Lecture 6: Introduction to ML in Practice

CMSC 25910

Winter 2026

The University of Chicago

THE UNIVERSITY OF CHICAGO

Goals and Intuition

Relationship Between Task & Methods

- Task: explain/describe data
  - Descriptive statistics (e.g., what percentage of students in the class are late based on today's attendance form?)
- Task: use observed data to infer information about a population
  - Inferential statistics (e.g., what fraction of the vote will this candidate receive based on this poll?)
- Task: draw a causal connection
  - Experiments (including on human subjects)
- Task: **predict** characteristics of **out-of-sample data**
  - Machine learning (prediction, forecasting, classification, etc.)

High-Level Intuition

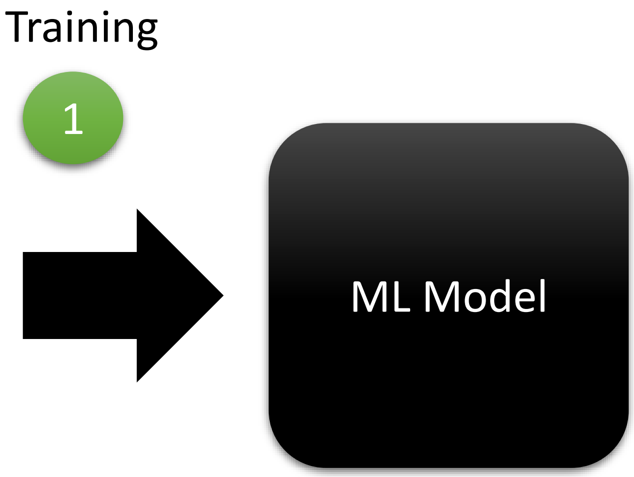
High-Level Intuition

Fox

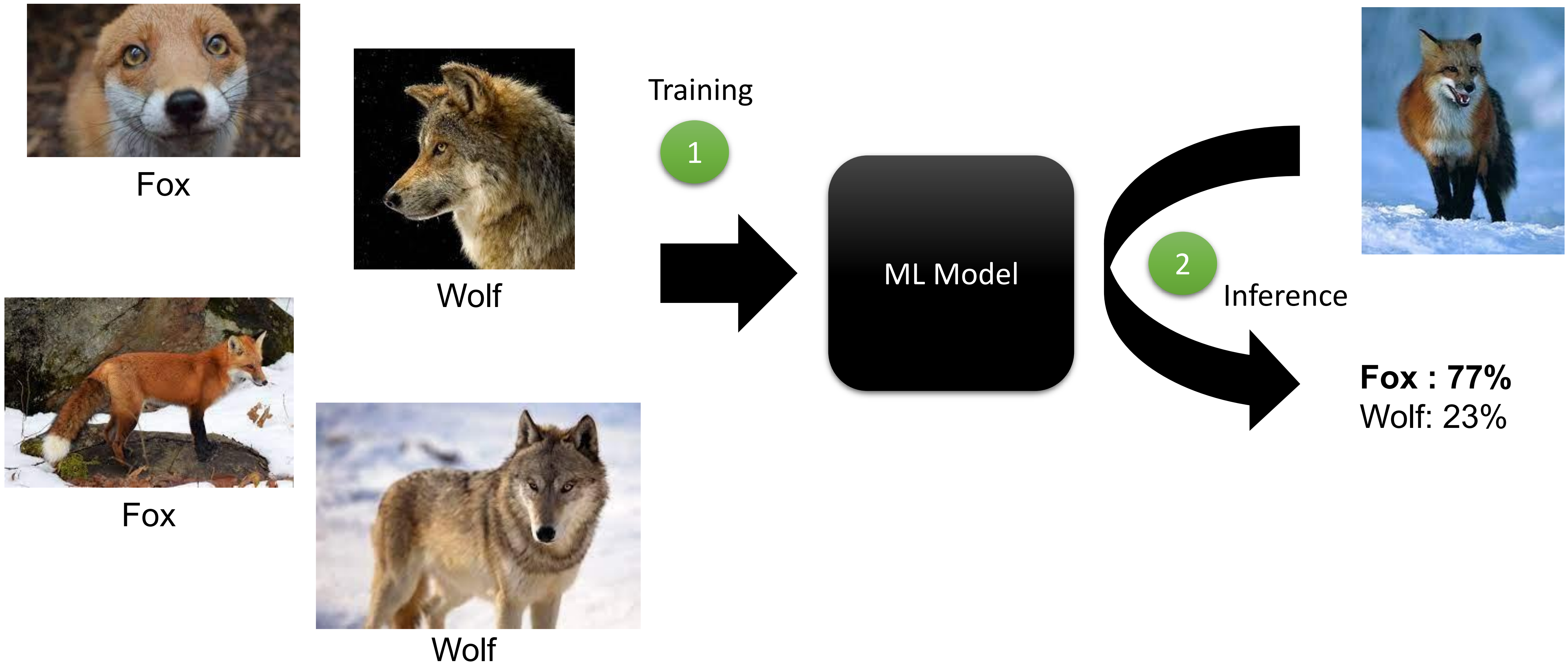
Wolf

Fox

Wolf



High-Level Intuition



Why We Build Models

- To understand data
- To make predictions about *out-of-sample* data

- We will focus on **supervised learning**, which is when the model is trained with a labeled dataset (e.g., “fox” and “wolf” in the previous slide are the labels, which you can informally think of as the “answers”)
- We will consider both **classification** (when the label is a category) and **regression** (when the label is a number)

Regression Example

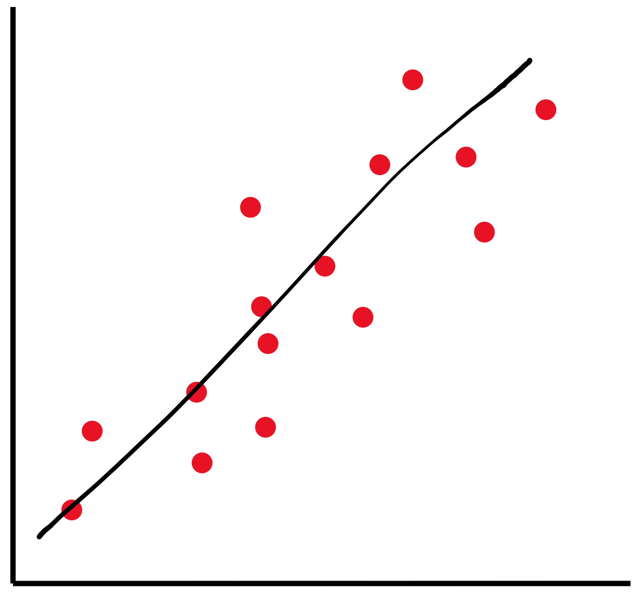
Let's Build a Model To Understand Data

- Running example: a regression problem
- Example:

Name	Age	Department	Gender	Title	Student Rating
Jack	55	CS	M	Professor	??
Jane	27	Stats	F	Assistant Professor	??

Given these input vectors… …predict this variable

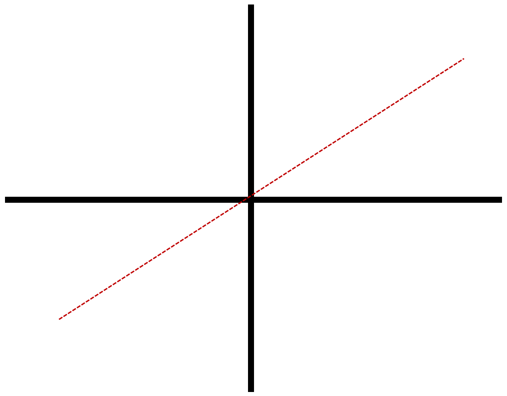
Building Intuition: Fitting a Line



Given Input Vector x, Predict y

- We need to choose a model to do that

$$\hat{y} = 0.3x$$



Parameters / weights

Output value / Explanatory

Input vector / predictor

$$\hat{y} = w^T x$$

$$x \in \mathbb{R}^n$$

$$y \in \mathbb{R}$$

$$w \in \mathbb{R}^n$$

Let’s Build a Model To Understand Data

- Running example: a regression problem
- Example:

Name	Age	Department	Gender	Title	Student Rating
Jack	55	CS	M	Professor	??
Jane	27	Stats	F	Assistant Professor	??

x_1, x_2, x_3, x_4, x_5 $\hat{y}$

Variables/Attributes/Columns become ‘features’ of the input vector

Linear Regression Model

- 'Linear' because of the relationship between x and y

$$\hat{y} = w^T x + b$$

Linear Regression Model

- ‘Linear’ because of the relationship between x and y

- A model is an assumption…
  - …of what function represents data *well*

$$\hat{y} = w^T x + b$$

- Once we’ve fixed a model…
  - …we find the parameters/weights *w* that make the model perform well

Linear Regression Model

- 'Linear' because of the relationship between x and y

- A model is an assumption…
  - …of what function represents data *well*

$$\hat{y} = w^T x + b$$

- Once we've fixed a model…
  - …we find the parameters/weights *w* that make the model perform well

We need a method to find those parameters

This suggests we need a performance metric

Our Data

- A dataset becomes a matrix
  - Each row is an input vector

Name	Age	Department	Gender	Title	Student Rating
Jack	55	CS	M	Professor	8.2
Jill	23	Econ	F	Professor	10.0
Josh	32	Bio	M	Staff	4.3
Jenn	44	Bio	F	Associate Professor	7.6
Jane	27	Stats	F	Assistant Professor	8.2

Our Data: Preparing It For ML

Probably drop this column (or perhaps calculate some other inferred numerical feature from it)

Name	Age	Department	Gender	Title	Student Rating
Jack	55	CS	M	Professor	8.2
Jill	23	Econ	F	Professor	10.0
Josh	32	Bio	M	Staff	4.3
Jenn	44	Bio	F	Associate Professor	7.6
Jane	27	Stats	F	Assistant Professor	8.2

Our Data: Preparing It For ML

Maybe keep as-is, but realize that the model may "make" certain assumptions (e.g., think of a linear model)

Name	Age	Department	Gender	Title	Student Rating
Jack	55	CS	M	Professor	8.2
Jill	23	Econ	F	Professor	10.0
Josh	32	Bio	M	Staff	4.3
Jenn	44	Bio	F	Associate Professor	7.6
Jane	27	Stats	F	Assistant Professor	8.2

Our Data: Preparing It For ML

Probably *dummy code* by turning N categories into N-1 binary columns (e.g., is_CS, is_Econ)

Name	Age	Department	Gender	Title	Student Rating
Jack	55	CS	M	Professor	8.2
Jill	23	Econ	F	Professor	10.0
Josh	32	Bio	M	Staff	4.3
Jenn	44	Bio	F	Associate Professor	7.6
Jane	27	Stats	F	Assistant Professor	8.2

Our Data: Preparing It For ML

(Why N-1 rather than N? Otherwise, you have multicollinearity, which can cause problems in some cases)

Name	Age	Department	Gender	Title	Student Rating
Jack	55	CS	M	Professor	8.2
Jill	23	Econ	F	Professor	10.0
Josh	32	Bio	M	Staff	4.3
Jenn	44	Bio	F	Associate Professor	7.6
Jane	27	Stats	F	Assistant Professor	8.2

Our Data: Preparing It For ML

Probably dummy code, but what is your *baseline* category? How do you handle small groups?

Name	Age	Department	Gender	Title	Student Rating
Jack	55	CS	M	Professor	8.2
Jill	23	Econ	F	Professor	10.0
Josh	32	Bio	M	Staff	4.3
Jenn	44	Bio	F	Associate Professor	7.6
Jane	27	Stats	F	Assistant Professor	8.2

Our Data: Preparing It For ML

(Wait a second! We need to think carefully about age and gender… are we using them to measure the current world or **make future predictions**?)

Name	Age	Department	Gender	Title	Student Rating
Jack	55	CS	M	Professor	8.2
Jill	23	Econ	F	Professor	10.0
Josh	32	Bio	M	Staff	4.3
Jenn	44	Bio	F	Associate Professor	7.6
Jane	27	Stats	F	Assistant Professor	8.2

Our Data: Preparing It For ML

(But even if we drop those columns, are their effects captured by other columns, which are often termed **proxy variables**)

Name	Age	Department	Gender	Title	Student Rating
Jack	55	CS	M	Professor	8.2
Jill	23	Econ	F	Professor	10.0
Josh	32	Bio	M	Staff	4.3
Jenn	44	Bio	F	Associate Professor	7.6
Jane	27	Stats	F	Assistant Professor	8.2

Train-Test Split

Dataset contains the target variable / label

- A dataset becomes a matrix
  - Each row is an input vector

	Name	Age	Department	Gender	Title	Student Rating
Training dataset	Jack	55	CS	M	Professor	8.2
	Jill	23	Econ	F	Professor	10.0
	Josh	32	Bio	M	Staff	4.3
Test dataset	Jenn	44	Bio	F	Associate Professor	7.6
	Jane	27	Stats	F	Assistant Professor	8.2

Performance Metric

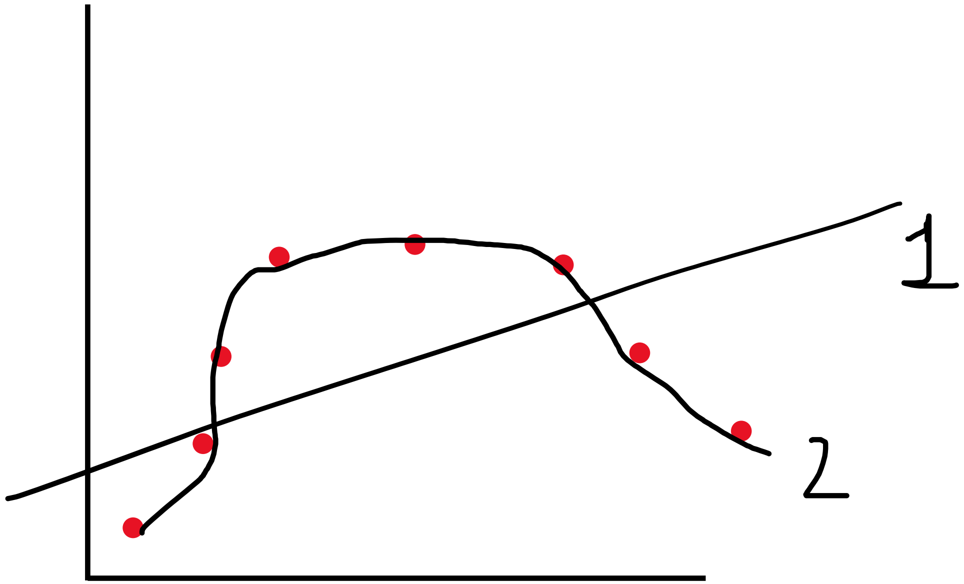
- Mean Squared Error (MSE)
  - Error decreases to 0 when *predicted y = ground-truth y*

$$\mathrm{MSE}_{\mathrm{test}} = \frac{1}{m} \sum_i (\hat{\boldsymbol{y}}^{(\mathrm{test})} - \boldsymbol{y}^{(\mathrm{test})})_i^2 .$$

m test examples

- Goal: We want the model to perform well on the test data, which has “never been seen before” (out-of-sample data)

Building Intuition…



Higher Capacity Models

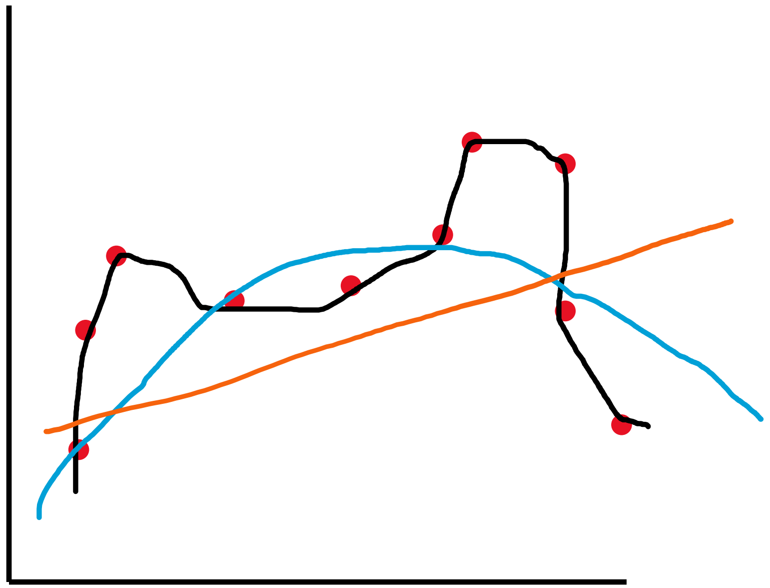
- We can increase the capacity of the model by adding more parameters; this will help with obtaining a “better” fit to the training data, but that is not always what we want

$$\hat{y} = w^T x$$

$$x \in \mathbb{R}^n$$

$$y \in \mathbb{R}$$

$$w \in \mathbb{R}^n$$



Optimization

- We want to find parameters *w* using the training dataset

$$\nabla_{\boldsymbol{w}} \mathrm{MSE}_{\mathrm{train}} = 0$$

- This is an optimization problem; we can find the minimum MSE
- Consider that we run this optimization with the training data. What will happen when we run it on the test data?

Some Key Challenges

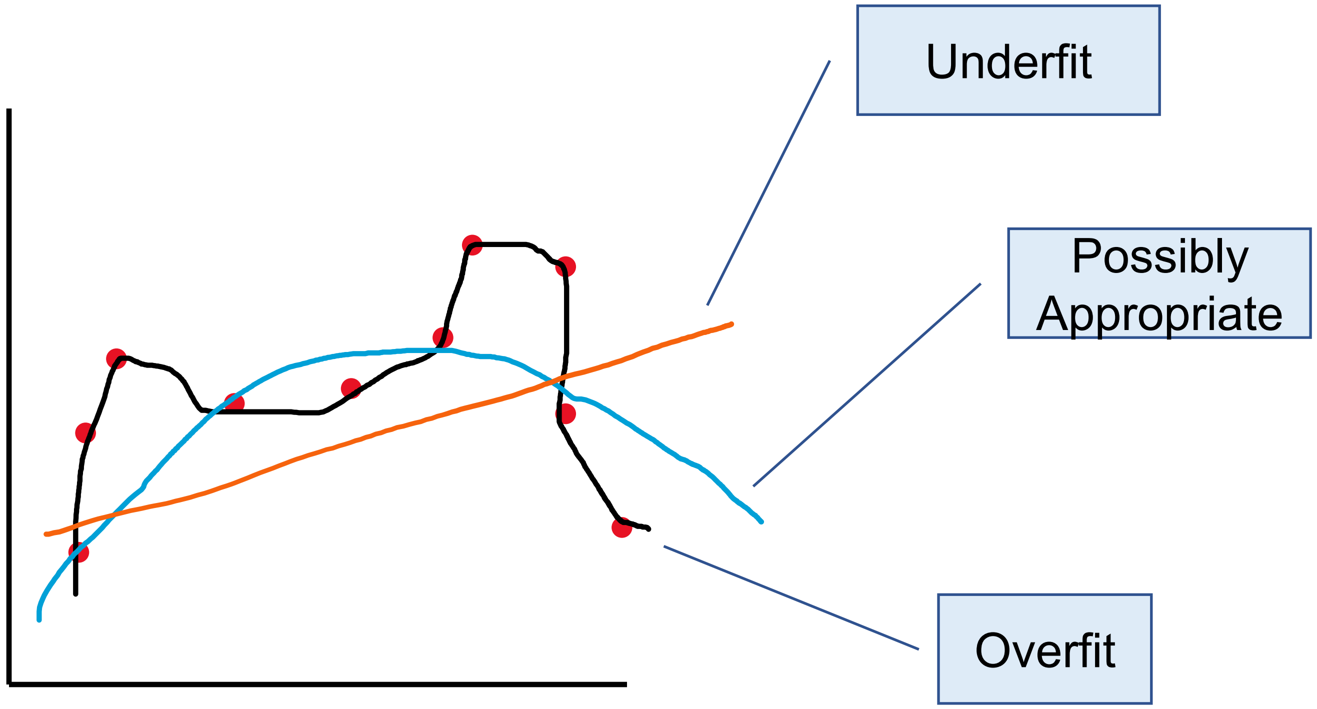
Challenges For Machine Learning

- Learn parameters so the model performs well on unseen data
  - Hope: *Generalize* to **unseen data**
  - Optimization process: Do well on the **training data**
- Remember why we build models:
  - To understand the process that generated the data (e.g., modeling)
  - To make predictions about out-of-sample data (e.g., automated decisions)

Underfitting, Overfitting

- Underfitting
  - Higher *training error* than necessary or desired
- Overfitting
  - A model achieves low *training error,* but high *test error*
- Ideally, we want low training error and small gap between training error and test error
  - That's a model that explains the data generation process
  - That's a model that helps us predict out-of-sample data

Underfitting, Overfitting…



So What Is Machine Learning?

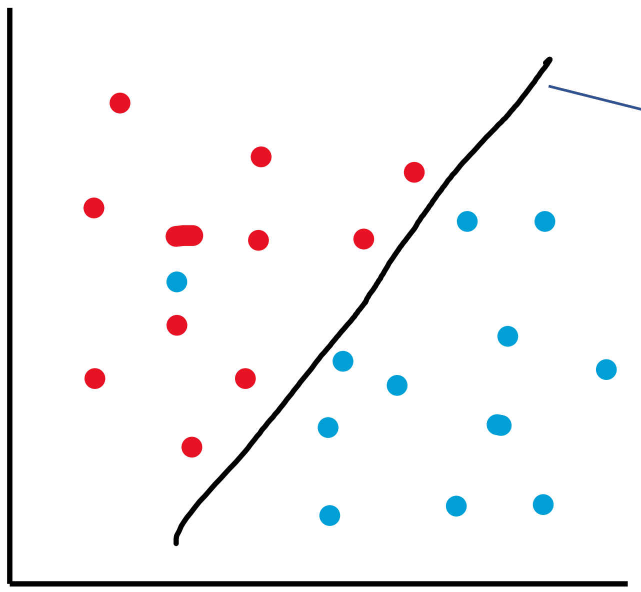
- A model
  - Linear regression, logistic regression, random forest, neural network,…
- Parameters
- A performance metric
  - For instance, MSE
- A training objective
  - Loss function
- A strategy to learn/fit the model parameters

One Common Task Formulation: Classification

[Some of the slides in this section were cannibalized from Elena Zheleva at UIC, and by the transitive property from the Berkeley DS 100 team, Marine Carpuat, Lise Getoor, Brian Ziebart. Please do not further distribute. Mistakes are my own.]

Classification Problem

- Given an input vector *x*, predict a class *c*
  - Binary classification problems
    - Spam vs. not spam
    - Give loan vs. don't
    - Admit student vs. don't
    - Will reoffend vs. won't

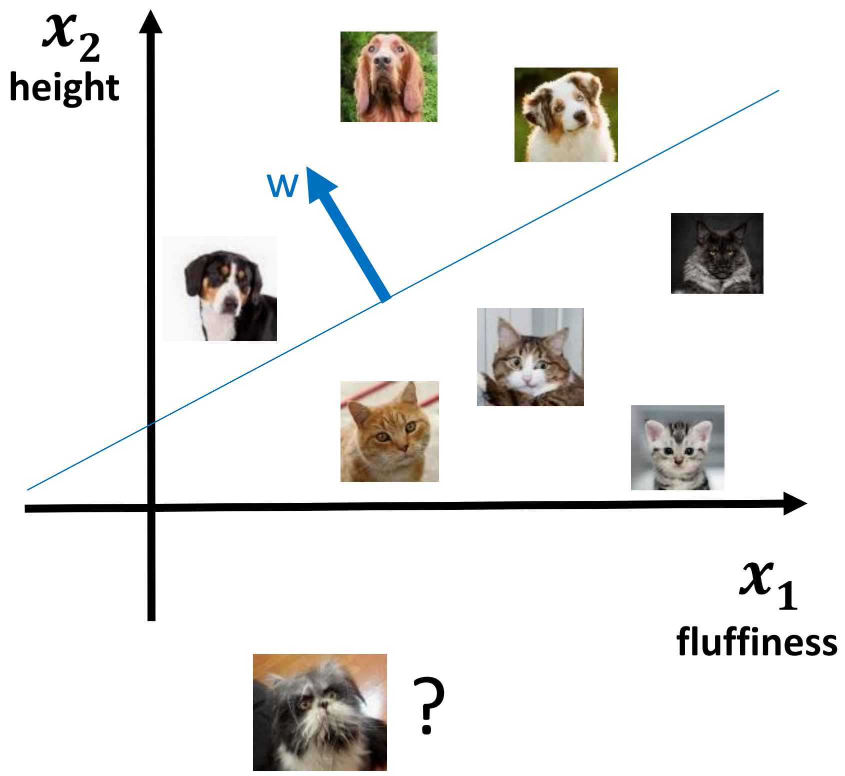


Find a hyperplane that separates the space of positive and negative samples

- How do you evaluate this?
  - Accuracy, false positives/negatives, …

Classification Problem

- Build a model that can predict the categorical value of an unseen object
- Problem setting
  - **X** – set of possible instances with features x_i
  - Y – target class
  - Unknown target function f: **X** $\rightarrow$Y
  - Set of function hypotheses H={h|h: **X** $\rightarrow$Y}
- Input
  - Training examples $\{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots (\boldsymbol{x}^{(N)}, y^{(N)})\}$ of unknown distribution
- Output
  - Hypothesis $h \in \mathrm{H}$ that best approximates target function f



Logistic Regression

- Widely used models for **binary classification:**

$x =$ "Get a FREE sample ..." $\Rightarrow$ $y = 1$

1 = "Spam"
0 = "Not spam"

$$\phi(x) = [2.0, 0.0, \ldots, 1.0, 0.5]$$

- Models P(y=1|x), the probability of *y=1* given *x*

$$\hat{\mathbf{P}}_\theta\left(y = 1 \,|\, x\right) = \sigma(\phi(x)^T\theta) = \frac{1}{1 + \exp\left(-\phi(x)^T\theta\right)}$$

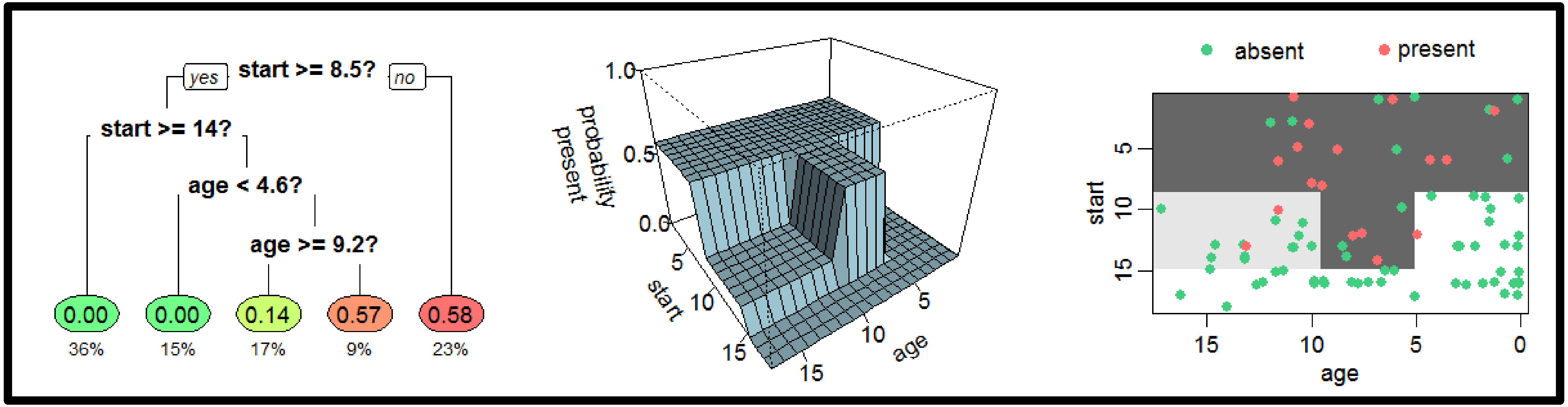
Model Architectures

[Some of the slides in this section were cannibalized from Elena Zheleva at UIC, and by the transitive property from the Berkeley DS 100 team, Marine Carpuat, Lise Getoor, Brian Ziebart. Please do not further distribute. Mistakes are my own.]

Some ML Model Architectures

- Regression models
- Decision trees
- Support Vector Machines (SVMs)
- Deep neural networks
- Many, many others

Example Decision Tree



Taken from https://en.wikipedia.org/wiki/Decision_tree_learning#/media/File:Cart_tree_kyphosis.png

Example Model Architectures

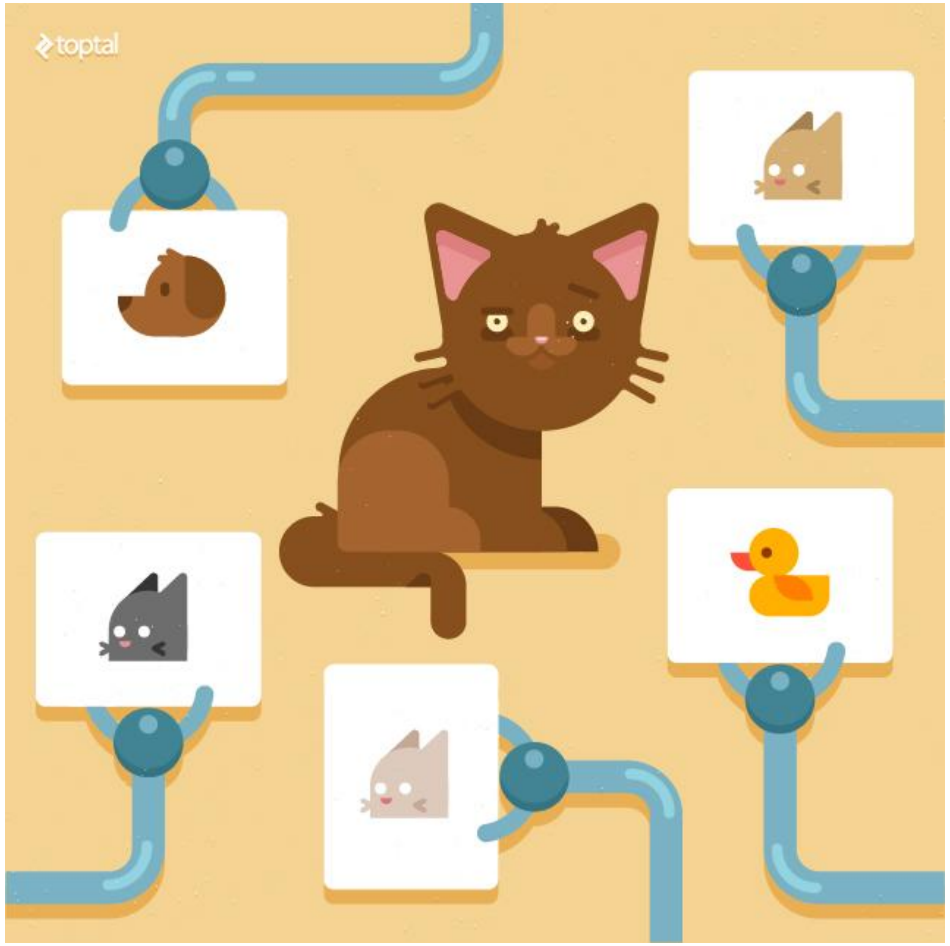
- **Support vector machine** (SVM)
    - Learning is **convex** (globally optimal weights)
- SVMs are good for medium-large data

Ensemble Methods

[Some of the slides in this section were cannibalized from Elena Zheleva at UIC, and by the transitive property from the Berkeley DS 100 team, Marine Carpuat, Lise Getoor, Brian Ziebart. Please do not further distribute. Mistakes are my own.]

Ensemble Methods

- Simplest approach:
  1. Generate **multiple classifiers**
  2. Each votes on test instance
  3. Take majority as classification
- Classifiers can be different due to
  - different sampling of training data
  - randomized parameters within the classification algorithm
  - inductive bias (e.g, decision tree + perceptron + kNN)



Random Forests

- Definition: Ensemble of decision trees
- Algorithm:
  - Divide training examples into multiple training sets (bagging)
  - Train a decision tree on each set
    - randomly select subset of variables to consider
  - Aggregate the predictions of each tree to make classification decision
    - e.g., can choose mode (most often) vote

XGBoost

- Developed by Chen and Guestrin (2016)
- Relies on gradient boosting

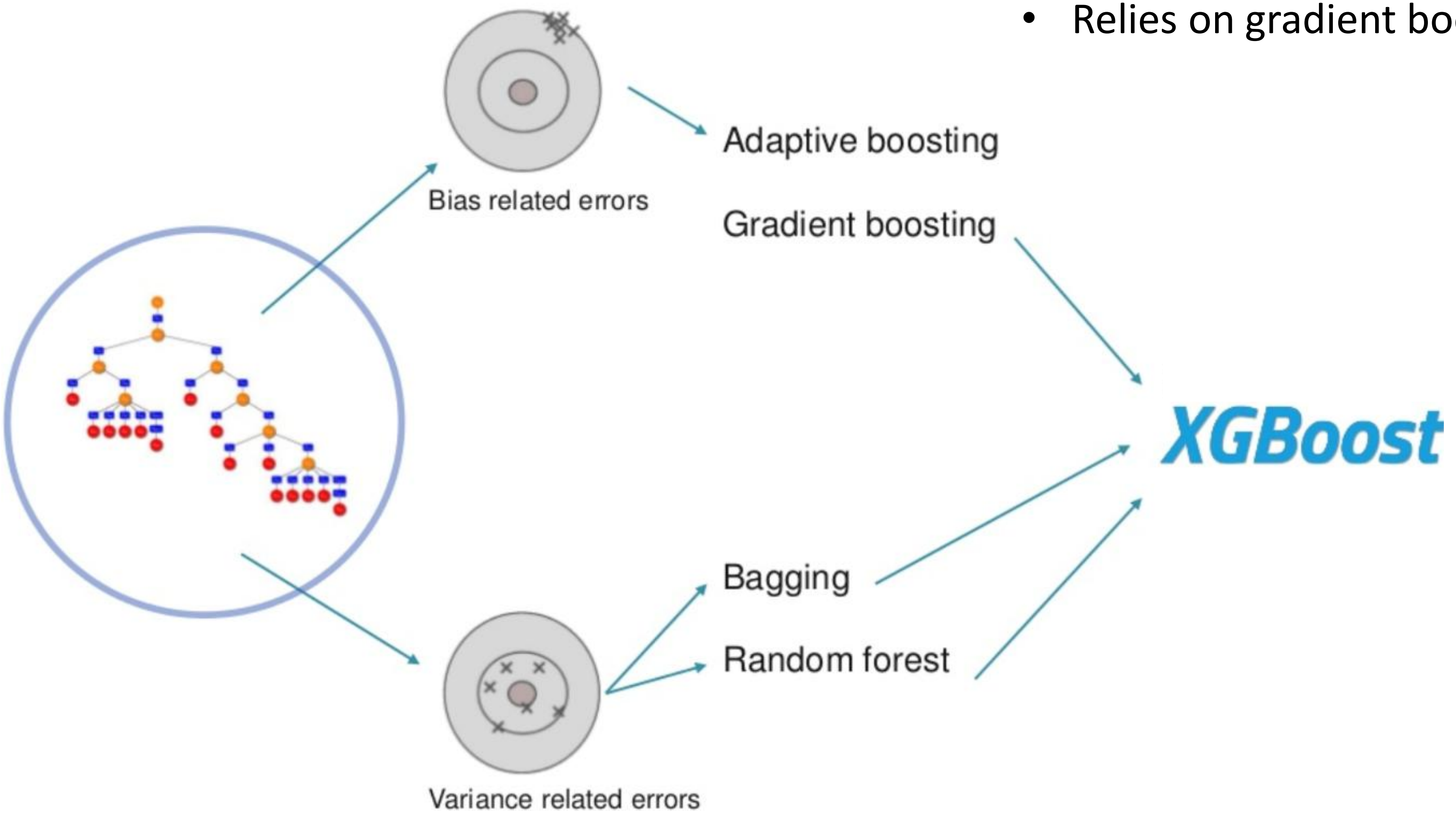


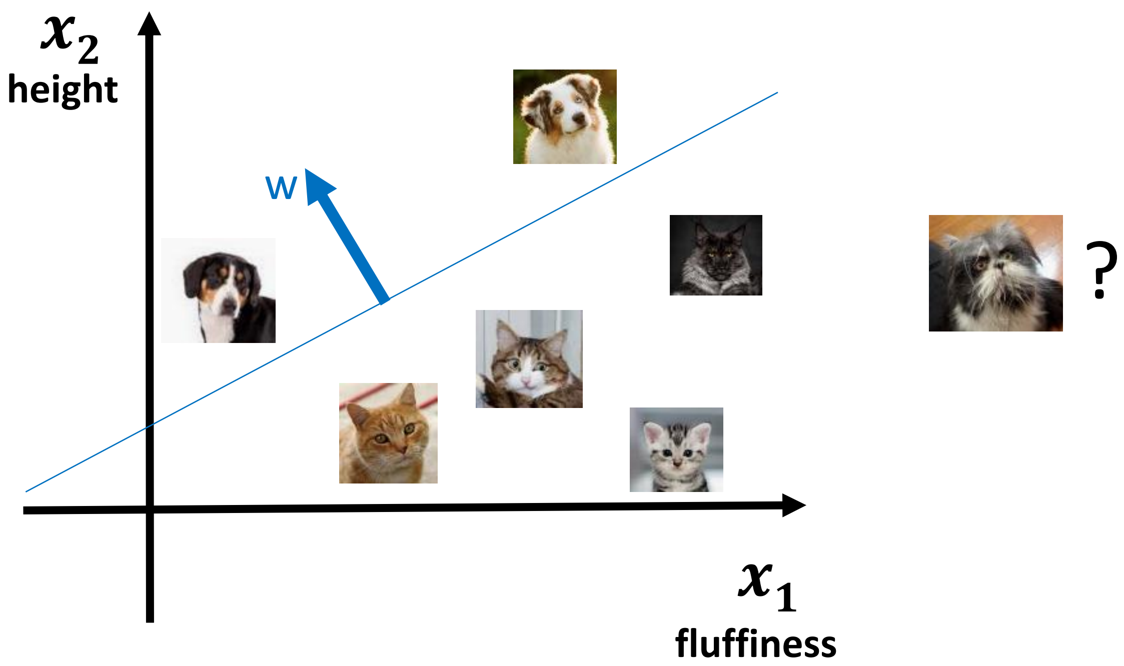
Figure: https://www.slideshare.net/JaroslawSzymczak1/xgboost-the-algorithm-that-wins-every-competition

Neural Networks

[Some of the slides in this section were cannibalized from Elena Zheleva at UIC, and by the transitive property from the Berkeley DS 100 team, Marine Carpuat, Lise Getoor, Brian Ziebart. Please do not further distribute. Mistakes are my own.]

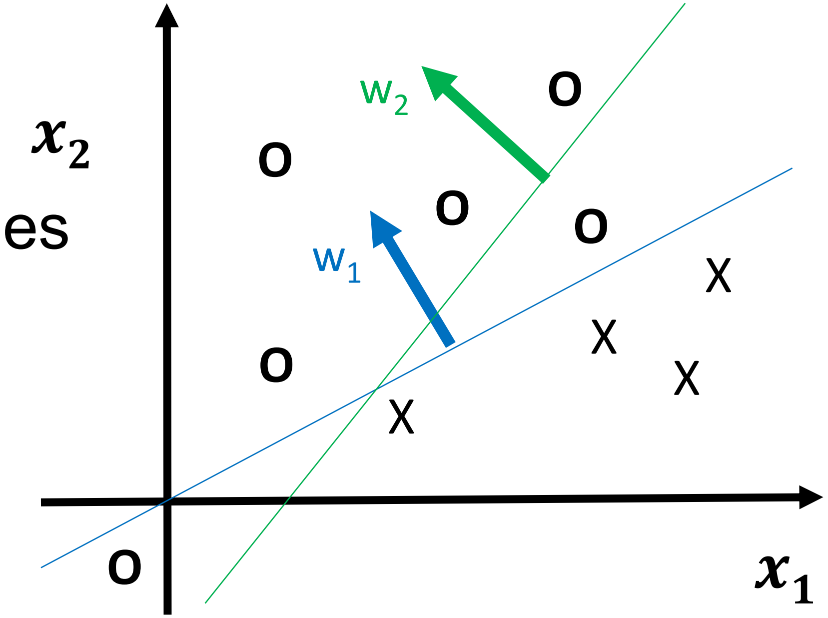
Predecessor: Perceptron (1958)

- Assume decision boundary is a hyperplane
- Training = find a hyperplane w that separates positive from negative examples
- See: https://en.wikipedia.org/wiki/Perceptron



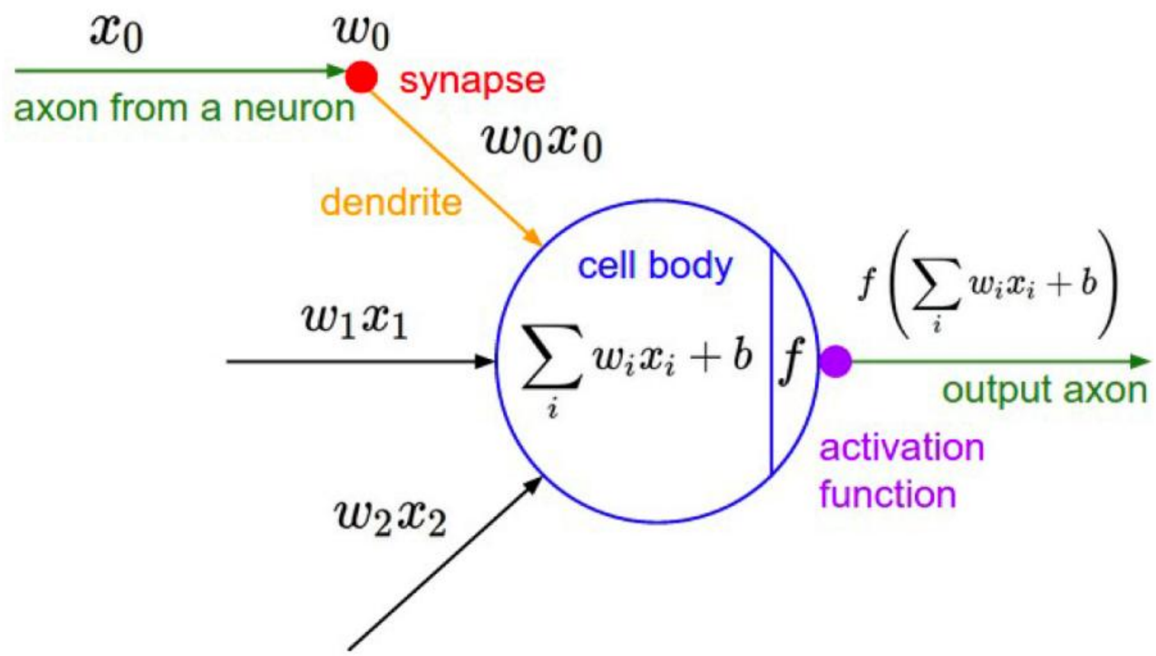
Neural Networks

- We can think of neural networks as combination of multiple linear models (perceptrons)
  - Multilayer perceptron
- Why would we want to do that?
  - Discover more complex decision boundaries
  - Learn combinations of features



Mathematical Model of a Neuron

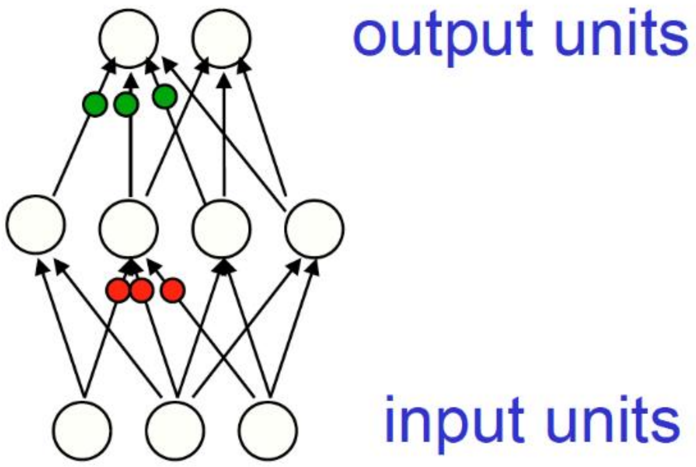
- We can think of neural networks as combination of multiple perceptrons
  - Hidden features define functions of the inputs, computed by neurons
  - Artificial neurons are called units
  - Vanilla perceptron: activation function is sign(z)

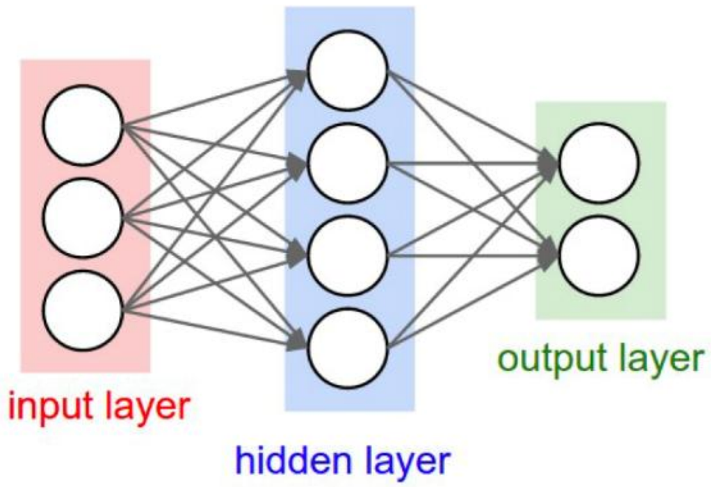


$$z = \sum_{d=1}^{D} w_d \, x_d + b$$

Neural Network Architecture

- Neural network with one layer of four hidden units:





  - Figure: Two different visualizations of a 2-layer neural network. In this example: 3 input units, 4 hidden units (layer 1) and 2 output units (layer 2)
- Each unit computes its value based on linear combination of values of units that point into it, and an activation function

Figure: http://cs231n.github.io/neural-networks-1/

Neural Network Architecture

- Going deeper: a 3-layer neural network with two layers of hidden units
- N-layer neural network:
    - N-1 layers of hidden units
    - One output layer

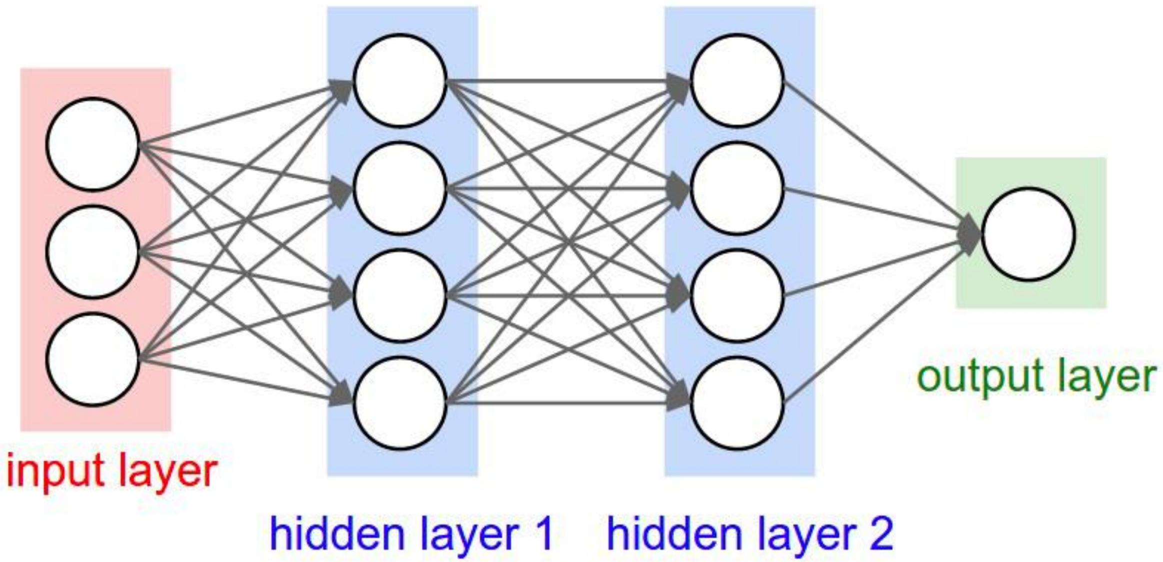
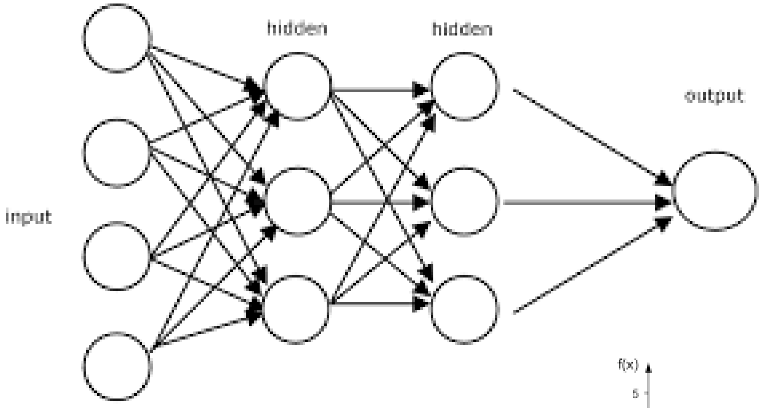


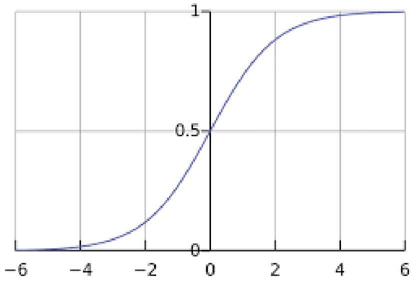
Figure : A 3-layer neural net with 3 input units, 4 hidden units in the first and second hidden layer and 1 output unit

Figure: http://cs231n.github.io/neural-networks-1/

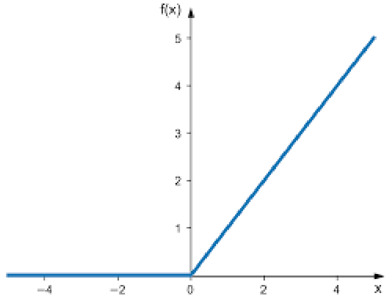
Neural Networks at 10,000 Feet

- Y = f(X)
  - F may be constructed by combining different functions
    - $\mathbf{h^1} = g^1 (W^1 \mathbf{x} + b^1)$
    - $h^2 = g^2 (W^2 \mathbf{h^1} + b^2)$
    - …
- Activation functions
  - Softmax
  - Relu
  - And many many more…
- Optimizers





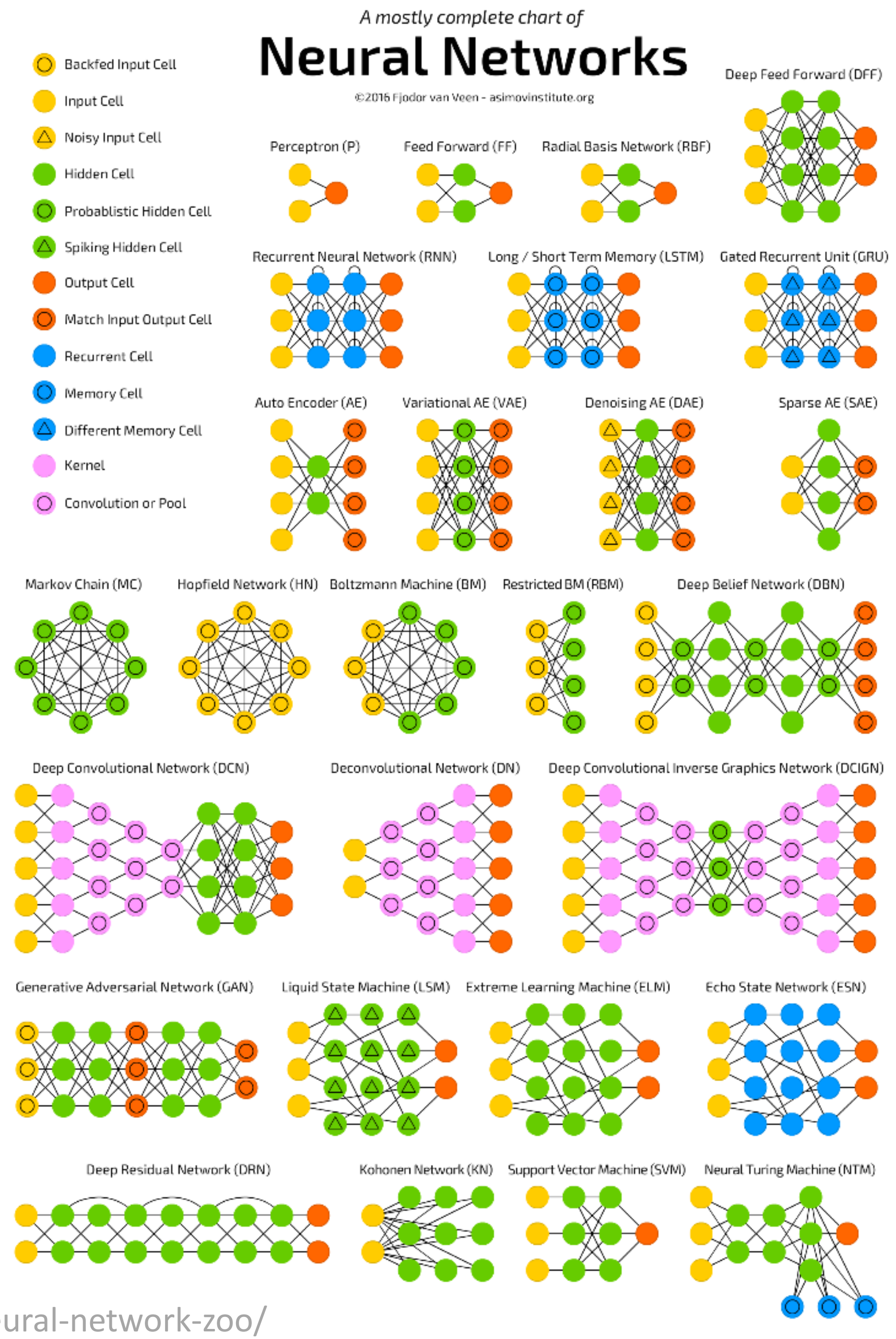
Softmax



Relu

Neural Networks: Backpropagation

- Goal: learn the weights of each layer
- Using backpropagation algorithm
  - Forward pass = prediction/inference
  - Backward pass = learning
    - Convert discrepancy between each output and its target value into an error derivative
    - Compute error derivatives in each hidden layer from error derivatives in layer above
- The optimization function is non-convex



Taken from http://www.asimovinstitute.org/neural-network-zoo/

A More Detailed Discussion of Feature Engineering

Feature Representation

- Transform categorical variables into a numerical representation
  - Dummy coding
- Normalization
- Standardization
- Binning
- Other transformations

- **All of these can have implications for ethics!**

Feature Engineering

- Preprocessing data
- What aspects of data matter?
  - What aspects **should** matter?

Pitfalls of Feature Engineering

- ML model performance depends on the input data
  - Is the training data representative of the population?
  - Are the transformations applied to the data correct?
  - Is there enough training data to learn a good model?
- Many potential pitfalls throughout the process
  - Even careful humans will make mistakes!
- AutoML and automatic augmentation techniques
  - Opportunity or threat?