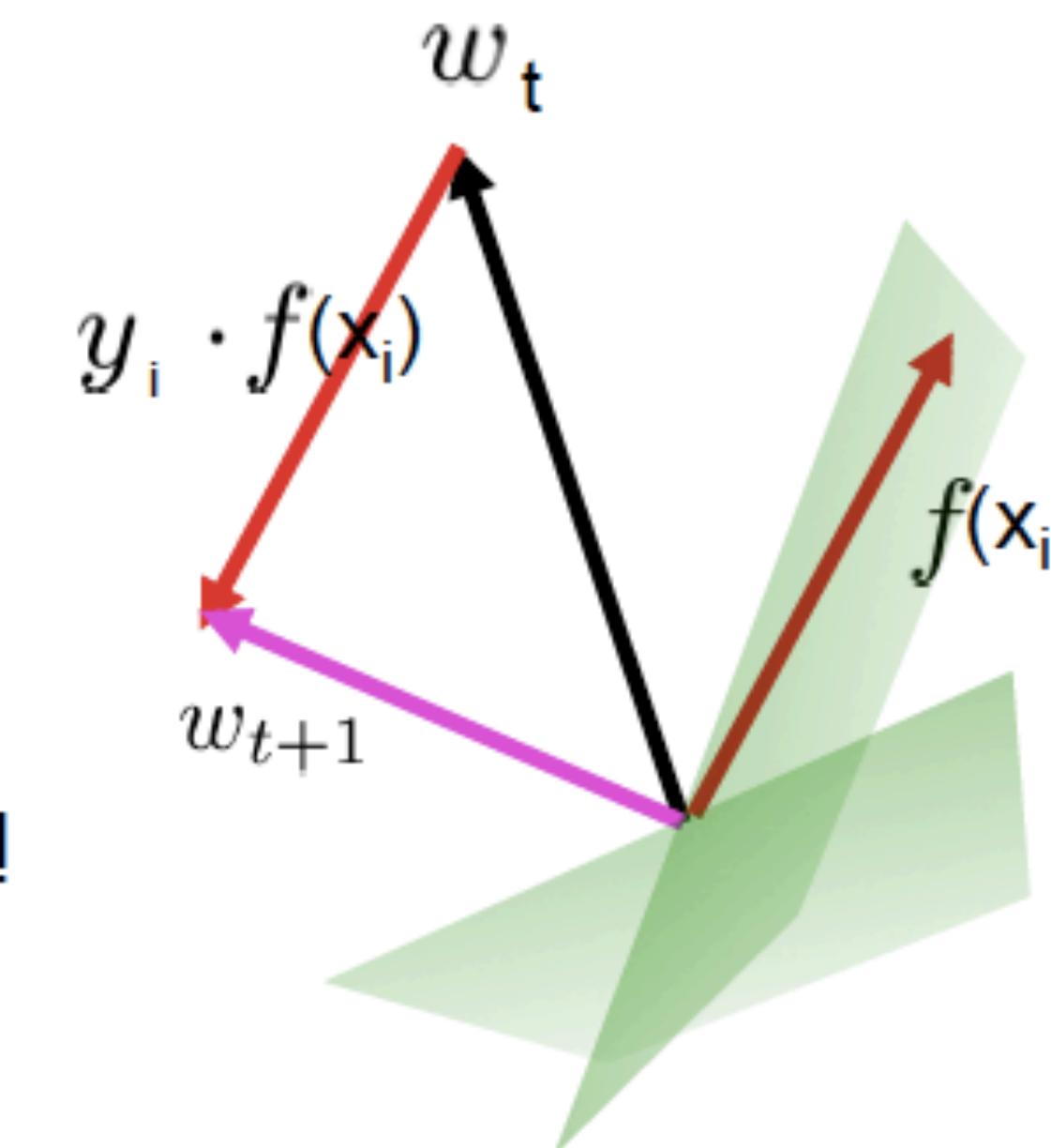
The Perception algorithm

- Start with weight vector = $\vec{0}$
- For each training instance (x_i, y_i) :
 - Classify with current weights

$$y = \begin{cases} +1 & \text{if } w \cdot f(x_i) \geq 0 \\ -1 & \text{if } w \cdot f(x_i) < 0 \end{cases}$$

- If correct (i.e., $y=y_i$), no change!
- If wrong: update

$$w = w + y_i f(x_i)$$



The Perceptron Algorithm:

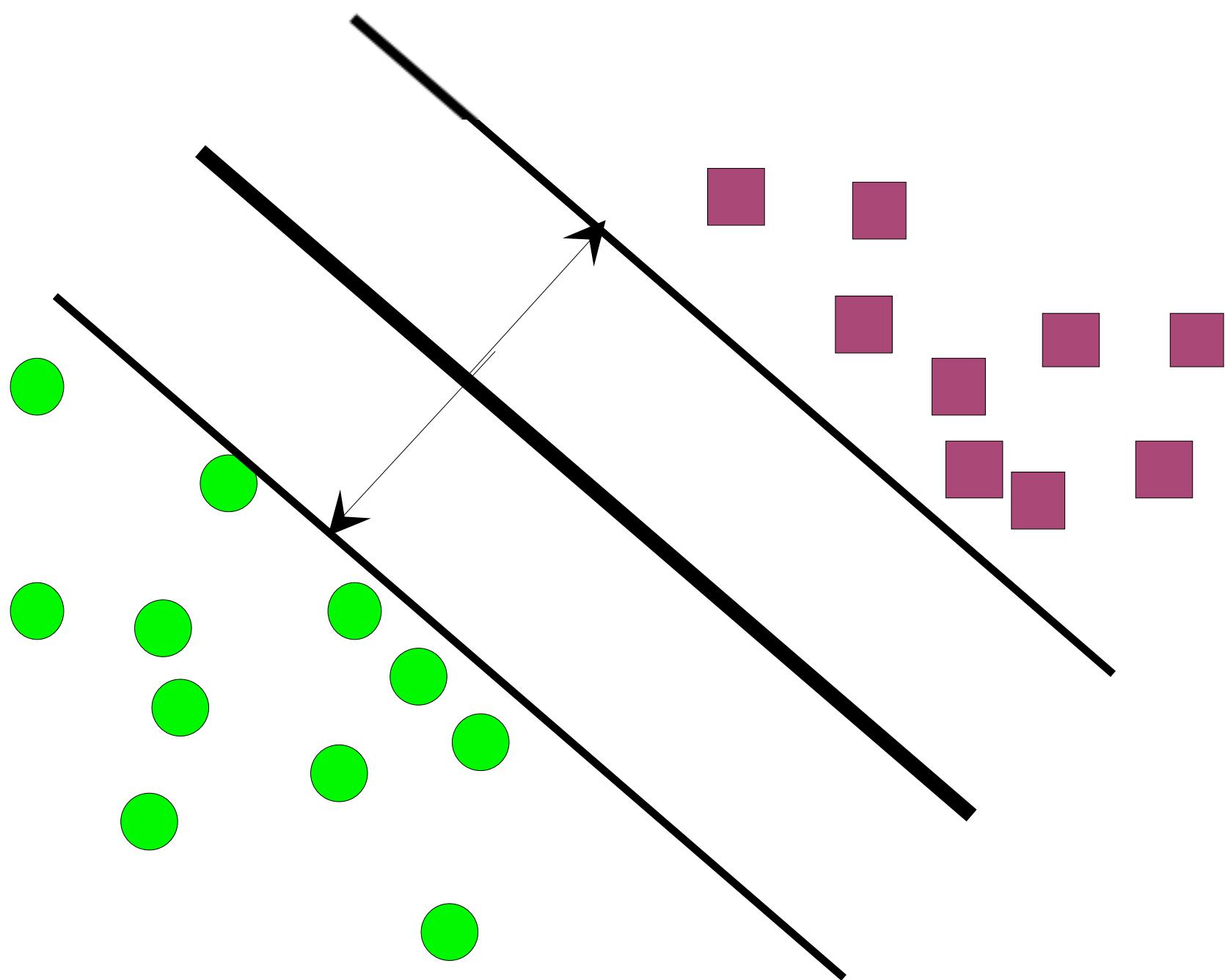
1. Start with the all-zeroes weight vector $\mathbf{w}_1 = \mathbf{0}$, and initialize t to 1. Also let's automatically scale all examples \mathbf{x} to have (Euclidean) length 1, since this doesn't affect which side of the plane they are on.
2. Given example \mathbf{x} , predict positive iff $\mathbf{w}_t \cdot \mathbf{x} > 0$.
3. On a mistake, update as follows:
 - Mistake on positive: $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \mathbf{x}$.
 - Mistake on negative: $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \mathbf{x}$.

$$t \leftarrow t + 1.$$

Linearly Separable

$\exists \mathbf{w}$ such that $\forall t$

$$y_t(\mathbf{w} \cdot \mathbf{x}_t) \geq \gamma > 0$$



Called the *functional margin*
with respect to the training set

Equivalently, for $y_t = +1$,

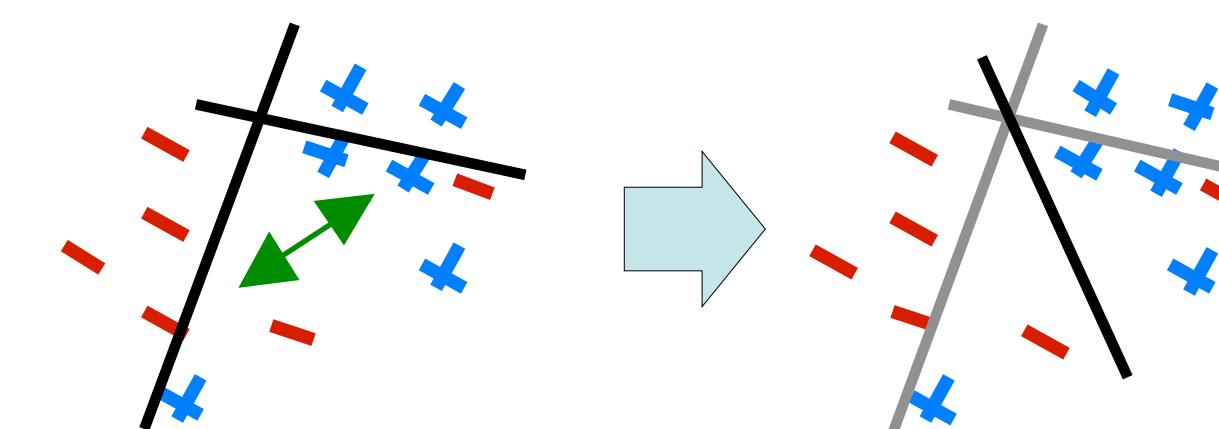
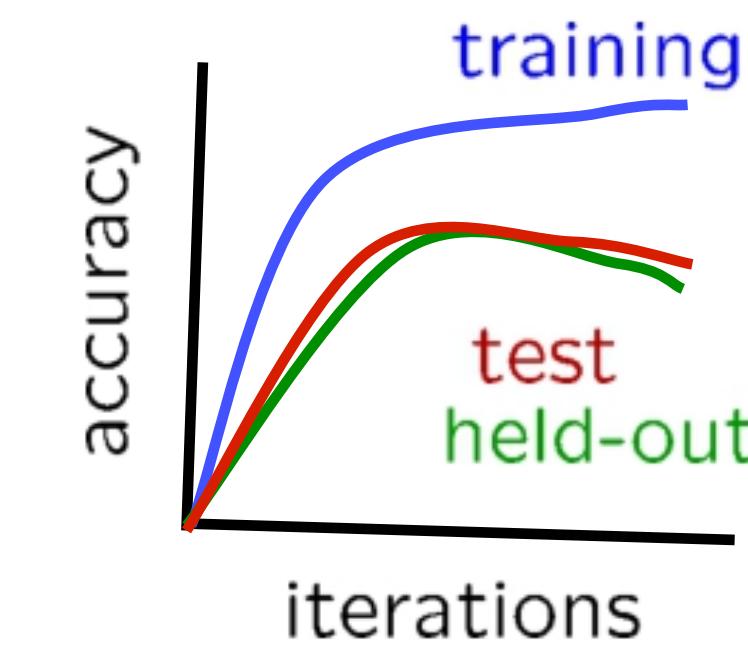
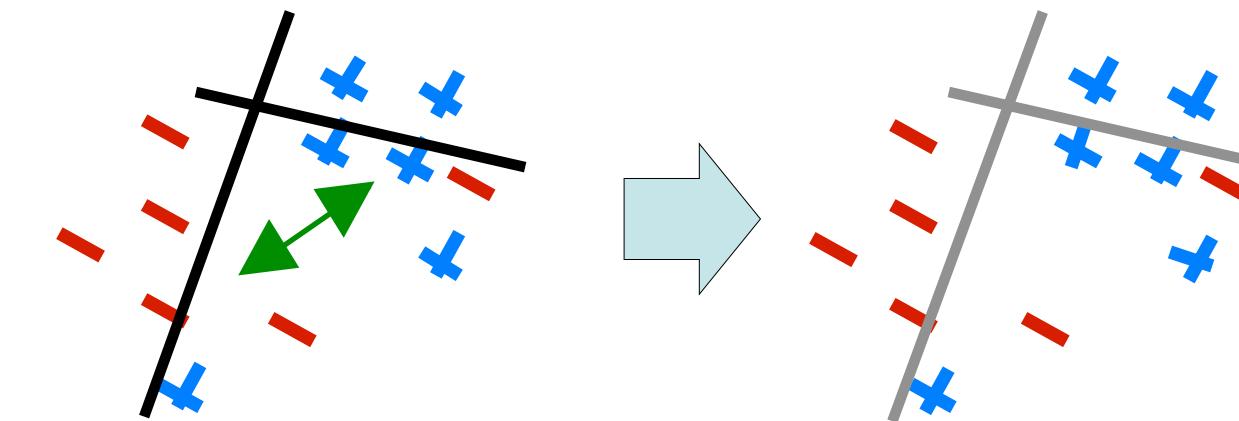
$$\mathbf{w} \cdot \mathbf{x}_t \geq \gamma$$

and for $y_t = -1$,

$$\mathbf{w} \cdot \mathbf{x}_t \leq -\gamma$$

Problems with the perceptron algorithm

- If the data isn't linearly separable, no guarantees of convergence or training accuracy
- Even if the training data is linearly separable, perceptron can overfit
- **Averaged** perceptron is an algorithmic modification that helps with both issues
 - Averages the weight vectors across all iterations

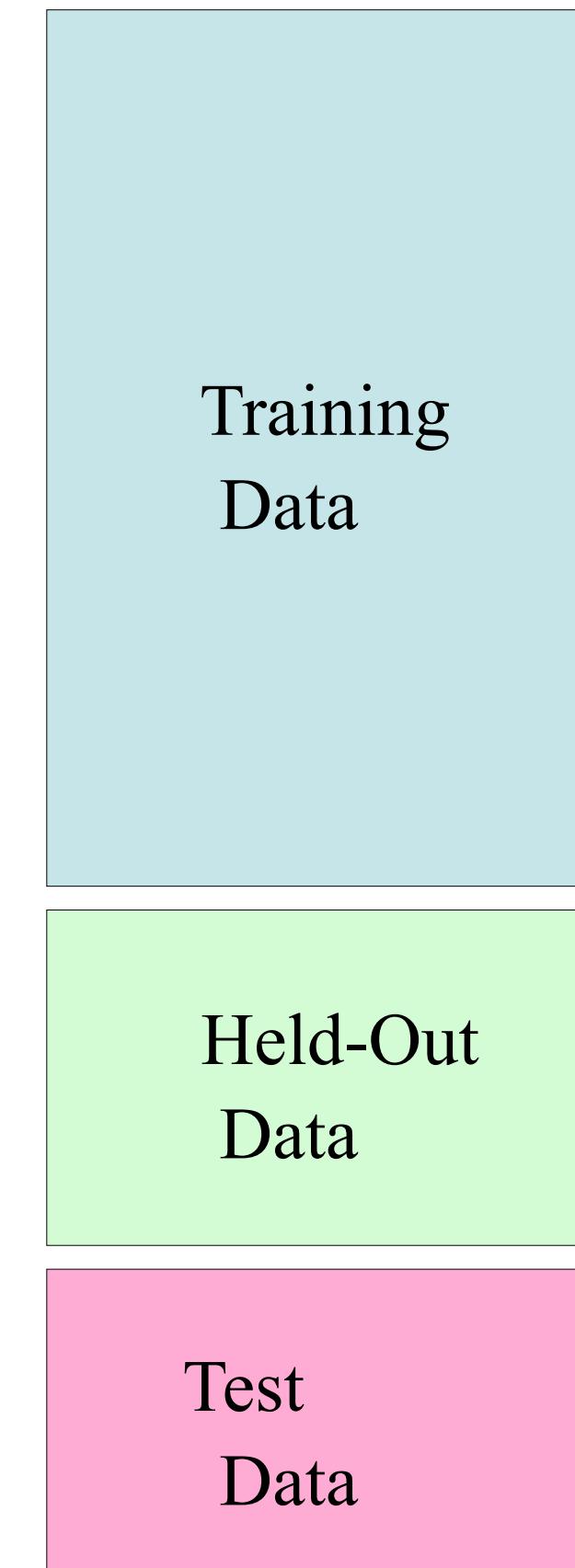


ML Methodology

- **Data:** labeled instances, e.g. emails marked spam/ham
 - Training set
 - Held out set (sometimes call Validation set)
 - Test set

Randomly allocate to these three, e.g. 60/20/20

- **Features:** attribute-value pairs which characterize each x
- **Experimentation cycle**
 - Select a hypothesis f
(Tune hyper parameters on held-out or *validation* set)
 - Compute accuracy of test set
 - Very important: never “peek” at the test set!
- **Evaluation**
 - Accuracy: fraction of instances predicted correctly



Multi-Layer Perceptron (MLP)

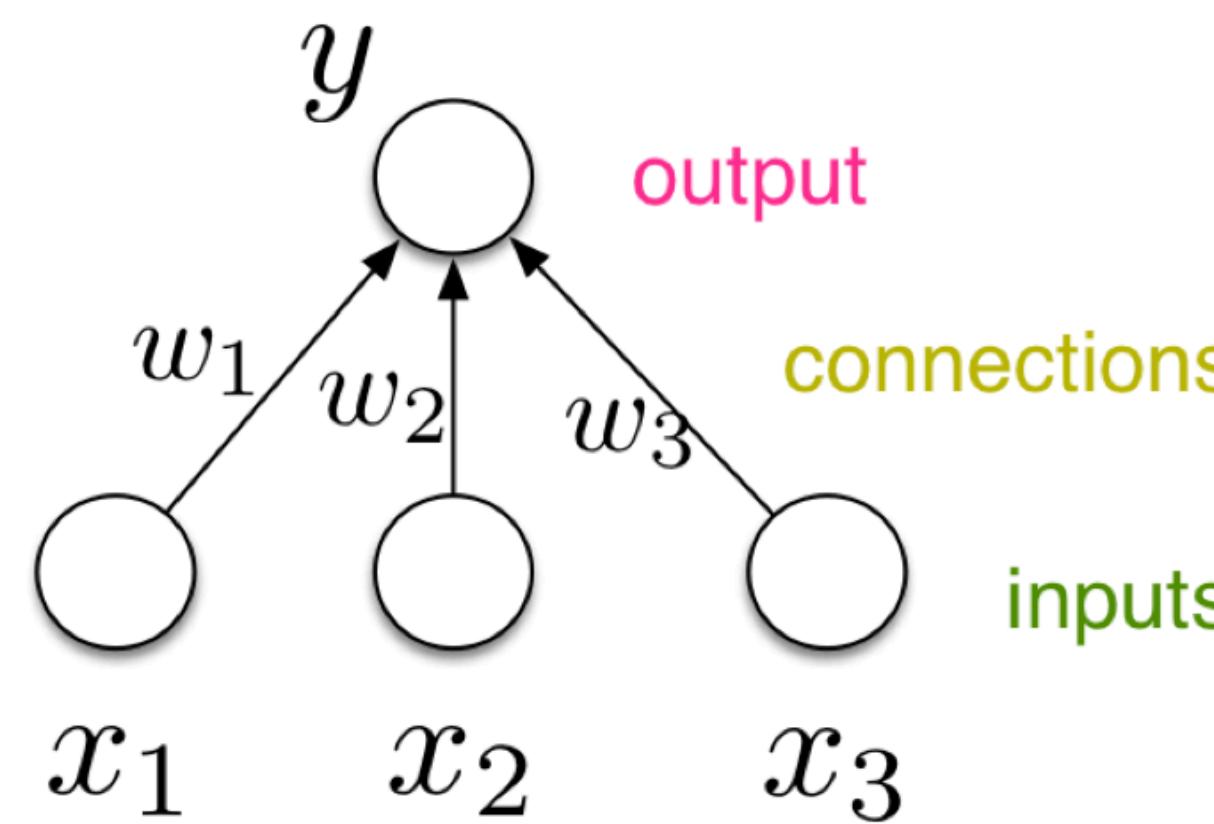
What are neural networks?

Why neural nets?

- inspiration from the brain
 - proof of concept that a neural architecture can see and hear!
- very effective across a range of applications (vision, text, speech, medicine, robotics, etc.)
- widely used in both academia and the tech industry
- powerful software frameworks (PyTorch, TensorFlow, etc.) let us quickly implement sophisticated algorithms

What are neural networks?

- Most of the biological details aren't essential, so we use vastly simplified models of neurons.
- While neural nets originally drew inspiration from the brain, nowadays we mostly think about math, statistics, etc.



$$y = \phi(\mathbf{w}^T \mathbf{x} + b)$$

Annotations for the equation:

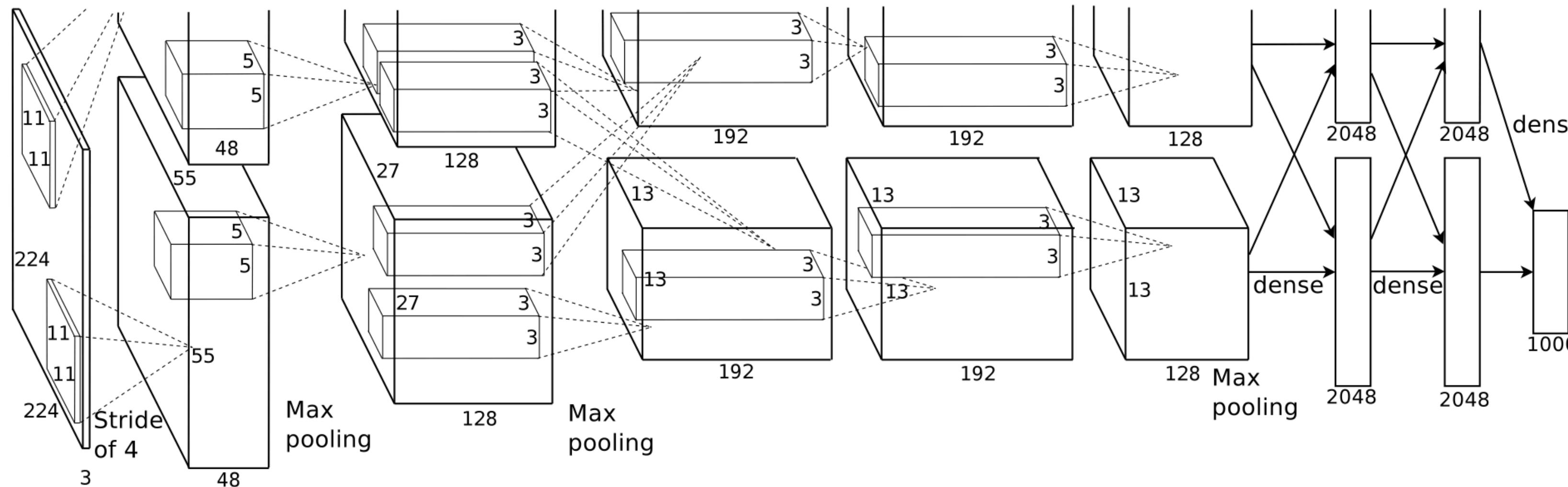
- "output" points to the variable y .
- "weights" points to the term \mathbf{w}^T .
- "bias" points to the term b .
- "activation function" points to the term ϕ .
- "inputs" points to the term \mathbf{x} .

- Neural networks are collections of thousands (or millions) of these simple processing units that together perform useful computations.

“Deep learning”

Deep learning: many layers (stages) of processing

E.g. this network which recognizes objects in images:

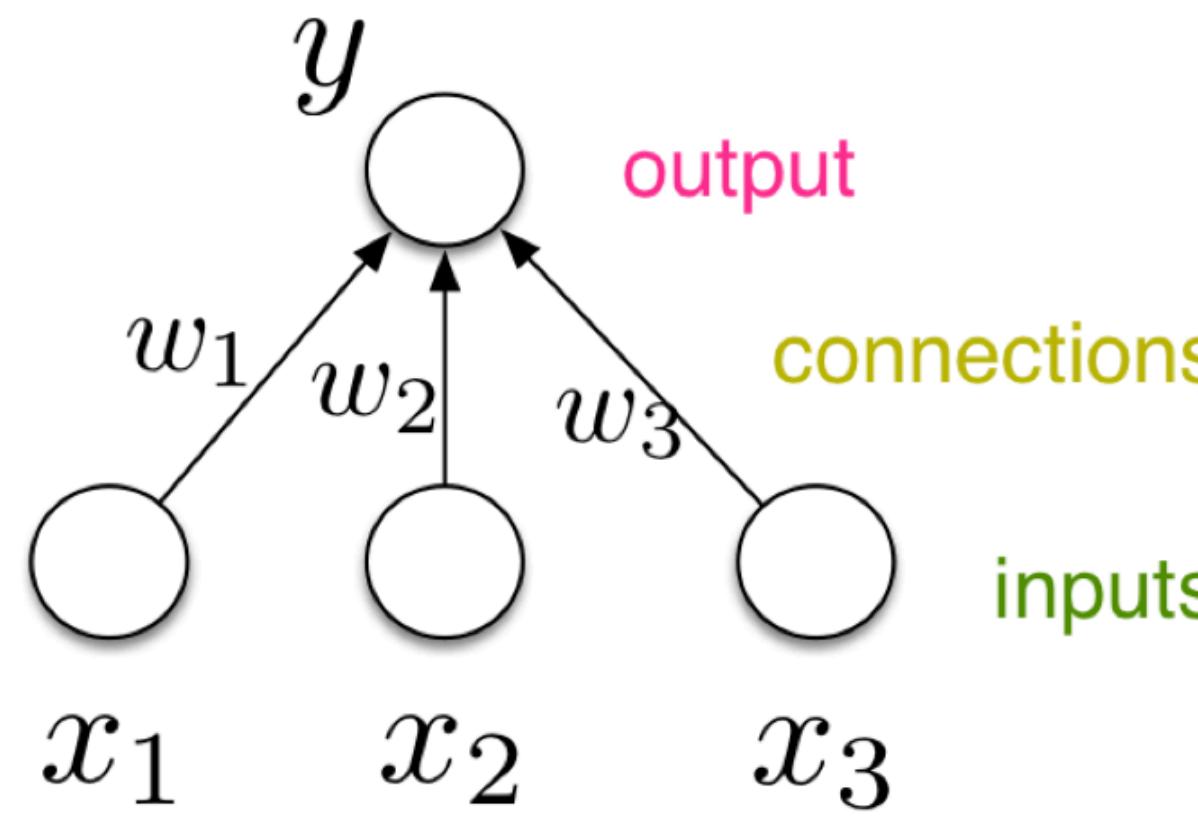


(Krizhevsky et al., 2012)

Each of the boxes consists of many neuron-like units similar to the one on the previous slide!

Overview

- Recall the simple neuron-like unit:



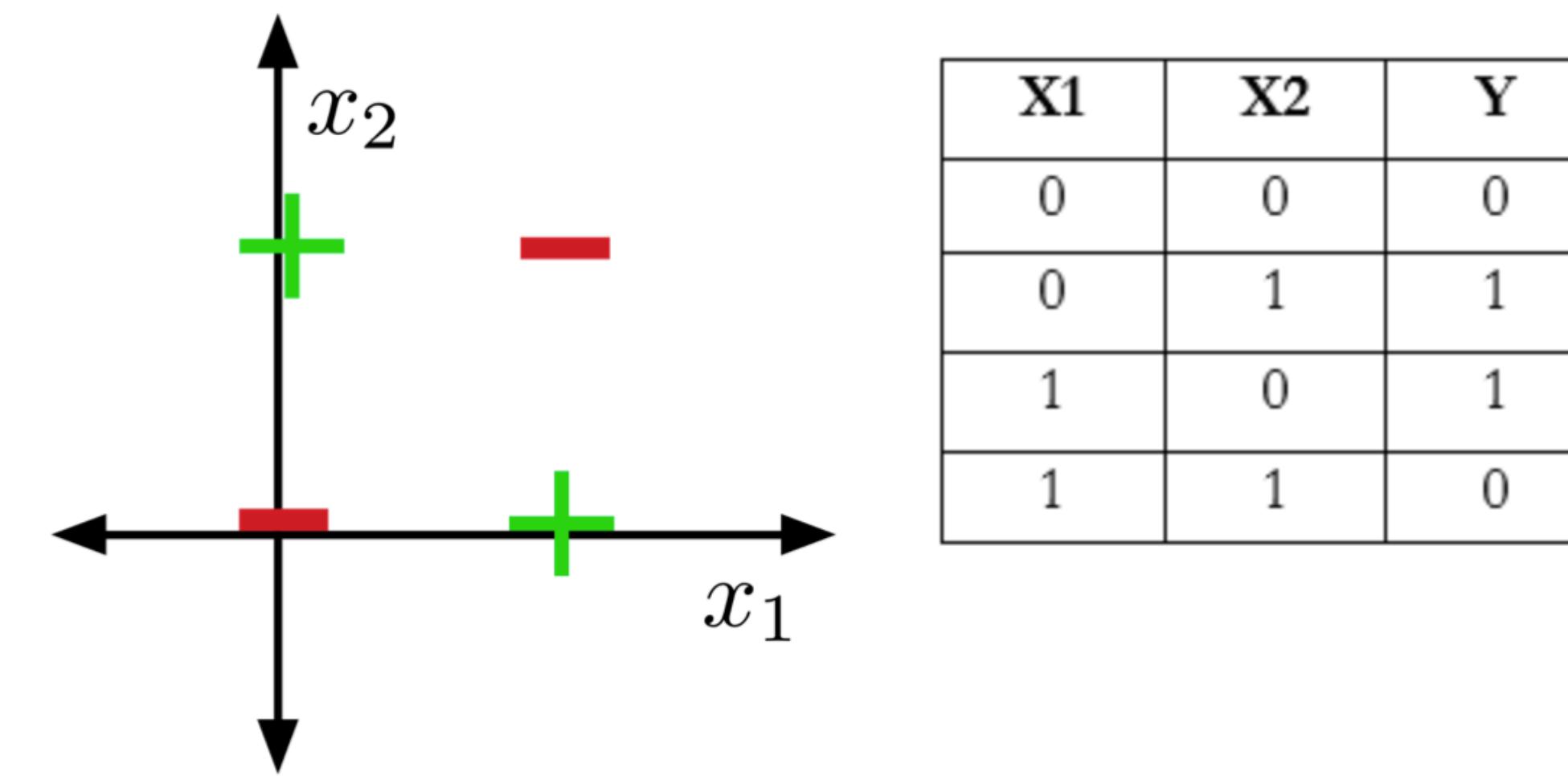
An equation for a simple neuron-like unit: $y = \phi(\mathbf{w}^\top \mathbf{x} + b)$. The equation is annotated with arrows pointing to its components:

- A pink arrow points down to the variable y , labeled "output".
- A blue arrow points down to the term $\mathbf{w}^\top \mathbf{x}$, labeled "weights".
- A blue arrow points down to the term b , labeled "bias".
- A red arrow points up to the activation function symbol ϕ , labeled "activation function".
- A green arrow points up to the input vector \mathbf{x} , labeled "inputs".

- Linear regression and logistic regression can each be viewed as a single unit.
- These units are much more powerful if we connect many of them into a neural network.

Limits of Linear Classification

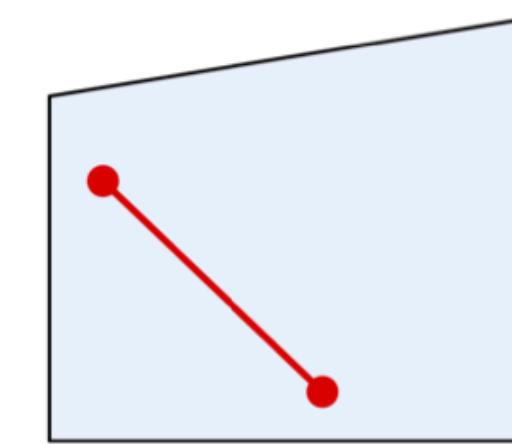
- Single neurons (linear classifiers) are very limited in expressive power.
- **XOR** is a classic example of a function that's not linearly separable.



- There's an elegant proof using convexity.

Limits of Linear Classification

Convex Sets



- A set \mathcal{S} is **convex** if any line segment connecting points in \mathcal{S} lies entirely within \mathcal{S} . Mathematically,

$$\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{S} \implies \lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2 \in \mathcal{S} \quad \text{for } 0 \leq \lambda \leq 1.$$

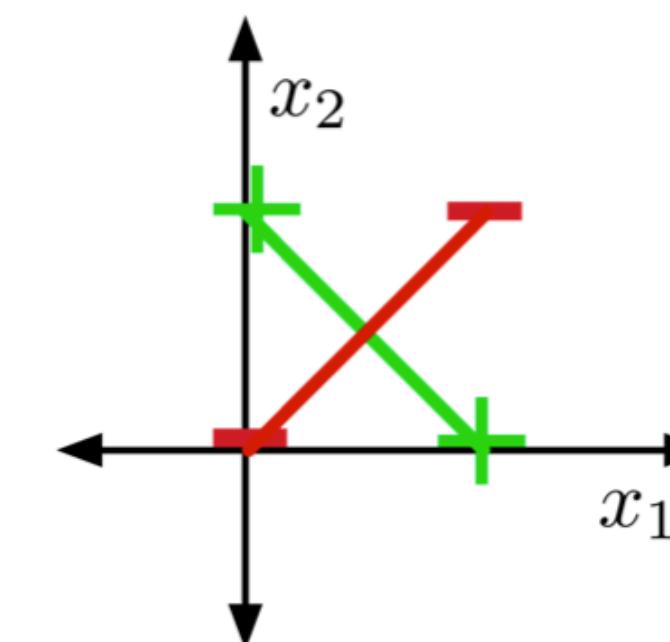
- A simple inductive argument shows that for $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathcal{S}$, **weighted averages**, or **convex combinations**, lie within the set:

$$\lambda_1 \mathbf{x}_1 + \dots + \lambda_N \mathbf{x}_N \in \mathcal{S} \quad \text{for } \lambda_i > 0, \lambda_1 + \dots + \lambda_N = 1.$$

Limits of Linear Classification

Showing that XOR is not linearly separable

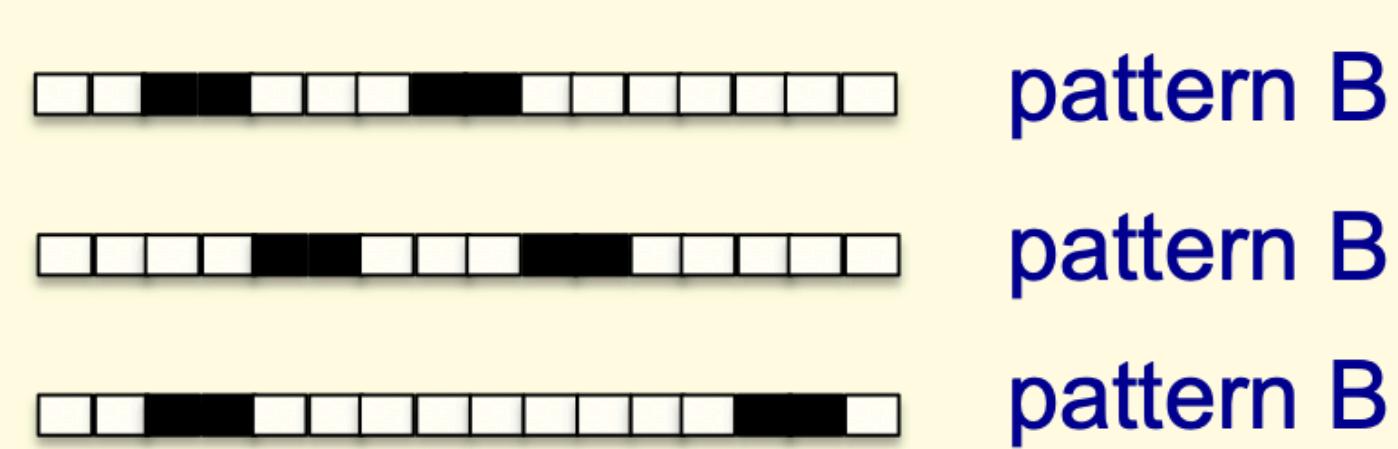
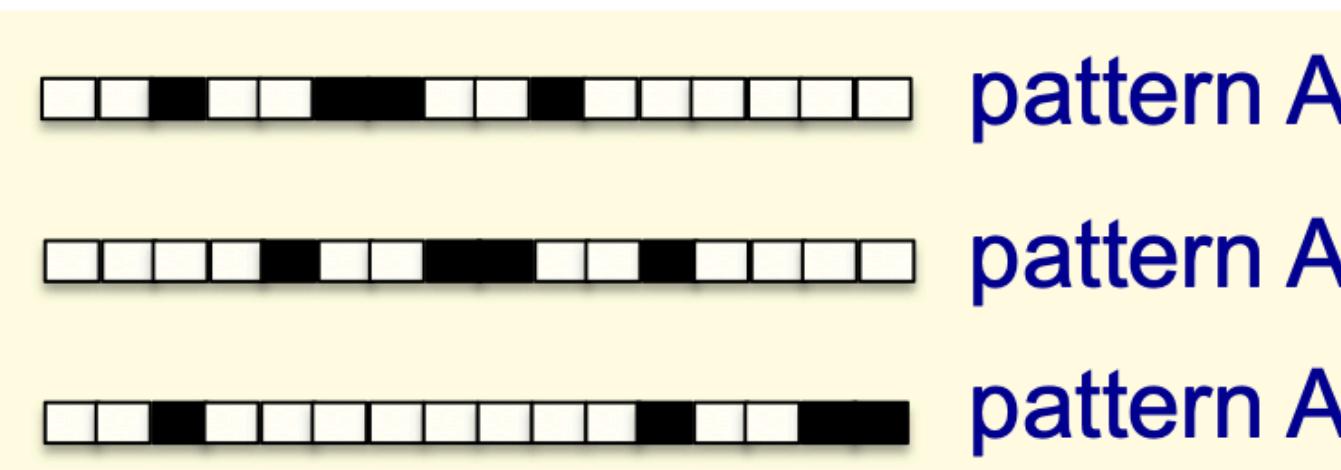
- Half-spaces are obviously convex.
- Suppose there were some feasible hypothesis. If the positive examples are in the positive half-space, then the green line segment must be as well.
- Similarly, the red line segment must lie within the negative half-space.



- But the intersection can't lie in both half-spaces. Contradiction!

Limits of Linear Classification

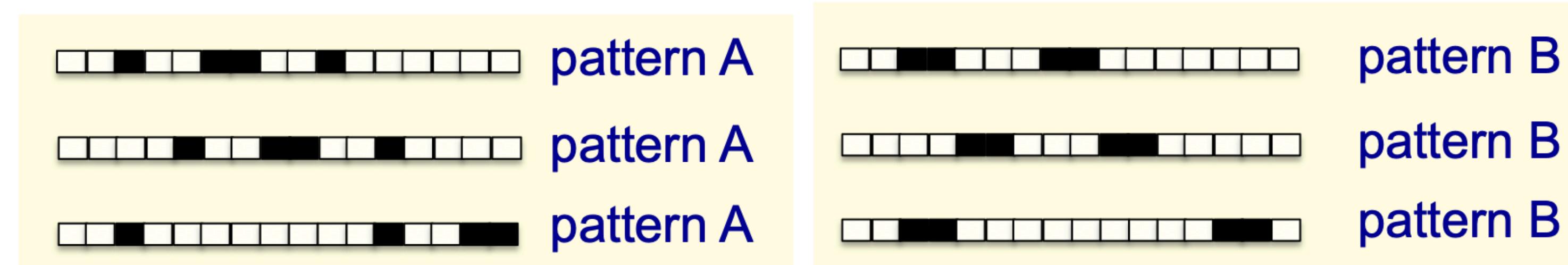
A more troubling example



- These images represent 16-dimensional vectors. White = 0, black = 1.
- Want to distinguish patterns A and B in all possible translations (with wrap-around)
- Translation invariance is commonly desired in vision!

Limits of Linear Classification

A more troubling example



- These images represent 16-dimensional vectors. White = 0, black = 1.
- Want to distinguish patterns A and B in all possible translations (with wrap-around)
- Translation invariance is commonly desired in vision!
- Suppose there's a feasible solution. The average of all translations of A is the vector $(0.25, 0.25, \dots, 0.25)$. Therefore, this point must be classified as A.
- Similarly, the average of all translations of B is also $(0.25, 0.25, \dots, 0.25)$. Therefore, it must be classified as B. Contradiction!

Limits of Linear Classification

- Sometimes we can overcome this limitation using feature maps, just like for linear regression. E.g., for **XOR**:

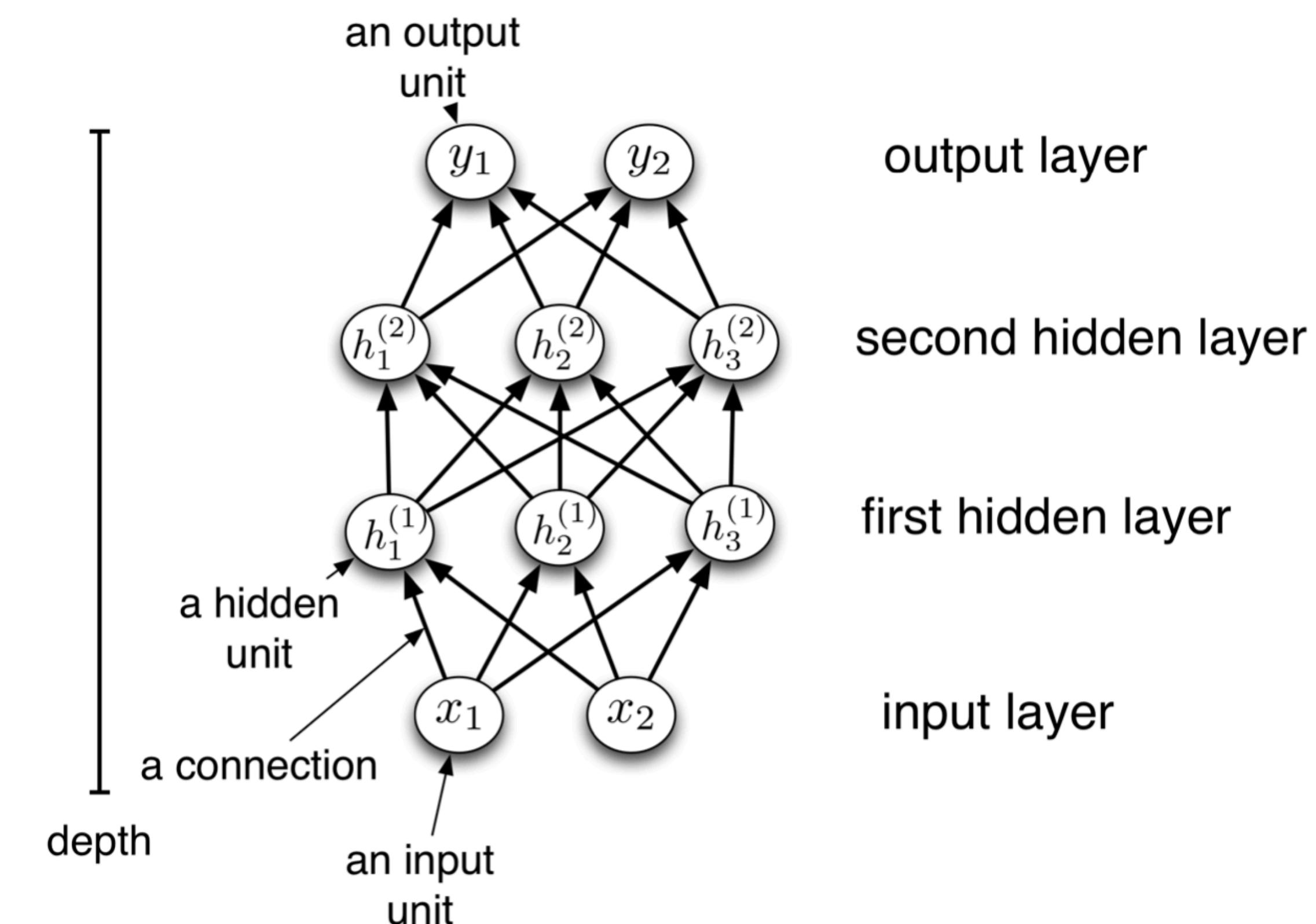
$$\psi(\mathbf{x}) = \begin{pmatrix} x_1 \\ x_2 \\ x_1x_2 \end{pmatrix}$$

x_1	x_2	$\phi_1(\mathbf{x})$	$\phi_2(\mathbf{x})$	$\phi_3(\mathbf{x})$	t
0	0	0	0	0	0
0	1	0	1	0	1
1	0	1	0	0	1
1	1	1	1	1	0

- This is linearly separable. (Try it!)
- Not a general solution: it can be hard to pick good basis functions.
Instead, we'll use neural nets to learn nonlinear hypotheses directly.

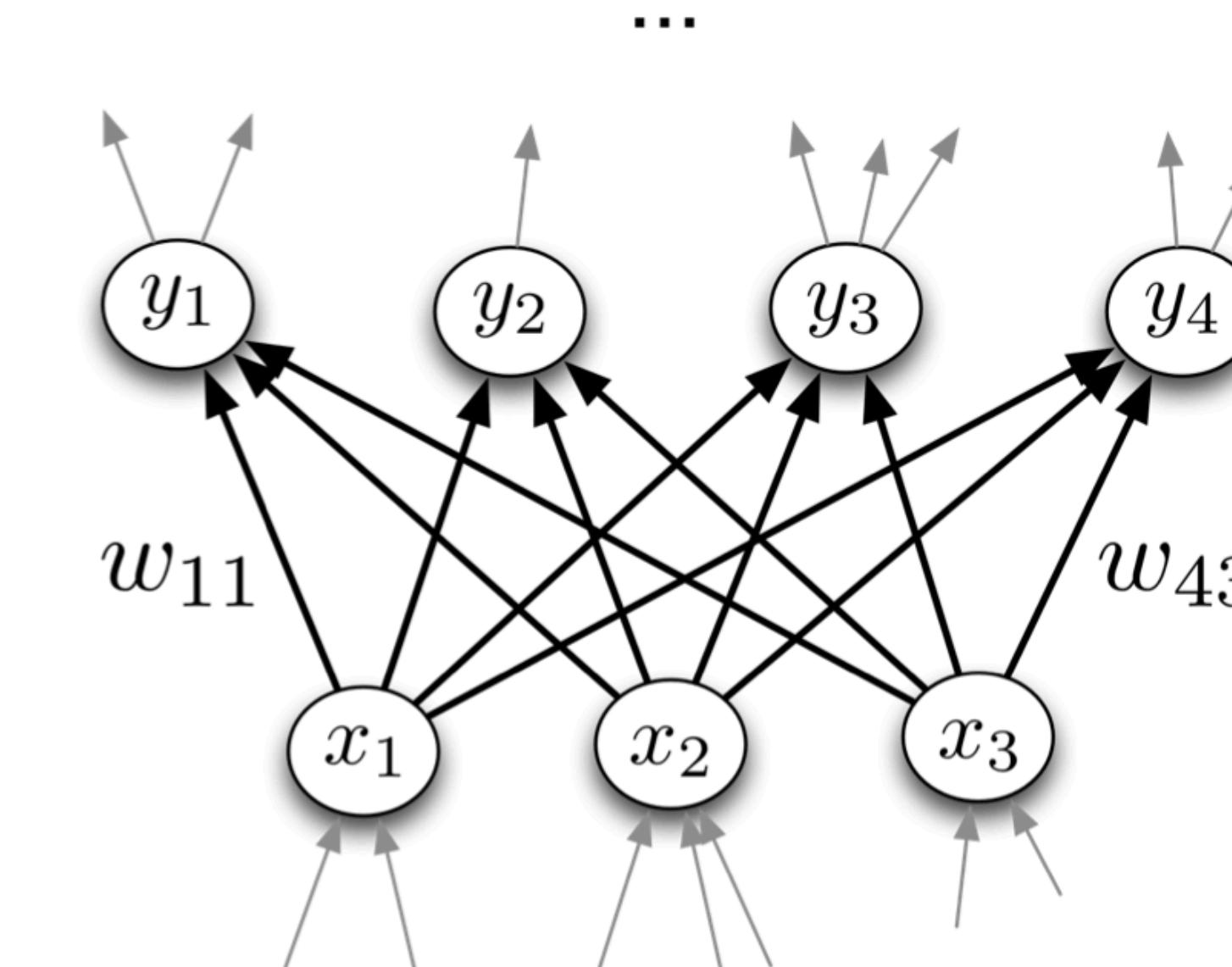
Multilayer Perceptrons

- We can connect lots of units together into a directed acyclic graph.
- This gives a **feed-forward neural network**. That's in contrast to **recurrent neural networks**, which can have cycles. (We'll talk about those later.)
- Typically, units are grouped together into layers.



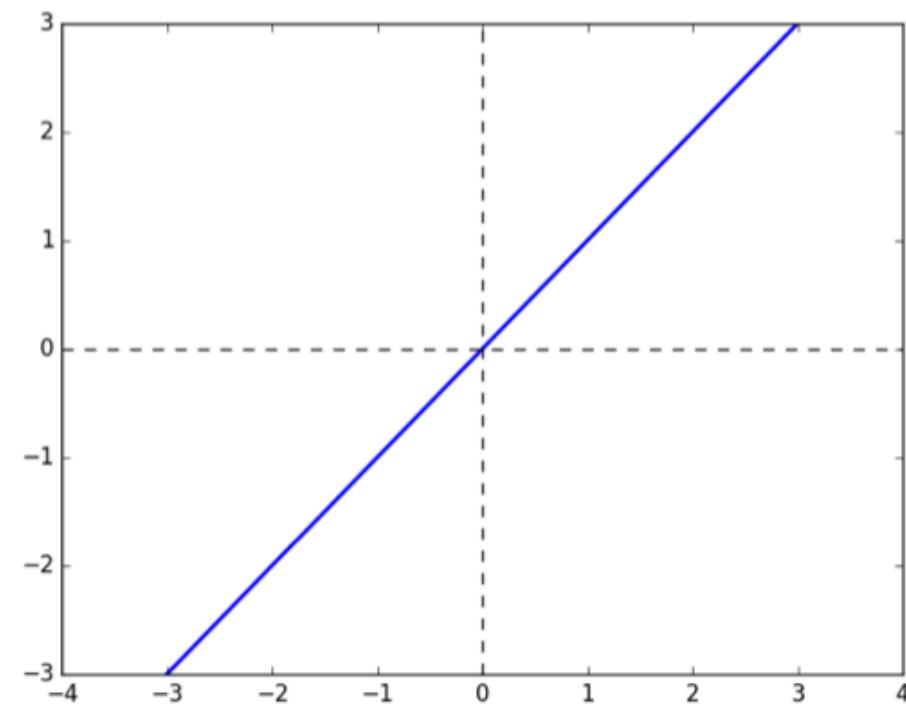
Multilayer Perceptrons

- Each layer connects N input units to M output units.
 - In the simplest case, all input units are connected to all output units. We call this a **fully connected layer**. We'll consider other layer types later.
 - Note: the inputs and outputs for a layer are distinct from the inputs and outputs to the network.
-
- Recall from softmax regression: this means we need an $M \times N$ weight matrix.
 - The output units are a function of the input units:
$$\mathbf{y} = f(\mathbf{x}) = \phi(\mathbf{Wx} + \mathbf{b})$$
 - A multilayer network consisting of fully connected layers is called a **multilayer perceptron**. Despite the name, it has nothing to do with perceptrons!



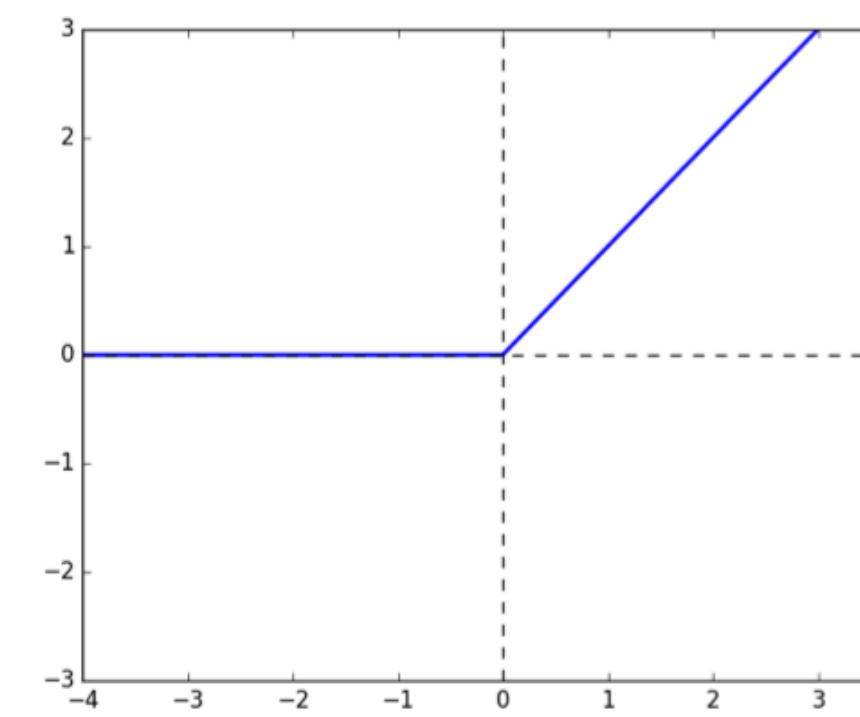
Multilayer Perceptrons

Some activation functions:



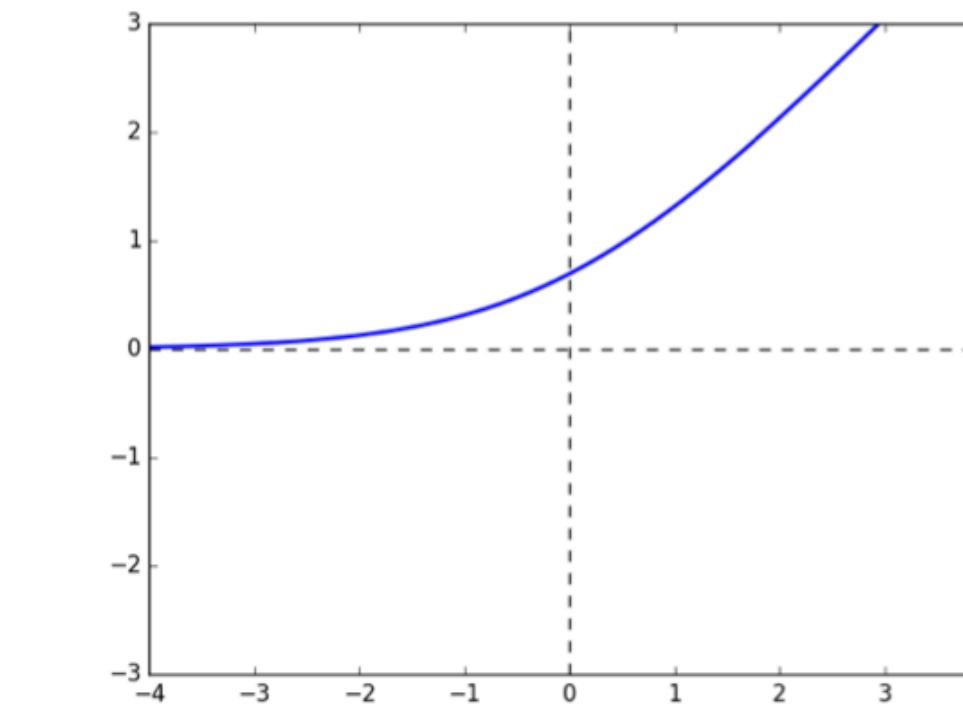
Linear

$$y = z$$



**Rectified Linear Unit
(ReLU)**

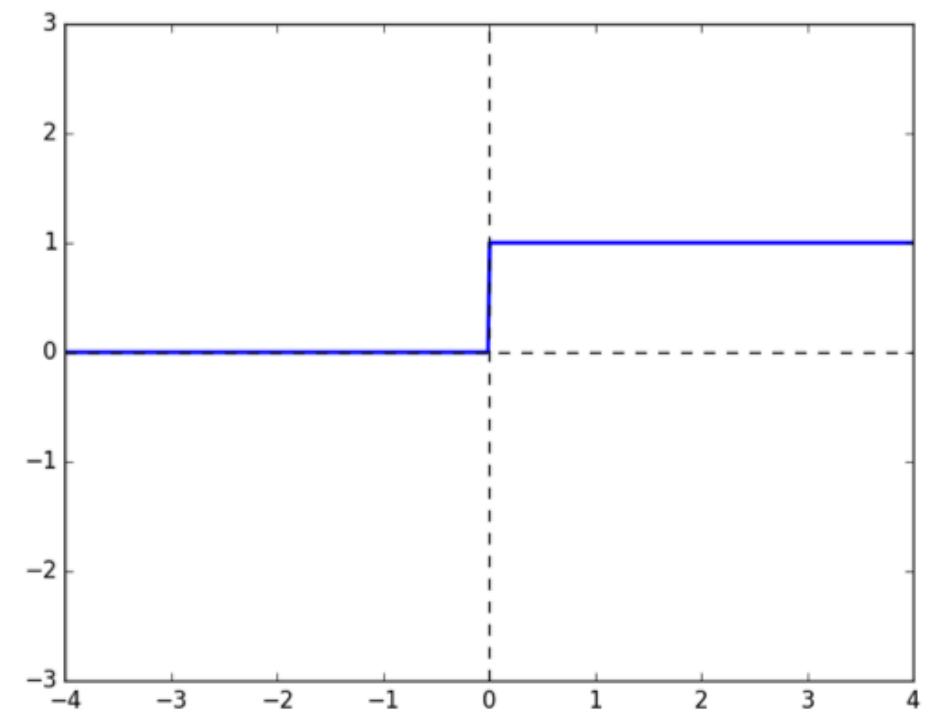
$$y = \max(0, z)$$



Soft ReLU

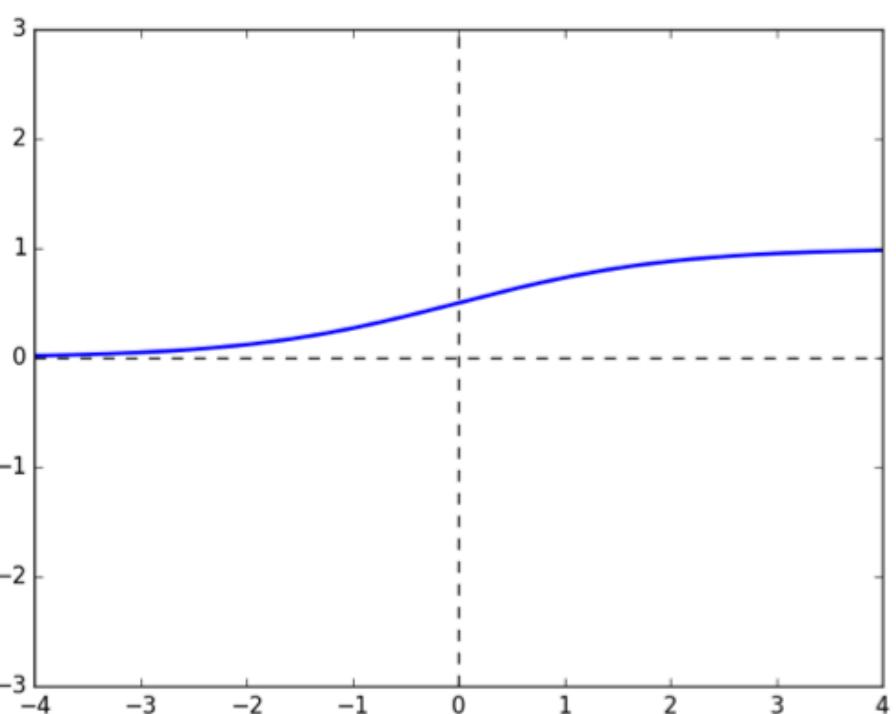
$$y = \log(1 + e^z)$$

Some activation functions:



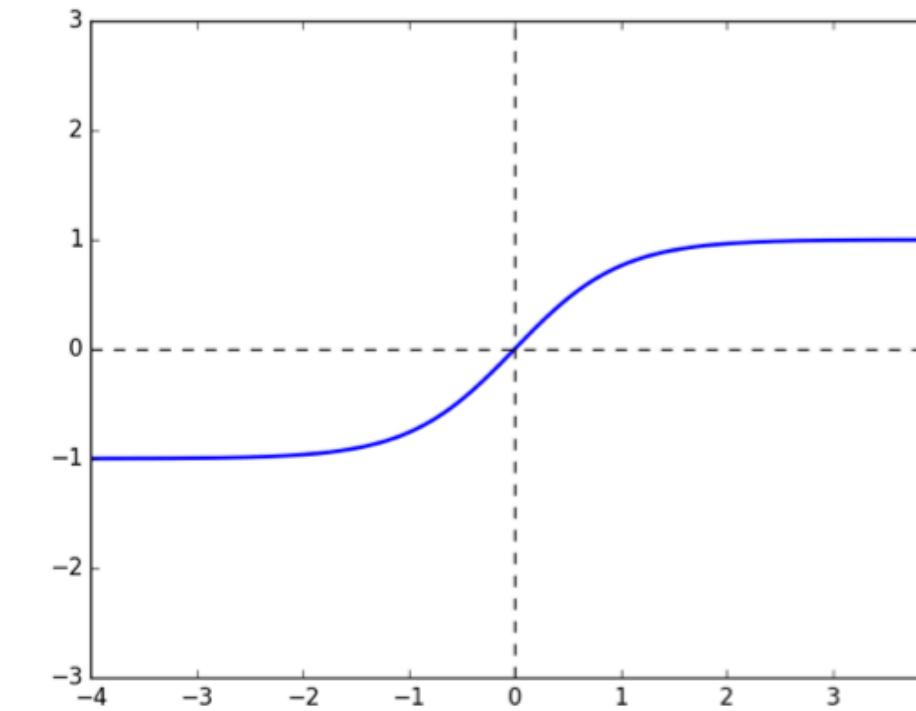
Hard Threshold

$$y = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$



Logistic

$$y = \frac{1}{1 + e^{-z}}$$



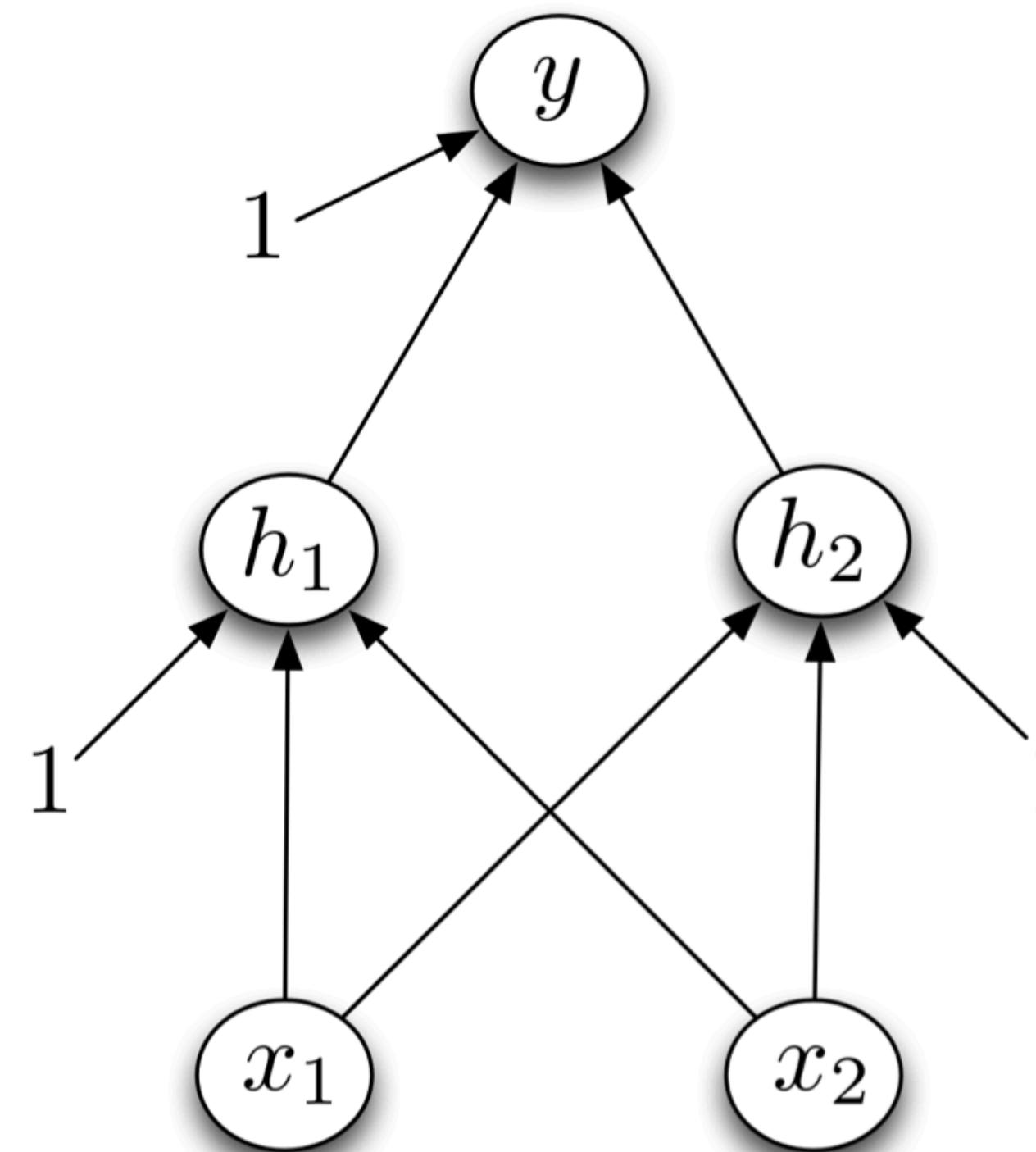
**Hyperbolic Tangent
(tanh)**

$$y = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Multilayer Perceptrons

Designing a network to compute XOR:

Assume hard threshold activation function



Multilayer Perceptrons

