# CM146
# Introduction to Machine Learning
# Winter 2022

Sriram Sankararaman

The instructor gratefully acknowledges Eric Eaton (UPenn), who assembled the original slides, Jessica Wu (Harvey Mudd), David Kauchak (Pomona), Kai-Wei Chang (UCLA) whose slides are also heavily used, and the many others who made their course materials freely available online.

# Machine learning is…

Machine Learning is the study of algorithms that

improve their performance P

at some task T

with experience E.

A well-defined learning task is given by <P, T, E>.

[Definition by Tom Mitchell (1998)]

# Machine Learning is…

Machine learning is about predicting the future based on the past.

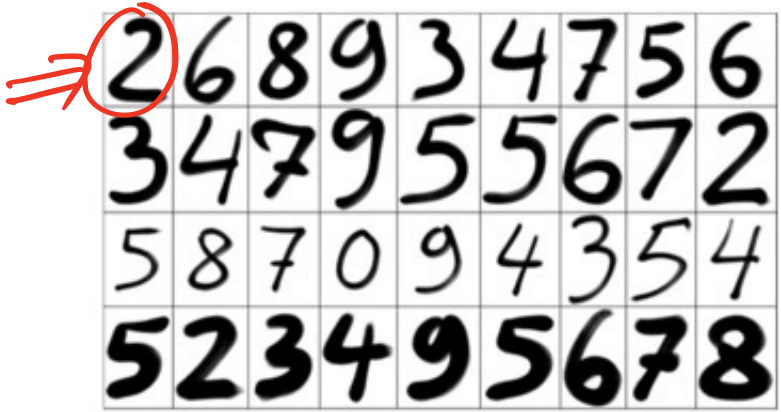
-- Hal Daume III

# Improve on task T with respect to performance P based on experience E

T: Recognizing hand-written words

P: Percentage of words correctly classified

E: Database of human-labeled images of hand written words



# Improve on task T with respect to performance P based on experience E

T: Driving on highways using vision sensors

P: Average distance traveled before a human-judge error

E: Demonstration of human drivers

(A sequence of images and steering commands recorded)

# When Do We Use Machine Learning?

Slide credit: Eric Eaton
Based on slide by E. Alpaydin

# When Do We Use Machine Learning?

ML is used when:

Human expertise does not exist (navigating on Mars)

Slide credit: Eric Eaton
Based on slide by E. Alpaydin

# When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans cannot explain their expertise (speech recognition)

Slide credit: Eric Eaton
Based on slide by E. Alpaydin

# When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans cannot explain their expertise (speech recognition)
- Algorithms must be customized (personalized medicine)

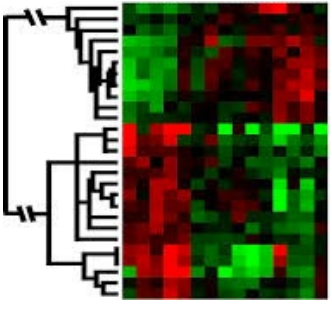
Slide credit: Eric Eaton
Based on slide by E. Alpaydin

# When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans cannot explain their expertise (speech recognition)
- Algorithms must be customized (personalized medicine)
- Data exists to acquire expertise (genomics)

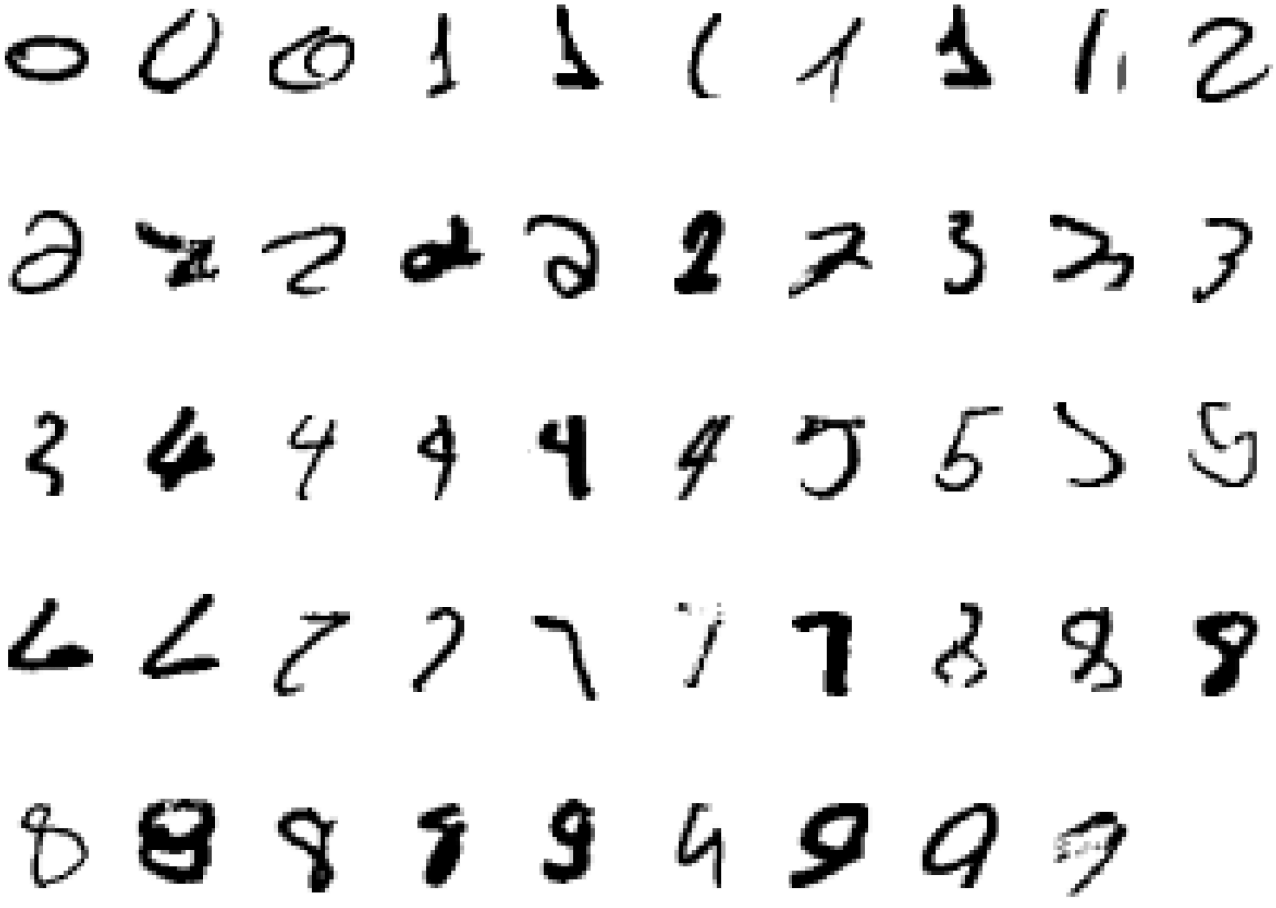


Slide credit: Eric Eaton
Based on slide by E. Alpaydin

A classic example of a task that requires machine learning:

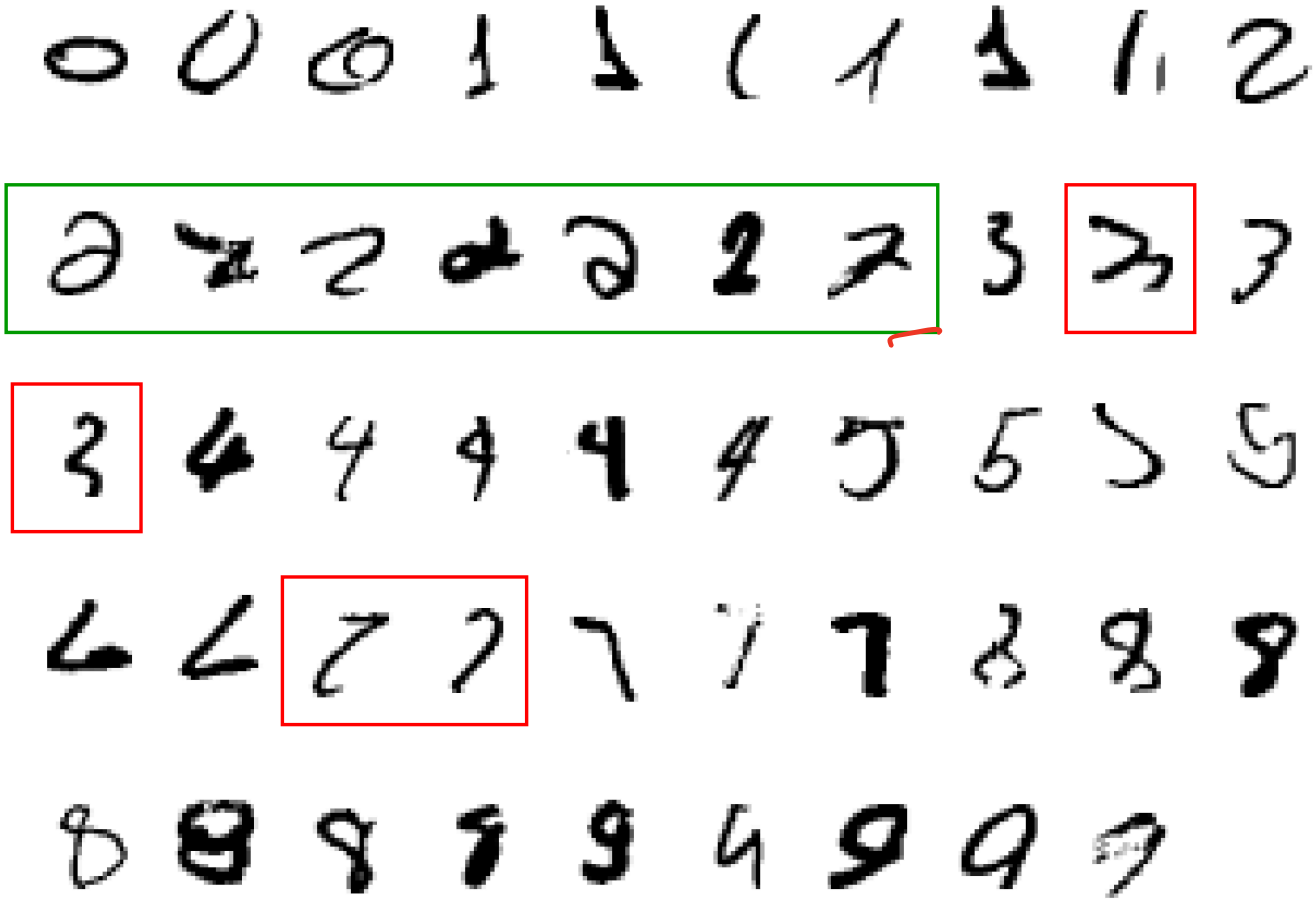
It is very hard to say what makes a 2



Slide credit: Geoffrey Hinton

A classic example of a task that requires machine learning:

It is very hard to say what makes a 2



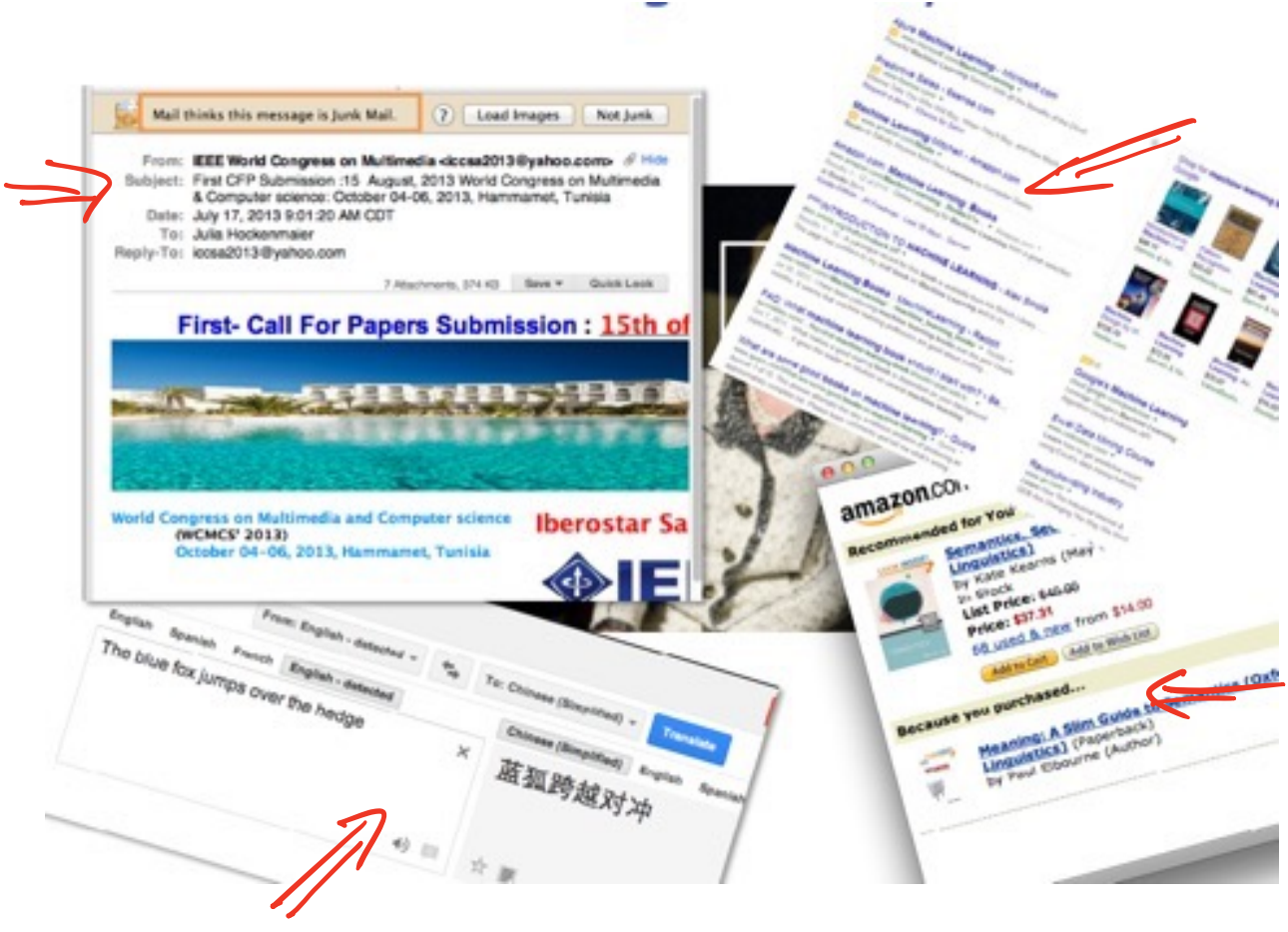
Slide credit: Geoffrey Hinton

# Why study ML now ?

An exciting time

- Initial algorithms and theory in place
- Growing amounts of data
- Growing computational power
- New applications

# Machine learning is everywhere



# ML is interdisciplinary

Makes use of

- Probability and statistics; Linear algebra; Calculus, Algorithms

Related to

- Philosophy, Psychology, Neuroscience, Linguistics, Vision, Robotics

Has applications in

- AI (Natural Language, Vision, HCI)
- Engineering
- Biology and medicine

# Administrivia

# CM146 Team

Sriram Sankararaman (sriram@cs.ucla.edu)

Monday 6:00-7:00 PM PST

TAs

| TA | Email | Office Hours | Location |
|---|---|---|---|
| Ulzee An | ulzee@ucla.edu | W/F 11am-12pm | https://ucla.zoom.us/u/abiJZ4Sq1t |
| Fan Yin | fanyin20@cs.ucla.edu | T/Th / 11am-12pm | Room# Building (or Zoom link ) |
| Boyang Fu | boyang19@ucla.edu | T/Th 1pm-2pm | Room# Building (or Zoom link ) |
| Da Yin | da.yin@cs.ucla.edu | T/Th 8am-9am | https://ucla.zoom.us/j/9011904726 |

# Registration

Course is currently full

Students on the waiting list will be enrolled in case some one drops.

No guarantees though

Only be giving PTEs till after the first math min-quiz.

# Registration

Expect several students will drop the course.

Course requires mathematical maturity

Last offering did see attrition later in the quarter.

Math mini-quiz intended to solve this.

Representative of the math needed for this course.

Course requirement.

Graded to assess background but not part of the final grade.

Encouraged to be honest/realistic about your background.

# Prerequisites

Probability and statistics

Linear algebra

Algorithms

Multivariate calculus

# Prerequisites

Probability and statistics

Linear algebra

Algorithms

Multivariate calculus

Programming experience needed

Python, numpy and scikit-learn (a machine learning library for python)

# Math background review

Math quiz on gradescope

Will be released on gradescope later this week

Due on Sunday 11:59 pm PST

You will have about 1 hour to complete the quiz once you begin

Score will not count towards your final grade but is intended to give you feedback

We will not grade any other problem sets/exams unless you attempt this quiz.

# Online lecture

First two weeks of lectures will be online.

When we get back to in-person class, we will also have concurrent zoom session.

Recorded videos of lectures will be made available.

Turn your video on if possible.

Participate in discussions and online polls.

Ask questions either using audio or chat.

# Textbooks

## No one textbook

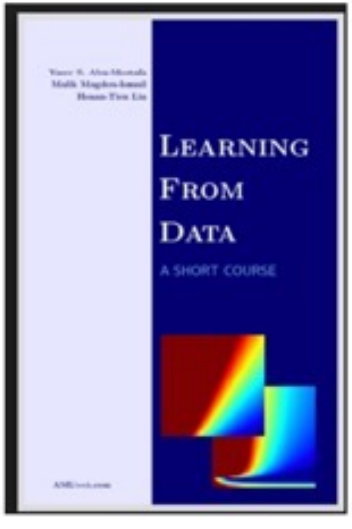
Primary reference: A course in machine learning by Hal Daume III (CIML). Freely available online : http://ciml.info/

See syllabus for reading list.

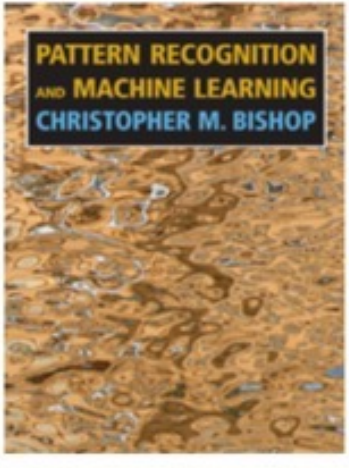
# Textbooks

No one textbook

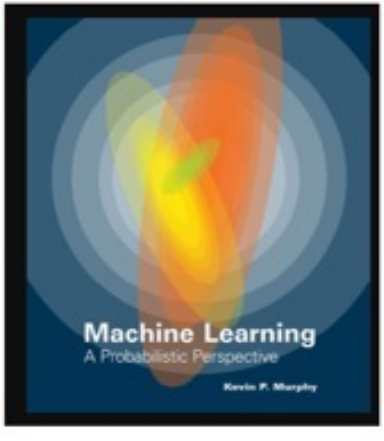
Other references




Basic




Comprehensive




Advanced

# Course format

## Problem sets (aka homeworks) (50%)

Four problem sets (numbered 1 to 4)

Due at 11:59pm EST on the due date

Late submissions not accepted

Will be using gradescope to manage submissions (will send out submission instructions)

All solutions must be clearly written or typed. Unreadable answers will not be graded. We encourage using LaTeX to typset answers.

Solutions will be graded on both correctness and clarity.

You are free to discuss homework problems. However, you must write up your own solutions. You must also acknowledge all collaborators.

# Course format

## Quizzes (30%)

- Starting week 2 (total of seven quizzes not including the math quiz at end of week 1)
- Answer quizzes in gradescope testing the material covered that week.
- Open book/notes.
- Will drop lowest scoring quiz.
- You will have an hour to finish the quiz once you start.

# Course format

## Exams (Final: 20%)

Scheduled for March 14 (11:30am - 2:30pm Franz Hall).

Exams are will cover material from the lectures and the problem sets.

Will cover all material.

No alternate or make-up exams will be administered, except for disability/medical reasons documented and communicated to the instructor prior to the exam date. In particular, exam dates and times cannot be changed to accommodate scheduling conflicts with other classes.

# Course format

Default cutoff for letter grades

| > 96 | 93 | 90 | 86 | 83 | 80 | 76 | 73 | 70 | < 70 |
|---|---|---|---|---|---|---|---|---|---|
| A + | A | A- | B+ | B | B- | C+ | C | C- | D |

We will not make adjustments for individuals.

No round-up.

This is a heavy course.

# Policies

## Attendance and class participation

Although not a formal component of the grade, attendance is important.

We look forward to your active participation.

If you are absent without a documented excuse, the instructor and TA will not be able to go over missed lecture material.

## Video lectures

We aim to make it available (no guarantees).

You should not rely on recordings as a substitute for lectures.

# Regrade requests

Regrade requests must be made within one week after the graded homeworks have been handed out, regardless of your attendance on that day and regardless of any intervening holidays such as Memorial Day.

We reserve the right to regrade all problems for a given regrade request.

# Academic integrity policy

## No cheating

You are free to discuss homework problems. However, you must write up your own solutions. You must acknowledge all collaborators.

Don’t use any old solutions.

Don’t post your HW/Exam solutions.

All incidents will be reported to the student offices.

# Forums

## Campuswire

Must have already got an email

Otherwise email me or your TA.

Strongly encourage students to post here rather than email course staff directly (you will get a faster response this way)

If you do need to contact the staff privately, campuswire allows you to do this.

# Forums

## Gradescope

For homework and quiz submissions.

For homeworks, you will need to upload pdfs of your submission to gradescope.

# Questions?

# Why take this course ?

## Build fundamentals

**A Regularized Framework for Sparse and Structured Neural Attention**

Vlad Niculae* Cornell University Ithaca, NY vlad@cs.cornell.edu

Mathieu Blondel NTT Communication Science Laboratories Kyoto, Japan mathieu@mblondel.org

**Abstract**

Modern neural networks are often augmented with an attention mechanism, which tells the network where to focus within the input. We propose in this paper a new framework for sparse and structured attention, building upon a smoothed max operator. We show that the gradient of this operator defines a mapping from real values to probabilities, suitable as an attention mechanism. Our framework includes softmax and a slight generalization of the recently-proposed sparsemax as special cases. However, we also show how our framework can incorporate modern structured penalties, resulting in more interpretable attention mechanisms, that focus on entire segments or groups of an input. We derive efficient algorithms to compute the forward and backward passes of our attention mechanisms, enabling their use in a neural network trained with backpropagation. To showcase their potential as a drop-in replacement for existing ones, we evaluate our attention mechanisms on three large-scale tasks: textual entailment, machine translation, and sentence summarization. Our attention mechanisms improve interpretability without sacrificing performance; notably, on textual entailment and summarization, we outperform the standard attention mechanisms based on softmax and sparsemax.

**1 Introduction**

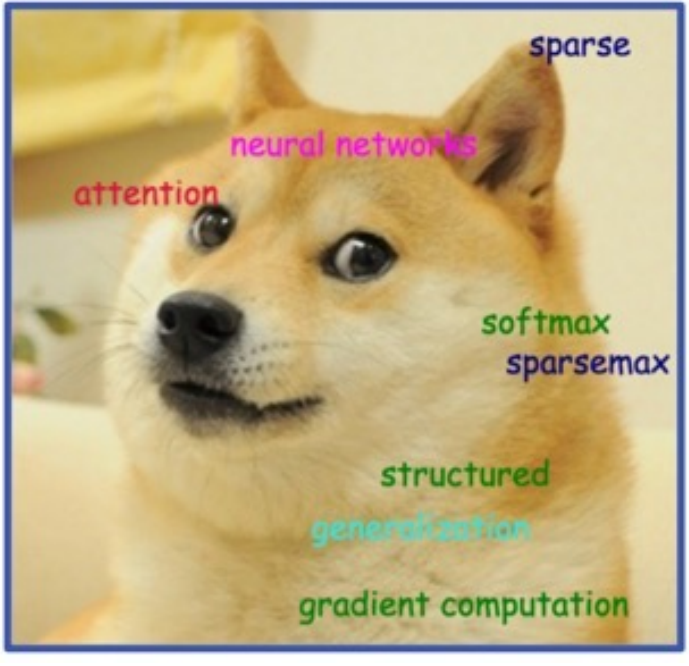
Modern neural network architectures are commonly augmented with an attention mechanism, which tells the network where to look within the input in order to make the next prediction. Attention-augmented architectures have been successfully applied to machine translation [2, 29], speech recognition [10], image caption generation [44], textual entailment [38, 31], and sentence summarization [39], to name but a few examples. At the heart of attention mechanisms is a mapping function that converts real values to probabilities, encoding the relative importance of elements in the input. For the case of sequence-to-sequence prediction, at each time step of generating the output sequence, attention probabilities are produced, conditioned on the current state of a decoder network. They are then used to aggregate an input representation (a variable-length list of vectors) into a single vector, which is relevant for the current time step. That vector is finally fed into the decoder network to produce the next element in the output sequence. This process is repeated until the end-of-sequence symbol is generated. Importantly, such architectures can be trained end-to-end using backpropagation.

Alongside empirical successes, neural attention—while not necessarily correlated with human attention—is increasingly crucial in bringing more **interpretability** to neural networks by helping explain how individual input elements contribute to the model's decisions. However, the most commonly used attention mechanisms, *softmax*, yield dense attention weights: all elements in the input always make at least a small contribution to the decision. To overcome this limitation, *sparsemax* was recently proposed [31], using the Euclidean projection onto the simplex as a sparse alternative to

*Work performed during an internship at NTT Communication Science Laboratories, Kyoto, Japan.

Modern neural networks ..... attention mechanism, ... We propose in this paper a new framework for sparse and structured attention, building upon a smoothed max operator. We show that the gradient of this operator defines a mapping from real values to probabilities, suitable as an attention mechanism. Our framework includes softmax and a slight generalization of the recently-proposed sparsemax as special cases. .....

# What this looks like to ML researchers

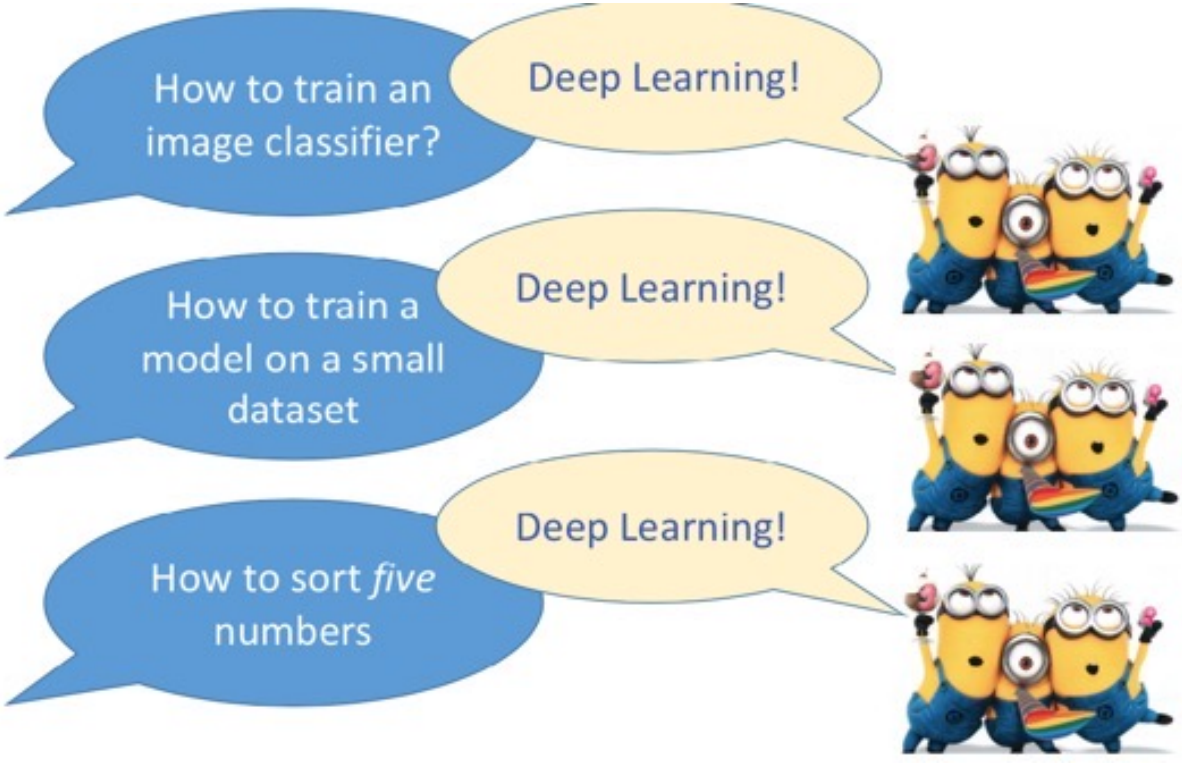


# What this looks like to normal people

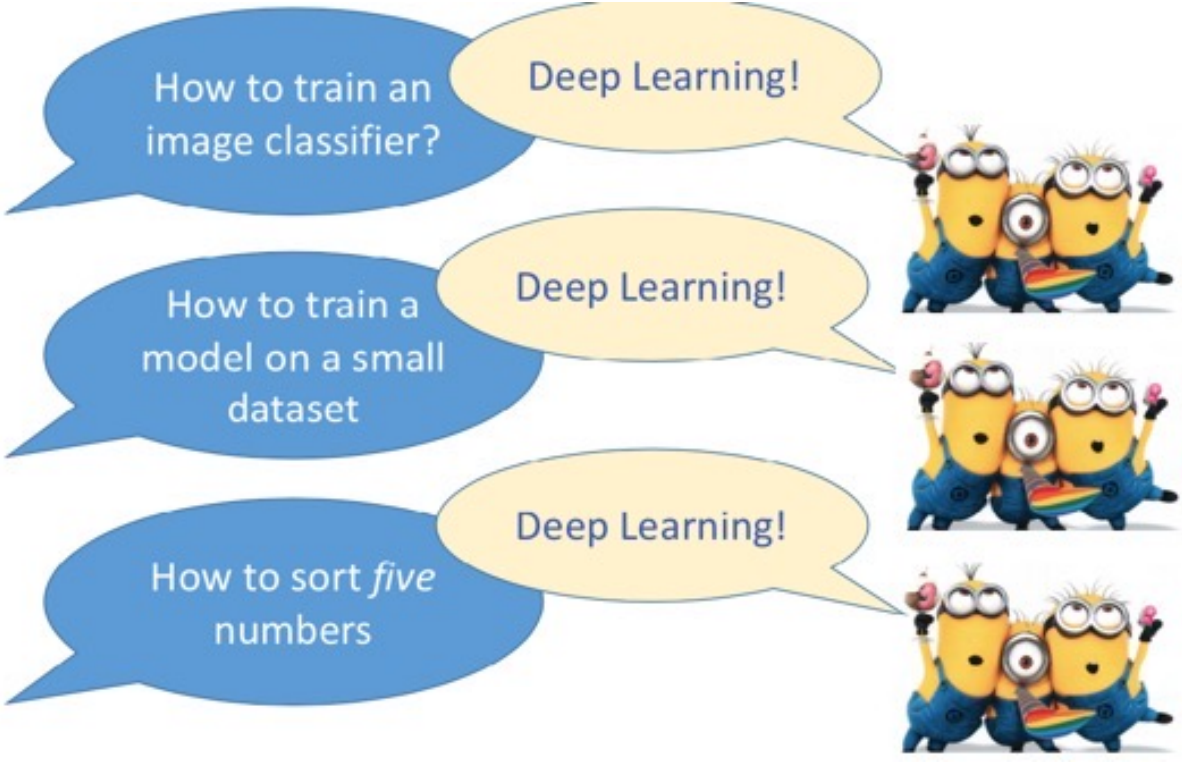
# What this looks like to normal people


After taking this course, hopefully you can begin to read ML papers

# What ML beginners may do



# What ML beginners may do




After taking this course, hopefully you can choose the right model for your application

# Goals of the course: Learn about…

Common techniques/tools used

# Goals of the course: Learn about…

Fundamental concepts and algorithms

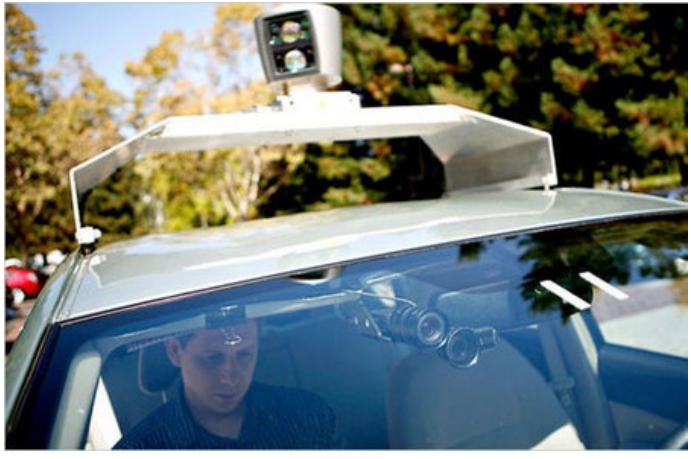
Common techniques/tools used

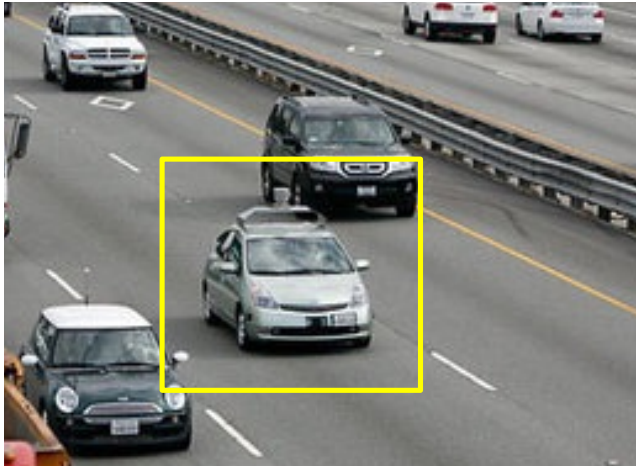
- theoretical understanding
- practical implementation
- best practices

# State of the Art Applications of Machine Learning

Slide credit: Eric Eaton

# Autonomous Cars



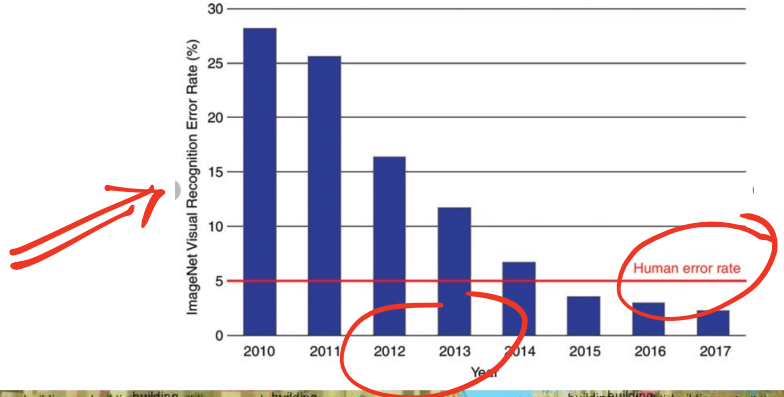


Nevada made it legal for autonomous cars to drive on roads in June 2011

As of 2013, four states (Nevada, Florida, California, and Michigan) have legalized autonomous cars

Slide credit: Eric Eaton

# Computer vision




# Speech recognition







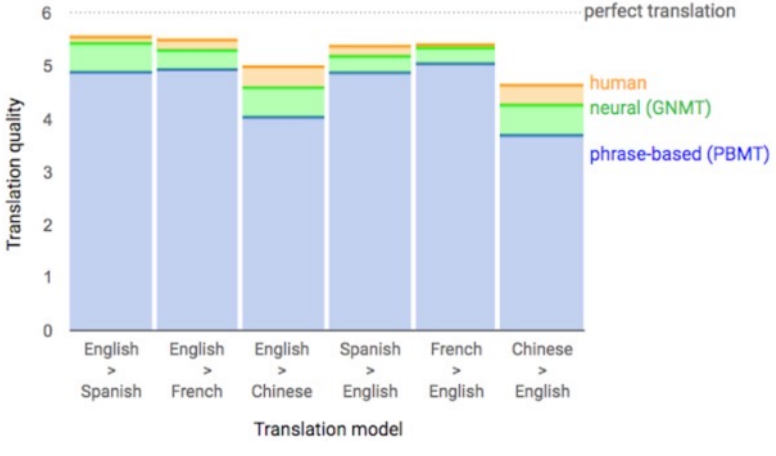
Slide credit: Li Deng, MS Research

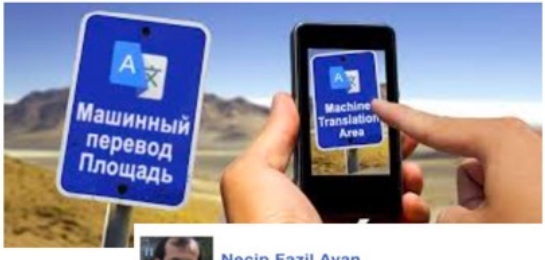
# Machine translation

## The Great A.I. Awakening

How Google used artificial intelligence to transform Google Translate, one of its more popular services — and how machine learning is poised to reinvent computing itself.

BY GIDEON LEWIS-KRAUS DEC. 14, 2016




Necip Fazil Ayan
1 hr

Onların, İzmir'in neden hayır dediğini anlamalarını beklemiyoruz.

We don't expect them to understand why Izmir said no.

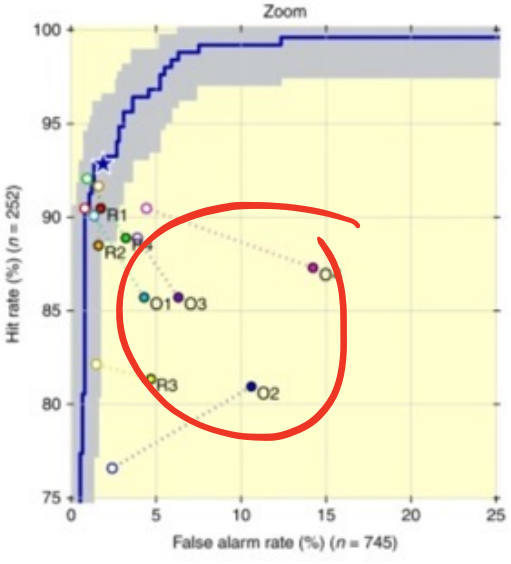
· Rate this translation

# Game playing


# Medicine

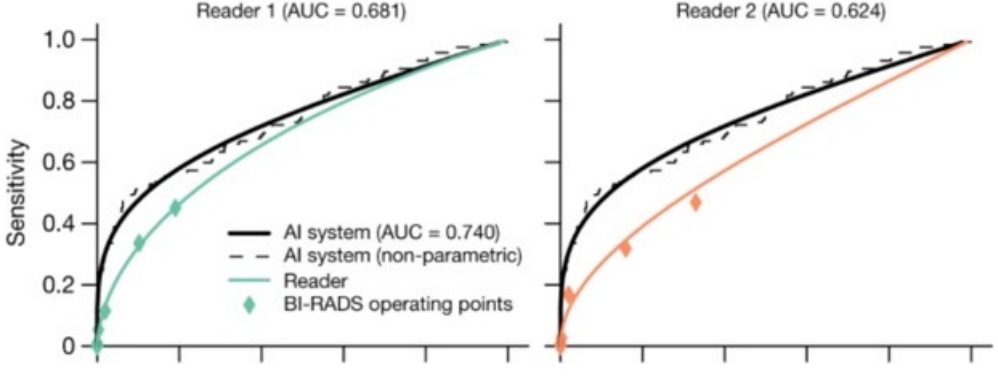
## Clinically applicable deep learning for diagnosis and referral in retinal disease

Jeffrey De Fauw, Joseph R. Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, George van den Driessche, Balaji Lakshminarayanan, Clemens Meyer, Faith Mackinder, Simon Bouton, Kareem Ayoub, Reena Chopra, Dominic King, Alan Karthikesalingam, Cían O. Hughes, Rosalind Raine, Julian Hughes, Dawn A. Sim, Catherine Egan, Adnan Tufail, Hugh Montgomery, Demis Hassabis, Geraint Rees, Trevor Back, Peng T. Khaw, Mustafa Suleyman, Julien Cornebise, Pearse A. Keane & Olaf Ronneberger




## *A.I. Is Learning to Read Mammograms*

Computers that are trained to recognize patterns and interpret images may outperform humans at finding cancer on X-rays.



# Types of Learning

Slide credit: Eric Eaton

# Types of Learning

**Supervised (inductive) learning :** Learn with a teacher

Given: labeled training instances (or examples)

Goal: learn mapping that predicts label for test instance

**Unsupervised learning :** Learn without a teacher

Given: unlabeled inputs

Goal: learn some intrinsic structure in inputs

**Reinforcement learning:** Learn by interacting

Given agent interacting in environment (having set of states)

Learn policy (state to action mapping) that maximizes agent's reward

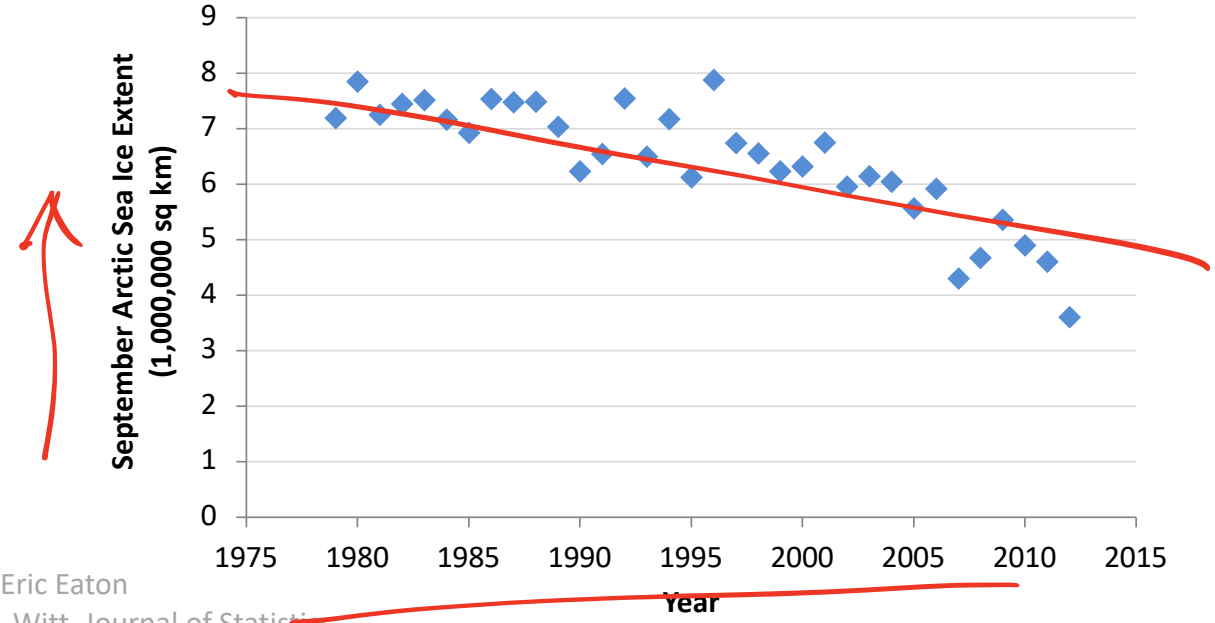
Slide credit: Eric Eaton
Based on slide by Pedro Domingos

# Supervised Learning: Regression

Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

Learn a function $f(x)$ to predict $y$ given $x$

$y$ is real-valued == regression




Slide credit: Eric Eaton

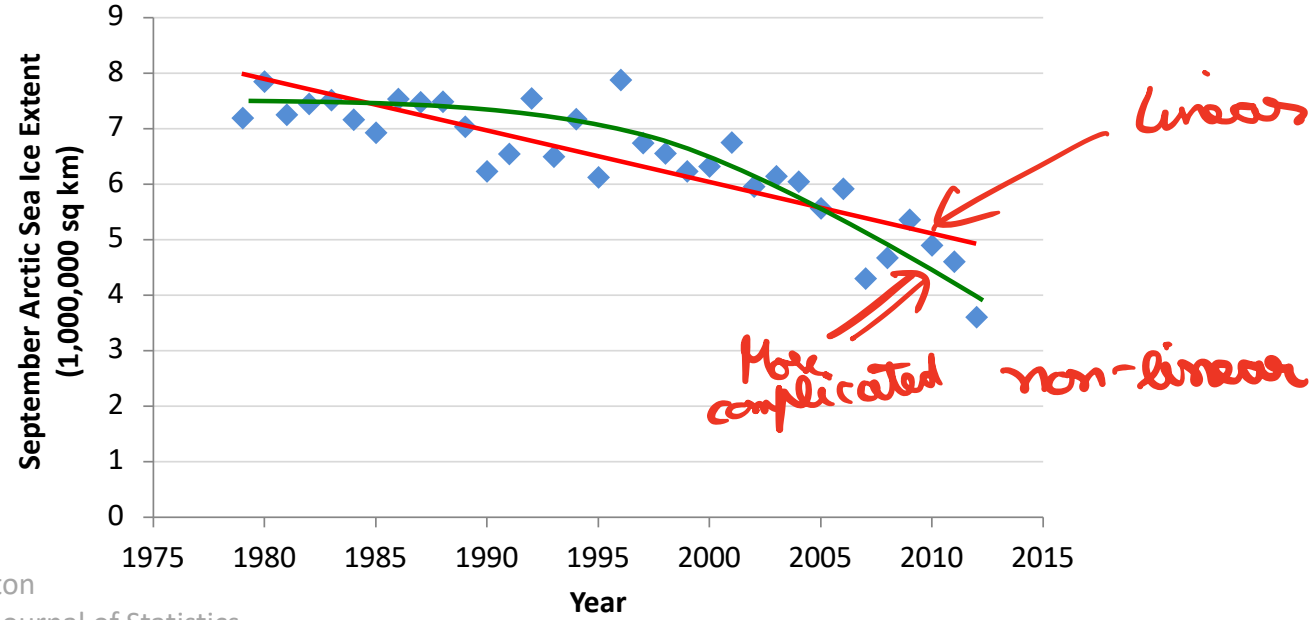
Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013)

# Supervised Learning: Regression

Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

Learn a function $f(x)$ to predict $y$ given $x$

$y$ is real-valued == regression




Slide credit: Eric Eaton

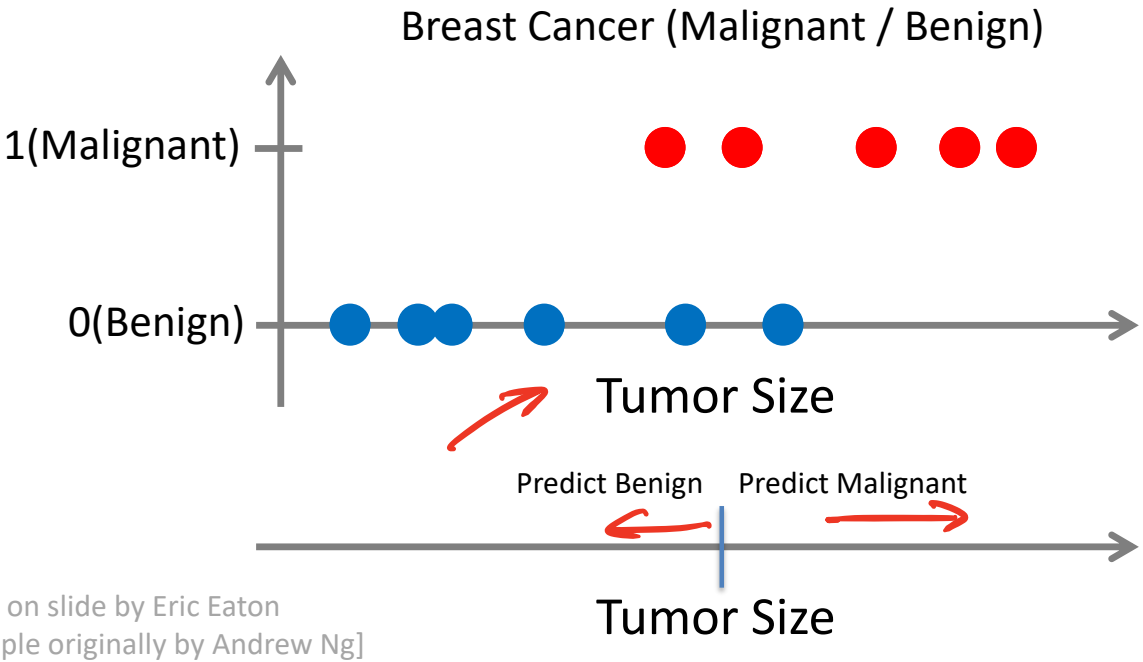
Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013)

# Supervised Learning: Classification

Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

Learn a function $f(x)$ to predict $y$ given $x$

$y$ is categorical == classification

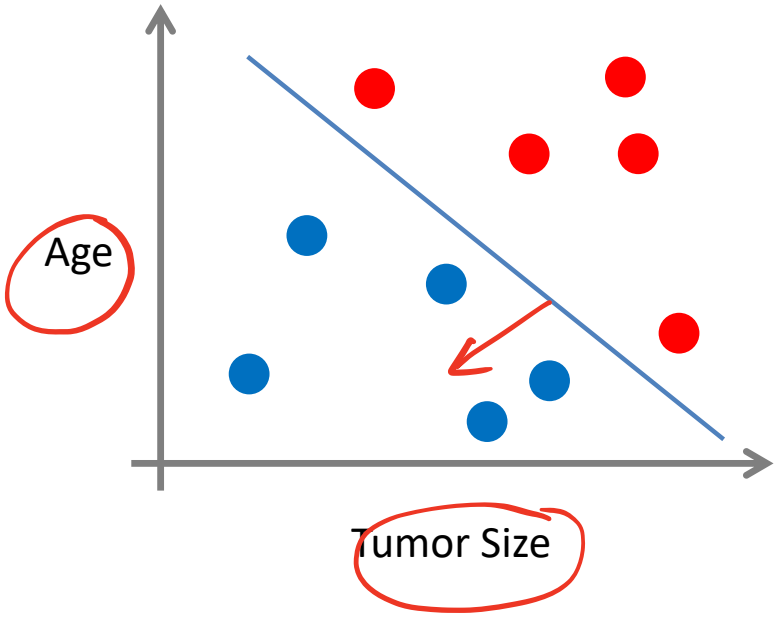



Based on slide by Eric Eaton
[example originally by Andrew Ng]

# Supervised Learning

*x* can be multi-dimensional

Each dimension corresponds to an attribute




- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape

…

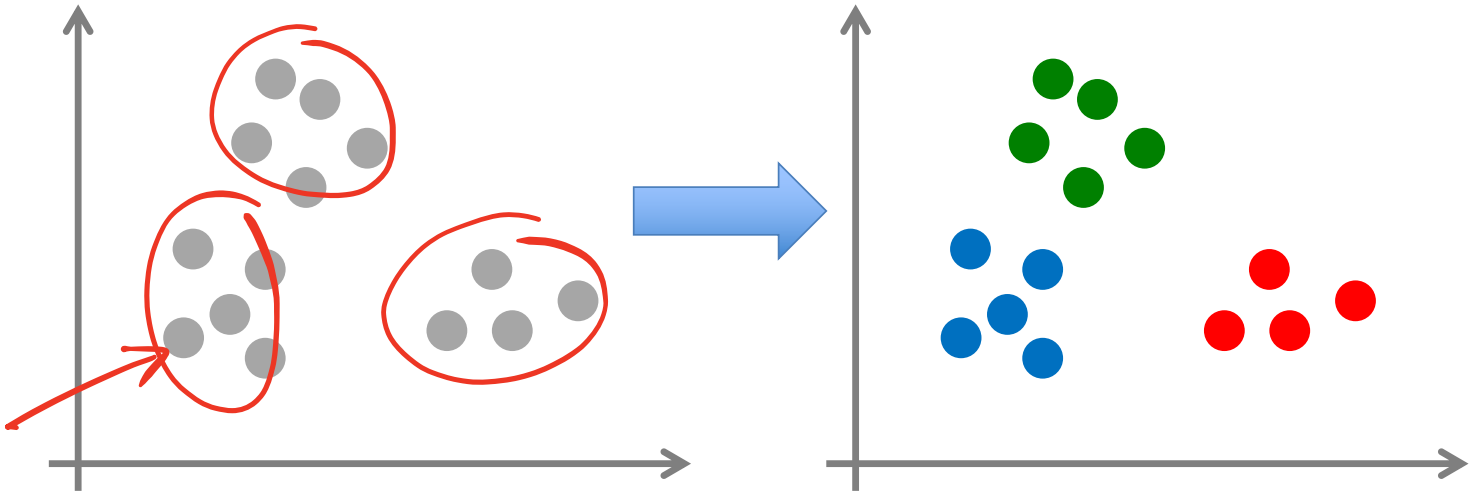
Based on slide by Eric Eaton
[example originally by Andrew Ng]

# Unsupervised Learning

Given $x_1, x_2, \ldots, x_n$ (without labels)

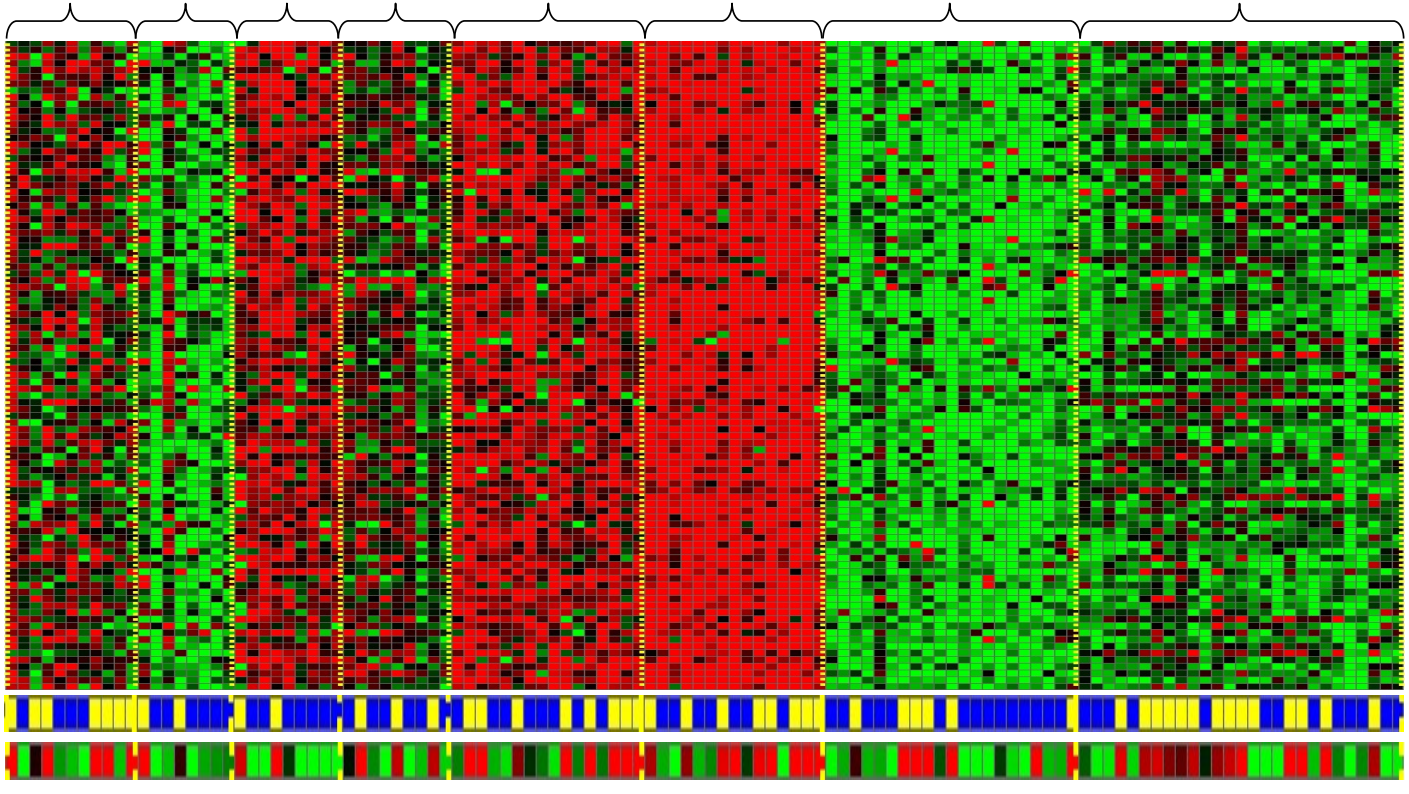
Output hidden structure behind the $x$'s

Clustering



# Unsupervised Learning

Genomics application: group individuals by genetic similarity




Slide credit: Eric Eaton
[Source: Daphne Koller]

# Reinforcement Learning

Given sequence of states and actions with (delayed) rewards

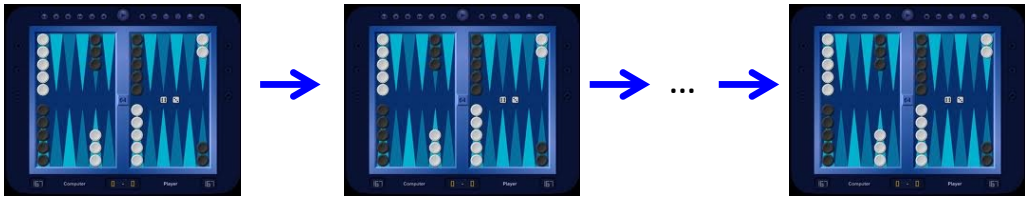
Learn policy that maximizes agent's reward

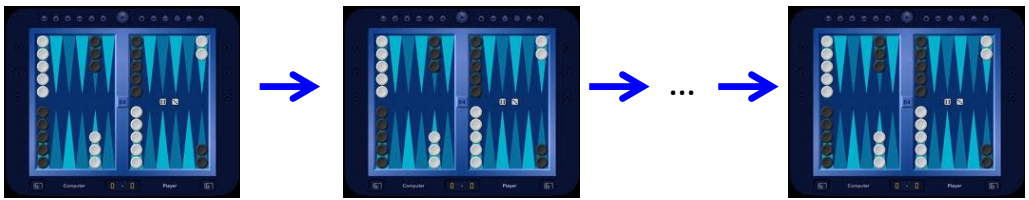
Examples:

- Game playing
- Robot in maze

# Reinforcement Learning

Backgammon




Given sequences of moves and whether or not the player won at the end, learn to make good moves

Slide credit: David Kauchak

# Framing a Learning Problem

Slide credit: Eric Eaton

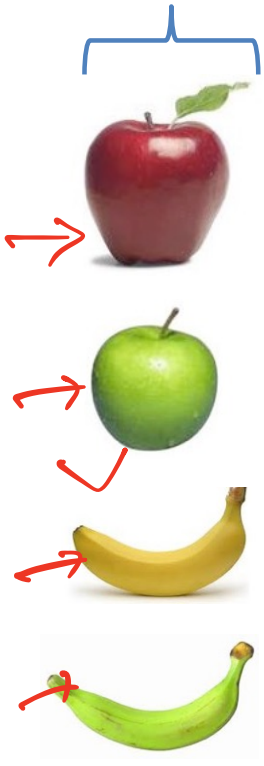
# Representing instances/examples

What is an instance?

How is it represented?

instances

features



$feat_1$, $feat_2$, $feat_3$, $feat_4$,…
red, round, leaf, 3oz, …

$feat_1$, $feat_2$, $feat_3$, $feat_4$,…
green, round, no leaf, 4oz, …

$feat_1$, $feat_2$, $feat_3$, $feat_4$,…
yellow, curved, no leaf, 4oz, …

$feat_1$, $feat_2$, $feat_3$, $feat_4$,…
green, curved, no leaf, 5oz, …

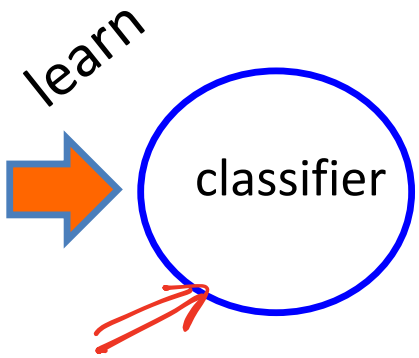
How our algorithms actually "view" the data

Features are the questions we can ask about the instances

Based on slide by David Kauchak

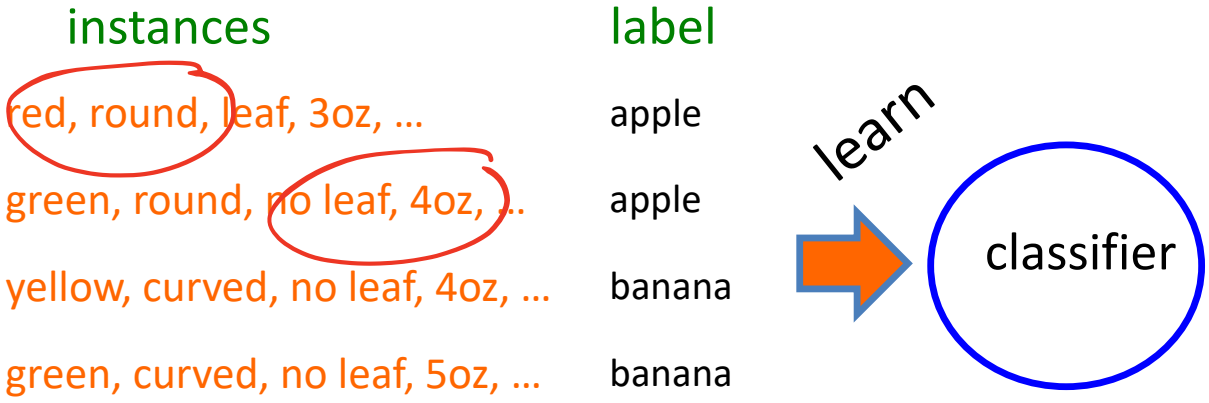
# Learning algorithm

| instances | label |
| --- | --- |
| red, round, leaf, 3oz, … | apple |
| green, round, no leaf, 4oz, … | apple |
| yellow, curved, no leaf, 4oz, … | banana |
| green, curved, no leaf, 5oz, … | banana |

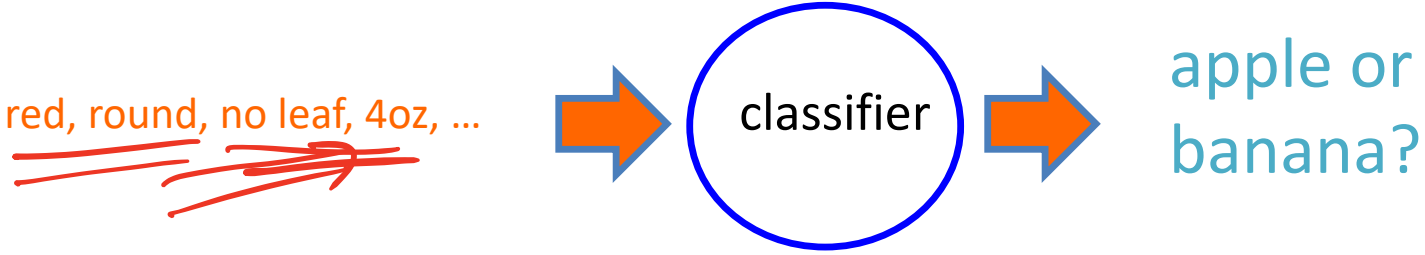



During learning/training/induction, learn a model of what distinguishes apples and bananas *based on the features*

# Learning algorithm




During learning/training/induction, learn a model of what distinguishes apples and bananas *based on the features*

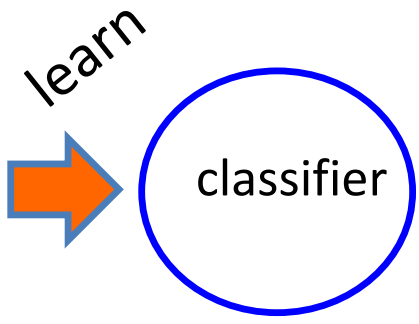



The classifier classifies a new instance *based on the features*

Based on slide by David Kauchak

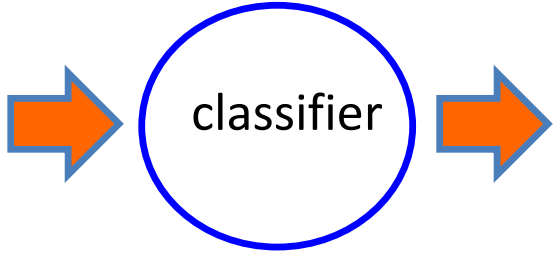
# Learning algorithm

| instances | label |
| --- | --- |
| red, round, leaf, 3oz, … | apple |
| green, round, no leaf, 4oz, … | apple |
| yellow, curved, no leaf, 4oz, … | banana |
| green, curved, no leaf, 5oz, … | banana |




During learning/training/induction, learn a model of what distinguishes apples and bananas *based on the features*

red, round, no leaf, 4oz, …




apple or banana?

apple

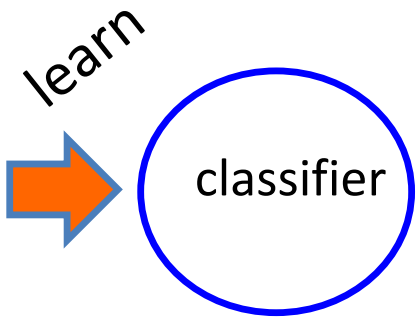
Why?

The classifier classifies a new instance *based on the features*

Based on slide by David Kauchak

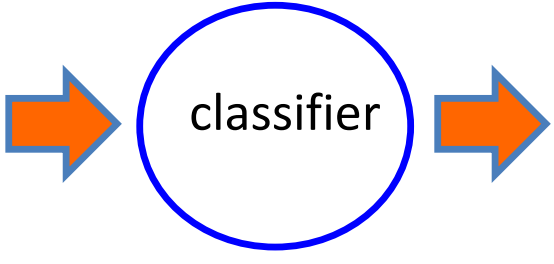
# Learning algorithm

| instances | label |
| --- | --- |
| red, round, leaf, 3oz, … | apple |
| green, round, no leaf, 4oz, … | apple |
| yellow, curved, no leaf, 4oz, … | banana |
| green, curved, no leaf, 5oz, … | banana |




During learning/training/induction, learn a model of what distinguishes apples and bananas *based on the features*

red, round, no leaf, 4oz, …




apple or banana?

apple

The classifier classifies a new instance *based on the features*

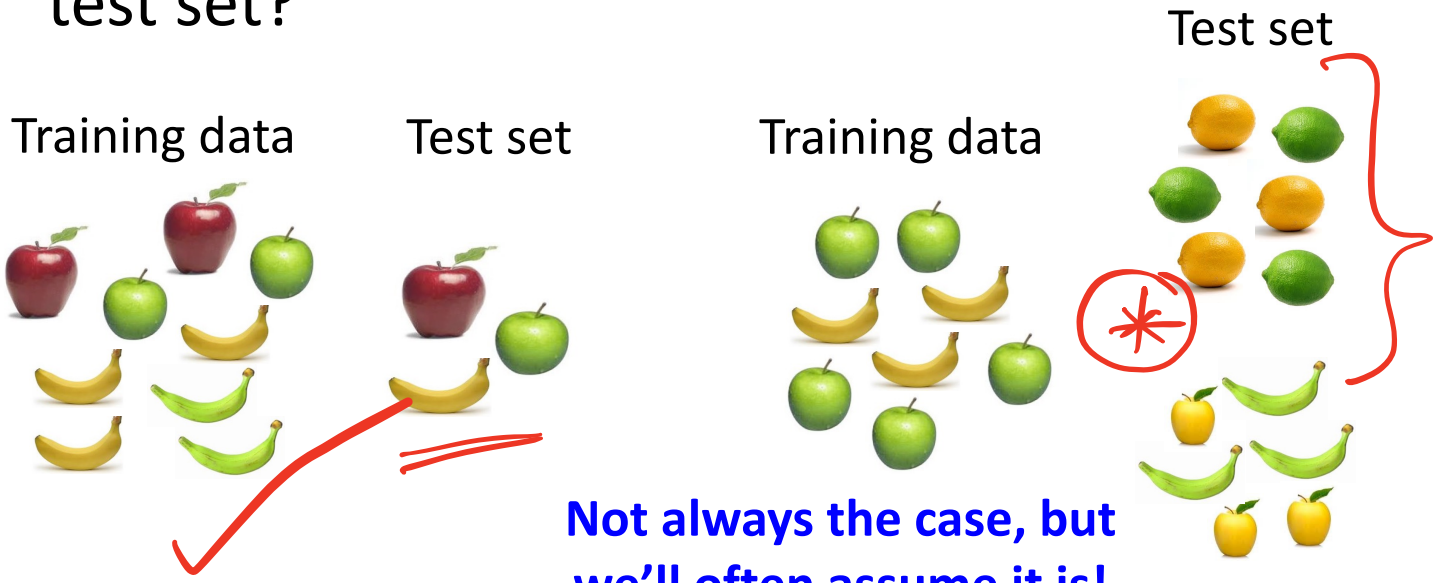
Based on slide by David Kauchak

# Learning algorithm

- Learning is about ***generalizing*** from training data
- What does this ***assume*** about training and test set?

# Learning algorithm

- Learning is about ***generalizing*** from training data
- What does this ***assume*** about training and test set?




**Not always the case, but we'll often assume it is!**

Based on slide by David Kauchak

We care about the performance of the learning algorithm on test data (generalization ability).

How we measure performance depends on the problem we are trying to solve

The training and test data should be strongly related.

# More Technically…

We start with a loss function $L(y, \hat{y})$

Tells us how bad the prediction of y is compared to the true value of y

A loss function for regression (squared loss)

$$L(y, \hat{y}) = (y - \hat{y})^2$$

A loss function for classification

$$L(y, \hat{y}) = \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{otherwise} \end{cases}$$

Based on slide by David Kauchak

# More Technically…

We are going to use the ***probabilistic model*** of learning

There is some **unknown** probability distribution $p$ over instance/label pairs called the ***data generating distribution***

$p(x, y)$

# Learning problem


Defined by

- Loss function : measures performance
- Data generating distribution : what data do we expect to see (characterizes experience)

# Learning problem

**Problem Setting**

- Set of possible instances X
- Set of possible labels Y
- Unknown target function $f$ : X ➔ Y
- Set of function hypotheses $H = \{h \mid h : X \rightarrow Y\}$

**Input**: Training instances drawn from data generating distribution *p*

$$\{(x_i, y_i)\}_{i=1}^{n} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

$p(x_1, y_1) = 0.1$ $p(x_2, y_2) = 0.01$

**Output**: Hypothesis *h* in *H* that best approximates *f*

Slide credit: Eric Eaton
Based on slide by Tom Mitchell

# Learning problem

**Output**: Hypothesis *h* in *H* that best approximates *f*

*h* should do well (as measured by the loss) on future instances

$$L(y, h(x))$$

$$p(x_1, y_1) L(y_1, h(x_1)) + p(x_2, y_2) L(y_2, h(x_2)) + p(x_3, y_3) L(y_3, h(x_3)) + \dots + \dots$$

Slide credit: Eric Eaton
Based on slide by Tom Mitchell

# Learning problem

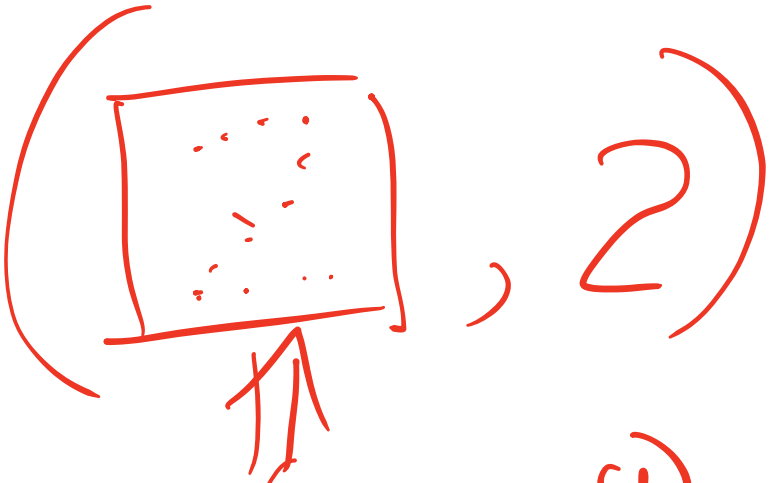
**Output**: Hypothesis *h* in *H* that best approximates *f*

*h* should do well (as measured by the loss) on future instances

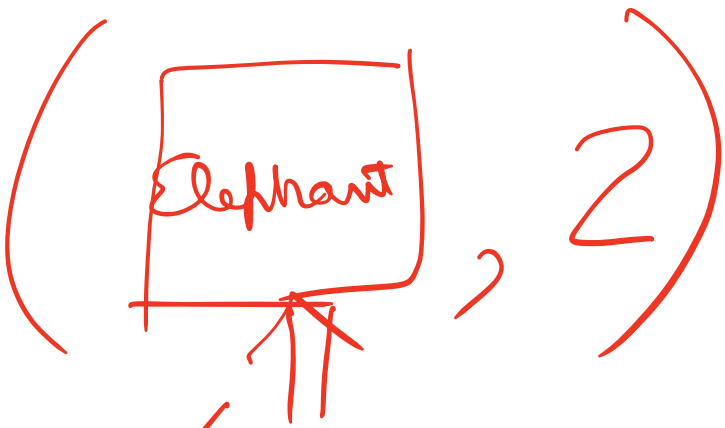
Formally, *h* should have low expected (test) loss/Risk

$$\mathbb{E}_{(x,y)\sim p}\left[L(y, h(x))\right] = \sum_{x,y} p(x, y) L(y, h(x))$$

$h_1$ $h_2$



$P(x, y) = 0.2$




$P(x, y) = 10^{-10}$

# Learning problem

**Output**: Hypothesis *h* in *H* that best approximates *f*

*h* should do well (as measured by the loss) on future instances

Formally, *h* should have low expected (test) loss/Risk

$$\mathbb{E}_{(x,y)\sim p}\left[L(y,h(x))\right] = \sum_{x,y} p(x,y)L(y,h(x))$$

Problem?

We don't know what *p* is

Slide credit: Eric Eaton
Based on slide by Tom Mitchell

# Learning problem

**Output**: Hypothesis *h* in *H* that best approximates *f*

*h* should do well (as measured by the loss) on future instances

Formally, *h* should have low expected (test) loss/Risk

$$\mathbb{E}_{(x,y)\sim p}\left[L(y,h(x))\right] = \sum_{x,y} p(x,y) L(y,h(x))$$

Problem?

We don't know what *p* is

But we are given samples drawn from *p*

Slide credit: Eric Eaton
Based on slide by Tom Mitchell

# Learning problem

We instead approximate the risk by the training error/Empirical risk

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i, h(x_i))$$

When is this reasonable ?

**Both** the training data **and** the test set are generated based on this distribution

Problem?

# Learning problem

We instead approximate the risk by the training error/empirical risk

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i, h(x_i))$$

When is this reasonable ?

**Both** the training data **and** the test set are generated based on this distribution

Problem?

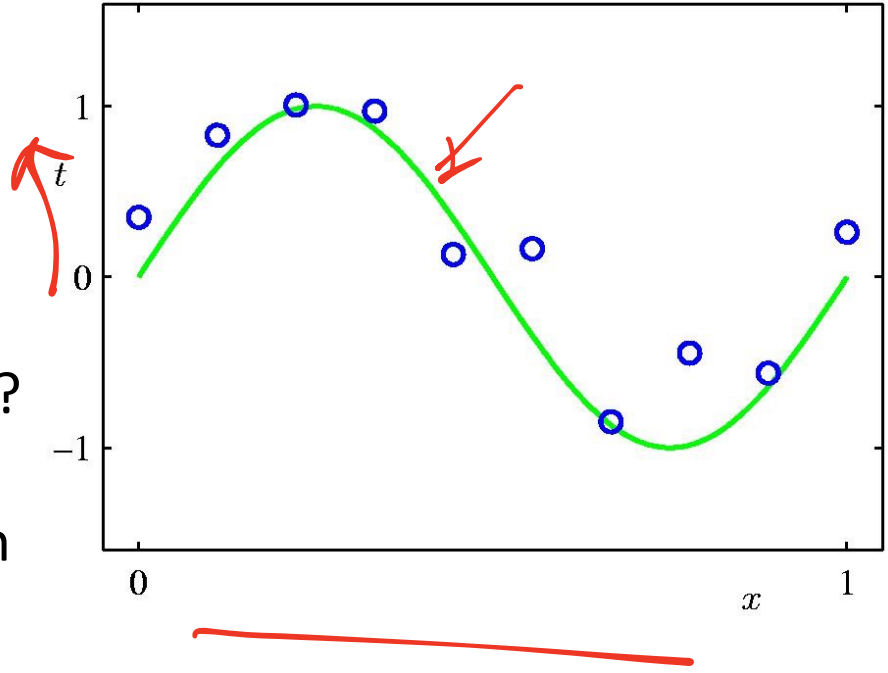
Can make the training error zero by memorizing

# Example Regression Problem

- Consider simple regression dataset
  - $f : X \rightarrow Y$
  - $x \in R$
  - $y \in R$

- Question 1: How should we pick the hypothesis space *H* ?
- Question 2: How do we find the best *h* in this space?

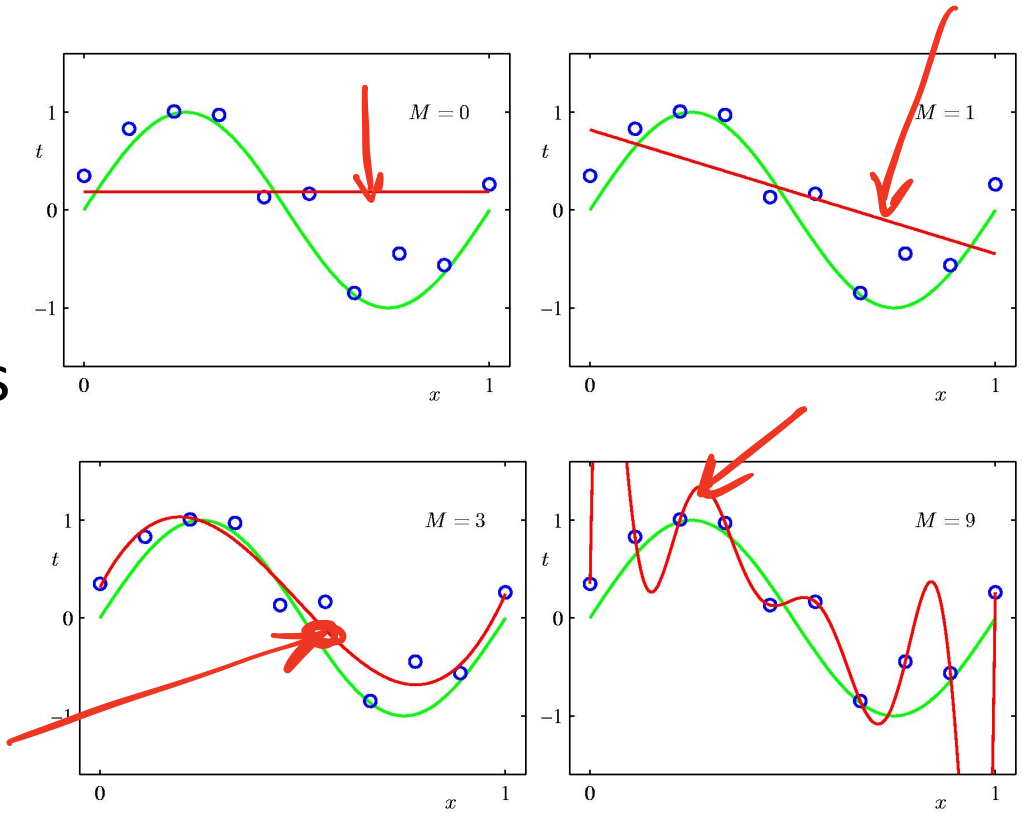
Dataset: 10 points generated from sin function with noise




Based on slide by David Sontag
Images from Bishop [PRML]

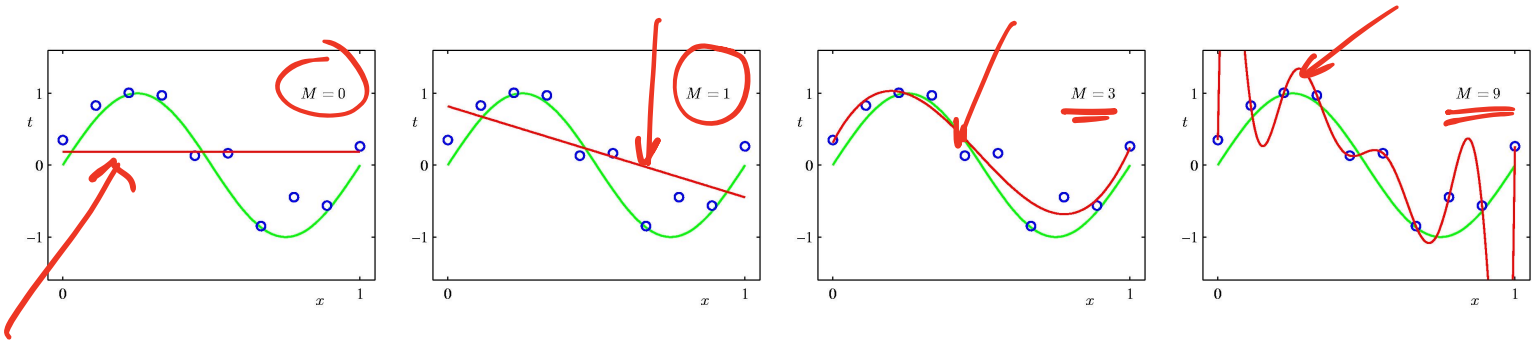
# Hypothesis Space: Degree-M Polynomials

- Infinitely many hypotheses
- Which one is **best**?




Based on slide by David Sontag
Images from Bishop [PRML]

# Hypothesis Space: Degree-M Polynomials




- For regression, common choice is squared loss

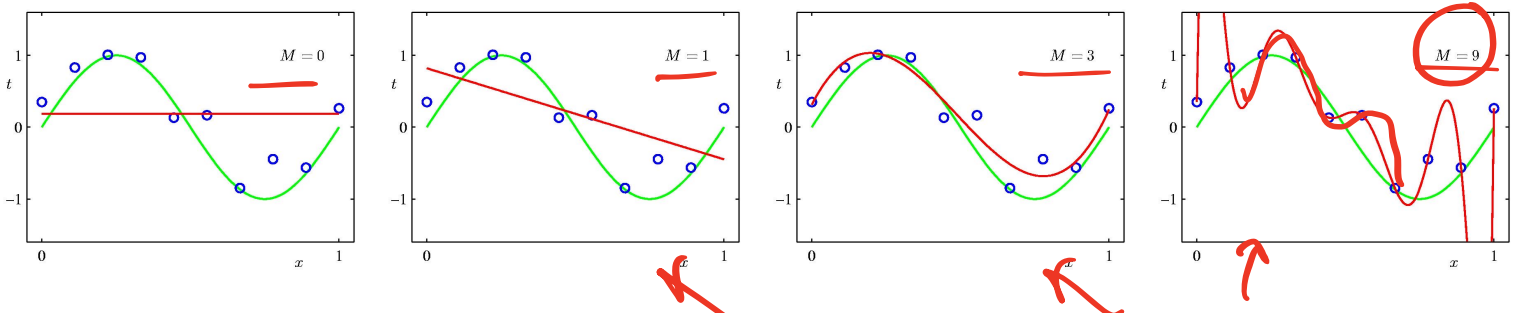
$$L(y_i, h(x_i)) = (y_i - h(x_i))^2$$

- *Empirical loss* of function *h* applied to training data is then

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

Based on slide by David Sontag
Images from Bishop [PRML]

# Hypothesis Space: Degree-M Polynomials



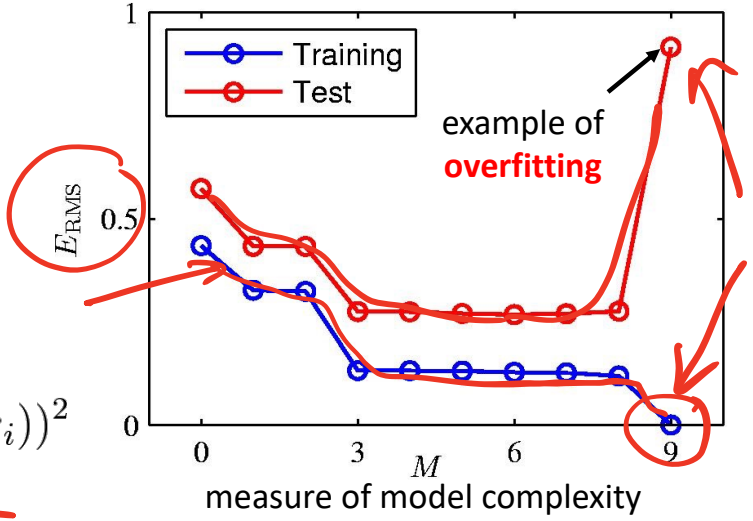

- For regression, common choice is squared loss

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

- *Empirical loss* of function *h* applied to training data is then

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

Learning Curve




Based on slide by David Sontag
Images from Bishop [PRML]

# Learning problem

The fundamental difficulty of machine learning

We have access to the training error but really care about the test error

Our learned function needs to generalize beyond the training data

# Key Issues in Machine Learning

**Representation** : How do we choose a hypothesis space?

Often we use **prior knowledge** to guide this choice

The ability to answer the next two questions also affects choice

**Optimization** : How do we find the best hypothesis within this space?

This is an **algorithmic** question, at the intersection of computer science and optimization research.

**Evaluation** : How can we gauge the accuracy of a hypothesis on unseen testing data?

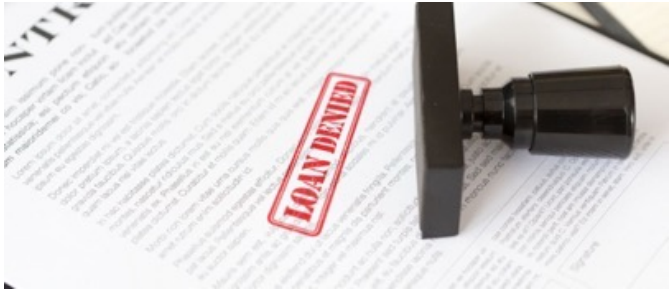
The previous example showed that choosing the hypothesis which simply minimizes training set error (i.e. empirical risk) **is not optimal**

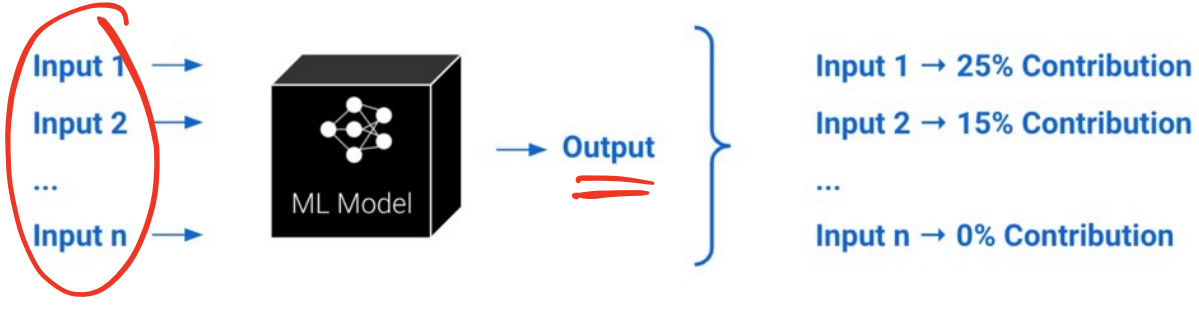
This question is the main topic of **learning theory**

# Challenges

## Interpretable ML

How can a lender explain why a loan application was rejected ?



# Challenges

## Debugging ML

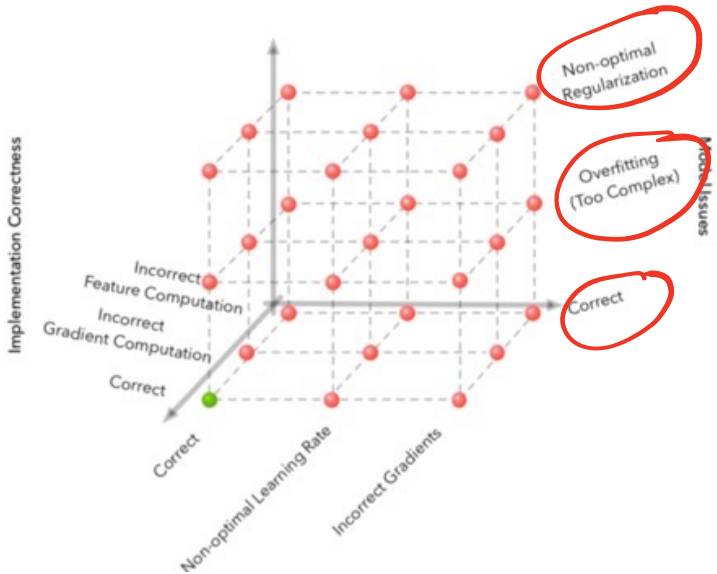
### Debugging a program


https://ai.stanford.edu/~zayd/why-is-machine-learning-hard.html

# Challenges

## Debugging ML

### Debugging ML model

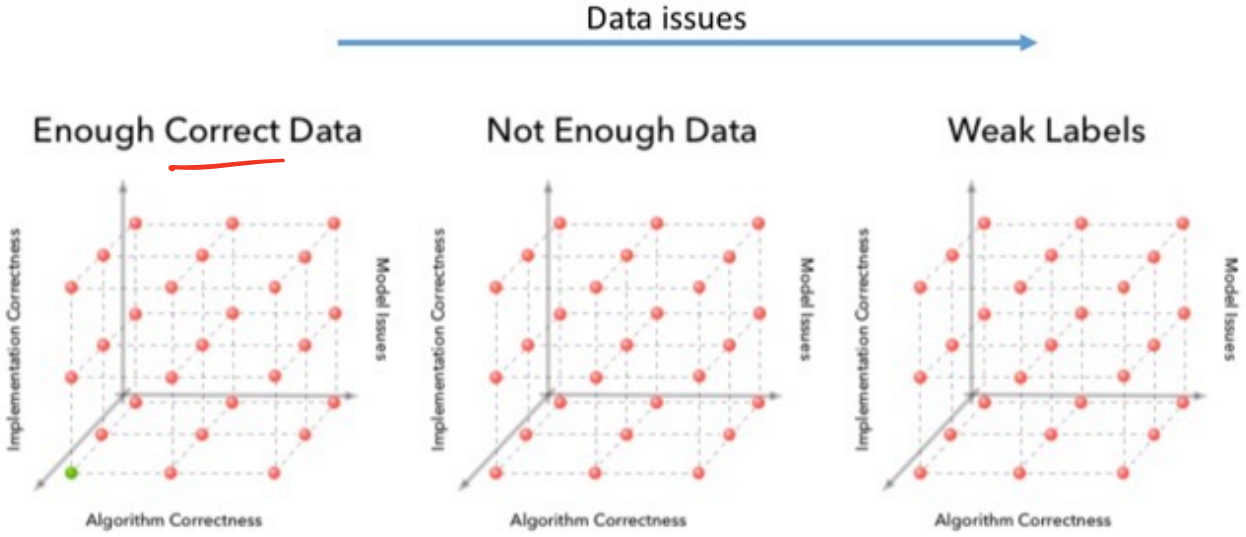



https://ai.stanford.edu/~zayd/why-is-machine-learning-hard.html

# Challenges

## Debugging ML

### Debugging ML model




https://ai.stanford.edu/~zayd/why-is-machine-learning-hard.html

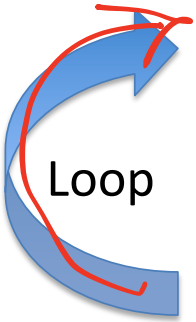
# Challenges

## Robustness

Adversarial attacks on ML models


https://arxiv.org/abs/1712.09327v1

# ML in Practice




Understand domain, prior knowledge, and goals

Data integration, selection, cleaning, pre-processing.

Learn models

Interpret results

Slide credit: Eric Eaton
Based on a slide by Pedro Domingos

# What we will Cover in this Course

**Supervised learning**

- Decision tree
- Perceptron
- Linear regression
- Logistic regression
- Neural networks
- Support vector machines & kernel methods
- Ensemble methods

**Unsupervised learning**

- PCA
- Clustering
- Hidden Markov Models

**Experimental evaluation**

- Cross-validation
- Metrics
- Real datasets!

Based on slide by Eric Eaton

# Summary

Formalizing a learning problem

Given a loss function *L* and a sample from an unknown data generating probability distribution *p*, find a function *h* that has low risk (expected loss)

# Summary

Learning can be viewed as approximating a function

Function approximation can be viewed as search through a space of hypotheses (representations of functions) for one that best fits a set of training data

Different learning methods assume different hypothesis spaces and/or employ different search techniques

# Summary

One of the difficulties is that we can only compute training error but we really want the expected test loss

We need the chosen function to generalize

# Summary

Next class: Decision trees

Take math quiz in gradescope later this week.