Introduction to Artificial Intelligence and Machine Learning

Patrick J. Martin, Ph.D.

Cognitive Science and Artificial Intelligence Department

MITRE Corporation

GAIN

GENERATION℠ AI NEXUS

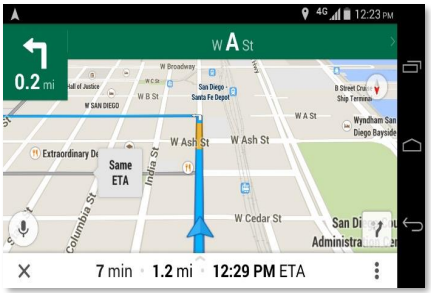
MITRE

Approved for public release. Distribution unlimited 19-01826-4

AI in the Real World

AI technology is incorporated into current consumer products…

Personal Assistants



Mapping Software

Web Platforms/Content Delivery

…and is foundational to developing industries

Autonomous Vehicles

Internet of Things

GAIN MITRE

Your Turn!

Write down your thoughts on AI! What do you know about it?

Applications: FaceApp  , Autocomplete

Mimic "intelligence"

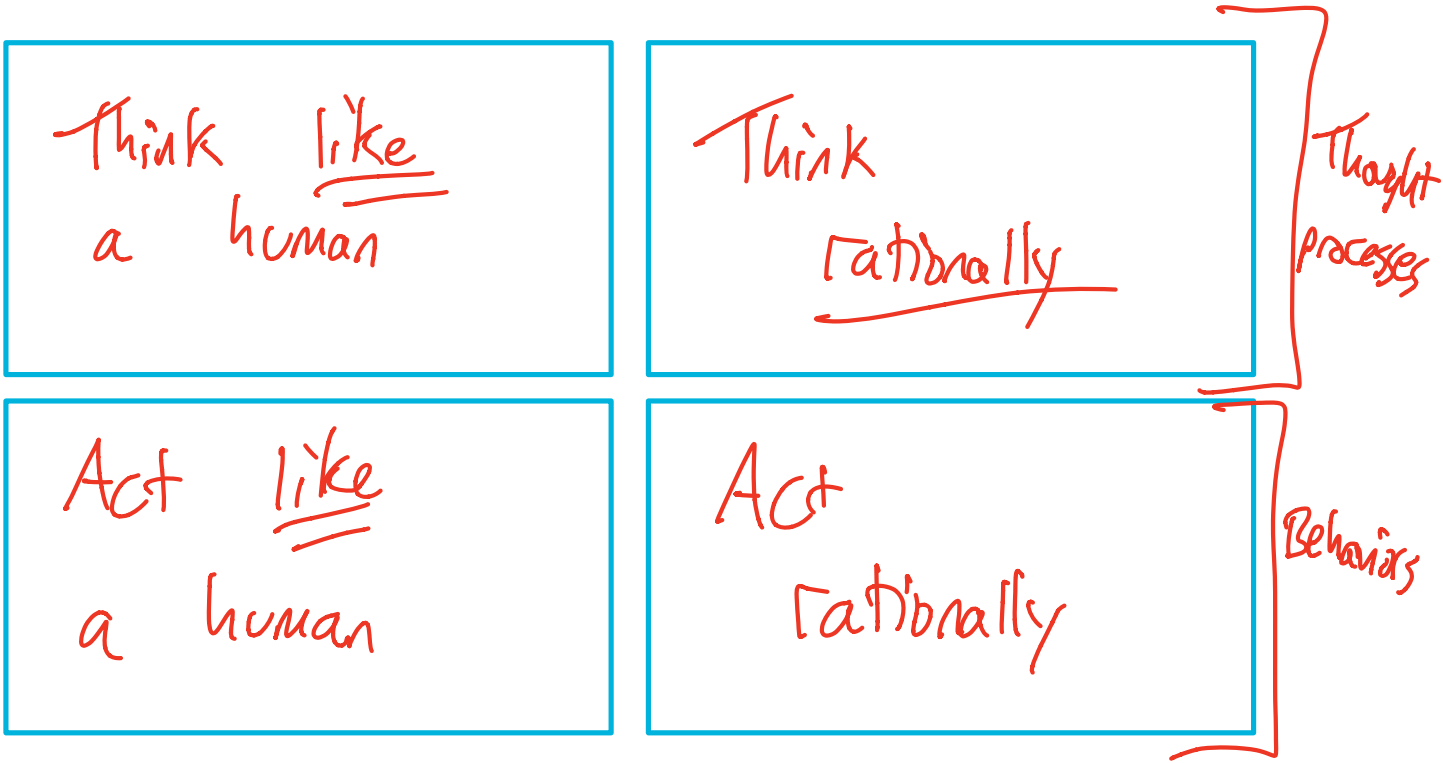
"Human like" behavior

Game Theory – economics!

Artificial Intelligence

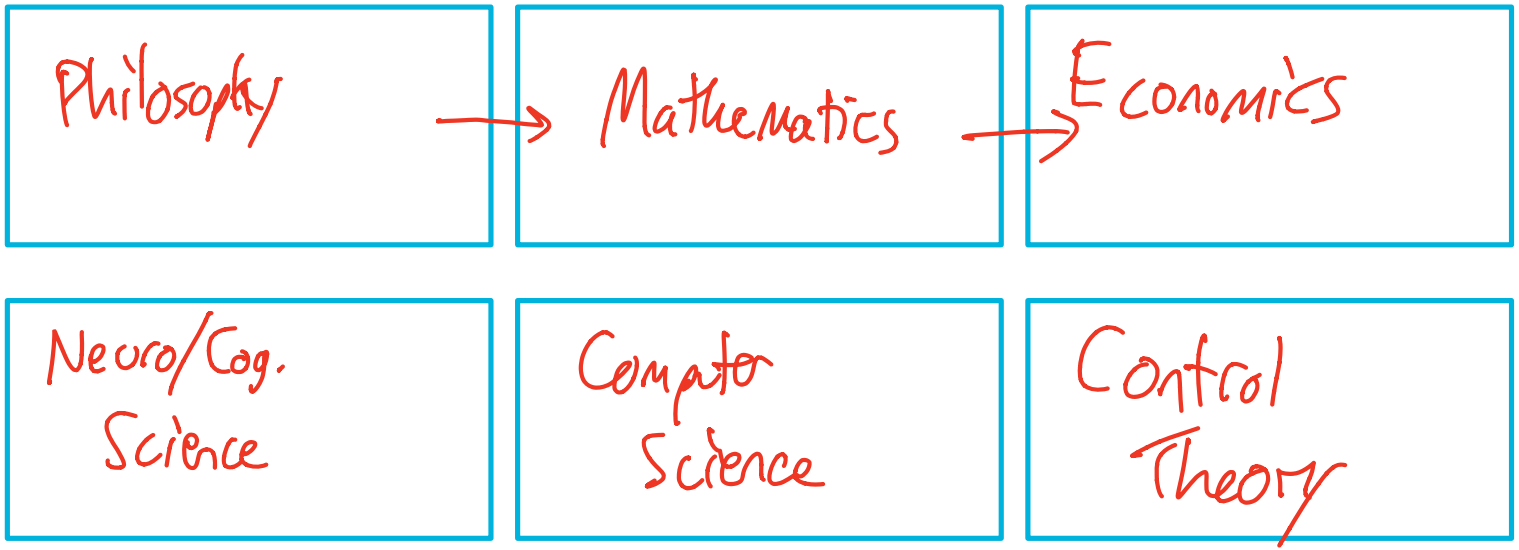
Artificial Intelligence

Algorithms and software that enable machines to:

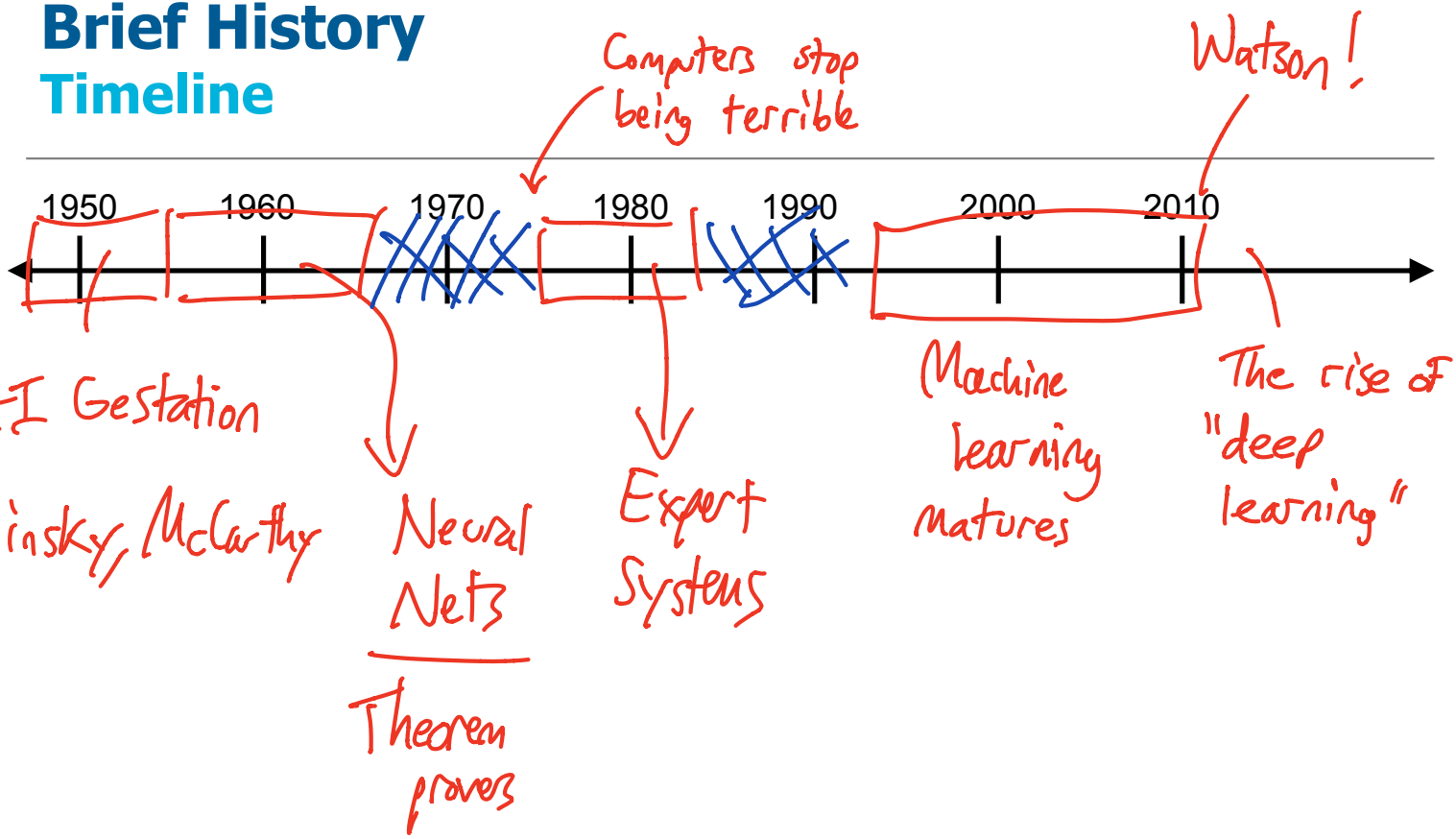


Brief History
Foundations

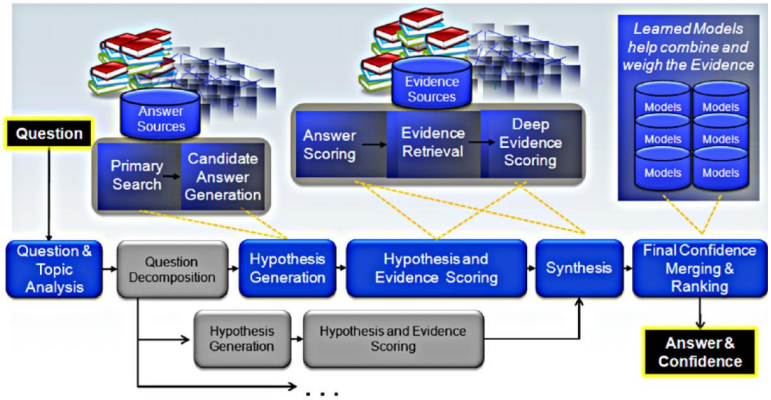
The field of AI builds upon many disciplines



Brief History
Timeline



Watson Jeopardy



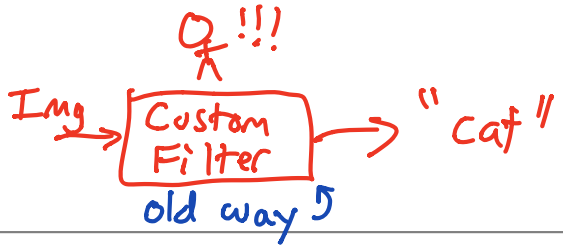
- **AI technologies used**
  - Graph search
  - Probabilistic reasoning
  - Machine learning
  - Natural language processing

- **Lessons learned**
  - Building real AI systems requires considerable engineering
  - Pushed the boundaries of Question Answering technology

BIG engineering project. Not magic!

D. Ferucci, et. al., Building Watson: An Overview of the DeepQA Project, *AI Magazine*, 2010

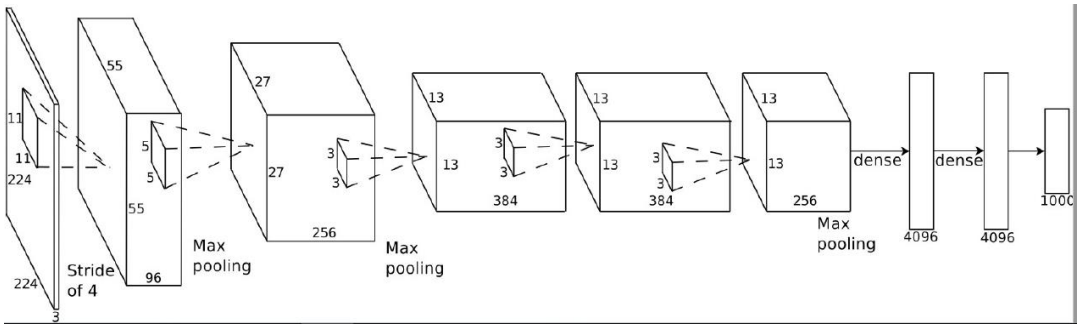
The Rise of Deep Learning

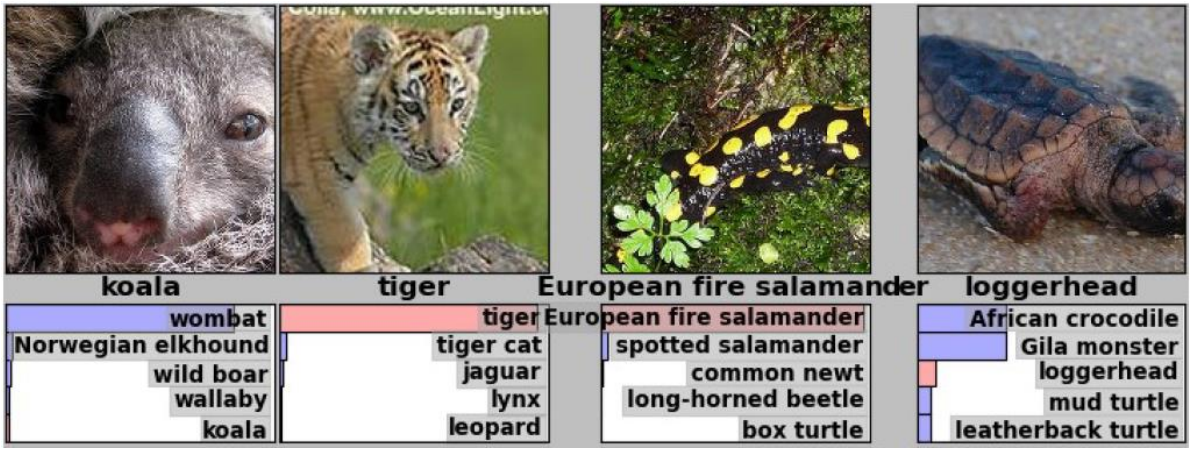


AlexNet (2012)

Convolutional neural net

Blew away all other high performance image classifiers!

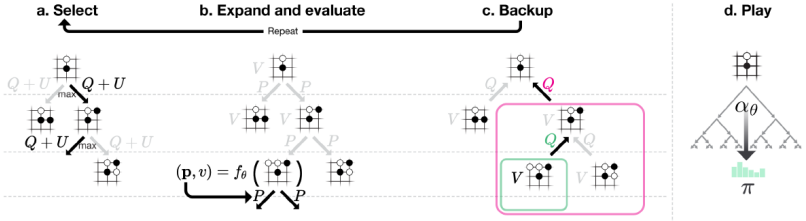




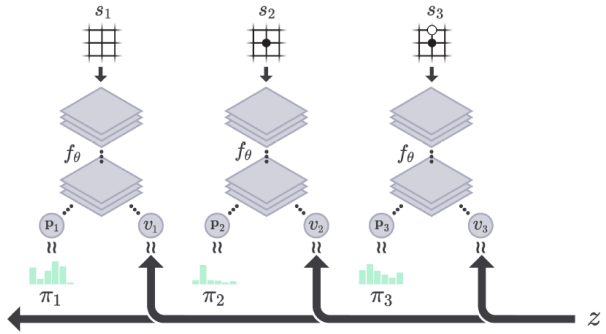
A. Krizhevsky et. al., ImageNet Classification with Deep Convolutional Neural Networks, NIPS, 2012

Alpha Go Zero (2016)

Tree search algorithm produces moves



Neural network trained with chosen moves



- **AI technologies used**
  - Deep reinforcement learning
  - Tree search

New search algorithm: Monte carlo tree search

- **Lessons learned**
  - RL can be scaled to complex, but known, interactions with humans
  - Not very useful in unknown, dynamic environments

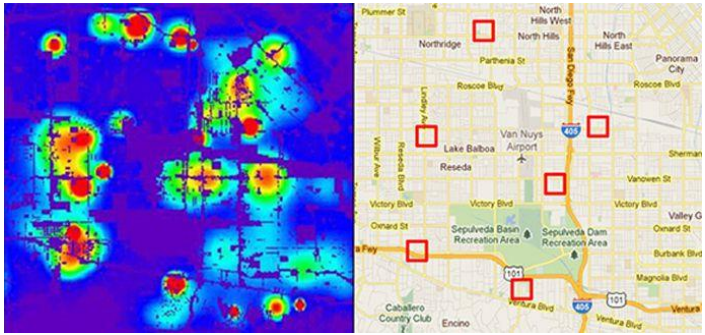
Compelling ... but not practical in many applications.

https://deepmind.com/blog/alphago-zero-learning-scratch/

MITRE

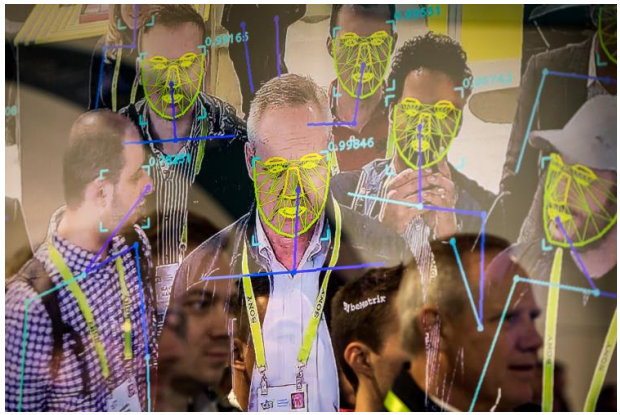
Challenges (really... Machine Learning)

- **Bias: At the mercy of our data**



Predictive Policing (Smithsonian)

- **Privacy: Our data is at the mercy of organizations**



Public Facial Precognition (NY Times)

Data is king... but data always has bias.

Computational Rationality (otherwise known as AI)

- **What do we mean by rational?**

Maximally achieving a pre-defined goal

- **What is a goal?**

Utility Function

State space : X

$J(X)$, $U(X)$

A rational agent

Maximizes

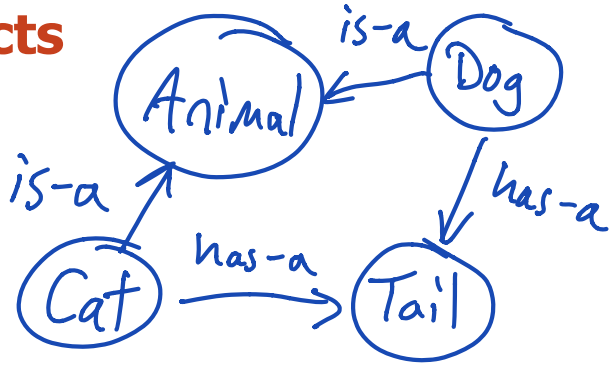
(expected)

utility

Knowledge, Reasoning, and Planning

- **Structures that maintain facts**

Ontologies: graph that captures relationships

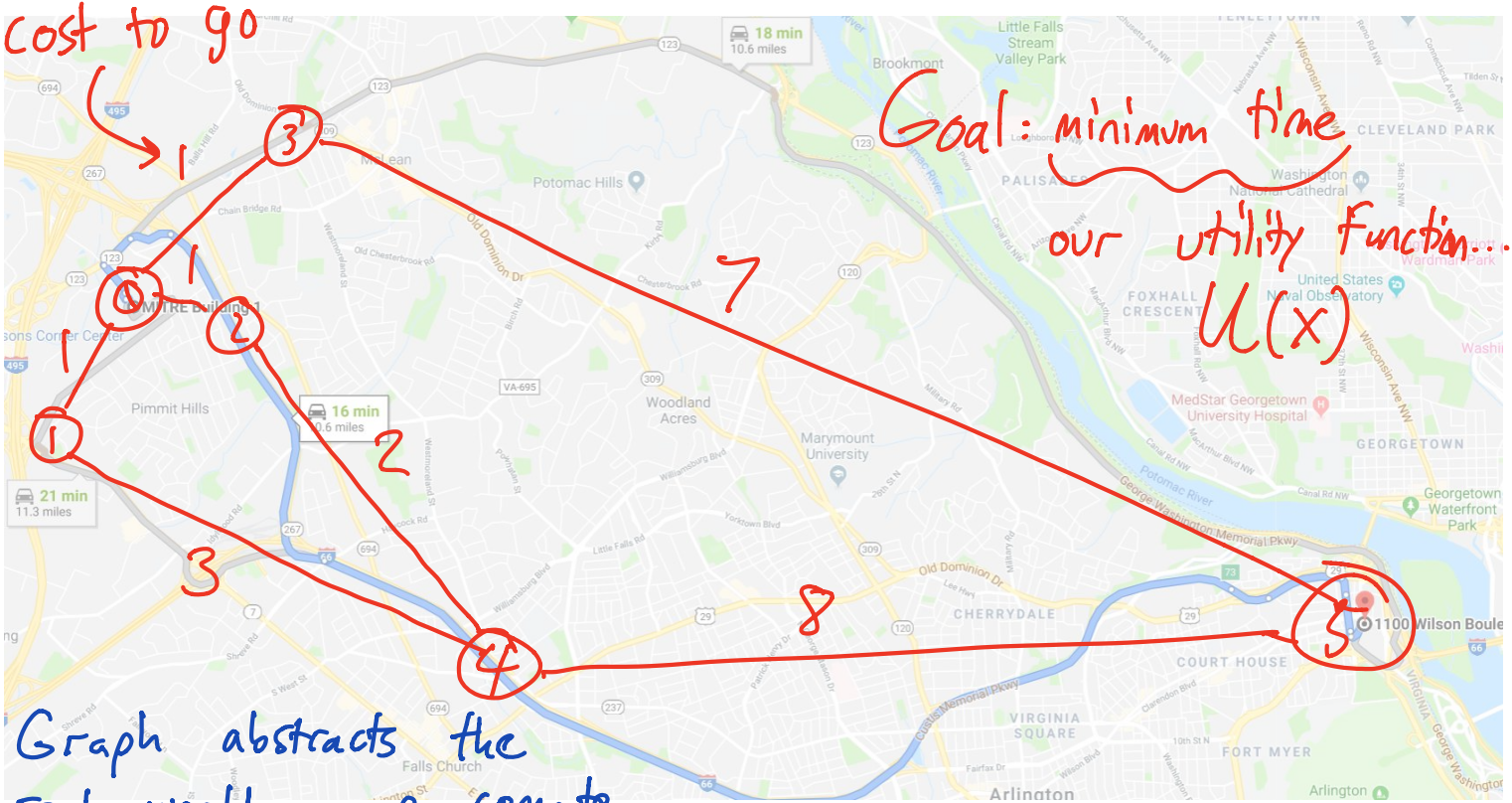


- **Algorithms that compute actions that achieve goals**

Search w/ queries

Compute pathways

Knowledge, Reasoning, and Planning
(Un)Informed Search



GAIN MITRE

Knowledge, Reasoning, and Planning
(Un)Informed Search

Search problems have the following pieces:

State space: X

actions: A (continuous / discrete)

start/goal: $x_0, x_g \in X$

utility (objective) Function

Search problems are solved with an algorithm

Breadth First search

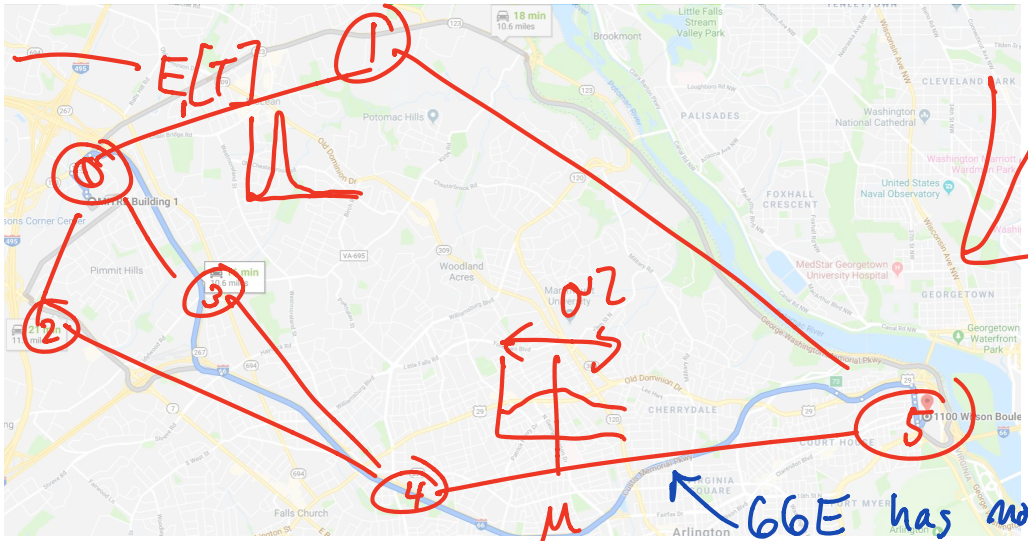
Depth First search

A* - informed by a heuristic Function

A few algorithms...

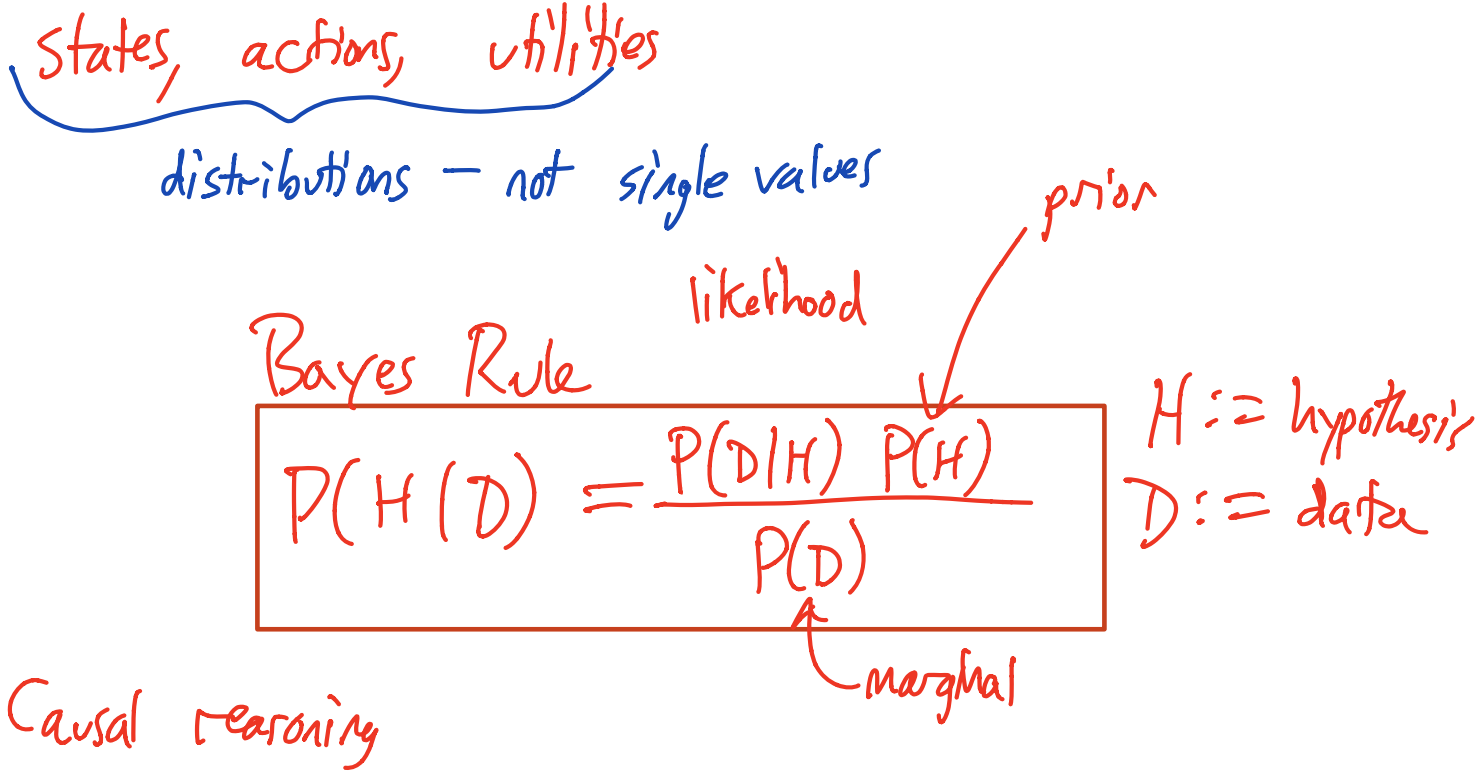
Probabilistic Reasoning

Consider the map search problem…



Probabilistic Reasoning
Bayesian Methods

Making decisions when we are uncertain about…



Probabilistic Reasoning
Example: Bayesian Network

Represents a complex, joint distribution

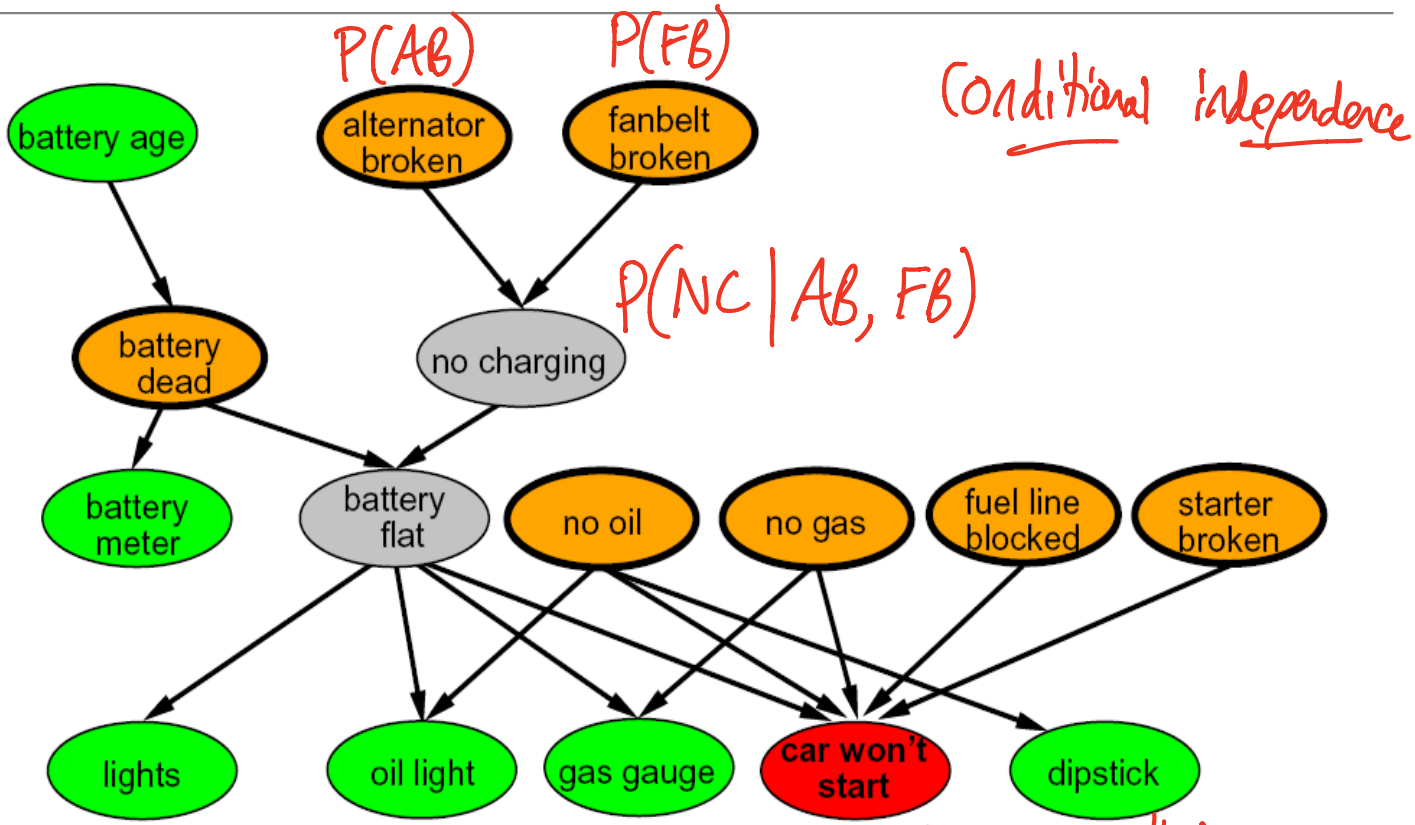
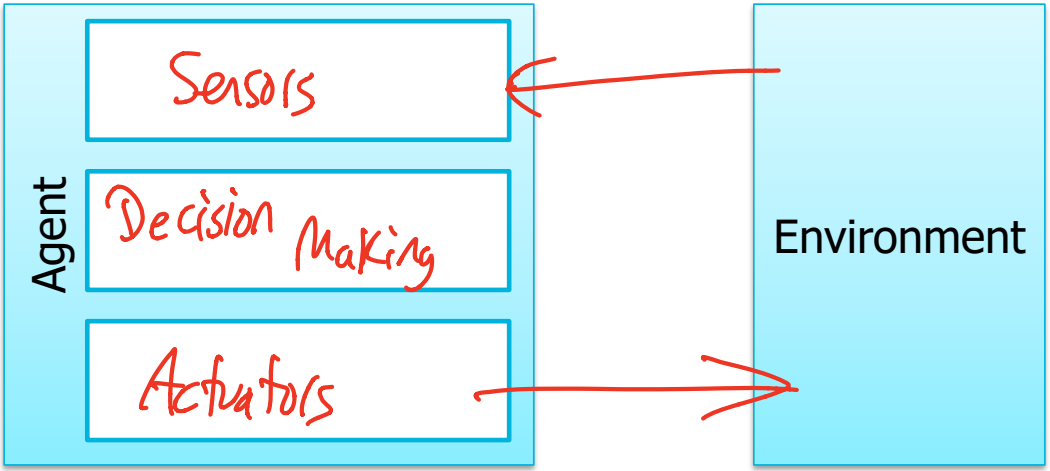


Image from http://ai.berkeley.edu

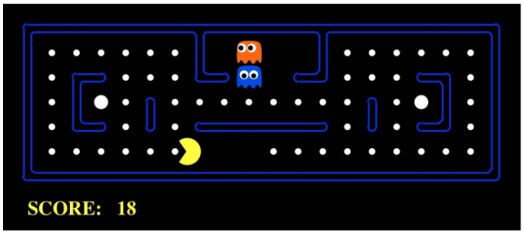
Communication, Perceiving, and Action

AI algorithms enable intelligent, rational agents



Communication, Perceiving, and Action
Applications

- **Pacman**



Pac-Man is a registered trademark of Namco-Bandai Games, used here for educational purposes

Actions? U, D, L, R

Sensors? Image production (eye, digital camera screenshot)

- **Autonomous Car**

Image courtesy of Toyota and used for educational purposes

Actions? acceleration, angle (θ) (continuous!)

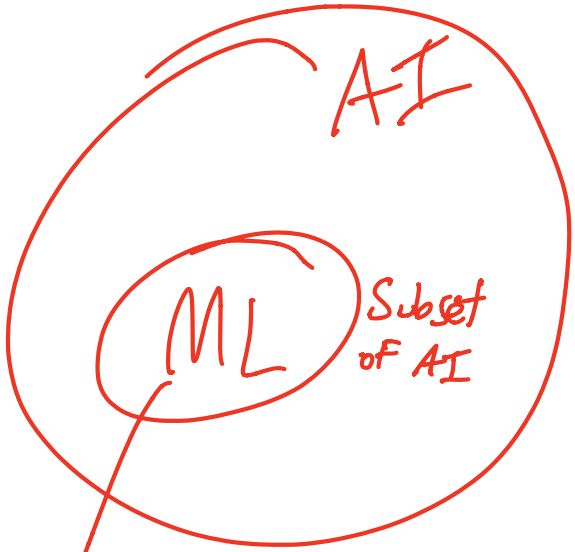
Sensors? vision, lidar, GPS, inertial

Machine Learning

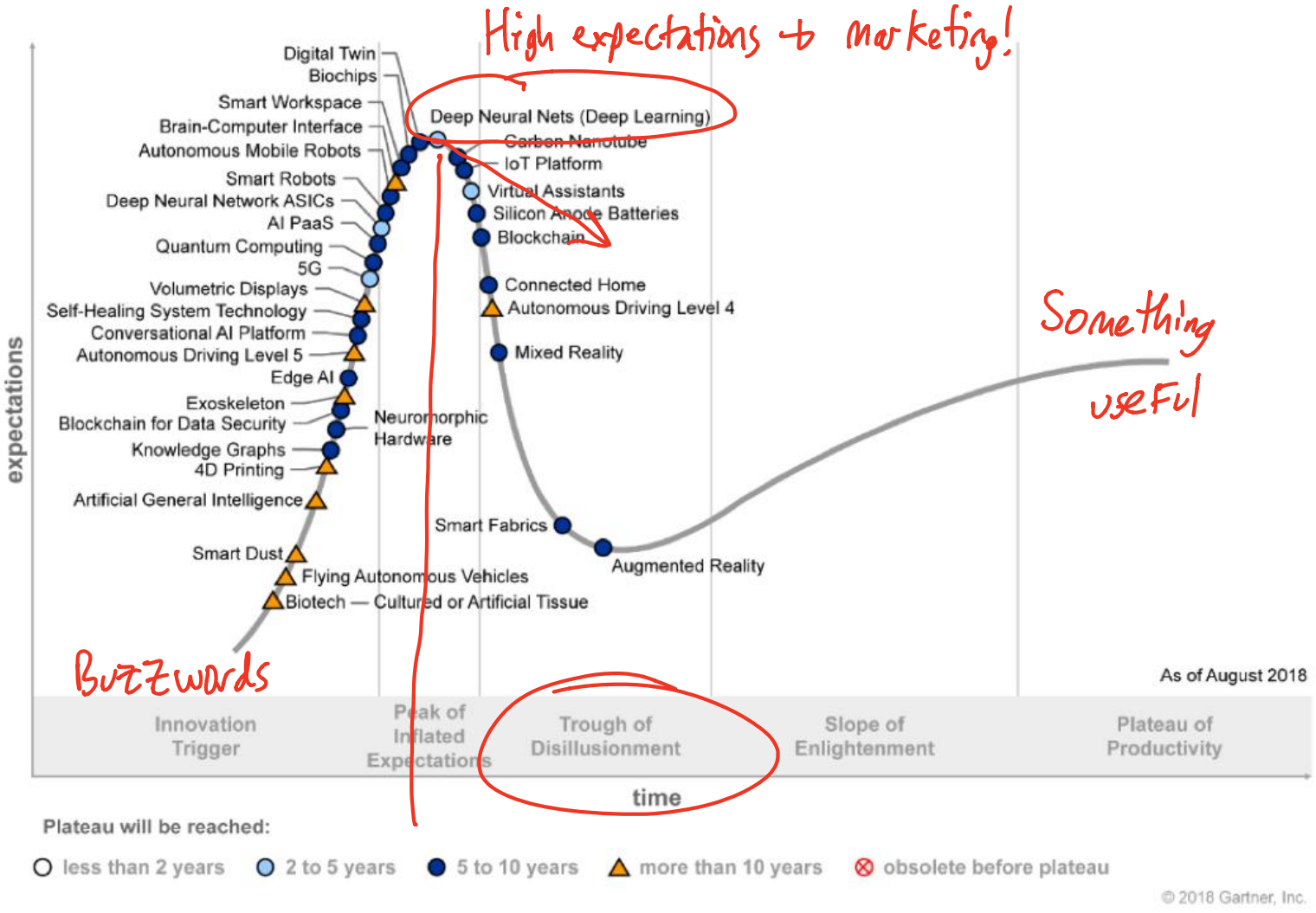
Marketing conflates them: "AI = ML"

But really...

https://xkcd.com/1838/



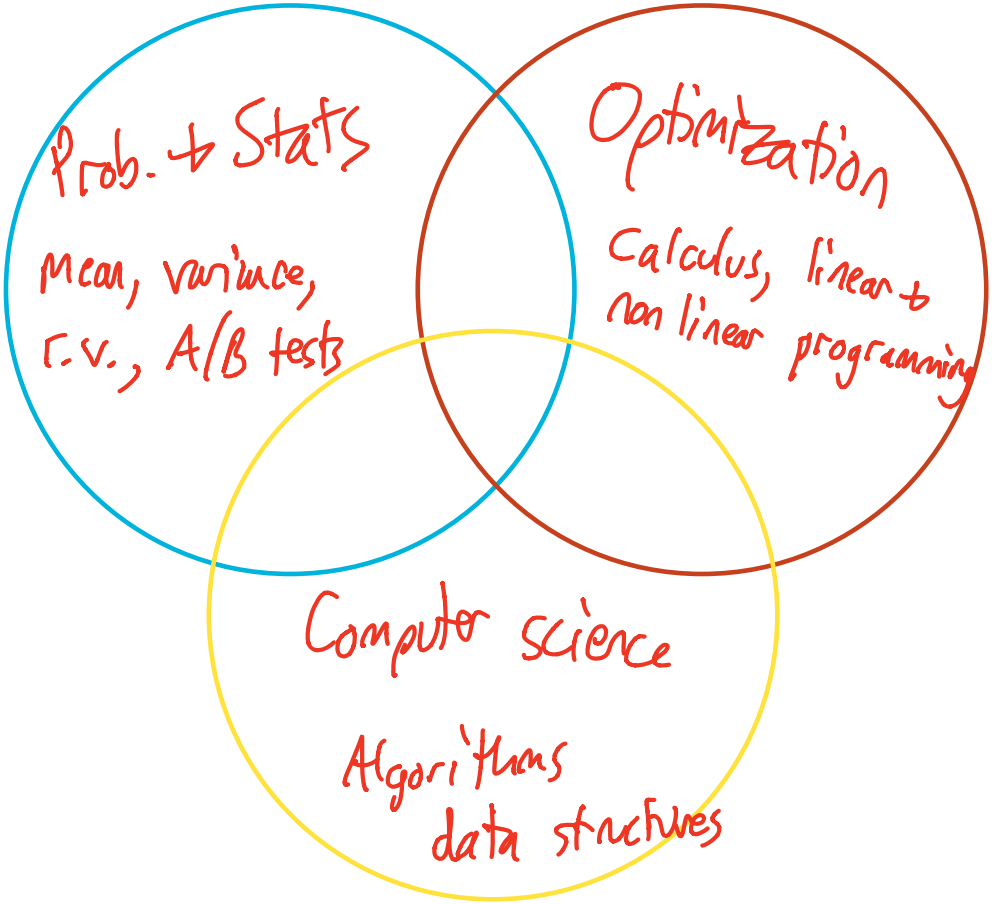
Popular one!



From Gartner Press Release, August 2018

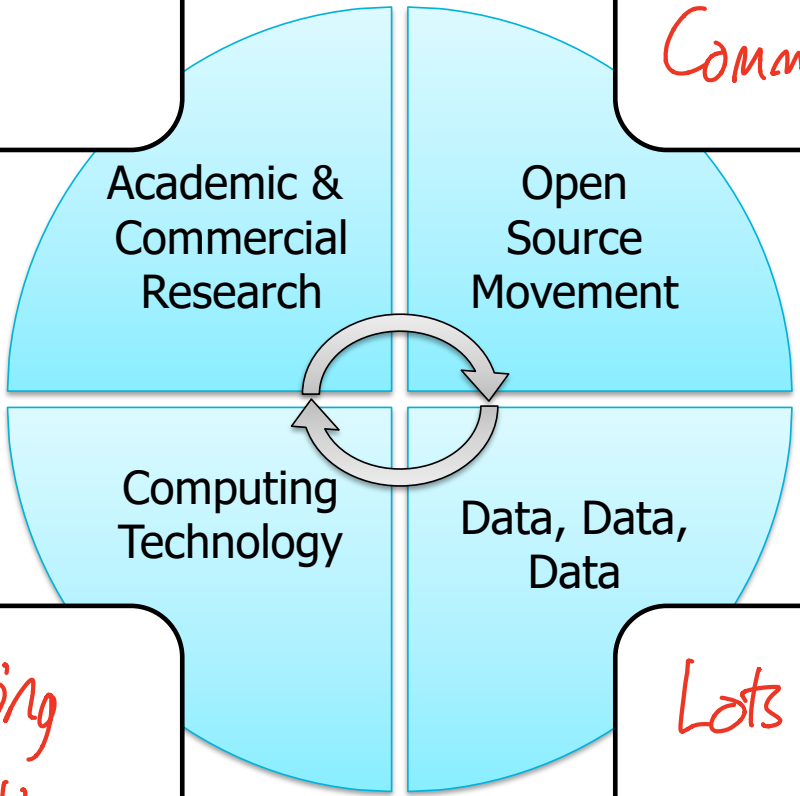
GAIN MITRE

Theoretical Underpinnings



Math
Algorithms
Architectures

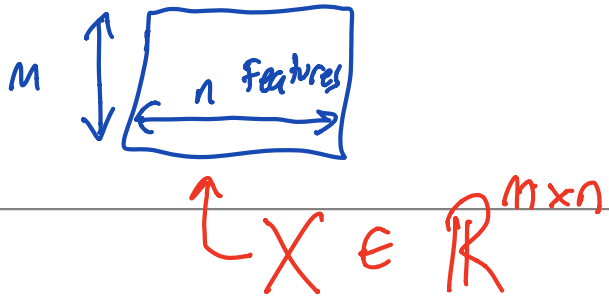
Free SW
Communities



Cloud computing
HW Acceleration

Lots of data

Supervised Learning



Definition

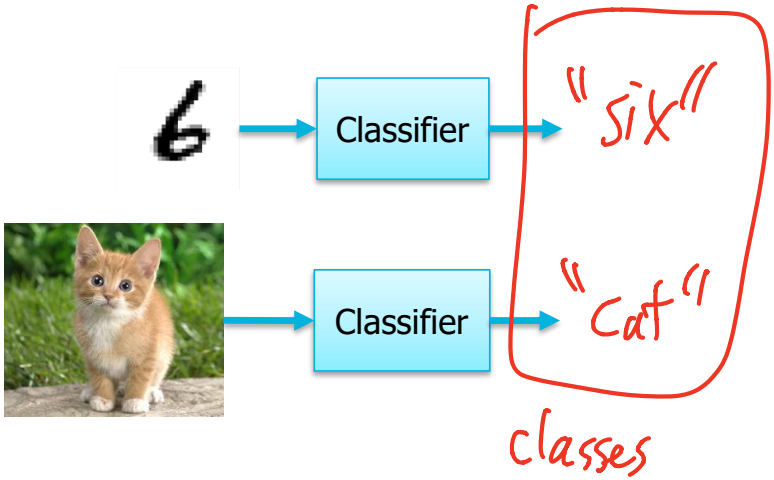
The process of learning a model from a set of **input vectors** and their **target vectors**

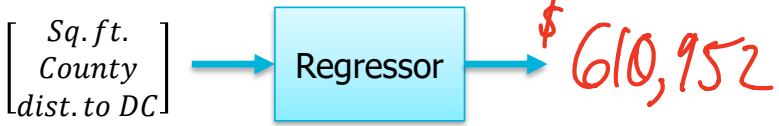
Classification

Finding discrete things

Regression

Finding continuous things





Unsupervised Learning

No ground truth

Definition

The process of learning a model from input data vectors with **no** target values/classes

imgflip.com

Applications

Clustering

Compression

In practice: Unsupervised learning is used in a pipeline of supervised learning

Frequently Used Algorithms

K Means

Principal Component Analysis

Reinforcement Learning

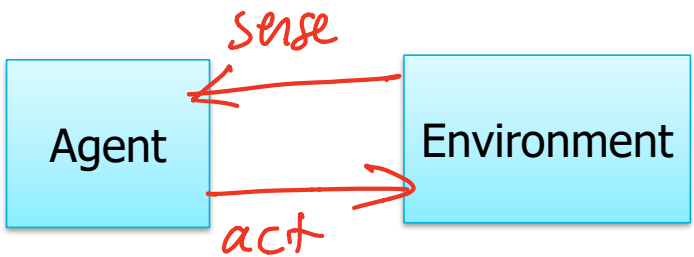
Definition

The process of learning an optimal output via trial and error in an environment; goal directed learning*

Applications

Task planning

Policy search (robots)



Frequently Used Algorithms

Q-Learning

[*] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*, MIT Press, 1998

Exercise

Which type of machine learning is applicable to the following scenarios? Be sure to include why.

1. Amazon segmenting customers in a database.

Unsupervised + supervised

2. Enabling a smart taxi to correctly make a right-on-red

RL! AND supervised

3. Detecting a person of interest in a surveillance video

supervised learning + unsupervised

Example Application

~~Sepsis Watch~~ ML For Pathology

- **Data Requirements**

Digitized biopsy slides

Pathologist labeling

- **Benefits**

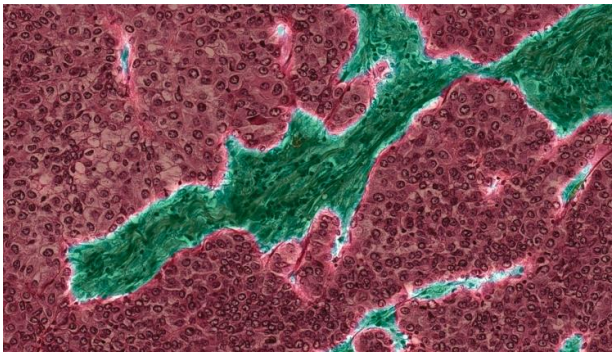
New identification tools

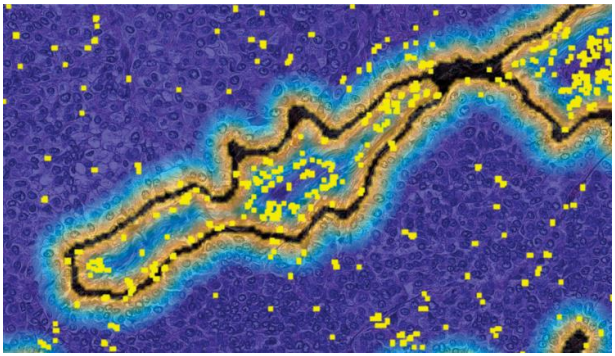
New workflow optimizations

Better outcomes

- **Challenges**

Expensive data collection + labeling



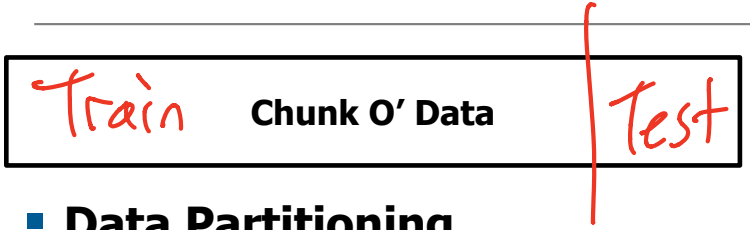


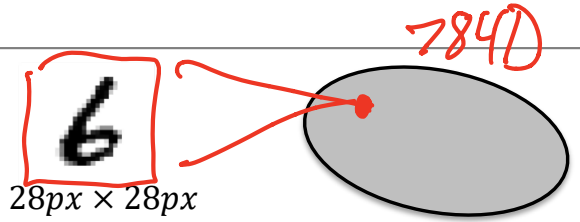
Images from https://spectrum.ieee.org/biomedical/diagnostics/the-first-frontier-for-medical-ai-is-the-pathology-lab

GAIN MITRE

1 Development Process

Data Partitioning and Analysis





- **Data Partitioning**

training data : For model building

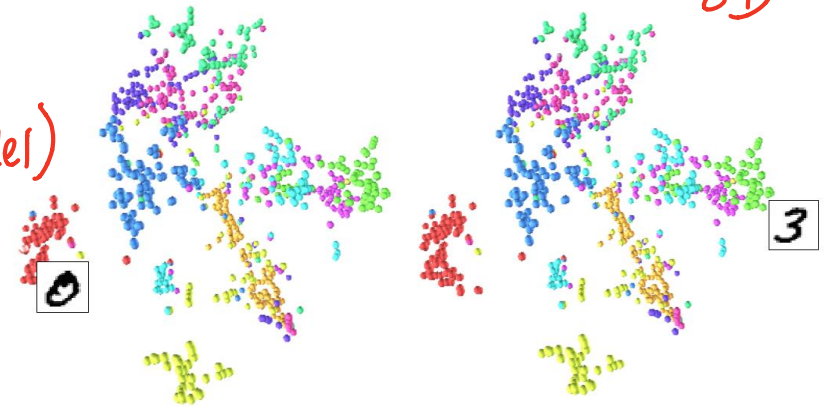
test data : For testing! (never use to build your model)

- **Analysis**

Examine Features & look For patterns

(unsupervised)

Neighbor embedding



http://colah.github.io

Development Process

ML Algorithm Assessment

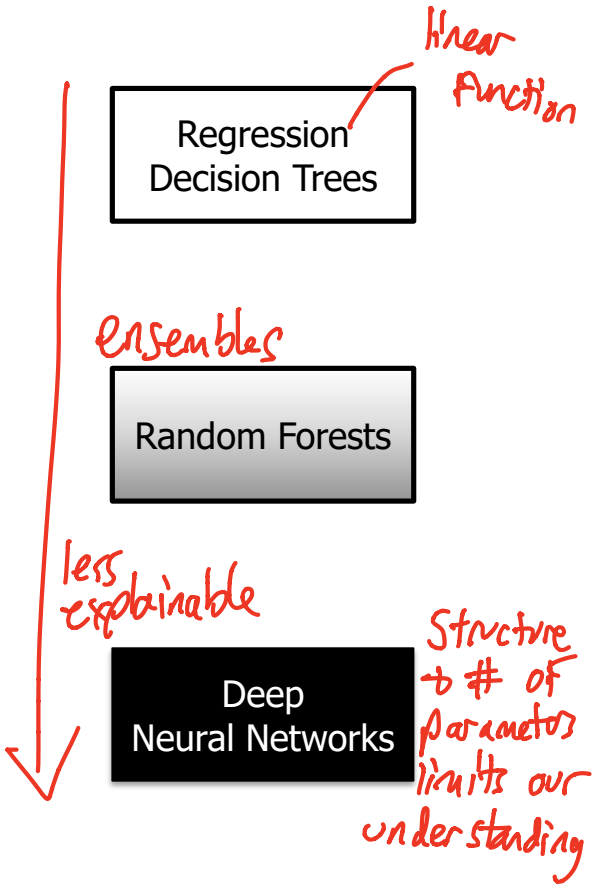
- **Training algorithms fit a model to the training data**

- **Things to think about…**

input shape + size

Complexity: runtime — training / inference

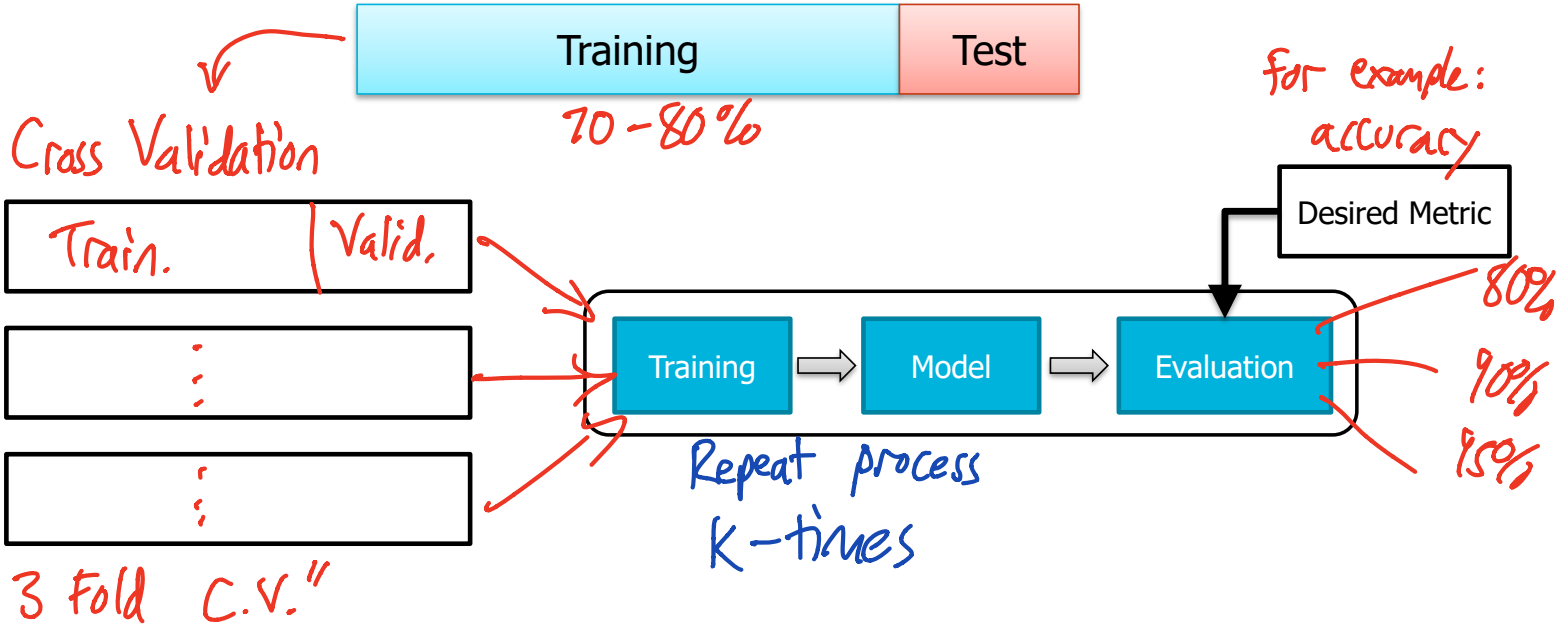
explainability: do you care about understanding the model fit to your data



Development Process

Model Training and Assessment

Goal: Learn a model that generalizes to data it has not seen before



*sklearn.model_selection has multiple CV tools to evaluate your models

Performance Metrics

Metrics allow you to compare candidate models

error: $(\hat{y}^i - y^i)$

Regression Metrics

Mean squared error

$$\frac{1}{M}\sum(\hat{y}^i - y^i)^2$$

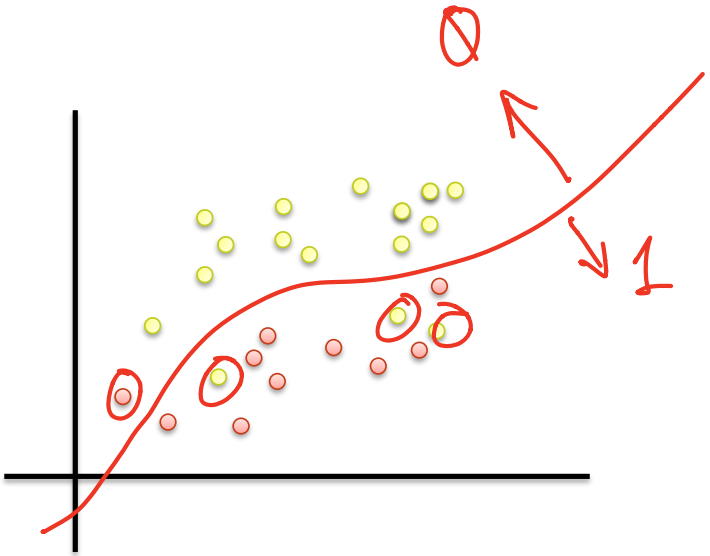
Classification Metrics

Accuracy

Precision

Recall

Classification Metrics (1)



Confusion Matrix

Predicted \ Actual	0	1
0	13 TP	1 FN
1	3 FP	9 TN

Classification Metrics (2)

Precision: How many classifications are true positives

$$P = \frac{TP}{TP + FP}$$

Recall: How many true positives were found

$$R = \frac{TP}{TP + FN}$$

Accuracy: Number of correctly classified items over all items

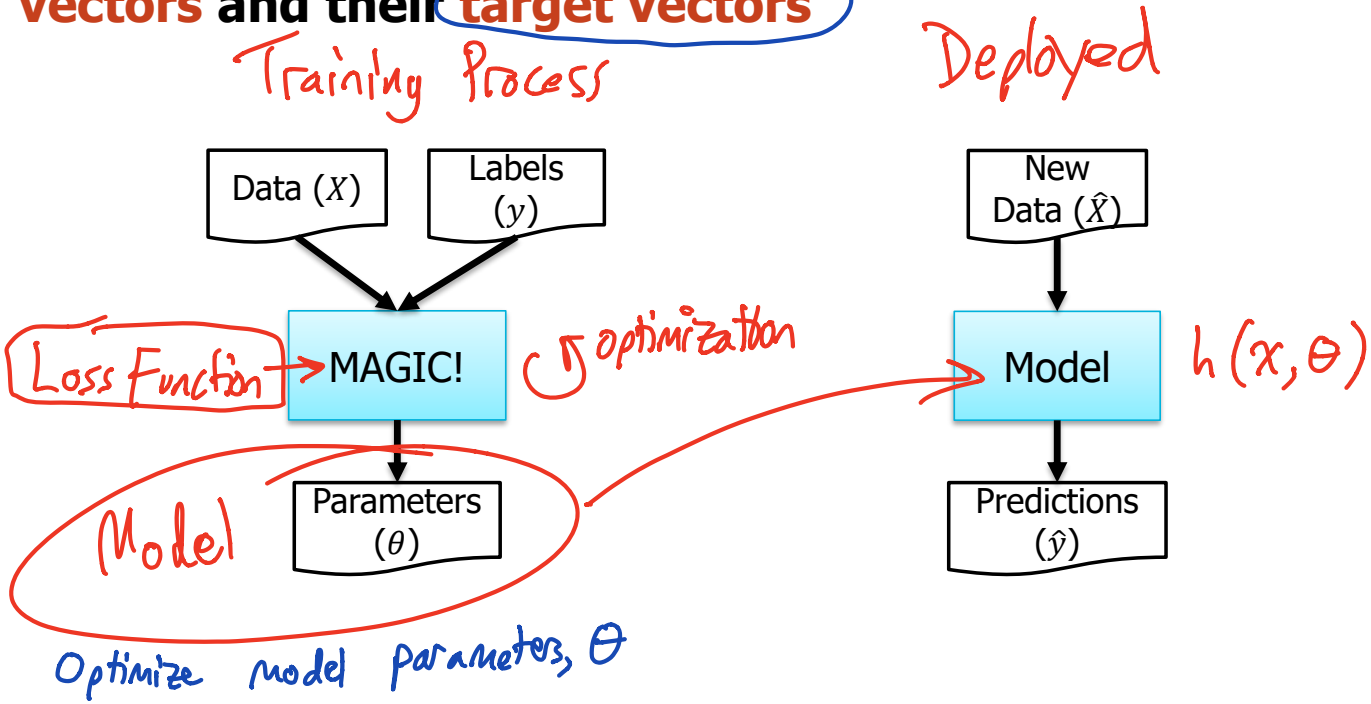
$$\frac{TP + TN}{ALL}$$

Recap: Supervised Learning

You MUST have ground truth on class/value you want to predict.

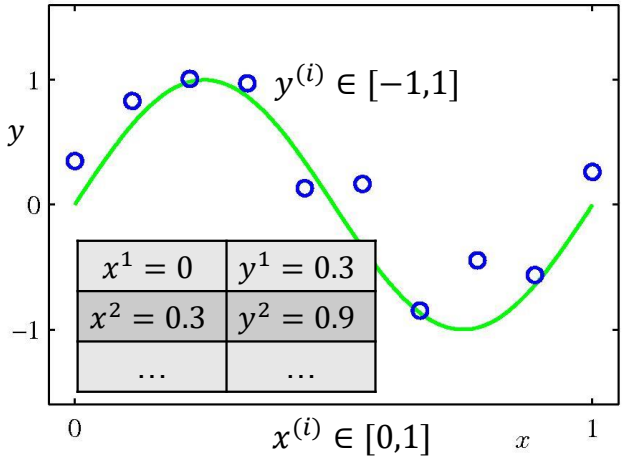
Definition

- **The process of learning a model from a set of input vectors and their target vectors**



Sounds Like Curve Fitting!

- **Start with a collection of data**
  - Each $x^{(i)} \in \mathbb{R}$ is an instance
- **Each input is associated with a target value/class**
  - $y^{(i)} \in \mathbb{R}$ (sometimes denoted $t^{(i)}$)



Choose model type: $h(\theta, x)$ — Could be linear regression, decision trees, neural networks, etc.

- **Learning Goal**
  - Predict the value of $\hat{y}$ for any new $\hat{x}$.

Sounds Like Curve Fitting!

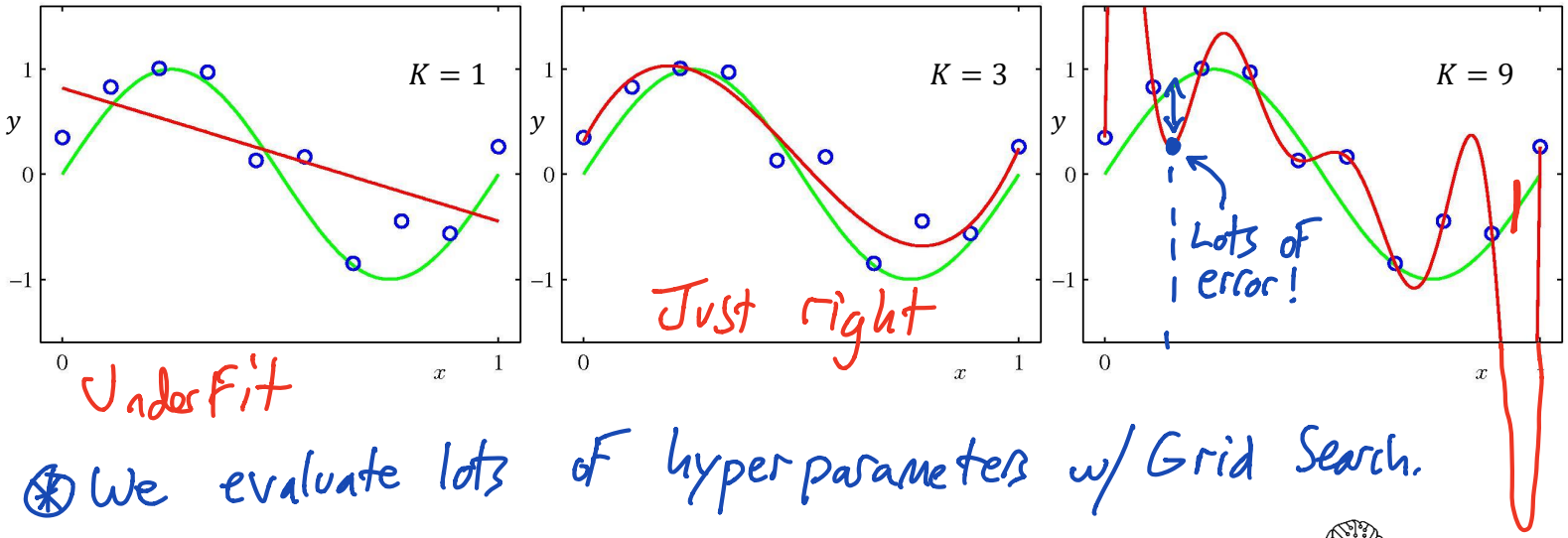
shape the optimization process

hyperparameter

Example: polynomial model

$$h(x, \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots = \sum_{k=0}^{K} \theta_k x^k$$

Some example models based on polynomial order, K overfit



* We evaluate lots of hyperparameters w/ Grid Search.

GAIN MITRE

Linear Regression

Problem Statement

- **Input: m instances of $x \in \mathbb{R}^n$, targets, $y \in \mathbb{R}$**
- **Goal: Compute parameters θ of a linear model $h(x, \theta)$ that predicts $\hat{y}$**

Using calculus & linear algebra...

optimal parameters:

$$\theta^* = (X^T X)^{-1} X^T y$$

expensive computation, $O(n^3)$

Linear Model

$$\hat{y} = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$

Loss Function

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left((\theta_1 x_1^i + \cdots + \theta_n x_n^i) - y^i \right)^2$$

$\hat{y}$ = truth

$\hat{y} = h(\theta, x)$

* Only works w/ linearly separable data.

Fitting the Model

Gradient Descent in a Nutshell

A critical algorithm for ML!

From regression to deep learning

Basic Steps:

① Choose the learning parameter $\alpha \in (0,1)$ and tolerance $\varepsilon \in (0,1)$

② Compute $\left.\frac{\partial J}{\partial \theta}\right|_x$

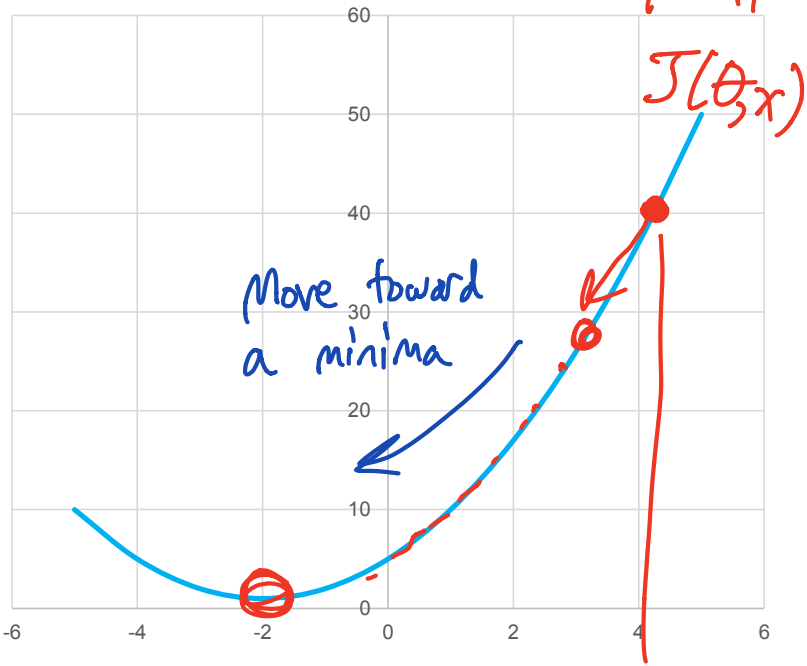
③ Step in opposite direction

$$\theta_{new} = \theta_{last} - \alpha \left.\frac{\partial J}{\partial \theta}\right|_x$$

④ Repeat until

$$\left|\frac{\partial J}{\partial \theta}\right| < \varepsilon$$

Convex, Linear : globally optimal



Regularization

- **Overfitting results in crazy model parameters, θ.**
  - Solution: Penalize the size of θ!
- **l_2 Regularization**

$$\frac{1}{2m}\sum_{i=1}^{m}\left(\left(\theta_1 x_1^i + \cdots + \theta_n x_n^i\right) - y^i\right)^2 + \boxed{\gamma\frac{1}{2}\sum_{i=1}^{n}\theta^2}$$

Benefit
Less overfitting; smaller θ overall.

- **l_1 Regularization**

$$\frac{1}{2m}\sum_{i=1}^{m}\left(\left(\theta_1 x_1^i + \cdots + \theta_n x_n^i\right) - y^i\right)^2 + \boxed{\gamma\sum_{i=1}^{n}|\theta|}$$

Benefit
Drives some $\theta \rightarrow 0$!
Reduces feature space.

Linear Regression Properties

Pros

- Versatile technique with lots of library support
- Scales well thanks to gradient descent

Cons

- In general, not globally optimal
- Nonlinear models require tricks…

Depends on loss function

$$\hat{y} = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$

Linear in θ!

We can make new features to represent polynomials.

ex. Quadratic. Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$

New feature vector

$$\begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix} \in \mathbb{R}^6$$

Can apply linear regression!

Works, but computationally & spatially expensive

Decision Trees

Where should we go out to lunch?

Indian Pizza Hamburgers

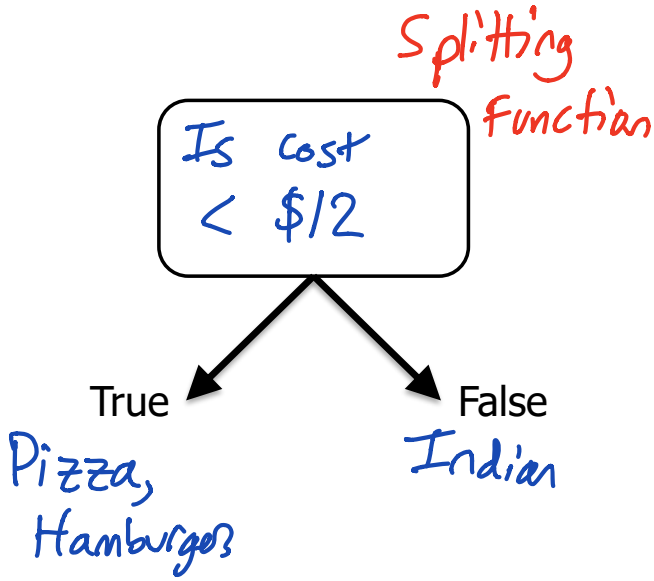
Ask questions based on a Feature...

What features do we care about?

Cost

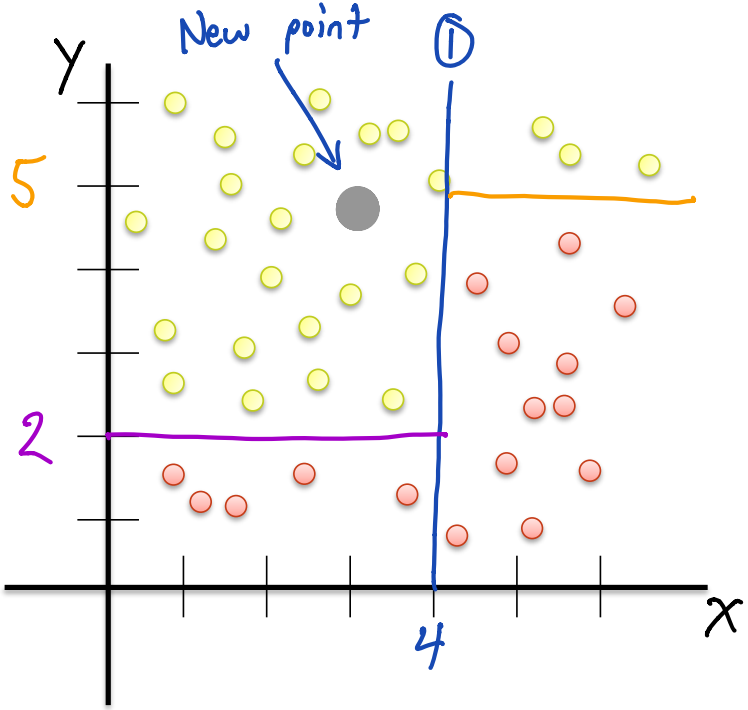
Noise

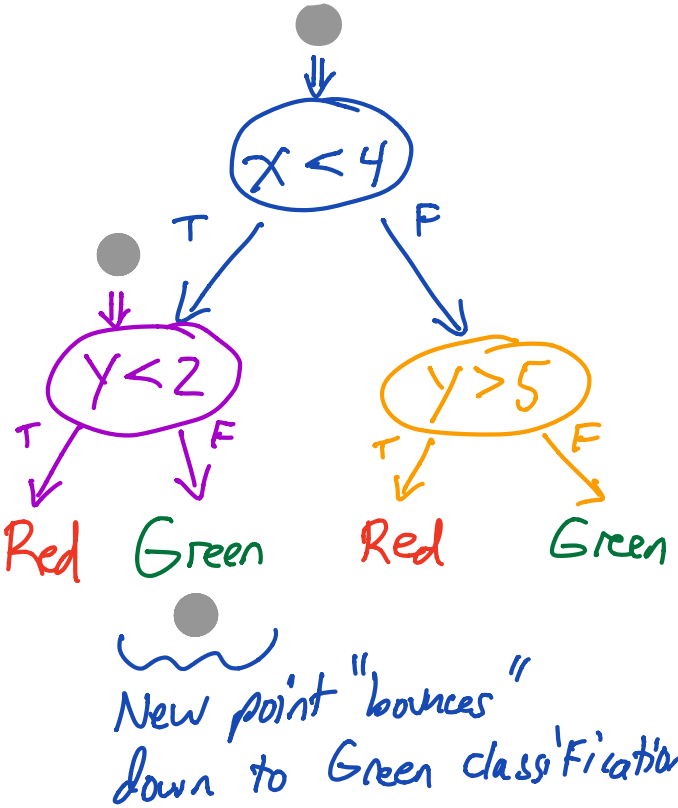
Menu



Decision Tree Example

What is the model for this data?





Split to maximize information gain.

Decision Tree Properties

Pros

- **Minimal data preparation required**
- **Fast inference**
  - $O(\log_2(m))$
- **Models are easier to understand**

Cons

Leads to overfitting

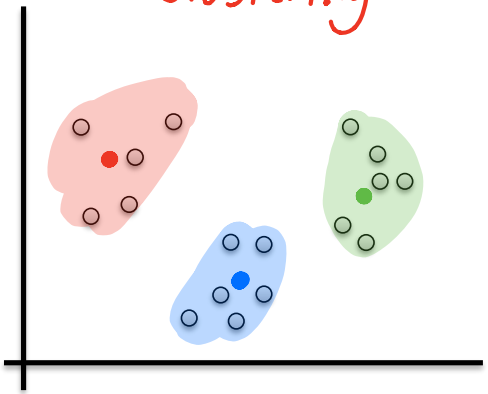
- **Sensitive to small variations in training data**
- **Models not globally optimal**
- **Training does not scale well with features**
  - $O(n \cdot m\ log(m))$

Recap: Unsupervised Learning

Definition

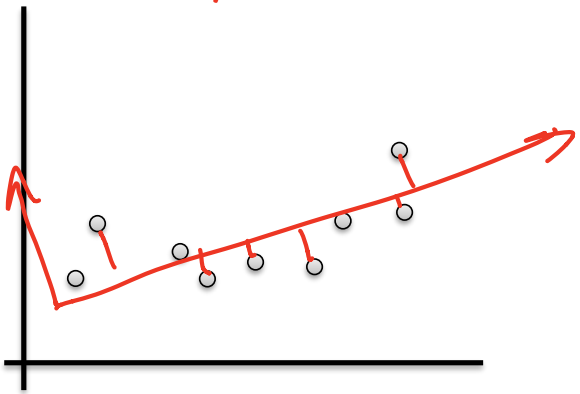
- **The process of learning a model from input data vectors with no target values/classes**

Tasks: Clustering



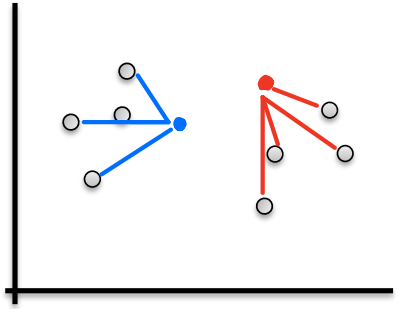
ex. K Means

Compression



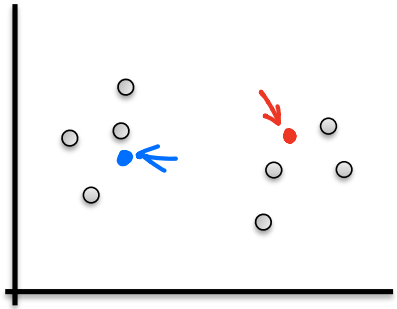
ex. PCA

Clustering via K-Means



Points are pulled to center

- **Minimize quadratic error between data and their cluster centers (μ)**
- **Gradient descent can result in local minima**



* Hyperparameter: # of clusters.

As there is no ground truth (no labels) we do not know the true # of clusters at the start!

K-Means Summary

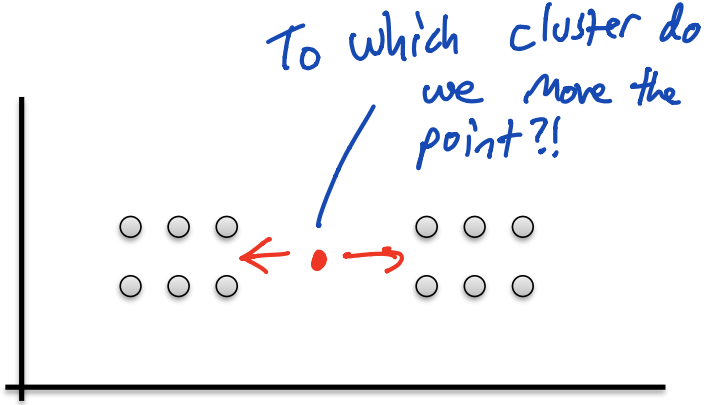
- **Efficient on large data sets**

- **Benefit:** Fast execution
  Provides initial intuition on data set classes

- **Things to keep in mind:**

Local minima are possible!

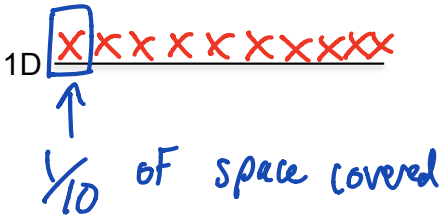
Initial cluster points start at random positions in the feature space.

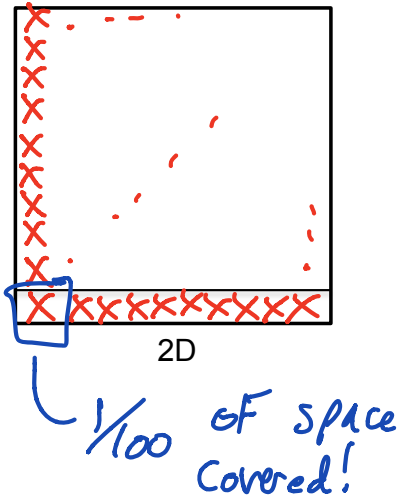


Motivation for Dimensionality Reduction

- **Curse of Dimensionality!**
  - As the number of dimensions grow, the amount of data we need grows exponentially
- **Let's sample some spaces…**

http://muppet.wikia.com/wiki/Count_von_Count



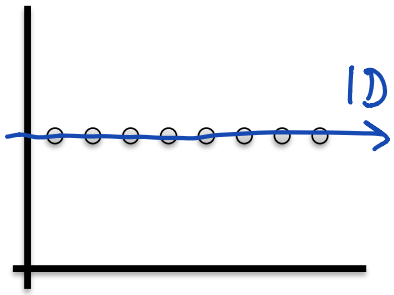


As # dimensions grows, our samples cover LESS!

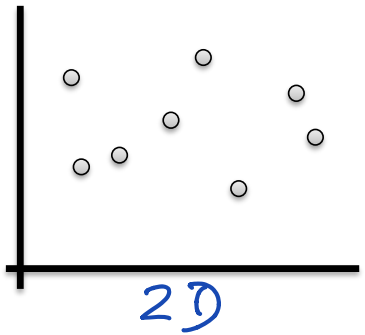
Image if we had 784 dimensions… (in just a moment)

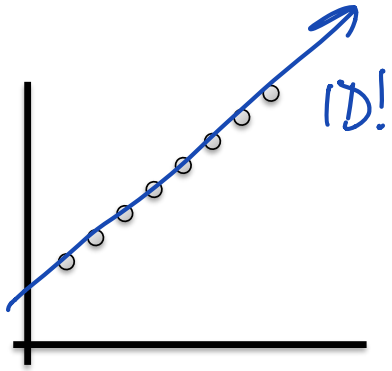
Principal Component Analysis (PCA)

- **PCA is a projection technique**
  - Hypothesis: The bulk of our data set's information resides in a lower dimensional state space!
- **In what dimension does the data reside?**



Yes points are on the plane — but really exist on a line.

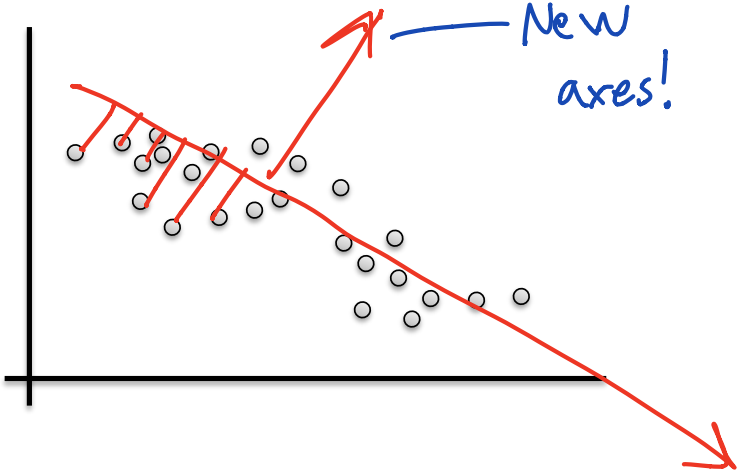




We can transform the points to 1D!

PCA Technical Approach

- **PCA transforms data of any shape into a new coordinate system**
- **Project with "minimal loss of information"**

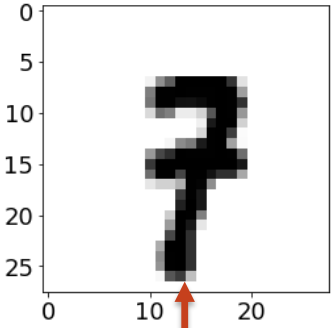


Linear algebra: SVD

Data (X) ⇒ PCA ⇒ principal Component (axes)

Features get mapped to a lower dimensional space.

PCA Example

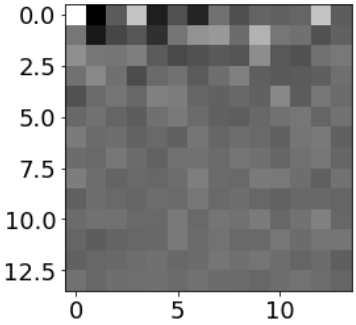


Each feature is a number between 0 and 255

28 × 28 = 784 D!

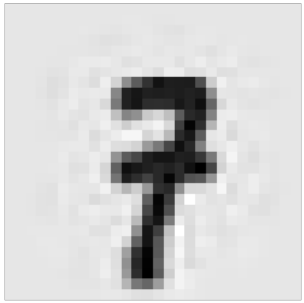
PCA Transform!

14 × 14 = 196 D



¯_(ツ)_/¯

Inverse Transform!



Still looks like a 7!

⊗ We can train a Follow on classifier w/ lower dimensional data.

→ Faster

→ less data to keep around

→ Similar performance

PCA Summary

- **Great for managing large, sparse data sets**
- **Benefit:** Less Features, less data
- **Things to keep in mind:**
  You must scale your data before using PCA!
  (Ideally, [-1,1])

Back to some algorithmic supervision!

What do we know now?

Supervised Methods

Must have linearly separable data

- Linear Regression
- Decision Trees

Work well with nonlinear, lower dimensional feature spaces.

Unsupervised Methods

Help us clean up our data...

- Principal Component Analysis
- K Means

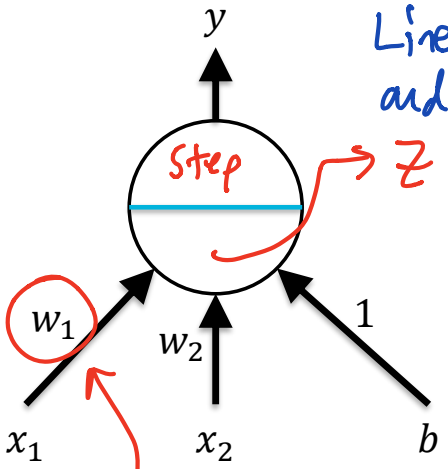
Neural Networks

To the (sort of) rescue!

Note: there are lots of activation Functions! LTU are not used much now.

Neural networks go back to the 1950s!

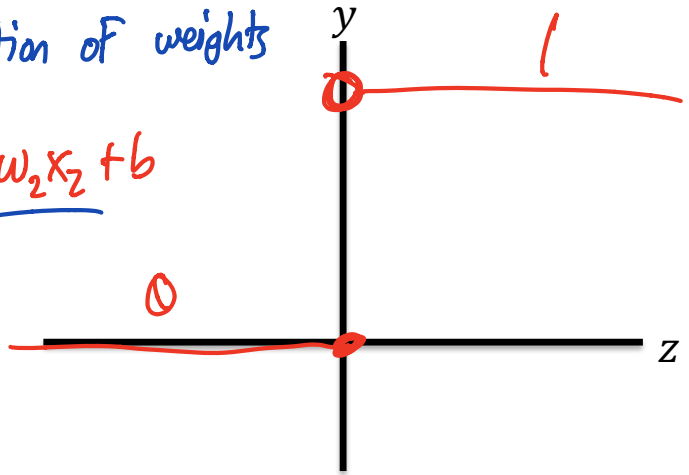
Linear threshold unit (LTU)



Linear combination of weights and inputs

$z = w_1 x_1 + w_2 x_2 + b$

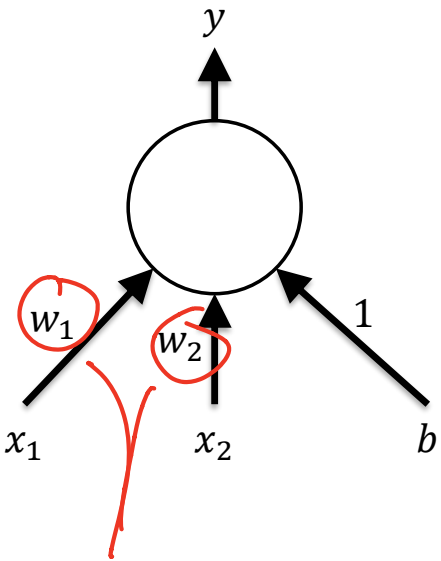
Notation For parameters are θ or w in the literature



$y = \text{step}(Wx + b)$

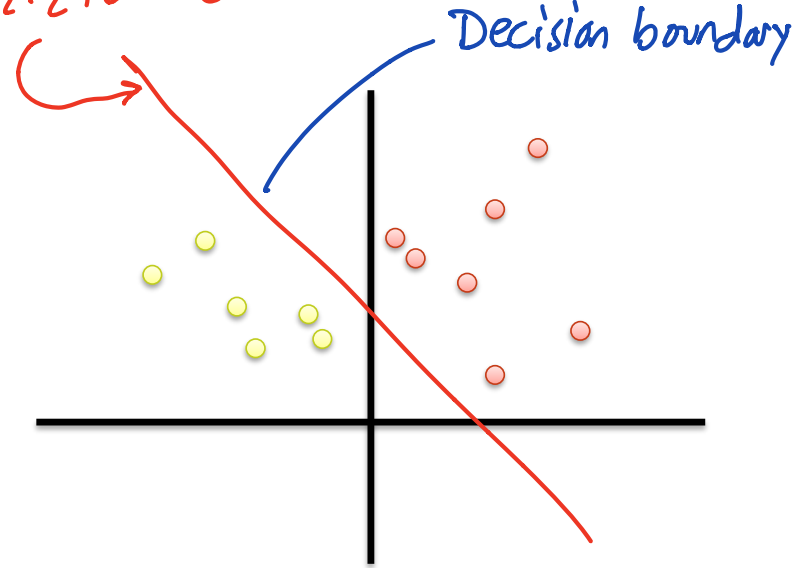
A Closer Look

A single LTU neuron can classify linearly separable data.



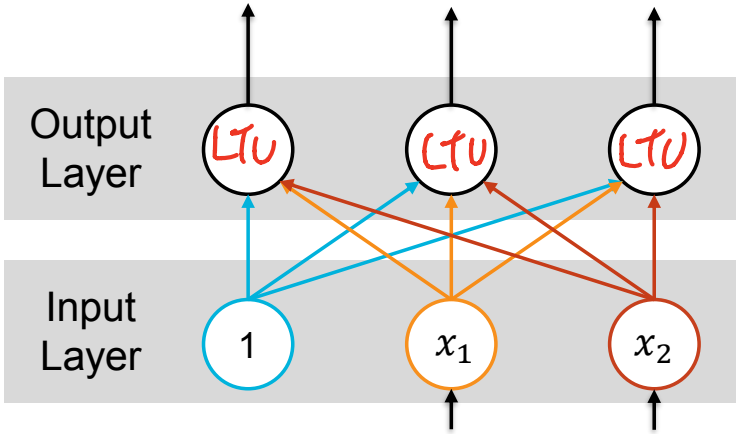
Weights (or, parameters) indicate "importance" of inputs x_1, x_2 to classification

$w_1 x_1 + w_2 x_2 + b = 0$



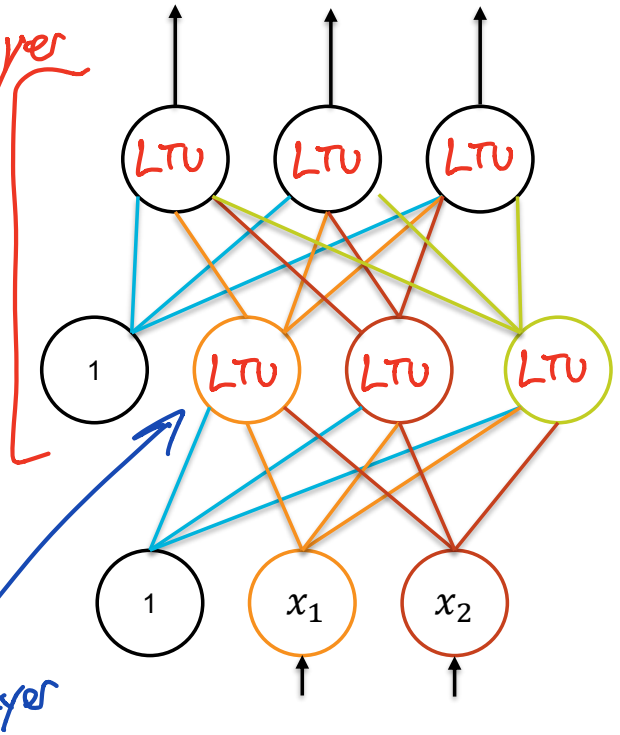
Neural Network Architectures

Classical single layer perceptron



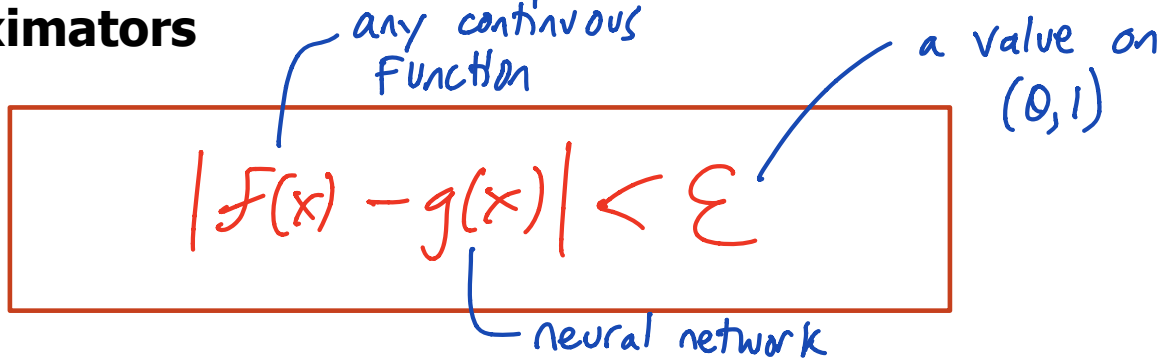
We choose # layers and # artificial neurons

Multi layer perceptron



Why Neural Networks?

- **NNs with one hidden layer universal function approximators**



- **NNs have efficient architectures and training algorithms**

NN hyperparameters: ① arch ② optimizer ③ depth (layers) ④ width (neurons)

Summary

- **Machine learning techniques provide powerful statistical tools**
  - Classification
  - Regression
  - Clustering
  - Dimensionality reduction

- **Remember:**

Garbage data IN
Garbage models OUT

MITRE

MITRE's mission-driven teams are dedicated to solving problems for a safer world. Through our federally funded R&D centers and public-private partnerships, we work across government to tackle challenges to the safety, stability, and well-being of our nation.

Learn more www.mitre.org