# CMSC 471: Machine Learning

Many slides courtesy Frank Ferraro

# Why study learning?

- **Discover** new things or structure previously unknown
  - Examples: data mining, scientific discovery
- Fill in skeletal or **incomplete specifications** in a domain
  - Large, complex systems can't be completely built by hand & require dynamic updating to incorporate new info.
  - Learning new characteristics expands the domain or expertise and lessens the "brittleness" of the system
- Acquire models automatically directly from data rather than by manual programming
- Build agents that can **adapt** to users, other agents, and their environment
- Understand and improve efficiency of **human learning**

# What does it mean to learn?

Wesley has been taking an AI course

Geordi, the instructor, needs to determine if Wesley has "learned" the topics covered, at the end of the course

What is a "reasonable" exam?

- (Bad) Choice 1: History of pottery
  - Wesley's performance is not indicative of what was learned in AI
- (Bad) Choice 2: Questions answered during lectures
  - Open book?

A good test should test ability to answer "related" but "new" questions on the exam

Generalization

# Model, parameters and hyperparameters

Model: mathematical formulation of system (e.g., classifier)

Parameters: primary "knobs" of the model that are set by a learning algorithm

Hyperparameter: secondary "knobs"



http://www.uiparade.com/wp-content/uploads/2012/01/ui-design-pure-css.jpg

Instance of data
(“datum”)

score( Instance of data (“datum”) )

scoring model

$$\text{score}_{\theta}(\ \text{Instance of dat}\ (\text{“datum”})\ )$$

$$\Downarrow$$

objective

$$F(\theta)$$

UMBC

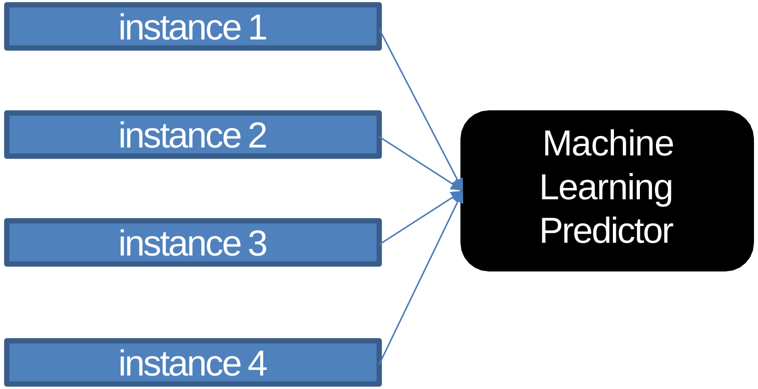
scoring model $\text{score}_\theta($ Instance of data ("datum") $)$

$\Downarrow$

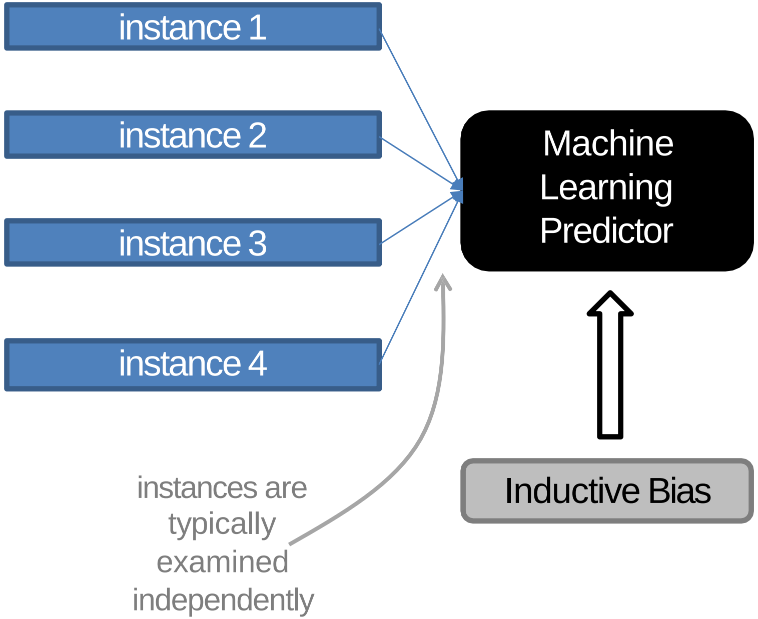
objective $F(\theta)$

*(implicitly) dependent on the observed data X=*

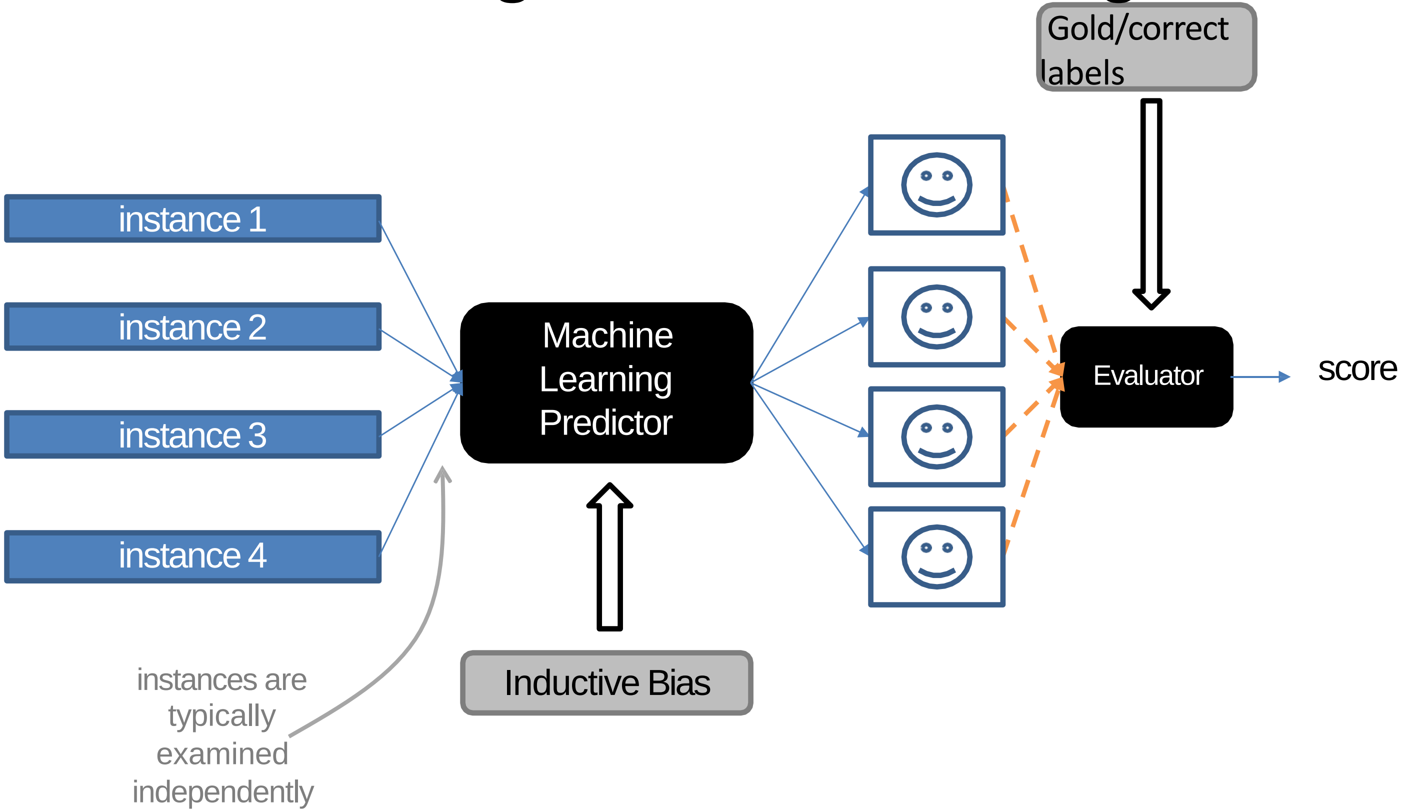
# Machine Learning Framework: Learning



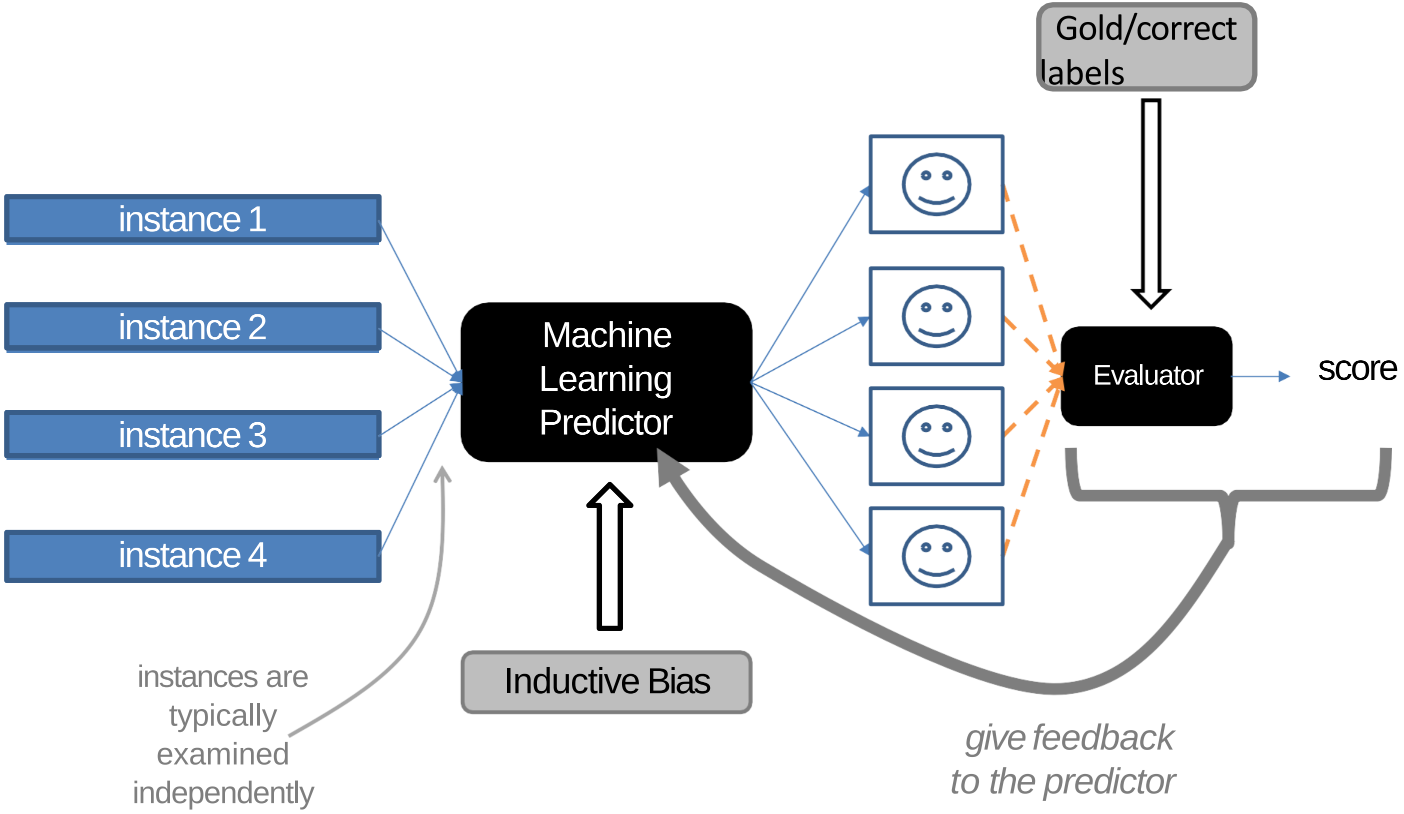
# Machine Learning Framework: Learning



# Machine Learning Framework: Learning



# Machine Learning Framework: Learning



# Classify with Goodness

$$\text{predicted label} = \underset{\text{label}}{\arg\max}\ \text{score}(\text{example}, \text{label})$$

# ML Framework Example

## Puppy classifier

*Training data*

TRAINING

Classifier (trained **model**)

# ML Framework Example

## Puppy classifier

*Training data*

Classifier (trained **model**)

TEST

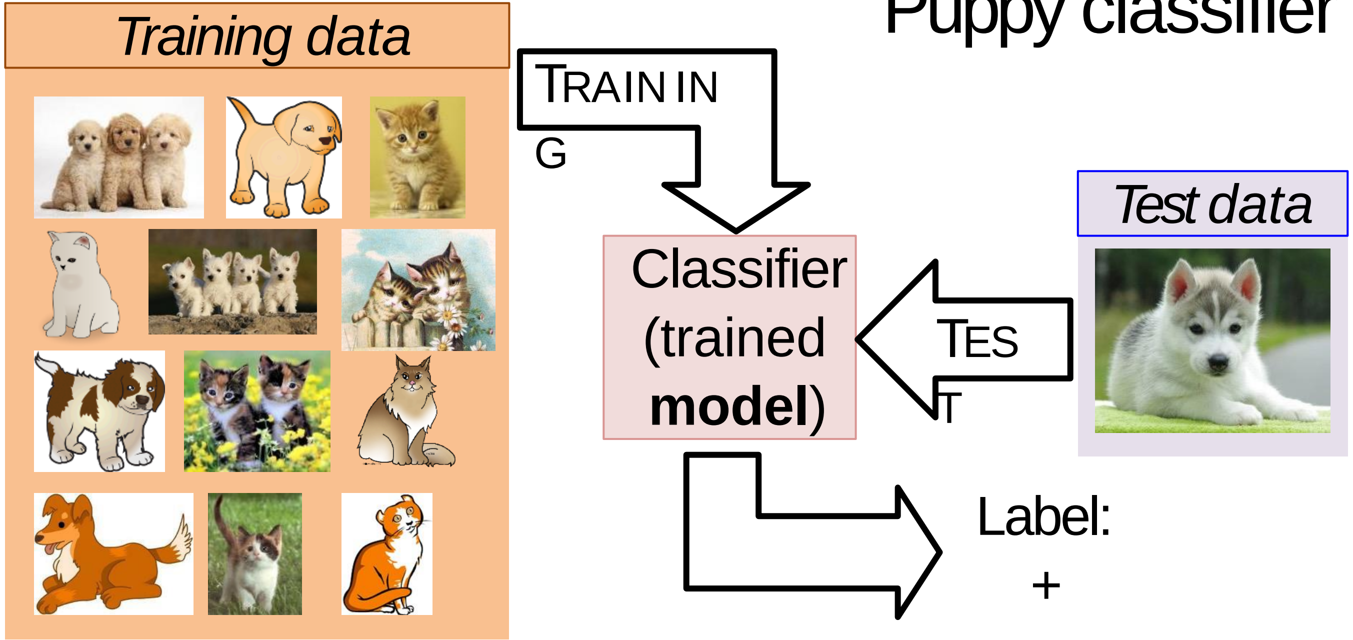
*Test data*

Label:

+

# ML Framework Example

## Puppy classifier



# ML Framework Example

## Puppy classifier

*Training data, X*

| *Text-ure* | *Ears* | *Legs* | *Class* |
|---|---|---|---|
| Fuzzy | Round | 4 | + |
| Slimy | Missing | 8 | - |
| Fuzzy | Pointy | 4 | - |
| Fuzzy | Round | 4 | + |
| Fuzzy | Pointy | 4 | + |
| … | | | |

TRAINING

Classifier (trained **model)**

TEST

*Test data*

$x_1$ = <Fuzzy, Pointy, 4>

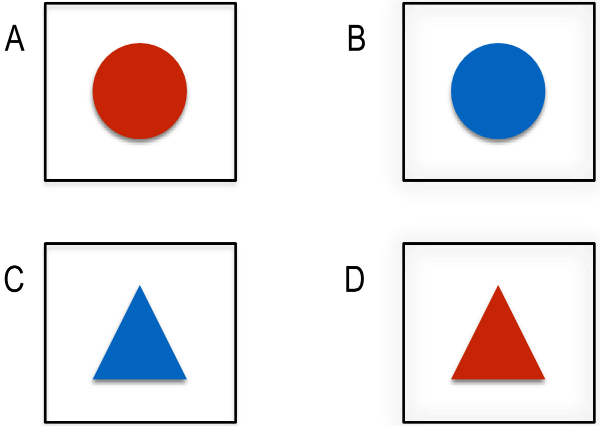
Label: +

# General ML Consideration: Inductive Bias

What do we know *before* we see the data, and how does that influence our modeling decisions?

# General ML Consideration:
# Inductive Bias

What do we know *before* we see the data, and how does that influence our modeling decisions?

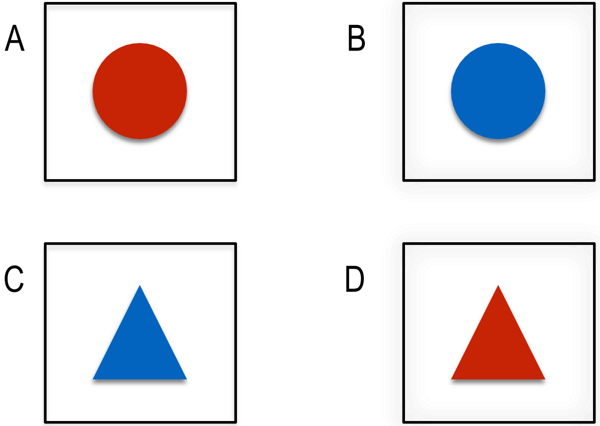



*Partition these into two groups…*

Courtesy Hamed Pirsiavash

# General ML Consideration: Inductive Bias

What do we know *before* we see the data, and how does that influence our modeling decisions?



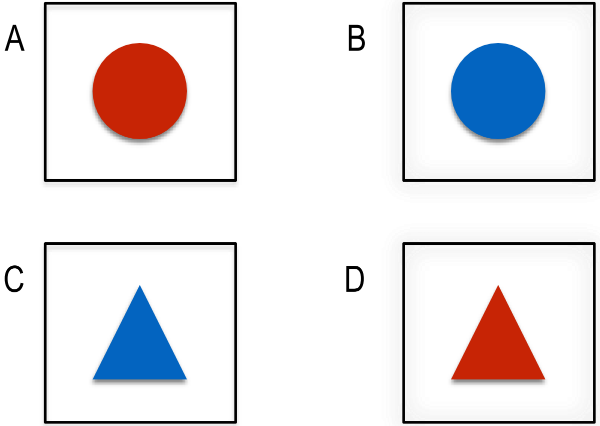

*Partition these into two groups*

*Who selected red vs. blue?*

Courtesy Hamed Pirsiavash

# General ML Consideration: Inductive Bias

What do we know *before* we see the data, and how does that influence our modeling decisions?




*Partition these into two groups*

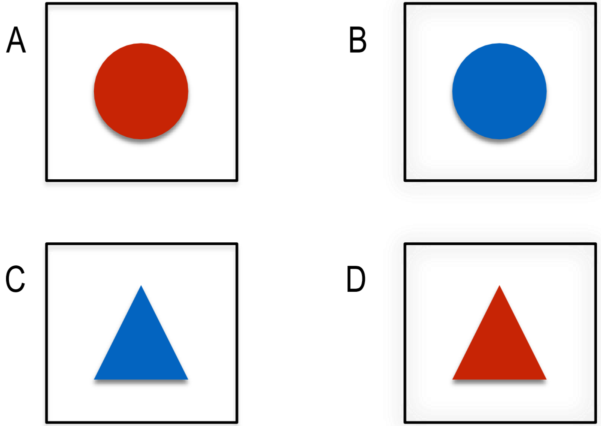
*Who selected red vs. blue?*

*Who selected ● vs. ▲ ?*

Courtesy Hamed Pirsiavash

# General ML Consideration: Inductive Bias

## What do we know *before* we see the data, and how does that influence our modeling decisions?




*Partition these into two groups*

*Who selected red vs. blue?*

*Who selected ● vs. ▲ ?*

Tip: Remember how your own biases/interpretation are influencing your approach

Courtesy Hamed Pirsiavash

# AI & ML

# AI and Learning Today

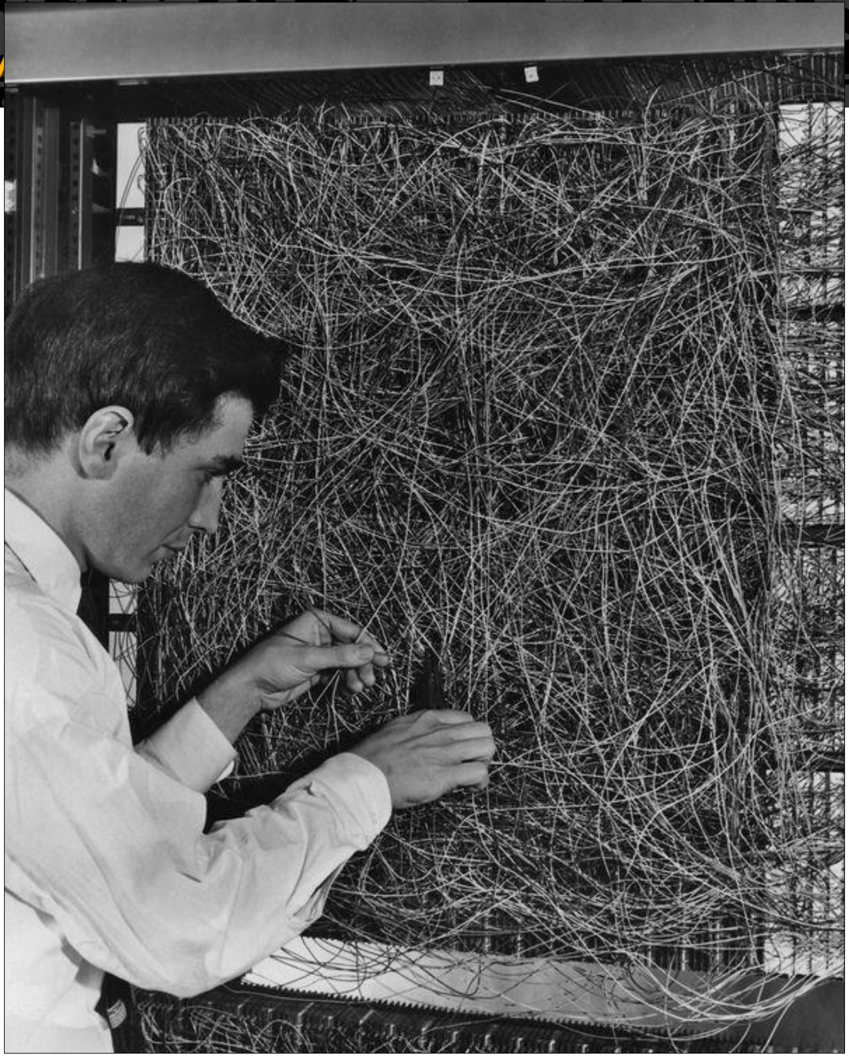
- 50s&60s: neural network learning popular

  Marvin Minsky did neural networks for his dissertation
- Mid 60s: replaced by paradigm of manually encoding & using symbolic knowledge

  Cf. Perceptrons, Minsky & Papert book showed limitations of perceptron model of neural networks
- 90s: more data & Web drove interest in statistical machine learning techniques & data mining
- Now: machine learning techniques & big data play biggest driver in almost all successful AI systems

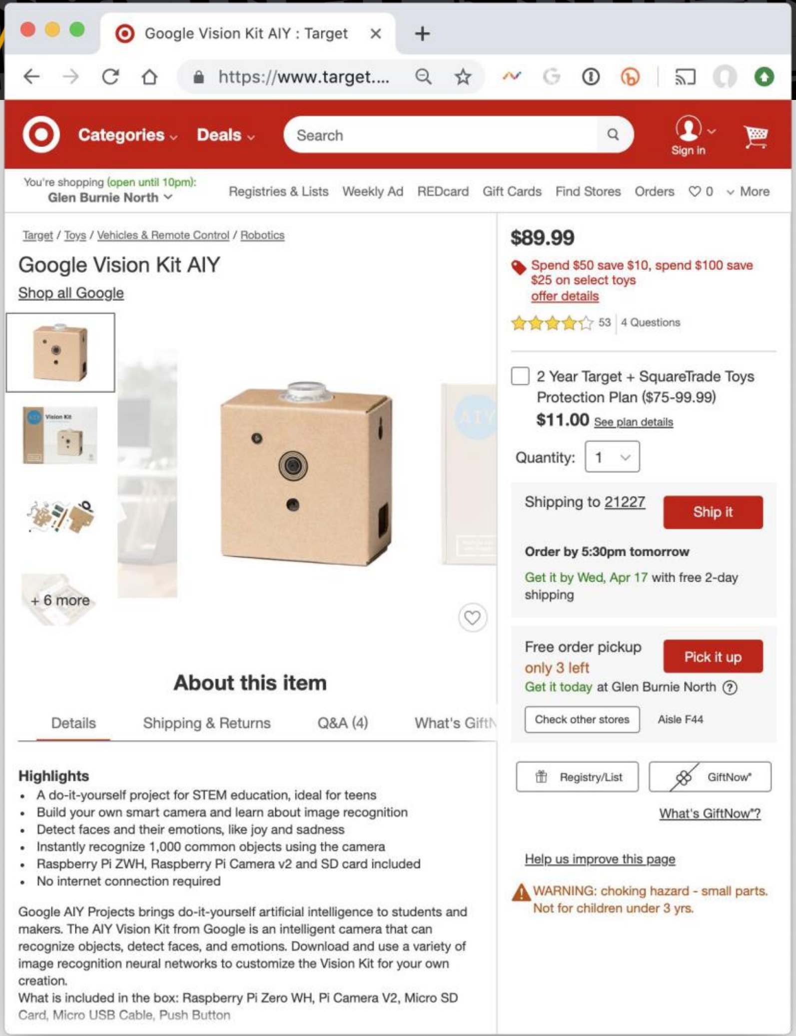
  …and neural networks are the current favorite approach

seeAlso: Timeline of machine learning



# Neural Networks 1960

A man adjusting the random wiring network between the light sensors and association unit of scientist Frank Rosen- blatt's Perceptron, or MARK 1 computer, at the Cornell Aeronautical Laboratory, Buffalo, New York, circa 1960. The machine is designed to use a type of artificial neural network, known as a perceptron.




# Neural Networks 2020

Google's AIY Vision Kit ($89.99 at Target) is an intelligent camera that can recognize objects, detect faces and emotions. Download and use a variety of image recognition neural networks to customize the Vision Kit for your own creation. Included in the box: Raspberry Pi Zero WH, Pi Camera V2, Micro SD Card, Micro USB Cable, Push Button.

**Currently $58.85 on [Amazon]**

# Machine Learning Successes

- Games: chess, go, poker
- Text sentiment analysis
- Email spam detection
- Recommender systems (e.g., Netflix, Amazon)
- Machine translation
- Speech understanding
- SIRI, Alexa, Google
- Assistant, …
- Autonomous vehicles
- Individual face recognition
- Understanding digital images
- Credit card fraud detection
- Showing annoying ads

# The Big Idea and Terminology

Given some data, learn a model of how the world works that lets you predict new data

- **Training Set:** Data from which you learn initially
- **Model:** What you learn; a “model” of how inputs are associated with outputs
- **Test set:** New data you test your model against
- **Corpus:** A body of text data (pl.: corpora)
- **Representation:** The computational expression of data

# Major Machine learning paradigms (1)

- **Rote**: 1-1 mapping from inputs to stored representation, learning by memorization, association-based storage & retrieval
- **Induction:** Use specific examples to reach general conclusions
- **Clustering**: Unsupervised discovery of natural groups in data

# Major Machine learning paradigms (2)

- **Analogy:** Find correspondence between different representations
- **Discovery**: Unsupervised, specific goal not given
- **Genetic algorithms:** *Evolutionary* search techniques, based on *survival of the fittest*
- **Reinforcement:** Feedback (positive or negative reward) given at the end of a sequence of steps
- **Deep learning:** *artificial neural networks* with *representation learning* for ML tasks

# CORE TERMINOLOGY

# Three Axes for Thinking About Your ML Problem

| the task | the data | the approach |
|---|---|---|
| Classification | Fully-supervised | Probabilistic |
| Regression | Semi-supervised | Generative |
| Clustering | Un-supervised | Conditional |
| | | Spectral |
| | | Neural |
| | | Memory-based |
| | | Exemplar |
| | | … |

*the **task**: what kind of problem are you solving?*

*the **data**: amount of human input/number of labeled examples*

*the **approach**: how any data are being used*

# Types of learning problems

- **Supervised**: learn from training examples
  - Regression:
  - Classification: Decision Trees, SVM
- **Unsupervised**: learn w/o training examples
  - Clustering
  - Dimensionality reduction
  - Word embeddings
- **Reinforcement learning:** improve performance using feedback from actions taken
- Lots more we won't cover
  - Hidden Markov models, Learning to rank, Semi-supervised learning, Active learning …

# Machine Learning Problems

| | *Supervised Learning* | *Unsupervised Learning* |
| --- | --- | --- |
| *Discrete* | classification or categorization | clustering |
| *Continuous* | regression | dimensionality reduction |

# Supervised learning

- Given training examples of inputs & corres- ponding outputs, produce “correct” outputs for new inputs
- Two important scenarios:
  - **Classification:** outputs typically labels (goodRisk, badRisk); learn decision boundary to separate classes
  - **Regression:** aka *curve fitting* or *function approxima- tion*; Learn a *continuous* input-output mapping from examples, e.g., for a zip code, predict house sale price given its square footage

# Unsupervised Learning

Given only *unlabeled* data as input, learn some sort of structure, e.g.:

- **Clustering**: group Facebook friends based on similarity of post texts and friends
- **Embeddings**: Find sets of words whose meanings are related (e.g., doctor, hospital)
- **Topic modelling**: Induce N topics and words most common in documents about each

# Inductive Learning Framework

- Raw input data from sensors or a database preprocessed to obtain **feature vector**, **X**, of **relevant** features for classifying examples
- Each **X** is a list of (attribute, value) pairs
- ***n*** attributes (a.k.a. features): fixed, positive, and finite
- Features have fixed, finite number # of possible values
  - Or continuous within some well-defined space, e.g., "age"
- Each example is a point in an *n*-dimensional feature space
  - X = [Person:Sue, EyeColor:Brown, Age:Young, Sex:Female]
  - X = [Cheese:*f*, Sauce:*t*, Bread:*t*]
  - X = [Texture:Fuzzy, Ears:Pointy, Purrs:Yes, Legs:4]

# ML Framework Example

## Puppy classifier

*Training data, X*

| *Text-ure* | *Ears* | *Legs* | *Class* |
|---|---|---|---|
| Fuzzy | Round | 4 | + |
| Slimy | Missing | 8 | - |
| Fuzzy | Pointy | 4 | - |
| Fuzzy | Round | 4 | + |
| Fuzzy | Pointy | 4 | + |
| … | | | |

TRAINING

Classifier (trained **model)**

TEST

*Test data*

$x_1$ = <Fuzzy, Pointy, 4>

Label: +

# Classification Examples

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

Sentiment analysis

…

# Classification Examples

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

Sentiment analysis

…

*Input*:

an instance

a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

*Output*: a predicted class $c$ from $C$

# Classification: Hand-coded Rules?

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

Sentiment analysis

…

Rules based on combinations of words or other features

spam: black-list-address OR (“dollars” AND “have been selected”)

Accuracy can be high

If rules carefully refined by expert

Building and maintaining these rules is expensive

Can humans faithfully assign uncertainty?

# Classification: Supervised Machine Learning

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

Sentiment analysis

…

*Input:*

an instance $d$

a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

A training set of $m$ hand-labeled instances $(d_1,c_1),\ldots,(d_m,c_m)$

*Output:*

a learned classifier $\gamma$ that maps instances to classes

# Classification: Supervised Machine Learning

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

Sentiment analysis

…

*Input:*

an instance $d$

a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

A training set of $m$ hand-labeled instances $(d_1,c_1),\ldots,(d_m,c_m)$

*Output:*

a learned classifier $\gamma$ that maps instances to classes

$\gamma$ learns to associate certain ***features*** of instances with their labels

# Classification: Supervised Machine Learning

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

Sentiment analysis

…

*Input:*

an instance $d$

a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

A training set of $m$ hand-labeled instances $(d_1,c_1),\ldots,(d_m,c_m)$

*Output:*

a learned classifier $\gamma$ that maps instances to classes

Naïve Bayes

Logistic regression

Support-vector machines

k-Nearest Neighbors

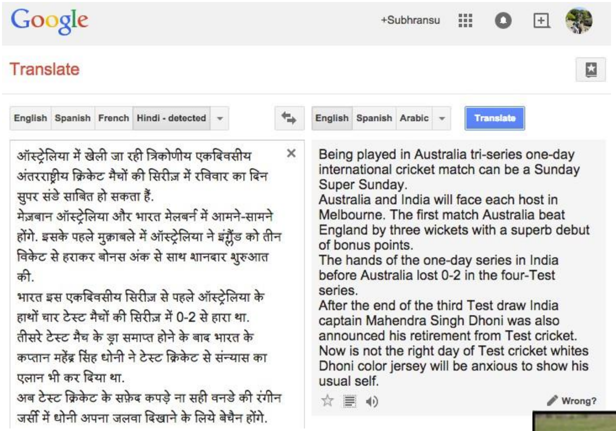
# Classification Example: Face Recognition

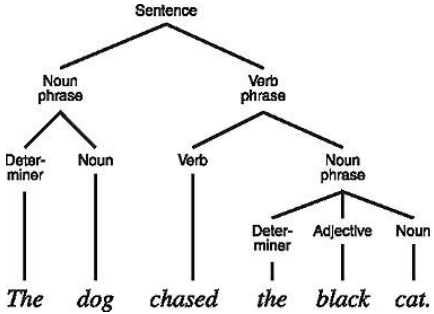

What is a good *representation* images?

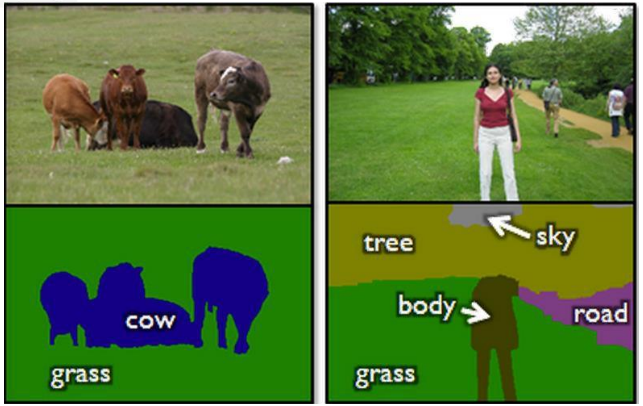
Pixel values?  Edges?

Courtesy from Hamed Pirsiavash

# Classification Example: Sequence & Structured Prediction




Courtesy Hamed Pirsiavash

# Ingredients for classification

Inject *your* knowledge into a learning system

*Feature representation*

*Training data: labeled examples*

*Model*

Courtesy Hamed Pirsiavash

# Ingredients for classification

Inject *your* knowledge into a learning system

Problem specific

Difficult to learn from bad ones

*Feature representation*

*Training data: labeled examples*

*Model*

Courtesy Hamed Pirsiavash

# Ingredients for classification

Inject *your* knowledge into a learning system

| Problem specific<br><br>Difficult to learn from bad ones | Labeling data == $$$<br><br>Sometimes data is available for "free" | |
|---|---|---|
| *Feature representation* | *Training data: labeled examples* | *Model* |

Courtesy Hamed Pirsiavash

# Ingredients for classification

Inject *your* knowledge into a learning system

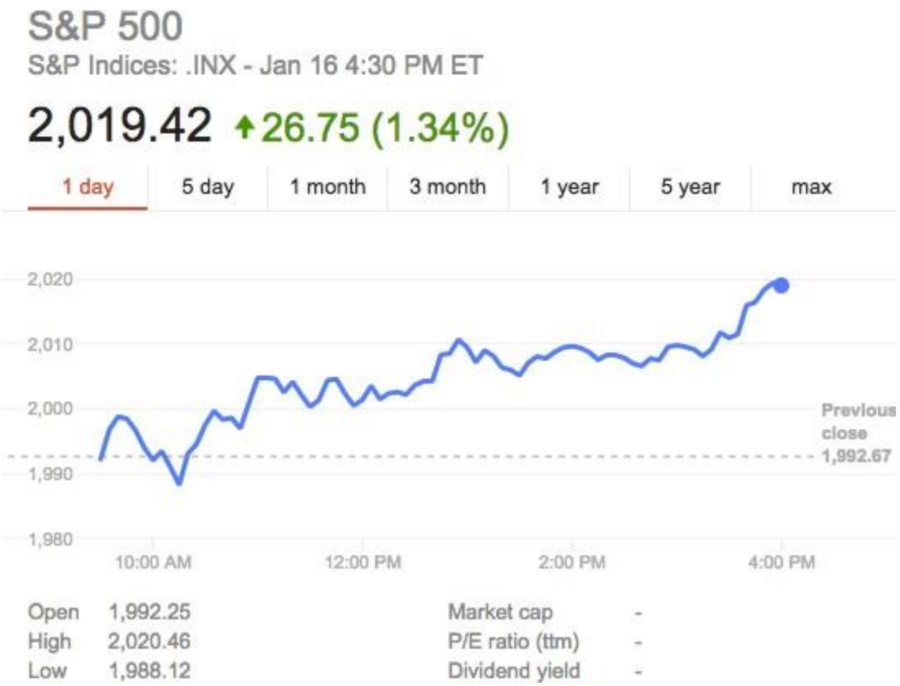
| Problem specific<br><br>Difficult to learn from bad ones | Labeling data == $$$<br><br>Sometimes data is available for "free" | No single learning algorithm is always good ("no free lunch")<br><br>Different learning algorithms work differently |
|---|---|---|
| *Feature representation* | *Training data: labeled examples* | *Model* |

Courtesy Hamed Pirsiavash

# Regression

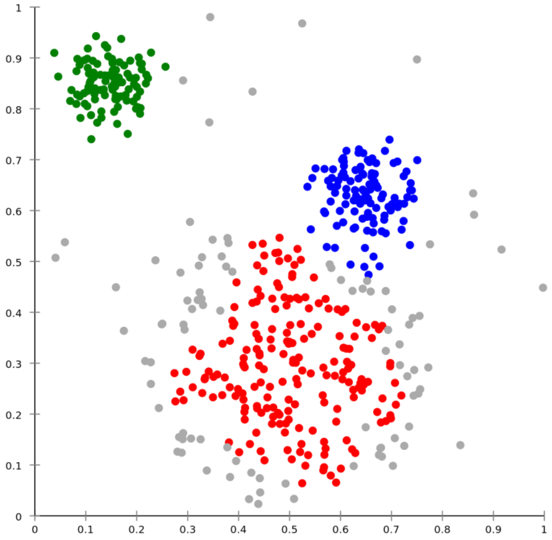
Like classification, but real-valued

# Regression Example: Stock Market Prediction



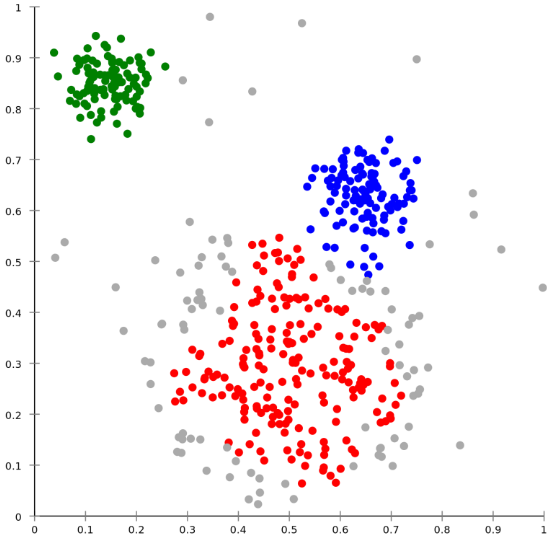

Courtesy Hamed Pirsiavash

# Unsupervised learning: Clustering




Courtesy Hamed Pirsiavash

# Unsupervised learning: Clustering




Courtesy Hamed Pirsiavash