CS 540 Introduction to Artificial Intelligence
Midterm Review

University of Wisconsin-Madison

Spring 2024

Announcements

- **Homeworks**:
  - HW6 deadline on **Thursday March. 14th at 11 AM**
  - While you can use study groups to discuss high level ideas, **you need to code independently.**

- Thank you for your feedback!

Midterm Information

- **Time:** March 13th 7:30-9 PM
- **Place:** Humanities 2340: A-K Humanities 3650: L-Z
- Format: multiple choice (20 questions)
- Cheat sheet: single piece of paper, front and back
- Calculator: fine if it doesn't have an Internet connection
- Detailed topic list + practice: https://piazza.com/class/lrjf9oinrox1zf/post/409

Reasoning With Conditional Distributions

- Evaluating probabilities:
  - Wake up with a sore throat.
  - Do I have the flu?
- Logic approach: $S \rightarrow F$
  - Too strong.
- **Inference**: compute probability given evidence $P(F|S)$
  - Can be much more complex!

Using **Bayes' Rule**

- Want: $P(F|S)$
- **Bayes' Rule:** $P(F|S) = \frac{P(F,S)}{P(S)} = \frac{P(S|F)P(F)}{P(S)}$
- Parts:
  - $P(S) = 0.1$ Sore throat rate
  - $P(F) = 0.01$ Flu rate
  - $P(S|F) = 0.9$ Sore throat rate among flu sufferers

So: $P(F|S) = 0.09$

Using Bayes' Rule

- Interpretation $P(F|S) = 0.09$
  - Much higher chance of flu than normal rate (0.01).
  - Very different from $P(S|F) = 0.9$
    - 90% of folks with flu have a sore throat
    - But, only 9% of folks with a sore throat have flu
- Idea: **update** probabilities from **evidence**

Bayesian **Inference**

- Fancy name for what we just did. Terminology:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

- *H* is the hypothesis
- *E* is the evidence

Bayesian **Inference**

- Terminology:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$ ← **Prior**

- Prior: estimate of the probability **without** evidence

Bayesian Inference

- Terminology:

Likelihood

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

- Likelihood: probability of evidence **given a hypothesis**

Bayesian Inference

- Terminology:

$$\color{magenta}{P(H|E)} = \frac{P(E|H)P(H)}{P(E)}$$

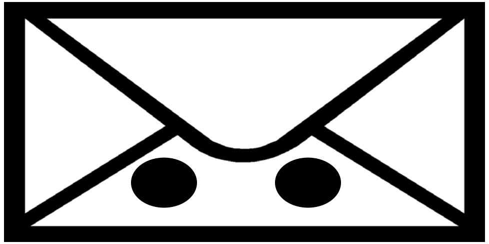
↑

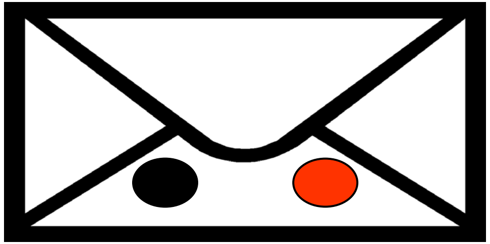
Posterior

- Posterior: probability of hypothesis **given evidence**.

Two Envelopes Problem

- We have two envelopes:
  - E_1 has two black balls, E_2 has one black, one red
  - The **red** one is worth \$100. Others, zero
  - Open an envelope, see one ball. Then, can switch (or not).
  - You see a black ball. **Switch?**





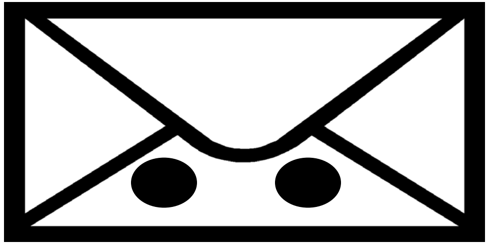
Two Envelopes Solution

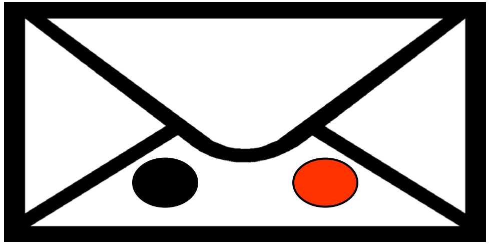
- Let's solve it. $$P(E_1|\text{Black ball}) = \frac{P(\text{Black ball}|E_1)P(E_1)}{P(\text{Black ball})}$$

- Now plug in: $$P(E_1|\text{Black ball}) = \frac{1 \times \frac{1}{2}}{P(\text{Black ball})}$$

$$P(E_2|\text{Black ball}) = \frac{\frac{1}{2} \times \frac{1}{2}}{P(\text{Black ball})}$$

So switch!





Naïve Bayes

- Conditional Probability & Bayes:

$$P(H|E_1, E_2, \ldots, E_n) = \frac{P(E_1, \ldots, E_n|H)P(H)}{P(E_1, E_2, \ldots, E_n)}$$

- If we further make the **conditional independence assumption (a.k.a. Naïve Bayes)**

$$P(H|E_1, E_2, \ldots, E_n) = \frac{P(E_1|H)P(E_2|H) \ \cdots \ P(E_n|H)P(H)}{P(E_1, E_2, \ldots, E_n)}$$

Naïve Bayes

- Expression

$$P(H|E_1, E_2, \ldots, E_n) = \frac{P(E_1|H)P(E_2|H)\cdots, P(E_n|H)P(H)}{P(E_1, E_2, \ldots, E_n)}$$

- *H*: some class we'd like to infer from evidence
  - We know prior *P*(*H*)
  - Estimate $P(E_i|H)$ from data! ("training")
  - Very similar to envelopes problem.

Break & Quiz

Q 3.1: 50% of emails are spam. Software has been applied to filter spam. A certain brand of software claims that it can detect 99% of spam emails, and the probability for a false positive (a non-spam email detected as spam) is 5%. Now if an email is detected as spam, then what is the probability that it is in fact a nonspam email?

A. 5/104
B. 95/100
C. 1/100
D. 1/2

Break & Quiz

Q 3.1: 50% of emails are spam. Software has been applied to filter spam. A certain brand of software claims that it can detect 99% of spam emails, and the probability for a false positive (a non-spam email detected as spam) is 5%. Now if an email is detected as spam, then what is the probability that it is in fact a nonspam email?

A. 5/104
B. 95/100
C. 1/100
D. 1/2

S : Spam
NS: Not Spam
DS: Detected as Spam

P(S) = 50 % spam email
P(NS) = 50% not spam email
P(DS|NS) = 5% false positive, detected as spam but not spam
P(DS|S) = 99% detected as spam and it is spam

Applying Bayes Rule
P(NS|DS) = (P(DS|NS)*P(NS)) / P(DS) = (P(DS|NS)*P(NS)) / (P(DS|NS)*P(NS) + P(DS|S)*P(S)) = 5/104

Example 1: Play outside or not?

- If weather is sunny, would you like to play outside?

Posterior probability p(Yes | ) vs. p(No | )

Example 1: Play outside or not?

- If weather is sunny, would you like to play outside?

Posterior probability p(Yes | ) vs. p(No | )

- Weather = {Sunny, Rainy, Overcast}
- Play = {Yes, No}
- Observed data {Weather, play on day *m*}, m={1,2,…,N}

Example 1: Play outside or not?

- If weather is sunny, would you like to play outside?

Posterior probability p(Yes | ) vs. p(No | )

- Weather = {Sunny, Rainy, Overcast}
- Play = {Yes, No}
- Observed data {Weather, play on day *m*}, m={1,2,…,N}

p(Play | ) = p( | Play) p(Play) / p()

Bayes rule

Example 1: Play outside or not?

- **Step 1**: Convert the data to a frequency table of Weather and Play

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Example 1: Play outside or not?

- **Step 1**: Convert the data to a frequency table of Weather and Play
- **Step 2**: Based on the frequency table, calculate **likelihoods** and **priors**

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

p(Play = Yes) = 0.64

p(| Yes) = 3/9 = 0.33

https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Example 1: Play outside or not?

- **Step 3**: Based on the likelihoods and priors, calculate posteriors

P(Yes|)

=P( |Yes)*P(Yes)/P()

?

P(No|)

=P( |No)*P(No)/P()

?

Example 1: Play outside or not?

- **Step 3**: Based on the likelihoods and priors, calculate posteriors

P(Yes| ☀)

=P(☀ |Yes)*P(Yes)/P(☀)

=0.33*0.64/0.36

=0.6

P(No|☀)

=P(☀ |No)*P(No)/P(☀)

=0.4*0.36/0.36

=0.4

P(Yes|☀) > P(No| ☀)

go outside and play!

Bayesian classification

What if $\mathbf{x}$ has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

$$\hat{y} = \arg\max_y p(y \mid X_1, \ldots, X_k)$$

(Posterior)

(Prediction)

Bayesian classification

What if **x** has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

$$\hat{y} = \arg\max_y p(y \mid X_1, \ldots, X_k) \quad \text{(Posterior)}$$

(Prediction)

$$= \arg\max_y \frac{p(X_1, \ldots, X_k \mid y) \cdot p(y)}{p(X_1, \ldots, X_k)} \quad \text{(by Bayes' rule)}$$

Independent of y

Bayesian classification

What if **x** has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

$$\hat{y} = \arg\max_y p(y \mid X_1, \ldots, X_k) \quad \text{(Posterior)}$$

(Prediction)

$$= \arg\max_y \frac{p(X_1, \ldots, X_k \mid y) \cdot p(y)}{p(X_1, \ldots, X_k)} \quad \text{(by Bayes' rule)}$$

$$= \arg\max_y p(X_1, \ldots, X_k \mid y)\, p(y)$$

Class conditional likelihood

Class prior

Naïve Bayes Assumption

Conditional independence of feature attributes

$$p(X_1, \ldots, X_k \mid y)p(y) = \Pi_{i=1}^{k} p(X_i \mid y)p(y)$$

Easier to estimate

(using MLE!)

Quiz break

Q3-2: Consider the following dataset showing the result whether a person has passed or failed the exam based on various factors. Suppose the factors are independent to each other.
We want to classify a new instance with
Confident=Yes, Studied=Yes, and Sick=No.

Confident	Studied	Sick	Result
Yes	No	No	Fail
Yes	No	Yes	Pass
No	Yes	Yes	Fail
No	Yes	No	Pass
Yes	Yes	Yes	Pass

- A Pass
- B Fail

Quiz break

Q3-2: Consider the following dataset showing the result whether a person has passed or failed the exam based on various factors. Suppose the factors are independent to each other.
We want to classify a new instance with
Confident=Yes, Studied=Yes, and Sick=No.

Confident	Studied	Sick	Result
Yes	No	No	Fail
Yes	No	Yes	Pass
No	Yes	Yes	Fail
No	Yes	No	Pass
Yes	Yes	Yes	Pass

- A Pass
- B Fail

Quiz break

We want to classify a new instance with Confident=Yes, Studied=Yes, and Sick=No.

- **A Pass**
- B Fail

Confident	Studied	Sick	Result
Yes	No	No	Fail
Yes	No	Yes	Pass
No	Yes	Yes	Fail
No	Yes	No	Pass
Yes	Yes	Yes	Pass

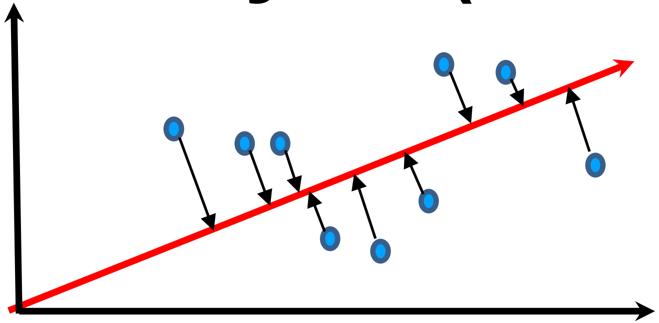
$$P(y=F|x_1=Y,x_2=Y,x_3=N)$$
$$=\frac{1}{2}*\frac{1}{2}*\frac{1}{2}*\frac{2}{5}/P(x_1=Y,x_2=Y,x_3=N)$$
$$\propto\frac{1}{4*5}$$

$$P(y=P|x_1=Y,x_2=Y,x_3=N)$$
$$=\frac{P(x_1=Y|Y=P)P(x_2=Y|Y=P)P(x_3=N|Y=P)P(y=P)}{P(x_1=Y,x_2=Y,x_3=N)}$$
$$=\frac{2}{3}*\frac{2}{3}*\frac{1}{3}*\frac{3}{5}/P(x_1=Y,x_2=Y,x_3=N)$$
$$\propto\frac{4}{9*5}$$

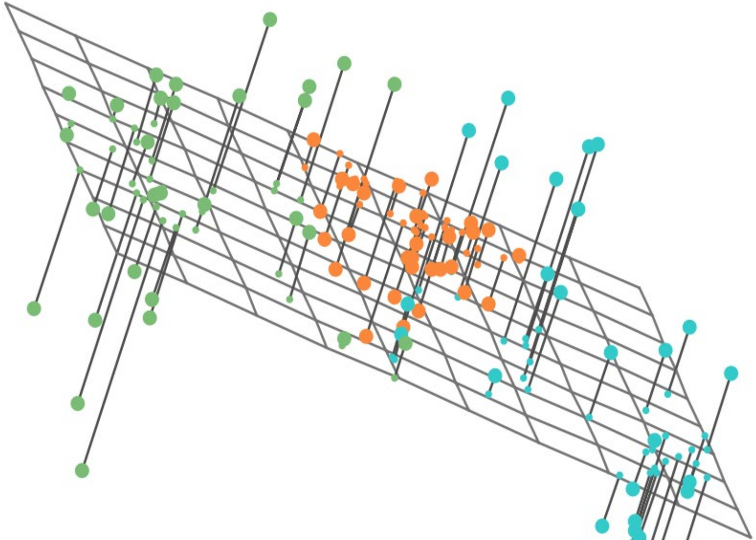
Larger!

Principal Components Analysis (PCA)

- A type of dimensionality reduction approach
  - For when data is **approximately lower dimensional**



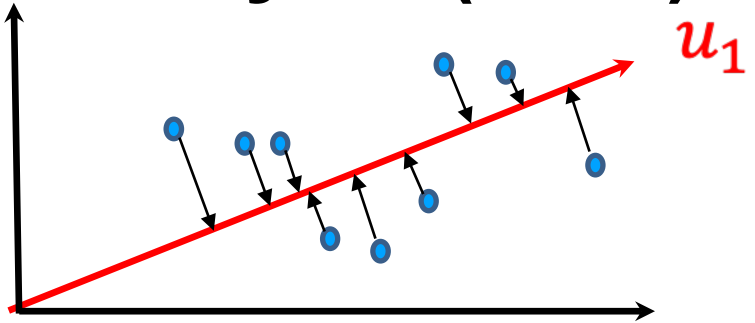
2D
↓
1D



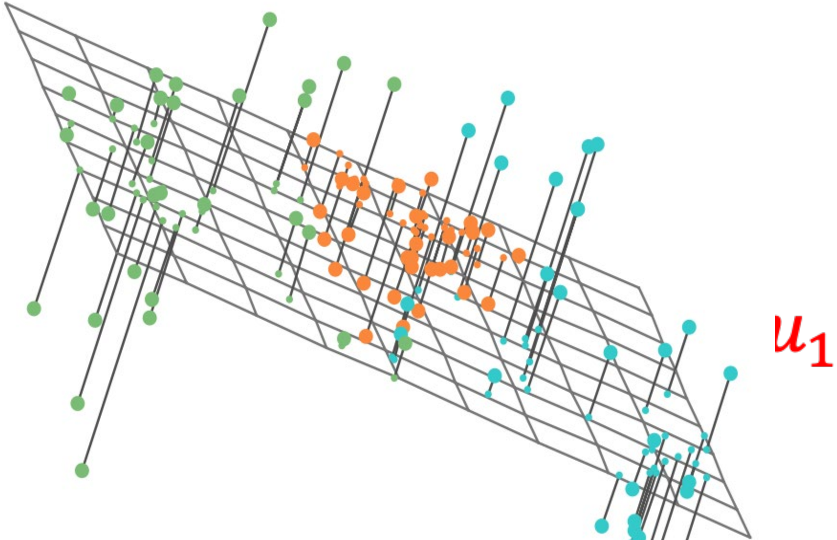
3D
↓
2D

Principal Components Analysis (PCA)

- Find **axes** $u_1, u_2, \ldots, u_m \in \mathbb{R}^d$ of a subspace
  - Will project to this subspace



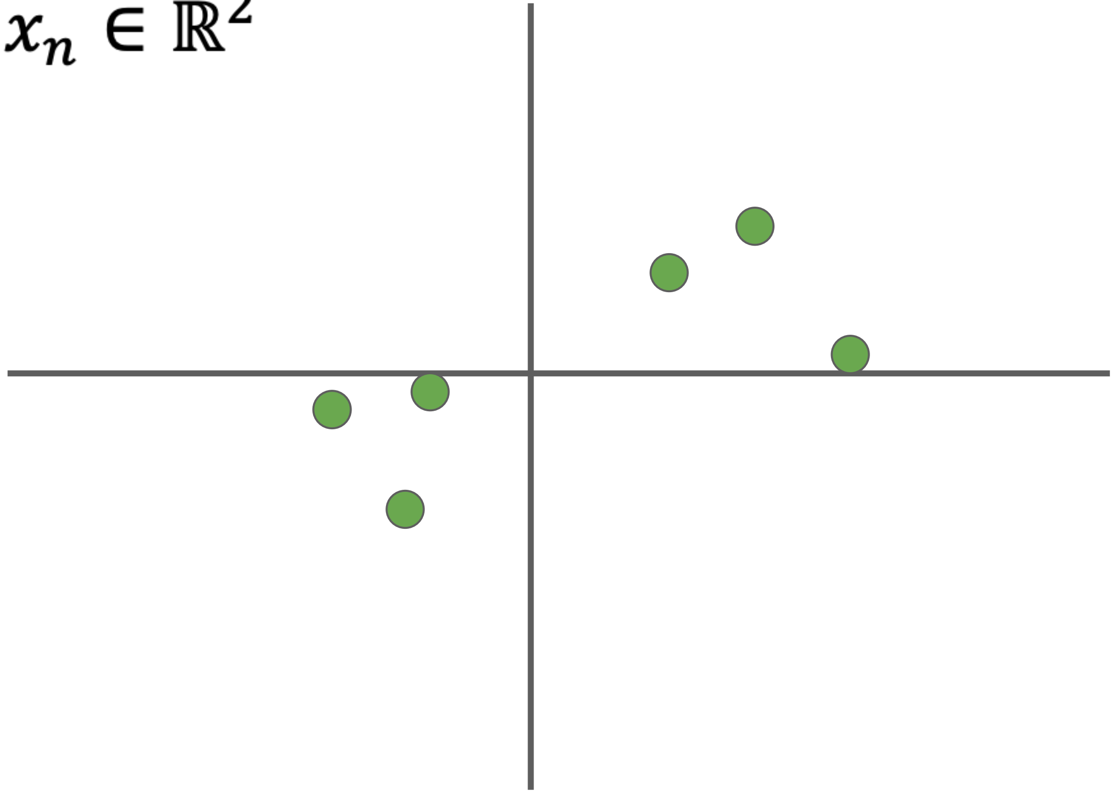
- Want to preserve data
  - minimize projection error



- These vectors are the **principal components**

Projection: An Example

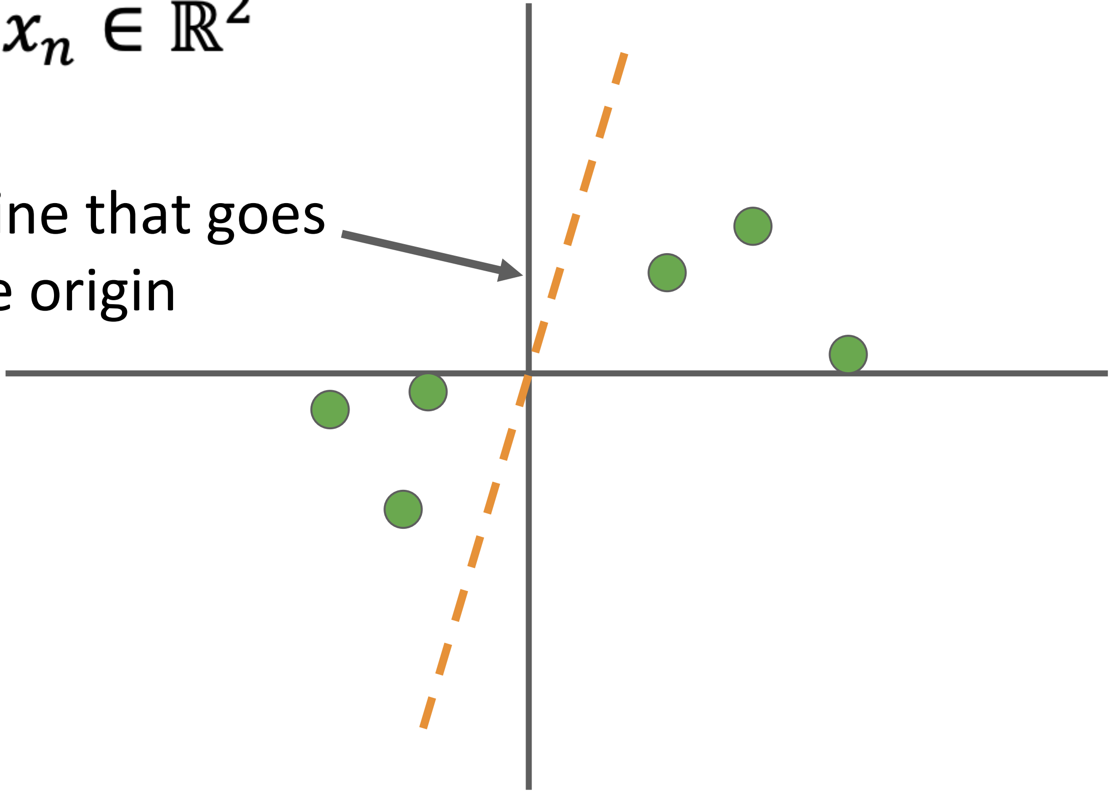
$x_1, x_2, \ldots, x_n \in \mathbb{R}^2$



Projection: An Example

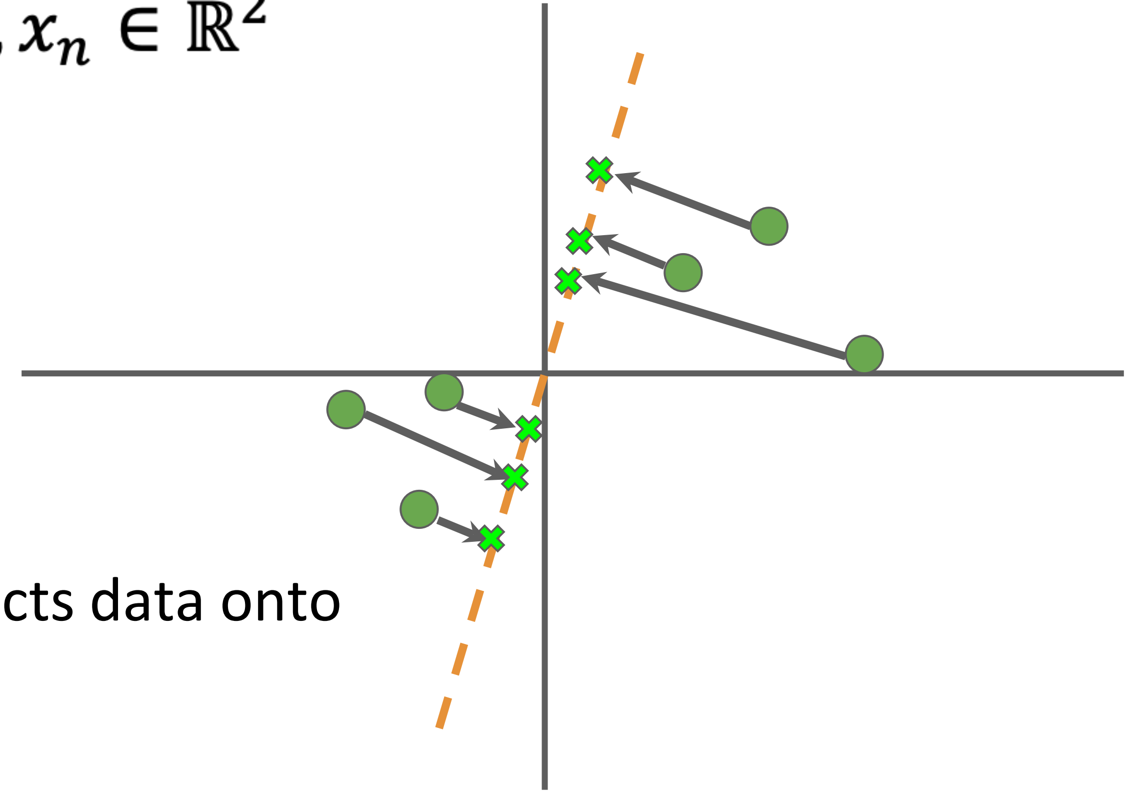
$x_1, x_2, \dots, x_n \in \mathbb{R}^2$

A random line that goes through the origin



Projection: An Example

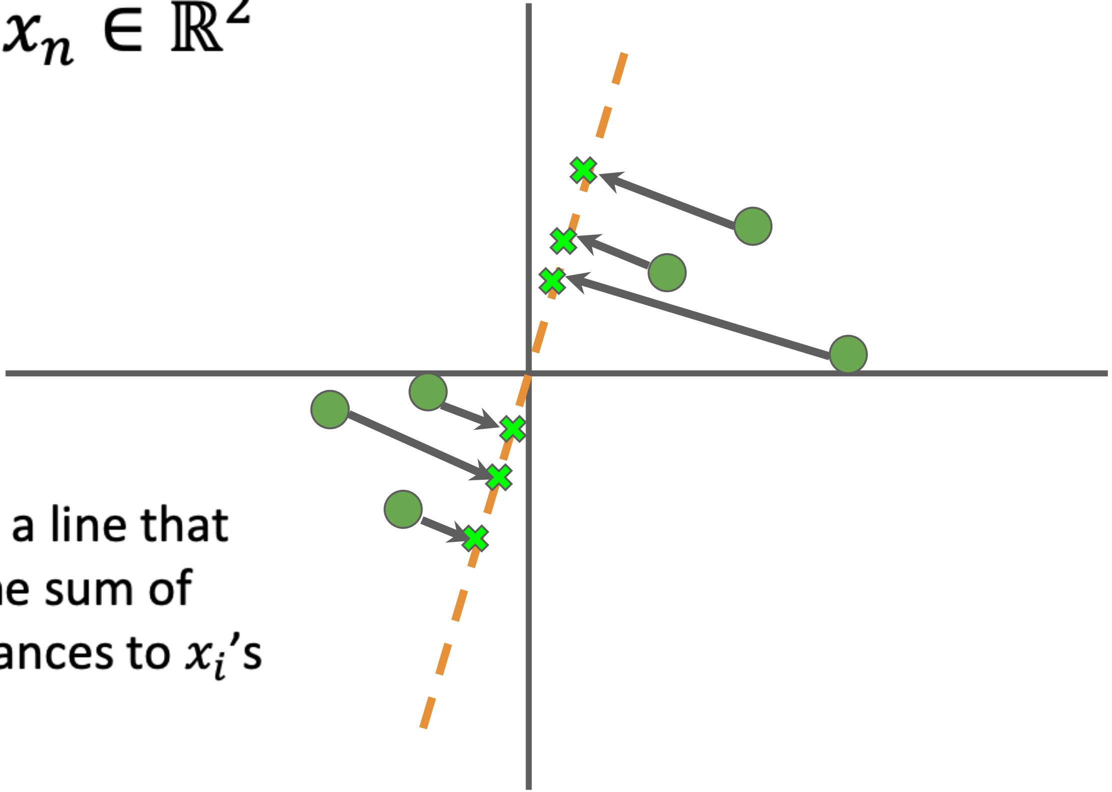
$x_1, x_2, \ldots, x_n \in \mathbb{R}^2$



PCA projects data onto this line

Projection: An Example

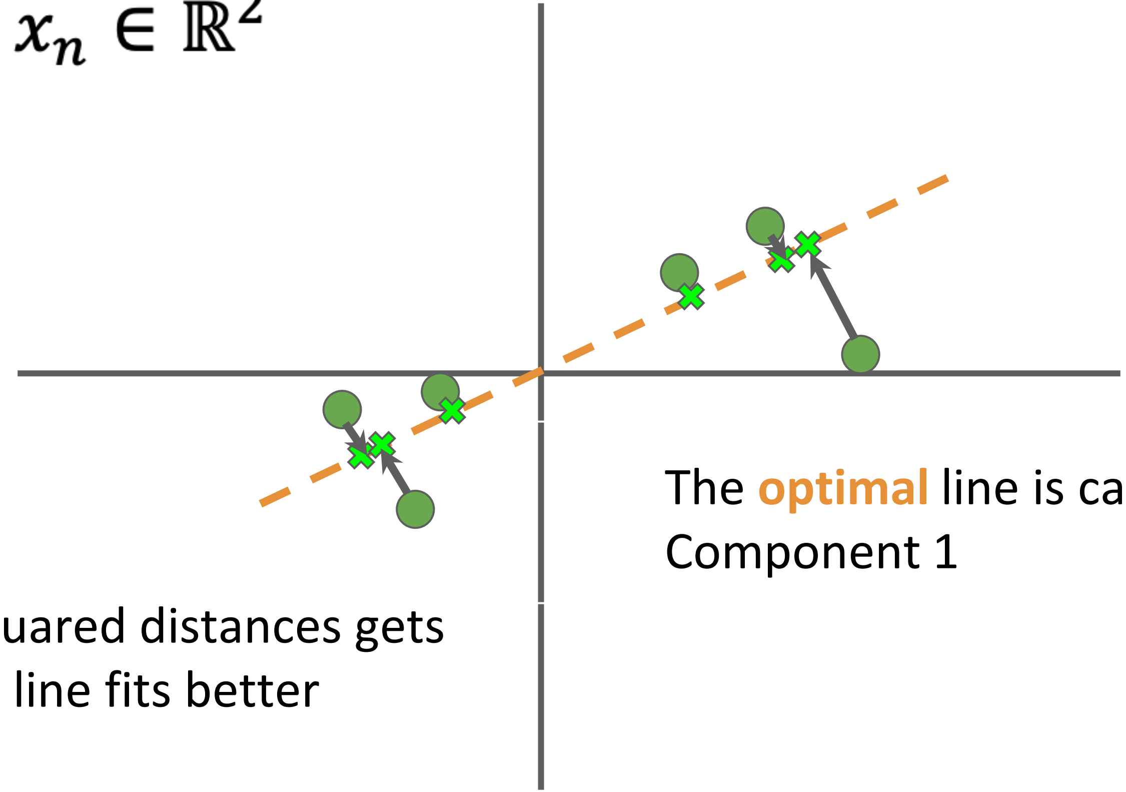
$x_1, x_2, \ldots, x_n \in \mathbb{R}^2$



Goal: finding a line that **minimizes** the sum of squared distances to x_i's

Projection: An Example

$x_1, x_2, \ldots, x_n \in \mathbb{R}^2$



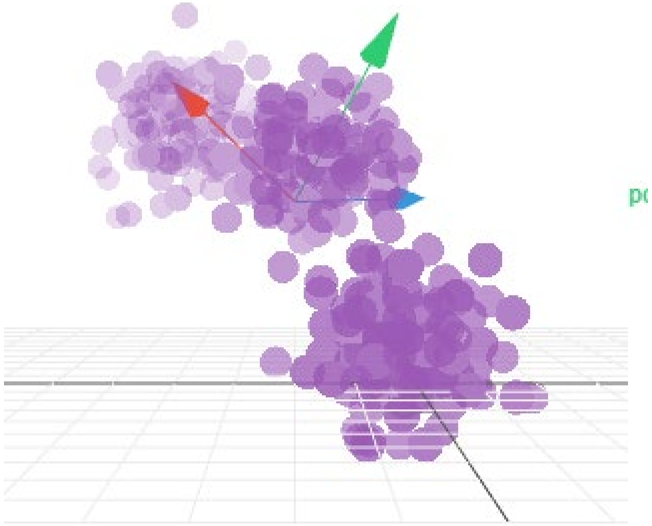
The **optimal** line is called Principal Component 1

The sum of squared distances gets smaller as the line fits better

PCA Procedure

Inputs: data $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$

- **Center data so that $\frac{1}{n}\sum_{i=1}^{n} x_i = 0$**



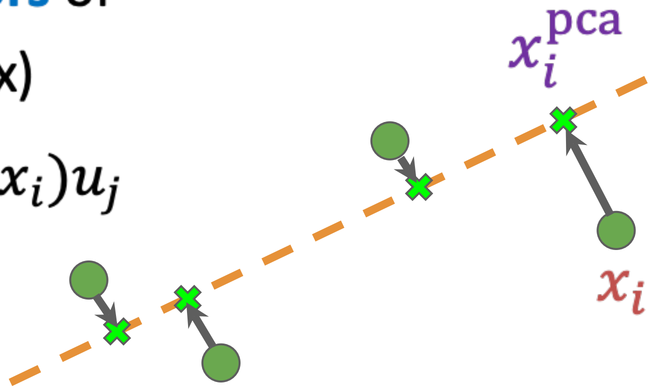
Victor Powell

PCA Procedure

Output:

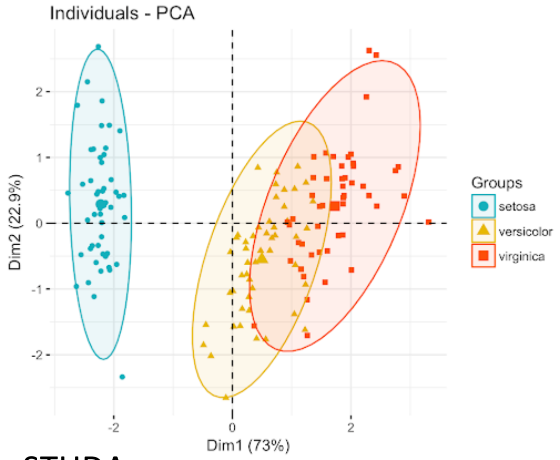
principal components $u_1, \dots, u_m \in \mathbb{R}^d$

- Orthogonal
- Can show: they are top-m **eigenvectors** of $S = \frac{1}{n-1}\sum_{i=1}^{n} x_i x_i^\top$ (covariance matrix)
- Each x_i projected to $x_i^{\text{pca}} = \sum_{j=1}^{m}(u_j^\top x_i)u_j$



Many Variations

- PCA, Kernel PCA, ICA, CCA
  - Extract structure from high dimensional dataset
- Uses:
  - **Visualization**
  - Efficiency
  - Noise removal
  - Downstream machine learning use



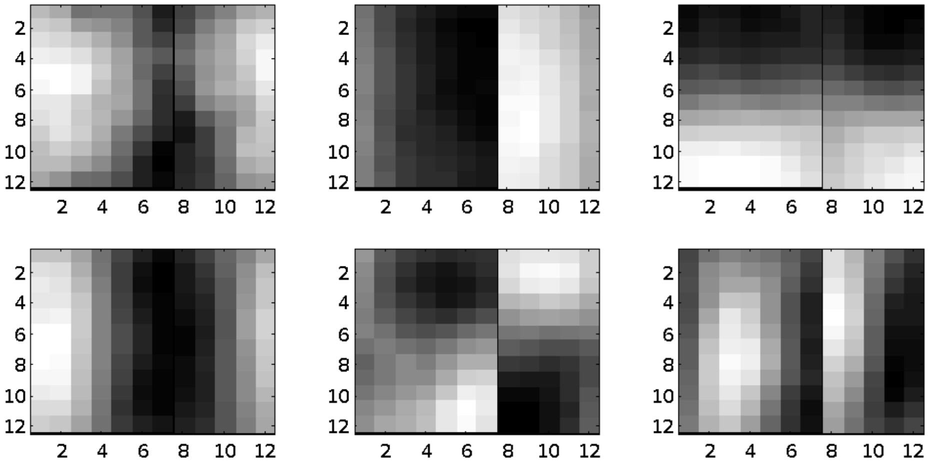
STHDA

Application: Image Compression

- Start with image; divide into 12x12 patches
  - That is, 144-D vector
  - **Original image:**

Application: Image Compression

- 6 principal components (as an image)



Application: Image Compression

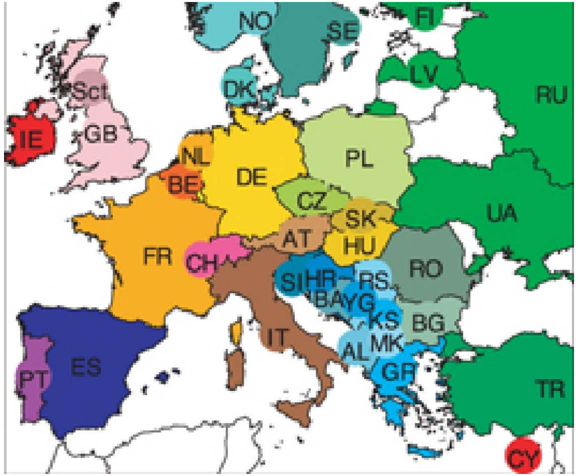
- Project to 6D

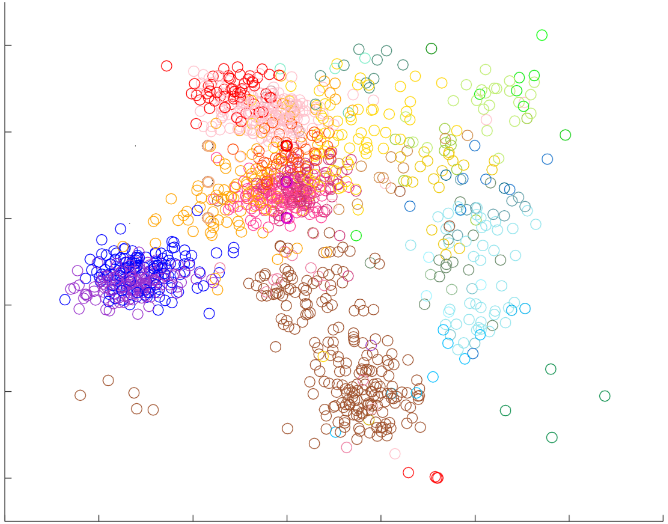
Compressed

Original

Application: Exploratory Data Analysis

- [**Novembre et al. '08**]: Take top two singular vectors of people x SNP matrix (POPRES)

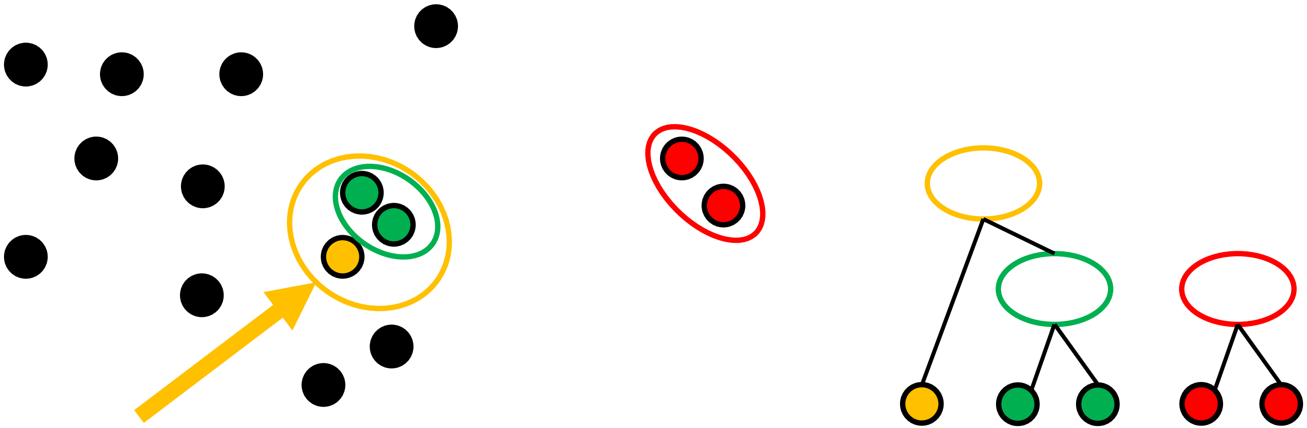




"Genes Mirror Geography in Europe"

Agglomerative Clustering Example

Repeat: Get pair of clusters that are closest and merge



Merging Criteria

Merge: use closest clusters. Define closest?

Single-linkage

$$d(A, B) = \min_{x_1 \in A, x_2 \in B} d(x_1, x_2)$$

Complete-linkage

$$d(A, B) = \max_{x_1 \in A, x_2 \in B} d(x_1, x_2)$$

Average-linkage

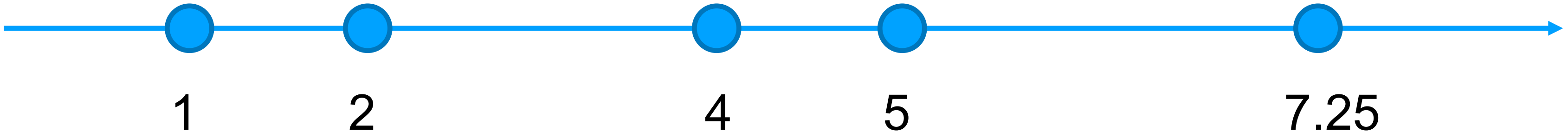
$$d(A, B) = \frac{1}{|A||B|} \sum_{x_1 \in A, x_2 \in B} d(x_1, x_2)$$

Single-linkage Example

We'll merge using single-linkage

1-dimensional vectors.

Initial: all points are clusters

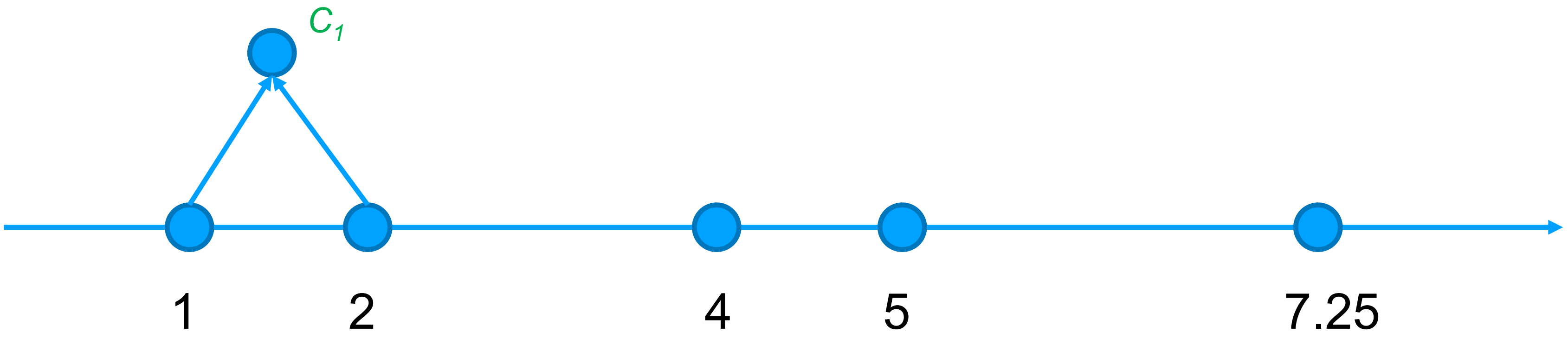


Single-linkage Example

We'll merge using single-linkage

$$d(C_1, \{4\}) = d(2, 4) = 2$$

$$d(\{4\}, \{5\}) = d(4, 5) = 1$$

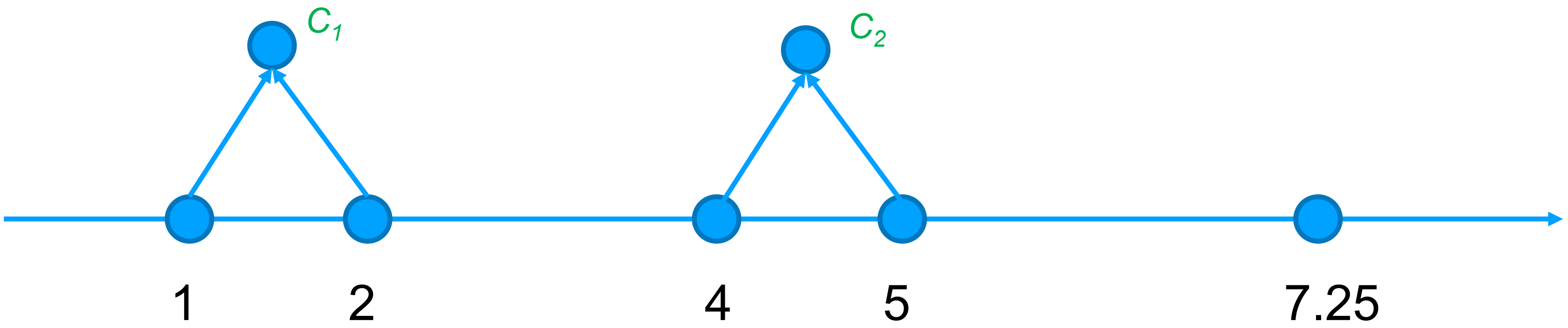


Single-linkage Example

Continue…

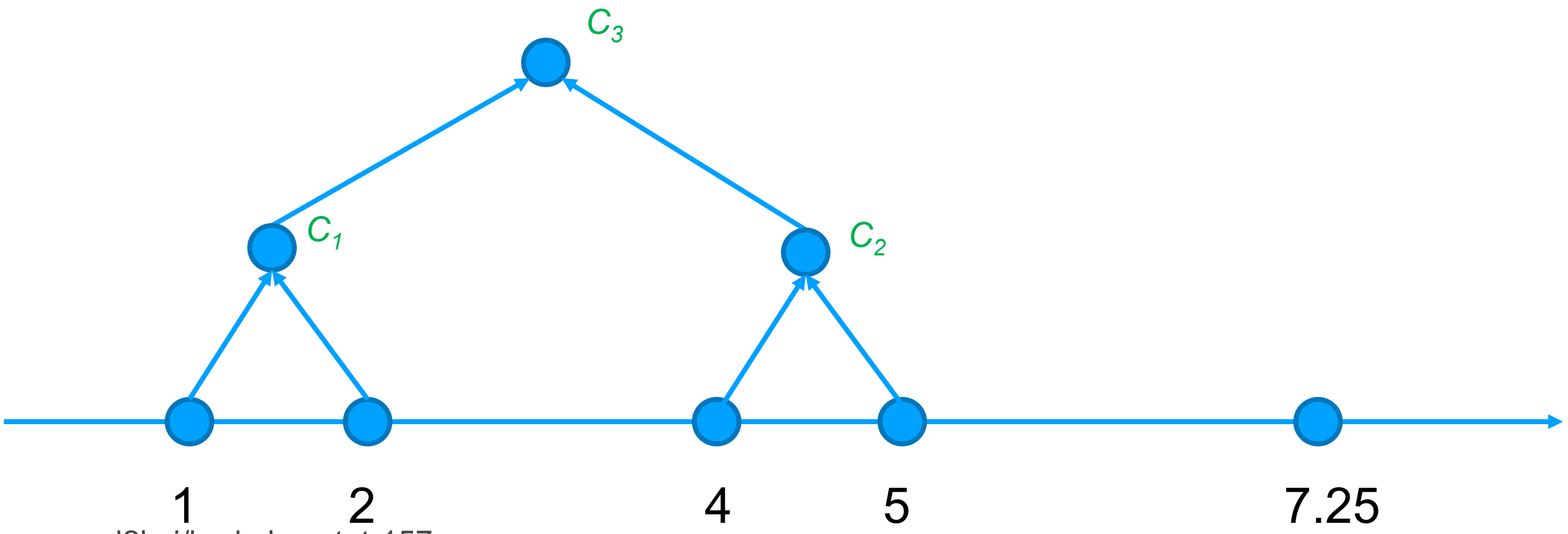
$$d(C_1, C_2) = d(2, 4) = 2$$

$$d(C_2, \{7.25\}) = d(5, 7.25) = 2.25$$

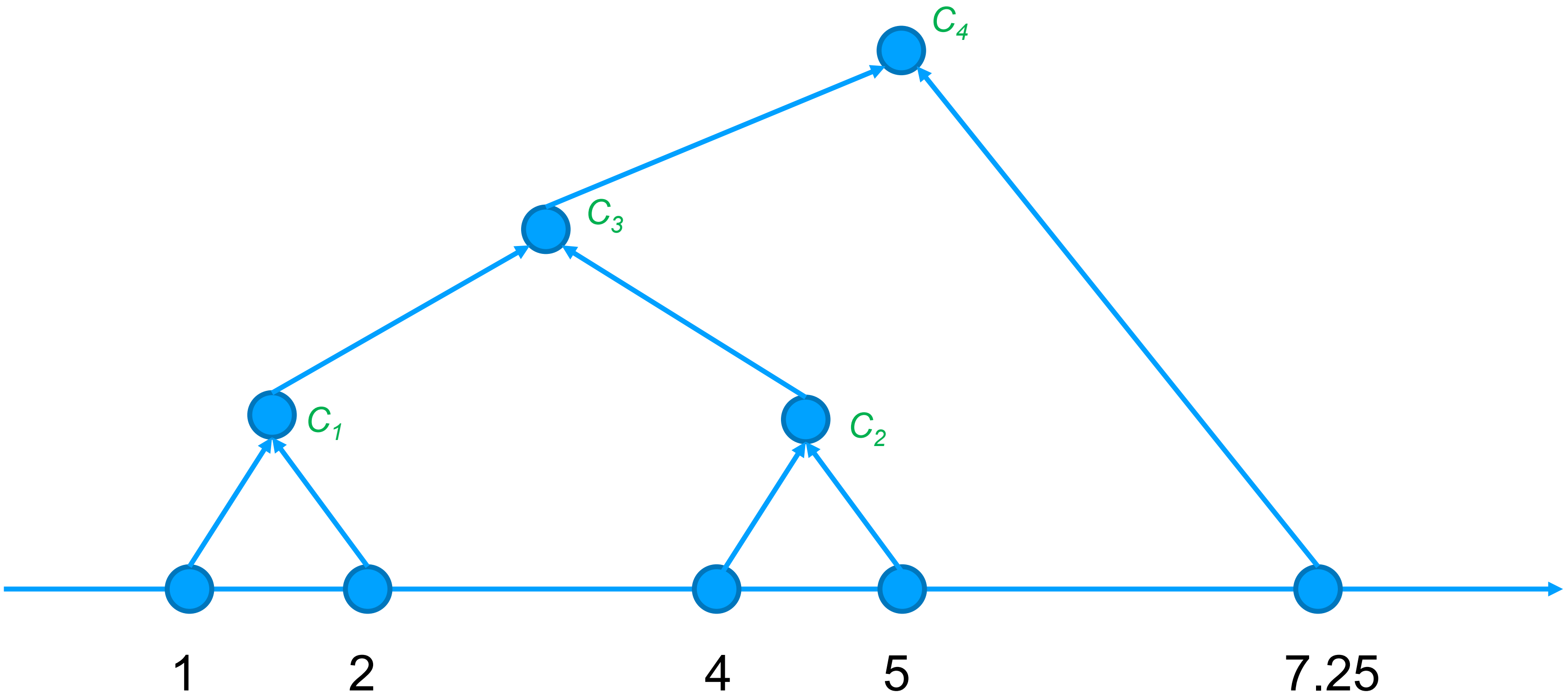


Single-linkage Example

Continue…



Single-linkage Example

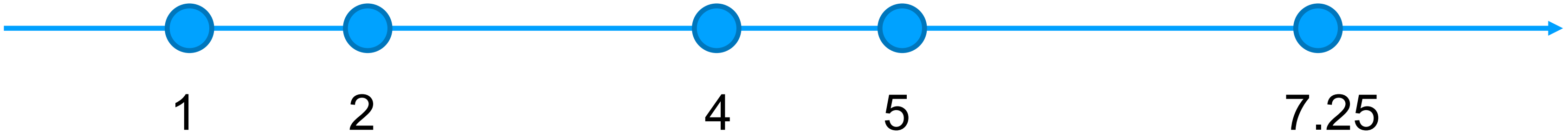


Complete-linkage Example

We'll merge using complete-linkage

1-dimensional vectors.

Initial: all points are clusters

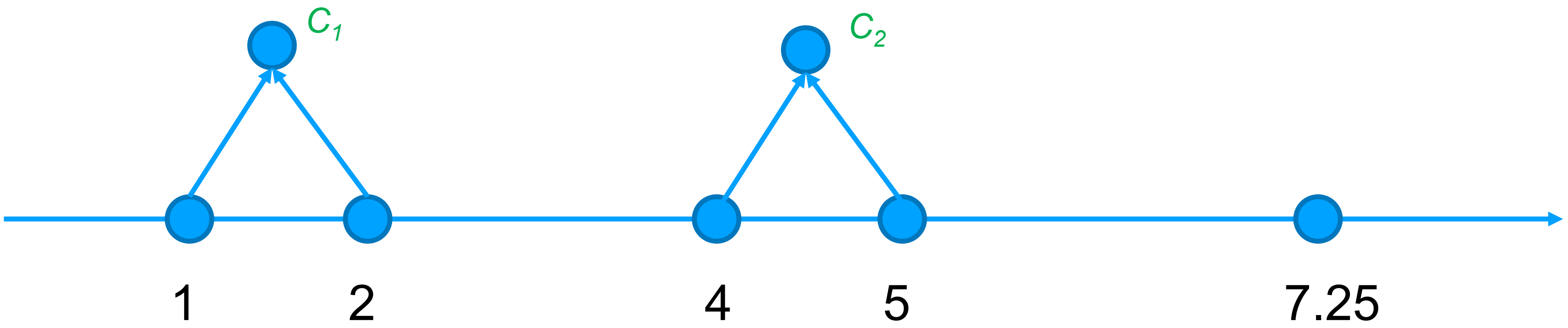


Complete-linkage Example

Beginning is the same…

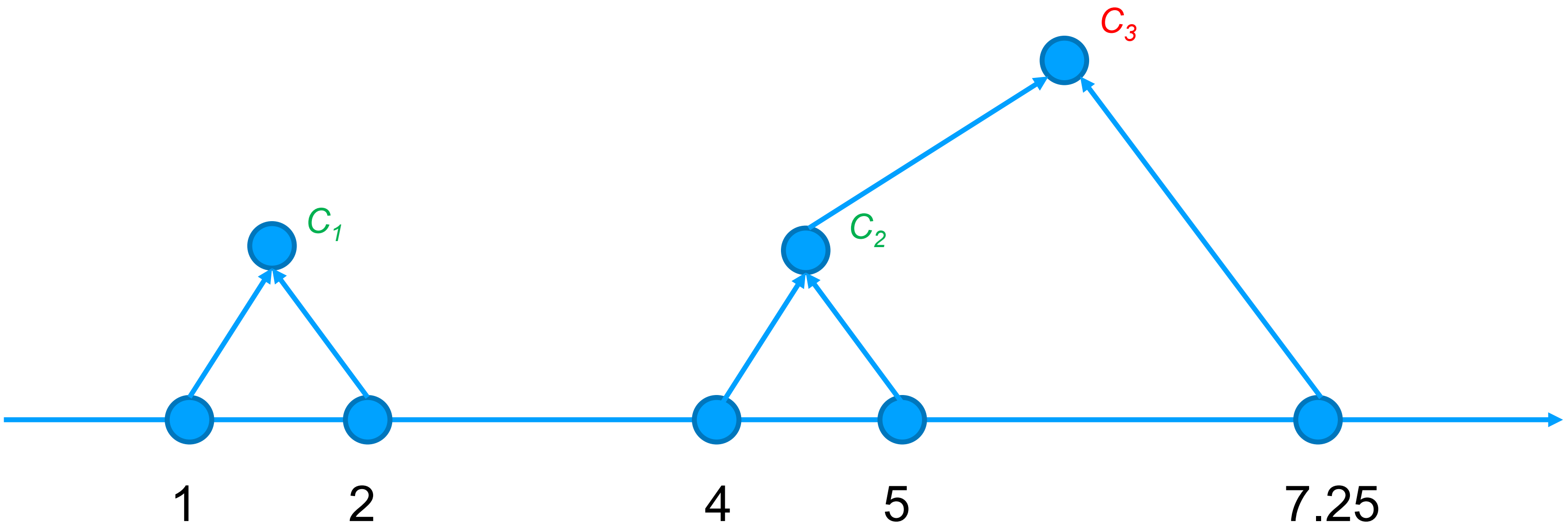
$$d(C_1, C_2) = d(1, 5) = 4$$

$$d(C_2, \{7.25\}) = d(4, 7.25) = 3.25$$

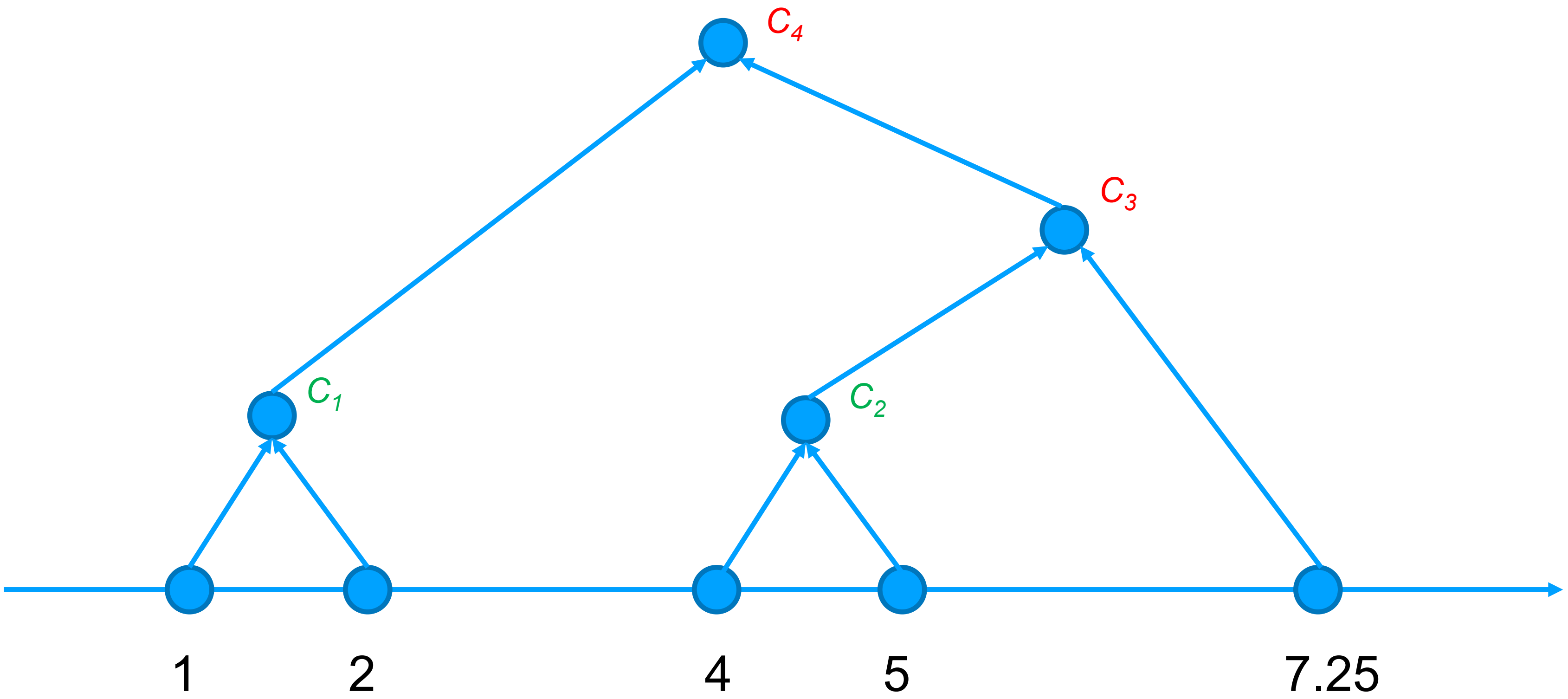


Complete-linkage Example

Now we diverge:



Complete-linkage Example

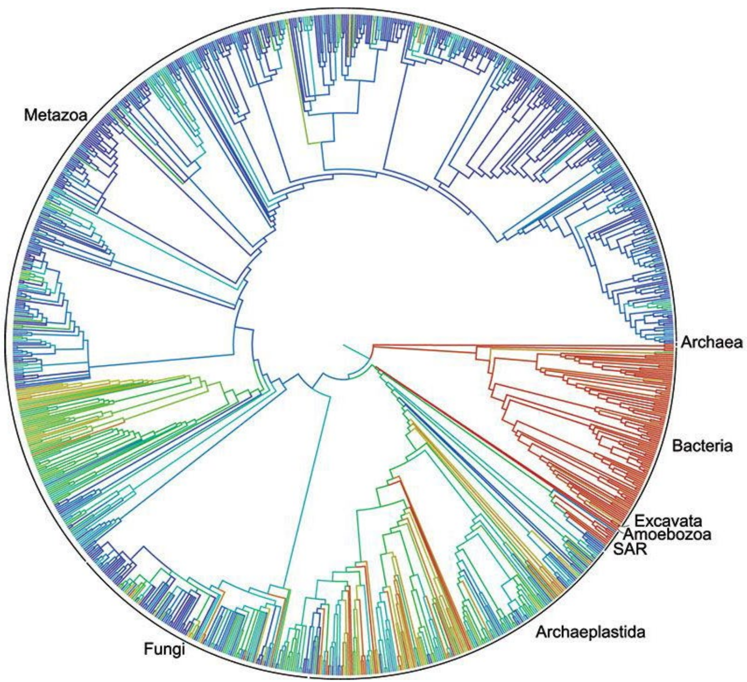


When to Stop?

No simple answer:

Use the binary tree
(a **dendogram**)

Cut at different levels (get different heights/depths)

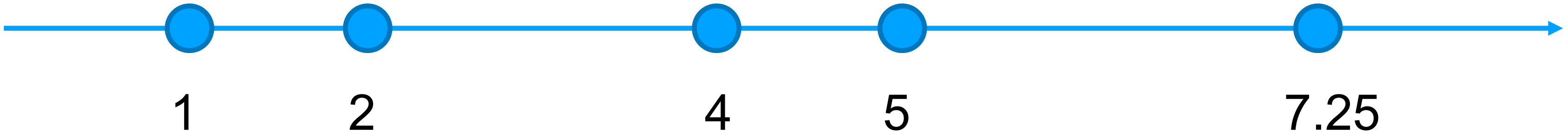


http://opentreeoflife.org/

Break & Quiz

Q 1.1: Let's do hierarchical clustering for two clusters with average linkage on the dataset below. What are the clusters?

A. {1}, {2,4,5,7.25}
B. {1,2}, {4, 5, 7.25}
C. {1,2,4}, {5, 7.25}
D. {1,2,4,5}, {7.25}

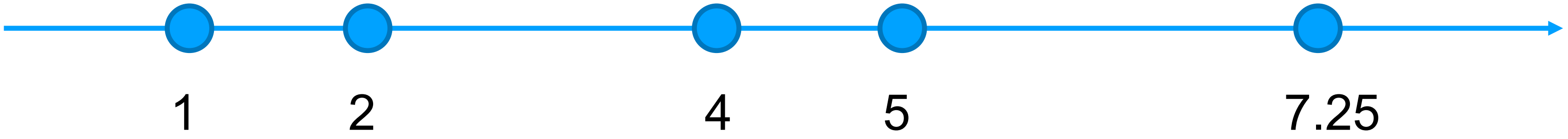


Break & Quiz

Q 1.1: Let's do hierarchical clustering for two clusters with average linkage on the dataset below. What are the clusters?

A. {1}, {2,4,5,7.25}
B. {1,2}, {4, 5, 7.25}
C. {1,2,4}, {5, 7.25}
D. {1,2,4,5}, {7.25}

Iteration 1: merge 1 and 2
Iteration 2: merge 4 and 5
Iteration 3: Now we have clusters {1,2}, {4,5}, {7.25}.
distance({1,2}, {4,5})= 3
distance({4,5}, {7.25}) = 2.75
distance({1,2}, {7.25}) is clearly larger than the above two.
So average linkage will merge {4,5} and {7.25}



Supervised Machine Learning

Statistical modeling approach

Labeled training data (n examples) → Learning algorithm → Classifier $\hat{f}$

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$$

drawn **independently** from
a fixed underlying distribution
(also called the i.i.d. assumption)

select $\hat{f}(\theta)$ from a pool of models $\mathcal{F}$
that **best describe the data observed**

How to select $\hat{f} \in \mathcal{F}$?

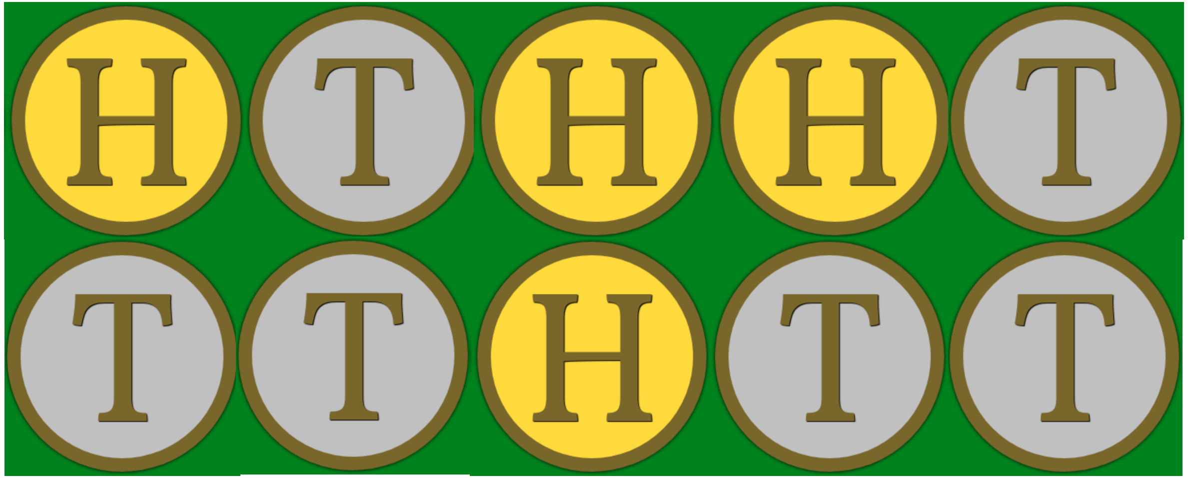
- **Maximum likelihood (best fits the data)**
- Maximum a posteriori

(best fits the data but incorporates prior assumptions)

- Optimization of ‘loss’ criterion (best discriminates the labels)

Maximum Likelihood Estimation: An Example

Flip a coin 10 times, how can you estimate $\theta = p(Head)$?



Intuitively, $\theta = 4/10 = 0.4$

How good is θ?

It depends on how likely it is to generate the observed data

$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$

(Let's forget about label for a second)

Likelihood function $L(\theta) = \Pi_i p(\mathbf{x}_i|\theta)$

Under i.i.d assumption

Interpretation: How **probable** (or how likely) is the data given the probabilistic model p_θ?

How good is θ?

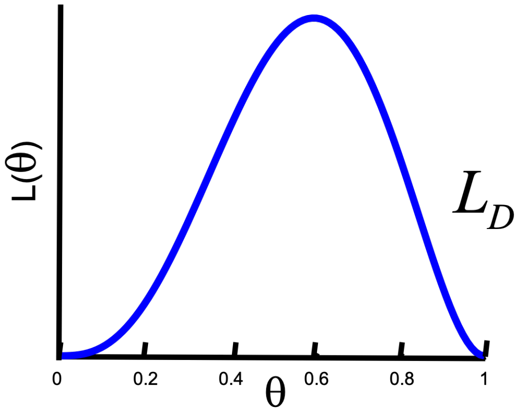
It depends on how likely it is to generate the observed data

$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ (Let's forget about label for a second)

Likelihood function $$L(\theta) = \Pi_i p(\mathbf{x}_i|\theta)$$

H,T, T, H, H

$$L_D(\theta) = \theta \cdot (1-\theta) \cdot (1-\theta) \cdot \theta \cdot \theta$$



Bernoulli distribution

Log-likelihood function

$$L_D(\theta) = \theta \cdot (1-\theta) \cdot (1-\theta) \cdot \theta \cdot \theta$$

$$= \theta^{N_H} \cdot (1-\theta)^{N_T}$$

Log-likelihood function

$$\ell(\theta) = \log L(\theta)$$

$$= N_H \log\theta + N_T \log(1-\theta)$$

Maximum Likelihood Estimation (MLE)

Find optimal θ^* to maximize the likelihood function (and log-likelihood)

$$\theta^* = \text{argmax} \; N_H \log\theta + N_T \log(1-\theta)$$

$$\frac{\partial l(\theta)}{\partial \theta} = \frac{N_H}{\theta} - \frac{N_T}{1-\theta} = 0 \;\Rightarrow\; \theta^* = \frac{N_H}{N_T + N_H}$$

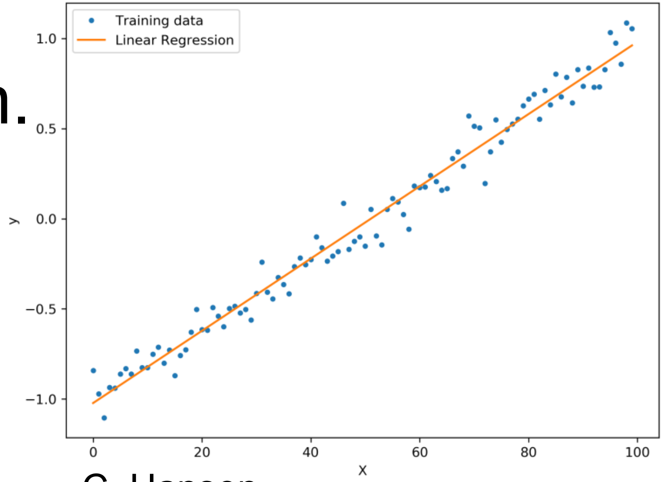
which confirms your intuition!

Linear Regression

Simplest type of regression problem.

Inputs: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$

- *x*'s are vectors, *y*'s are scalars.
- "**Linear**": predict a linear combination of x components + intercept



C. Hansen

$$f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_d x_d = \theta_0 + x^T \theta$$

Want: parameters θ

Linear Regression Setup

Problem Setup

Goal: figure out how to minimize square loss

Let’s organize it. Train set $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$

- Since $f(x) = \theta_0 + x^T\theta$, use a notational trick by augmenting feature vector with a constant dimension of 1:

$$x = \begin{bmatrix} 1 \\ x \end{bmatrix}$$

- Then, with this one more dimension we can write (θ contains θ_0 now)

$$f(x) = x^T\theta$$

Linear Regression Setup

Problem Setup

Train set $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$

Take train features and make it a n*(d+1) matrix, and y a vector:

$$X = \begin{bmatrix} x_1^T \\ \ldots \\ x_n^T \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ \ldots \\ y_n \end{bmatrix}$$

Then, the empirical risk is $\frac{1}{n}\|X\theta - y\|^2$

Finding The Estimated Parameters

Have our loss: $\frac{1}{n}\|X\theta - y\|^2$

Could optimize it with SGD, etc…

But the minimum also has a closed-form solution (vector calculus):

Hat: indicates an estimate

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

Not always invertible…

"Normal Equations"

How Good are the Estimated Parameters?

Now we have parameters $\hat{\theta} = (X^T X)^{-1} X^T y$

How good are they?

Predictions are $f(x_i) = \hat{\theta}^T x_i = ((X^T X)^{-1} X^T y)^T x_i$

Errors ("residuals")

$$|y_i - f(x_i)| = |y_i - \hat{\theta}^T x_i| = |y_i - ((X^T X)^{-1} X^T y)^T x_i|$$

If data is linear, residuals are 0. Almost never the case!

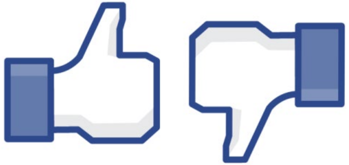
Mean squared error on a test set

$$\frac{1}{m} \sum_{i=n+1}^{n+m} (\hat{\theta}^T x_i - y_i)^2$$

Linear Regression → Classification?

What if we want the same idea, but *y* is 0 or 1?

Need to convert the $\theta^T x$ to a probability in [0,1]



$$p(y = 1|x) = \frac{1}{1 + \exp(-\theta^T x)}$$

Logistic function

Why does this work?

If $\theta^T x$ is really big, $\exp(-\theta^T x)$ is really small → *p* close to 1

If really negative exp is huge → *p* close to 0

"Logistic Regression"

Break & Quiz

Q 2.1: You have a dataset for regression given by $(x_1, y_1) = ([-1,0,1], 2)$ and $(x_2, y_2) = ([2,3,1], 4)$.

What are the labels, number of points (n), and dimension of the features (d)?

- A. labels are 2 and 4; n=3, and d=2.
- B. labels are 2 and 4; n=2, and d=3.
- C. labels are [-1,0,1] and [2,3,1]; n=2, and d=4.
- D. labels are 2 and 3; n=4, and d=2.

Break & Quiz

Q 2.1: You have a dataset for regression given by $(x_1, y_1) = ([-1,0,1], 2)$ and $(x_2, y_2) = ([2,3,1], 4)$.

What are the labels, number of points (n), and dimension of the features (d)?

- A. labels are 2 and 4; n=3, and d=2.
- **B. labels are 2 and 4; n=2, and d=3.**
- C. labels are [-1,0,1] and [2,3,1]; n=2, and d=4.
- D. labels are 2 and 3; n=4, and d=2.

Break & Quiz

Q 2.1: You have a dataset for regression given by $(x_1, y_1) = ([-1,0,1], 2)$ and $(x_2, y_2) = ([2,3,1], 4)$.

What are the labels, number of points (n), and dimension of the features (d)?

- A. labels are 2 and 4; n=3, and d=2.
- **B. labels are 2 and 4; n=2, and d=3.**
- C. labels are [-1,0,1] and [2,3,1]; n=2, and d=4.
- D. labels are 2 and 3; n=4, and d=2.

There are two data points, each x has 3 features, and the labels are the y-values.

Break & Quiz

Q 2.2: You have a dataset for regression given by $(x_1, y_1) = ([-1,0,1], 2)$ and $(x_2, y_2) = ([2,3,1], 4)$.

We have the weights $\beta_0 = 0, \beta_1 = 2, \beta_2 = 1, \beta_3 = 1$. Predict $\hat{\boldsymbol{y}}$ for $x = [1, 10, 1]$

- A. 15
- B. 9
- C. 13
- D. 21

Break & Quiz

Q 2.2: You have a dataset for regression given by $(x_1, y_1) = ([-1,0,1], 2)$ and $(x_2, y_2) = ([2,3,1], 4)$.

We have the weights $\beta_0 = 0, \beta_1 = 2, \beta_2 = 1, \beta_3 = 1$. Predict $\hat{\boldsymbol{y}}$ for $x = [1, 10, 1]$

- A. 15
- B. 9
- **C. 13**
- D. 21

Break & Quiz

Q 2.2: You have a dataset for regression given by $(x_1, y_1) = ([-1,0,1], 2)$ and $(x_2, y_2) = ([2,3,1], 4)$.

We have the weights $\beta_0 = 0, \beta_1 = 2, \beta_2 = 1, \beta_3 = 1$. Predict $\hat{y}$ for $x = [1, 10, 1]$

- A. 15
- B. 9
- **C. 13**
- D. 21

$\hat{y} = 1 * \beta_0 + 1 * \beta_1 + 10 * \beta_2 + 1 \ * \beta_3 = 13$

Break & Quiz

Q 2.3: You have a dataset for regression given by $(x_1, y_1) = ([-1,0,1], 2)$ and$(x_2, y_2) = ([2,3,1], 4)$.

We have the weights $\beta_0 = 0, \beta_1 = 2, \beta_2 = 1, \beta_3 = 1$. What is the mean squared error (MSE) on the training set?

- A. 9
- B. 13/2
- C. 25/2
- D. 25

Break & Quiz

Q 2.3: You have a dataset for regression given by $(x_1, y_1) = ([-1,0,1], 2)$ and $(x_2, y_2) = ([2,3,1], 4)$.

We have the weights $\beta_0 = 0, \beta_1 = 2, \beta_2 = 1, \beta_3 = 1$. What is the mean squared error (MSE) on the training set?

- A. 9
- B. 13/2
- C. **25/2**
- D. 25

Break & Quiz

Q 2.3: You have a dataset for regression given by $(x_1, y_1) = ([-1,0,1], 2)$ and $(x_2, y_2) = ([2,3,1], 4)$.

We have the weights $\beta_0 = 0, \beta_1 = 2, \beta_2 = 1, \beta_3 = 1$. What is the mean squared error (MSE) on the training set?

- A. 9
- B. 13/2
- C. **25/2**
- D. 25

Compute the predicted label for each data point, then compute the squared error for each data point, then take the mean error of the two points:

$$\hat{y}_1 = -1 * \beta_1 + 0 * \beta_2 + 1 * \beta_3 = -1$$

$$\ell(\hat{y}_1, y_1) = (-1 - 2)^2 = 9$$

$$\hat{y}_2 = 2 * \beta_1 + 3 * \beta_2 + 1 * \beta_3 = 8$$

$$\ell(\hat{y}_1, y_1) = (8 - 4)^2 = 16$$

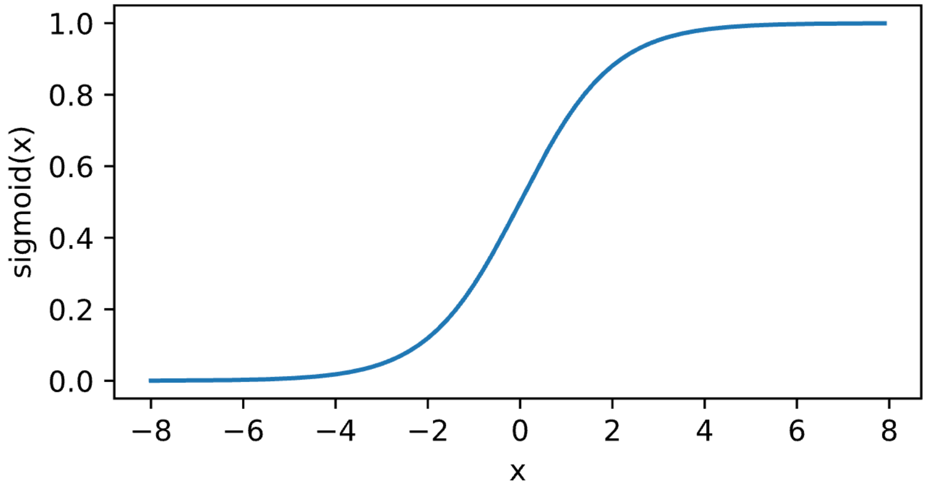
MSE = (16 + 9) / 2 = 25 / 2

Logistic regression

$\mathbf{x} \in \mathbb{R}^d, y = \{-1, +1\}$

$$p(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

$$p(y = -1|\mathbf{x}) = 1 - \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + \exp(\mathbf{w}^T\mathbf{x})}$$



Logistic regression

Given: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ $\quad\quad$ $\mathbf{x} \in \mathbb{R}^d, y = \{-1, +1\}$

Training: maximize likelihood estimate (on the conditional probability)

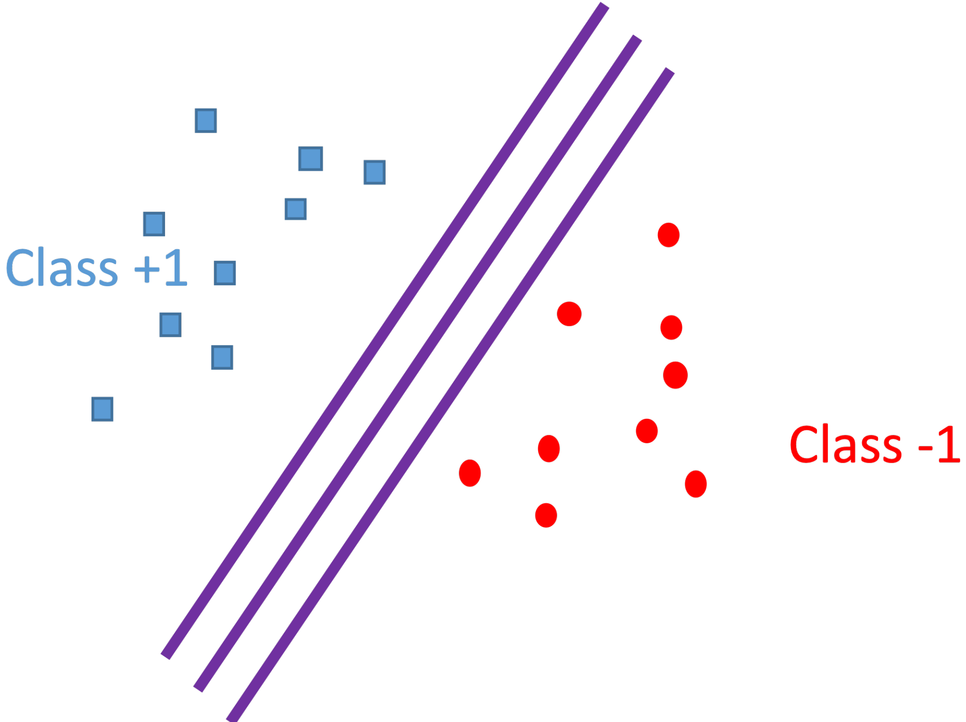
$$\max_{\mathbf{w}} \sum_i \log \frac{1}{1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)}$$

Logistic regression

Given: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ $\quad\quad$ $\mathbf{x} \in \mathbb{R}^d, y = \{-1, +1\}$

Training: maximize likelihood estimate (on the conditional probability)

When training data is linearly separable, many solutions



Logistic regression

Given: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ $\mathbf{x} \in \mathbb{R}^d, y = \{-1, +1\}$

Training: maximum A posteriori (MAP)

$$\min_{\mathbf{w}} \sum_i -\log \frac{1}{1+\exp(-y_i \mathbf{w}^T \mathbf{x}_i)} + \frac{\lambda}{2} \| \mathbf{w} \|_2^2$$

- Convex optimization
- Solve via (stochastic) gradient descent

How to train a neural network? Binary classification

$\mathbf{x} \in \mathbb{R}^d$ One training data point in the training set D

$\hat{y} \in [0,1]$ Model output for example **x**

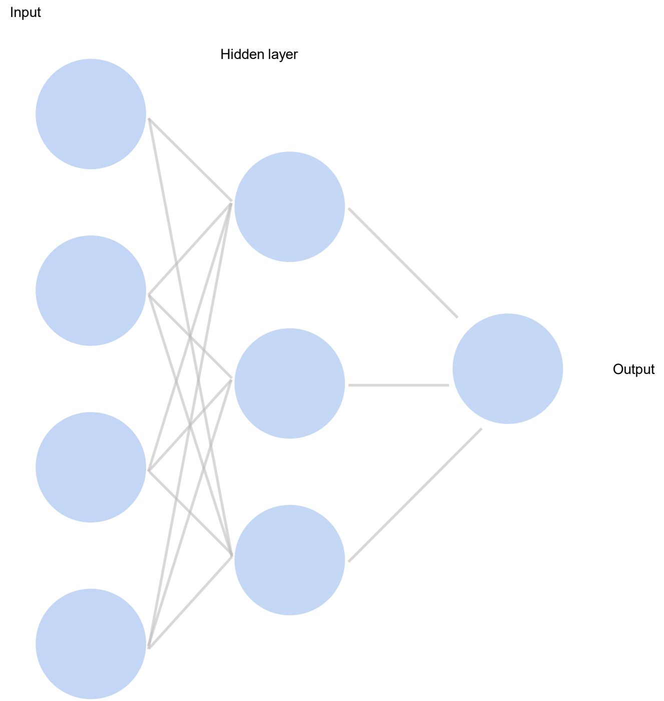
(This is a function of all weights W: $\hat{y} = g(W)$)

y Ground truth label for example **x**

Learning by matching the output to the label

We want $\hat{y} \rightarrow 1$ **when** $y = 1$**,**

and $\hat{y} \rightarrow 0$ **when** $y = 0$

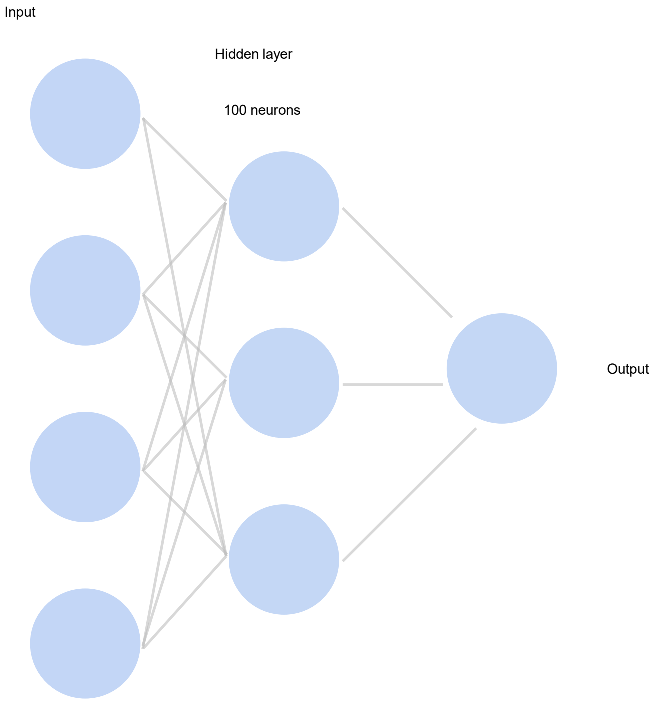


How to train a neural network? Binary classification

Loss function: $$\frac{1}{|D|}\sum_{(\mathbf{x},y)\in D}\ell(\mathbf{x},y)$$

Per-sample loss:

$$\ell(\mathbf{x},y) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$



Negative log likelihood
Minimizing NLL is equivalent to Max Likelihood Learning (MLE)
Also known as binary cross-entropy loss

How to train a neural network? Multiclass

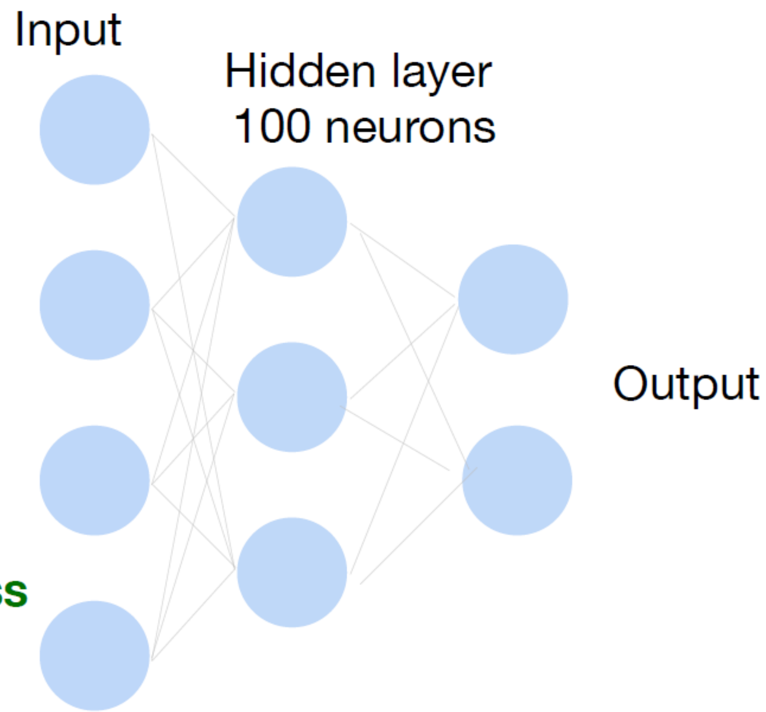
Loss function: $\frac{1}{|D|} \sum_{(\mathbf{x},y)\in D} \ell(\mathbf{x}, y)$

Per-sample loss:

$$\ell(\mathbf{x}, y) = \sum_{k=1}^{K} - Y_k \log p_k = - \log p_y$$

where Y is one-hot encoding of y

Also known as cross-entropy loss or softmax loss

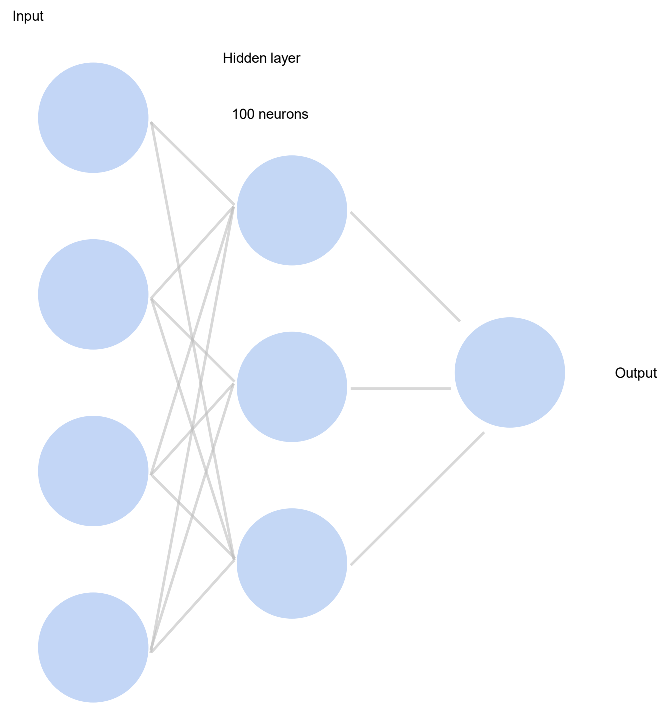


How to train a neural network?

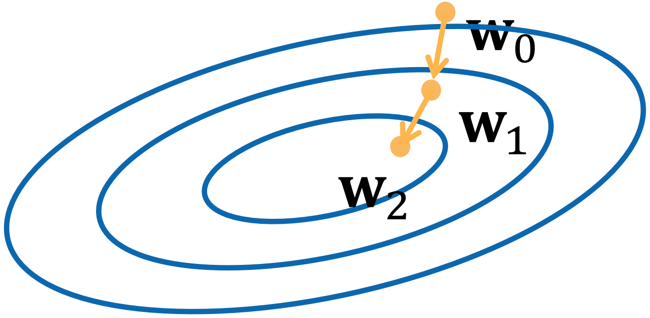
Update the weights W to minimize the loss function

$$L = \frac{1}{|D|} \sum_{(\mathbf{x},y)\in D} \ell(\mathbf{x}, y)$$

Use gradient descent!



Gradient Descent



- Choose a learning rate $\alpha > 0$
- Initialize the model parameters w_0
- For t =1,2,…
  - Update parameters:

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \alpha \frac{\partial L}{\partial \mathbf{w}_{t-1}}$$

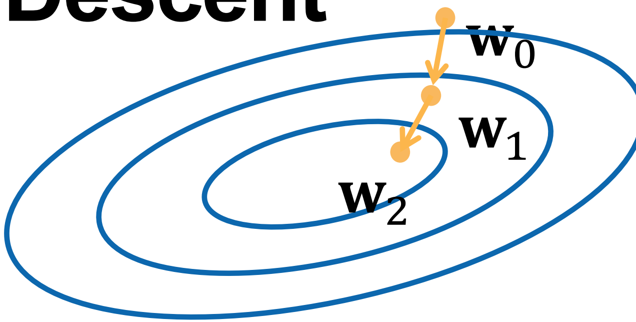
D can be very large. Expensive per iteration

$$= \mathbf{w}_{t-1} - \alpha \frac{1}{|D|} \sum_{(\mathbf{x},y)\in D} \frac{\partial \ell(\mathbf{x}, y)}{\partial \mathbf{w}_{t-1}}$$

The gradient w.r.t. all parameters is obtained by concatenating the partial derivatives w.r.t. each parameter

- Repeat until converges

Minibatch Stochastic Gradient Descent

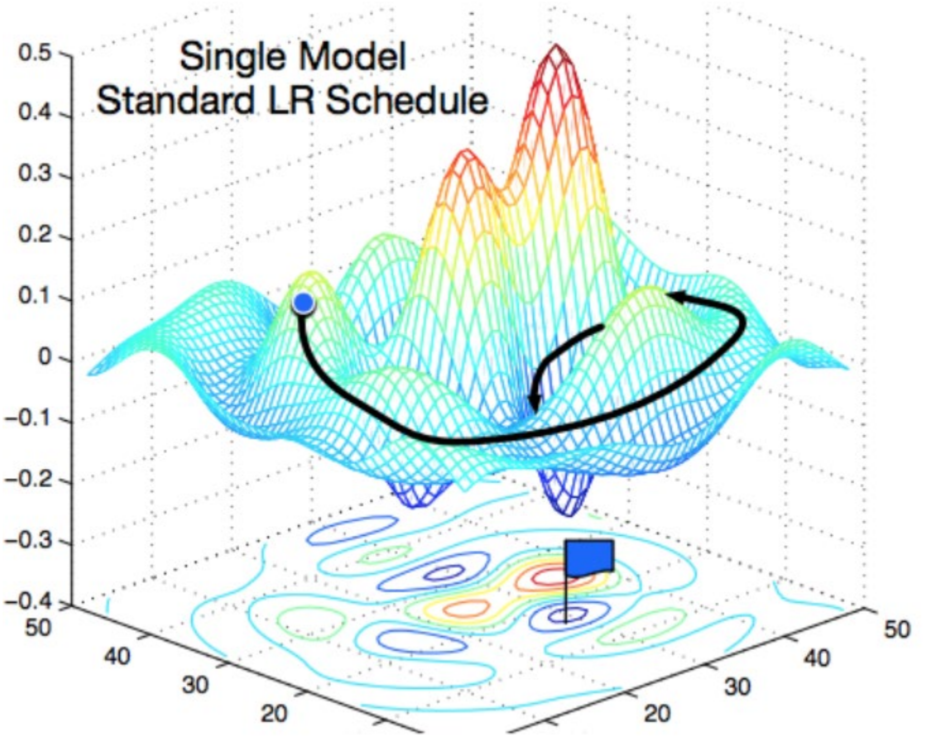


- Choose a learning rate $\alpha > 0$
- Initialize the model parameters w_0
- For t =1,2,…
  - **Randomly sample a subset (mini-batch)** $B \subset D$ Update parameters:

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \alpha \frac{1}{|B|} \sum_{(\mathbf{x},y)\in B} \frac{\partial \ell(\mathbf{x}, y)}{\partial \mathbf{w}_{t-1}}$$

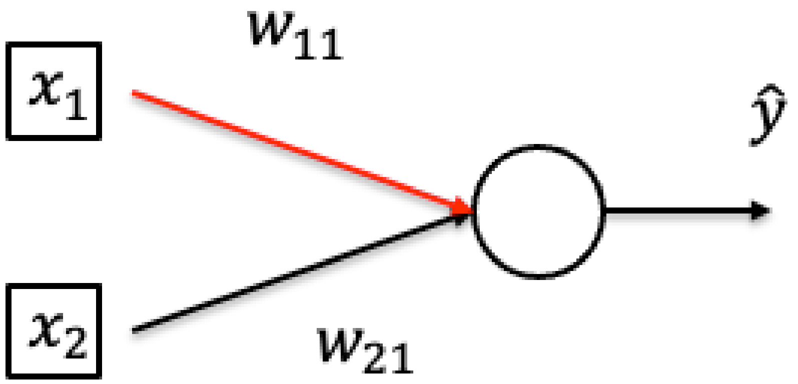
- Repeat until converges

Non-convex Optimization



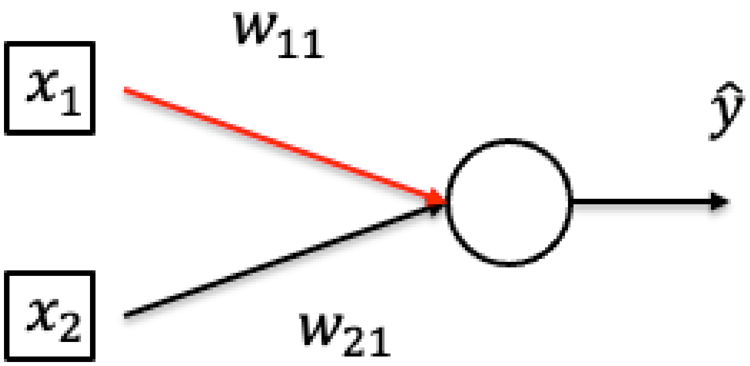
[Gao and Li et al., 2018]

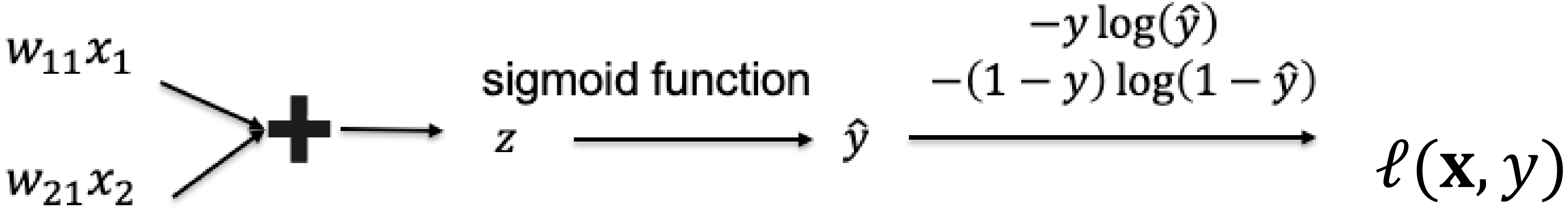
Calculate Gradient (on one data point)



- Want to compute $\frac{\partial \ell(\mathbf{x}, y)}{\partial w_{11}}$
- Data point: $((x_1, x_2), y)$

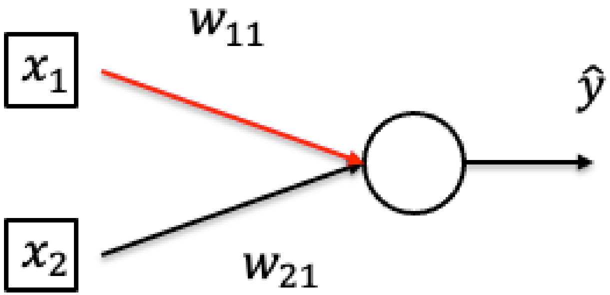
Calculate Gradient (on one data point)

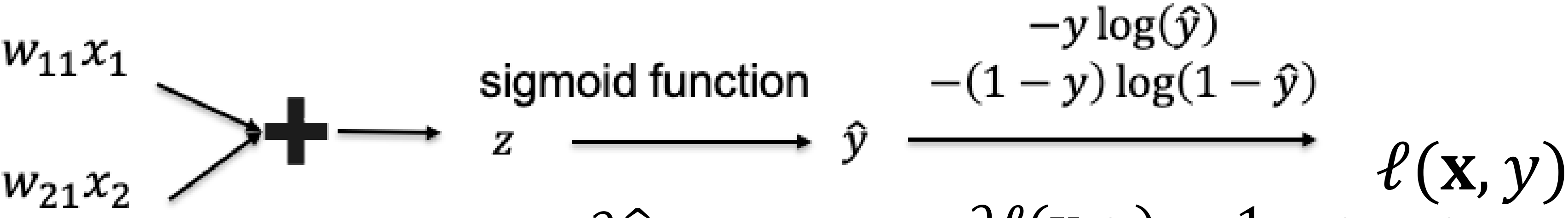




Use chain rule!

Calculate Gradient (on one data point)

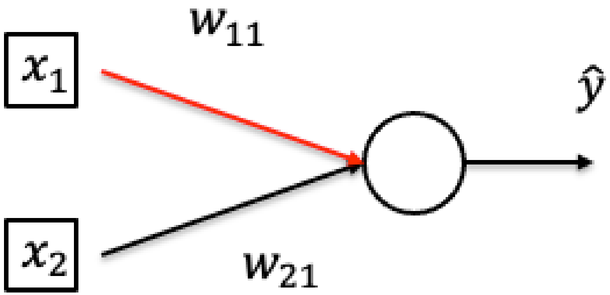


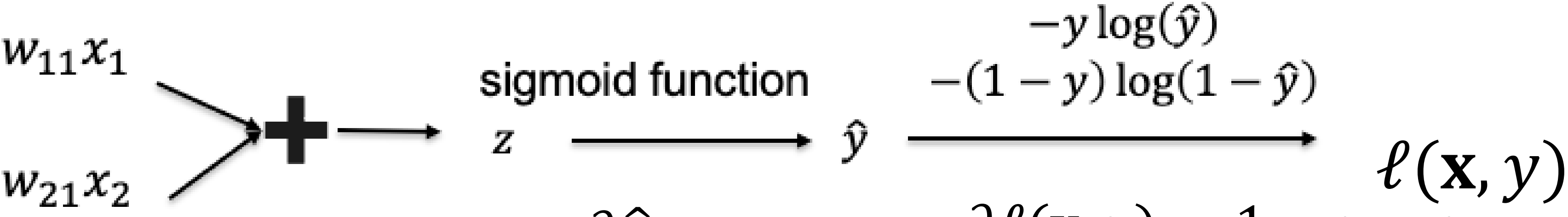


$$\frac{\partial \hat{y}}{\partial z} = \sigma'(z) \qquad \frac{\partial \ell(\mathbf{x}, y)}{\partial \hat{y}} = \frac{1-y}{1-\hat{y}} - \frac{y}{\hat{y}}$$

- By chain rule: $\frac{\partial l}{\partial w_{11}} = \frac{\partial l}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial w_{11}}$

Calculate Gradient (on one data point)

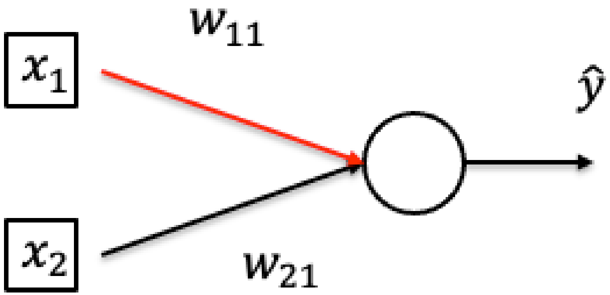


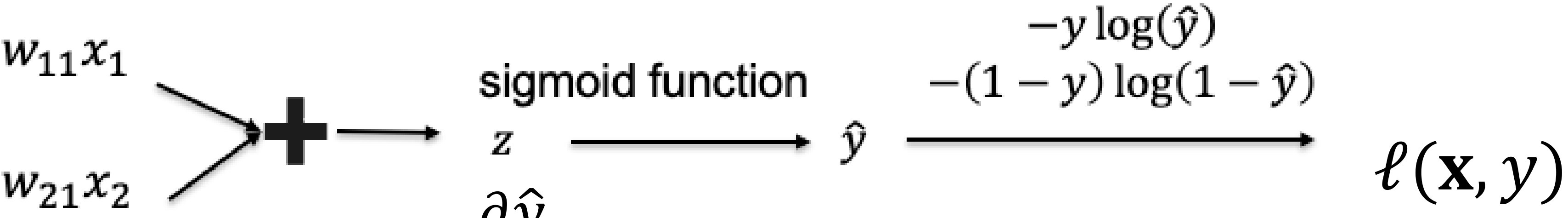


$$\frac{\partial \hat{y}}{\partial z} = \sigma'(z) \qquad \frac{\partial \ell(\mathbf{x}, y)}{\partial \hat{y}} = \frac{1-y}{1-\hat{y}} - \frac{y}{\hat{y}}$$

- By chain rule: $\dfrac{\partial l}{\partial w_{11}} = \dfrac{\partial l}{\partial \hat{y}} \dfrac{\partial \hat{y}}{\partial z} \; x_1$

Calculate Gradient (on one data point)

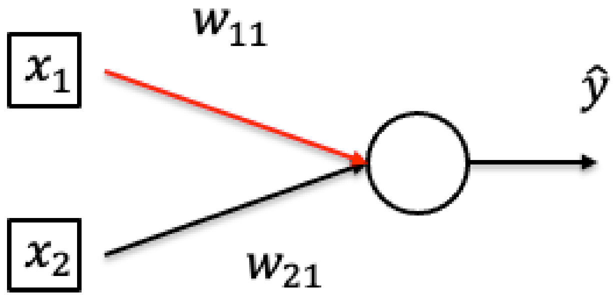


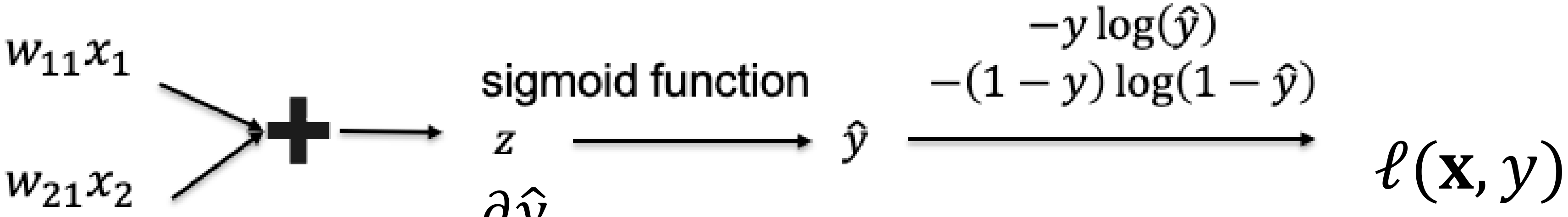


$$\frac{\partial \hat{y}}{\partial z} = \sigma'(z) = \sigma(z)(1-\sigma(z))$$

- By chain rule: $\dfrac{\partial l}{\partial w_{11}} = \dfrac{\partial l}{\partial \hat{y}} \;\; \hat{y}(1-\hat{y})x_1$

Calculate Gradient (on one data point)

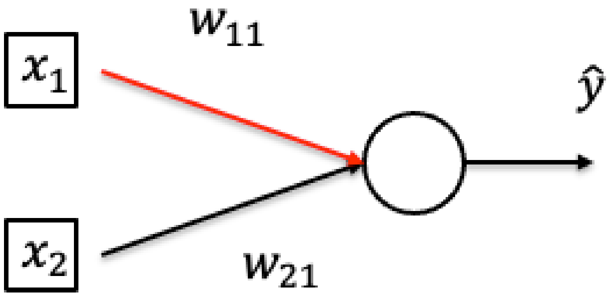


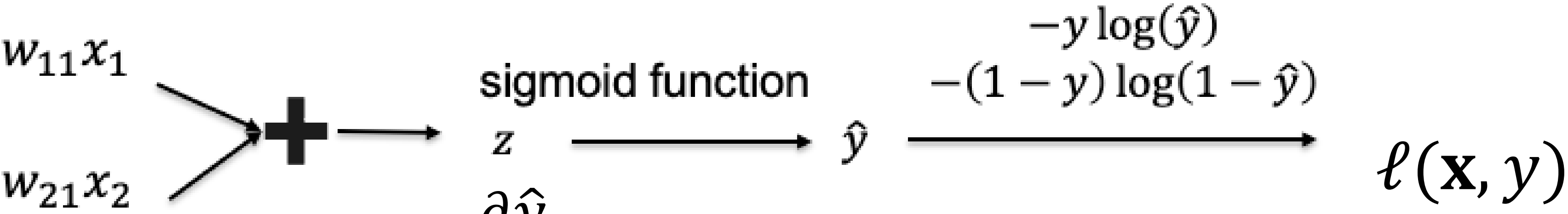


$$\frac{\partial \hat{y}}{\partial z} = \sigma'(z) = \sigma(z)(1-\sigma(z))$$

- By chain rule: $\dfrac{\partial l}{\partial w_{11}} = \left(\dfrac{1-y}{1-\hat{y}} - \dfrac{y}{\hat{y}}\right)\hat{y}(1-\hat{y})x_1$

Calculate Gradient (on one data point)

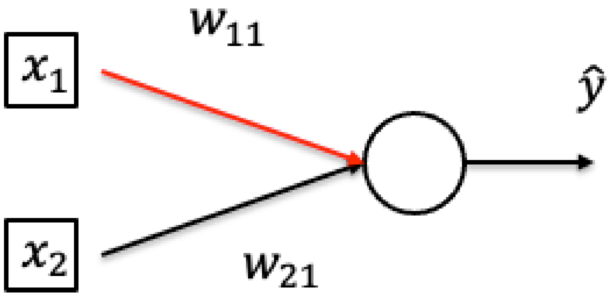


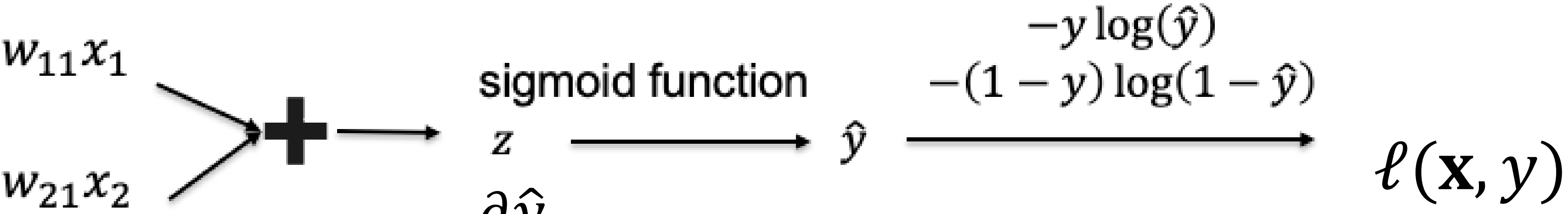


$$\frac{\partial \hat{y}}{\partial z} = \sigma'(z) = \sigma(z)(1-\sigma(z))$$

- By chain rule: $\frac{\partial l}{\partial w_{11}} = (\hat{y} - y)x_1$

Calculate Gradient (on one data point)

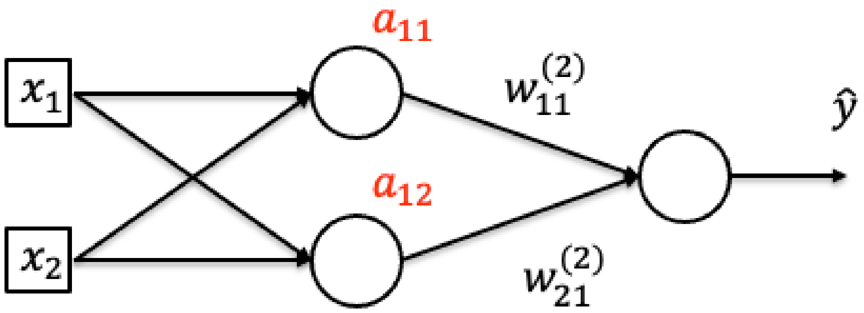




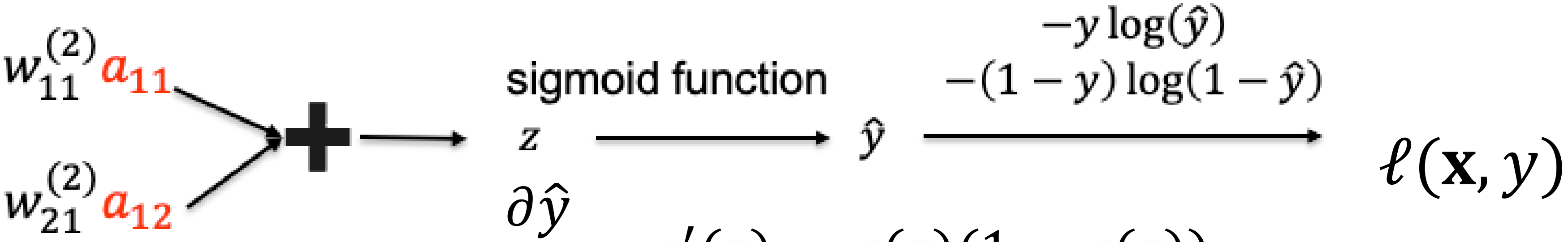
$$\frac{\partial \hat{y}}{\partial z} = \sigma'(z) = \sigma(z)(1-\sigma(z))$$

- By chain rule $\frac{\partial l}{\partial x_1} = \frac{\partial l}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} w_{11} = (\hat{y} - y)w_{11}$

Calculate Gradient (on one data point)



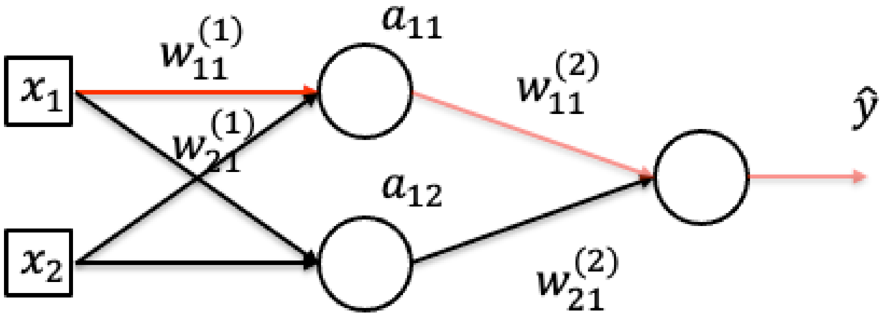
Make it deeper



$$\frac{\partial \hat{y}}{\partial z} = \sigma'(z) = \sigma(z)(1-\sigma(z))$$

- By chain ru $\dfrac{\partial l}{\partial a_{11}} = (\hat{y} - y)w_{11}^{(2)}, \ \dfrac{\partial l}{\partial a_{12}} = (\hat{y} - y)w_{21}^{(2)}$

Calculate Gradient (on one data point)

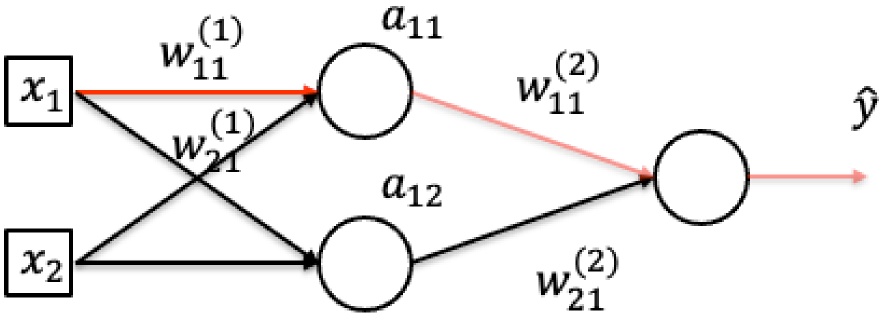


$w_{11}^{(1)} x_1$, $w_{21}^{(1)} x_2$ → + → z_{11} $\xrightarrow{\sigma(z_{11})}$ a_{11} → $l(x, y)$

$$\frac{\partial a_{11}}{\partial z_{11}} = \sigma'(z_{11}) \qquad \frac{\partial l}{\partial a_{11}} = (\hat{y} - y) w_{11}^{(2)}$$

- By chain rule $\frac{\partial l}{\partial w_{11}^{(1)}} = \frac{\partial l}{\partial a_{11}} \frac{\partial a_{11}}{\partial w_{11}^{(1)}} = (\hat{y} - y) w_{11}^{(2)} \frac{\partial a_{11}}{\partial w_{11}^{(1)}}$

Calculate Gradient (on one data point)

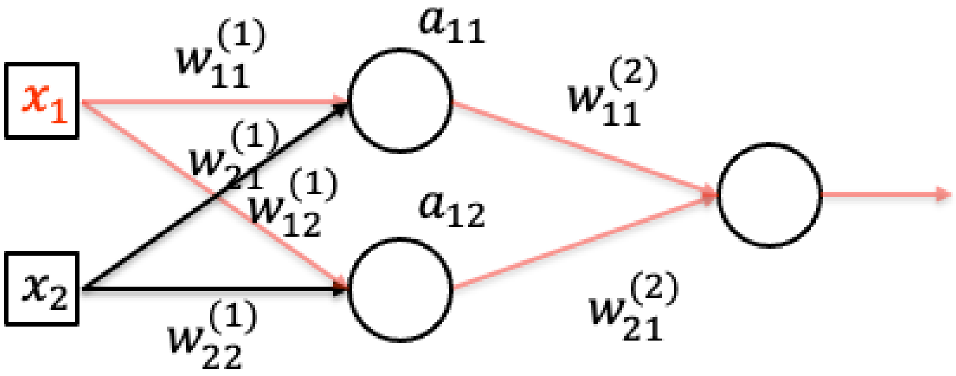


$w_{11}^{(1)} x_1$, $w_{21}^{(1)} x_2$ $\rightarrow$ $+$ $\rightarrow$ $z_{11} \xrightarrow{\sigma(z_{11})} a_{11} \longrightarrow l(\boldsymbol{x}, y)$

$$\frac{\partial a_{11}}{\partial z_{11}} = \sigma'(z_{11}) \qquad \frac{\partial l}{\partial a_{11}} = (\hat{y} - y) w_{11}^{(2)}$$

- By chain ru $\frac{\partial l}{\partial w_{11}^{(1)}} = \frac{\partial l}{\partial a_{11}} \frac{\partial a_{11}}{\partial w_{11}^{(1)}} = (\hat{y} - y) w_{11}^{(2)} a_{11}(1 - a_{11}) x_1$

Calculate Gradient (on one data point)



$w_{11}^{(1)} x_1$, $w_{21}^{(1)} x_2$ → + → z_{11} $\xrightarrow{\sigma(z_{11})}$ a_{11} → $l(x, y)$

$$\frac{\partial a_{11}}{\partial z_{11}} = \sigma'(z_{11}) \qquad \frac{\partial l}{\partial a_{11}} = (\hat{y} - y) w_{11}^{(2)}$$

- By chain rule: $\dfrac{\partial l}{\partial x_1} = \dfrac{\partial l}{\partial a_{11}} \dfrac{\partial a_{11}}{\partial x_1} + \dfrac{\partial l}{\partial a_{12}} \dfrac{\partial a_{12}}{\partial x_1}$