CS 360: Machine Learning

Prof. Sara Mathieson

Fall 2020

HAVERFORD
COLLEGE

Admin

- Office hours **today 4:30-6pm**
- **Lab 8** posted today, due Friday Nov 20
  - Keep working on getting logged in to lab machines
  - Can start Part 1 (data pre-processing)
- After Thanksgiving – **two options for capstone**
  - Midterm 2
  - Final project (posted soon)

Outline for November 10

- Ways to repair biased data algorithmically

- Validation best practices

- Introduction to neural networks

Outline for November 10

- Ways to repair biased data algorithmically
- Validation best practices
- Introduction to neural networks

How can we tell if an algorithm is biased?

D: dataset with attributes X, Y

* X is protected

* Y is unprotected (other features)

“Certifying and Removing Disparate Impact” Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, Suresh Venkatasubramanian

How can we tell if an algorithm is biased?

D: dataset with attributes X, Y

* X is protected
* Y is unprotected (other features)

Goal: determine outcome C (hired, admitted, etc)

“Certifying and Removing Disparate Impact” Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, Suresh Venkatasubramanian

How can we tell if an algorithm is biased?

D: dataset with attributes X, Y

* X is protected
* Y is unprotected (other features)

Goal: determine outcome C (hired, admitted, etc)

Direct discrimination: $C = f(X)$

* Female instrumentalist not hired for orchestra
* Some ethnic groups not allowed to eat at a restaurant

"Certifying and Removing Disparate Impact" Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, Suresh Venkatasubramanian

How can we tell if an algorithm is biased?

D: dataset with attributes X, Y

* X is protected
* Y is unprotected (other features)

Goal: determine outcome C (hired, admitted, etc)

Indirect discrimination: C = f(Y)

* but strong correlation between X and Y

* Ex: housing loans
* Ex: programming experience

"Certifying and Removing Disparate Impact" Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, Suresh Venkatasubramanian

Disparate Impact

X = protected attribute ← $\begin{cases} X = 0 & \text{minority group} \\ X = 1 & \text{majority group} \end{cases}$

Y = other attributes

C = binary outcome ← $\underbrace{\text{hired, admitted}}_{C=1}$, or $\underbrace{\text{not}}_{C=0}$

legal definition

$$P(C=1 \mid \underbrace{X=0}_{}) \le \underbrace{0.8}_{} P(C=1 \mid X=1)$$

example

40% women hired
60% men hired } years (?)

$$0.40 \le \underbrace{0.8(0.6)}_{0.48}$$

} $\boxed{\text{yes}}$

is disparate impact

Disparate Impact

C = outcome

Idea: if we can predict X from Y, there could be disparate impact

predictor "try" → $\boxed{f : Y \to X}$ ← could be more than one

Balanced Error Rate (BER), ε

$$\underbrace{\varepsilon = BER}_{\text{want: high!}} = \frac{P[\overbrace{f(Y)=0}^{\text{error}} \mid X=\overset{\text{majority}}{1}] + P[\overbrace{f(Y)=1}^{\text{error}} \mid X=\overset{\text{minority}}{0}]}{2}$$

threshold

max = 0.5

pred

	f	0	1
true	0	0.5	0.5
	1	0.5	0.5

$$\frac{0.5 + 0.5}{2} = 0.5$$

Disparate Impact

C = outcome

Idea: if we can predict X from Y, there could be disparate impact

predictor "try" → $\boxed{f : Y \rightarrow X}$ ← could be more than one

Balanced Error Rate (BER), ε

relationship between X & Y

$$\varepsilon = BER = \frac{P[\underbrace{f(Y)=0}_{\text{error}} \mid \overset{\text{majority}}{X=1}] + P[\underbrace{f(Y)=1 \mid X}_{\text{error}} = \overset{\text{minority}}{0}]}{2}$$

actually happened

outcome	$X=0$	$X=1$
$C=0$	a	b
$C=1$	c	d

$$\beta = \frac{c}{a+c}$$

$$\varepsilon' = \frac{1}{2} - \frac{\beta}{8}$$

if $\boxed{\varepsilon > \varepsilon'} \Rightarrow$ no disparate impact

Example of repair

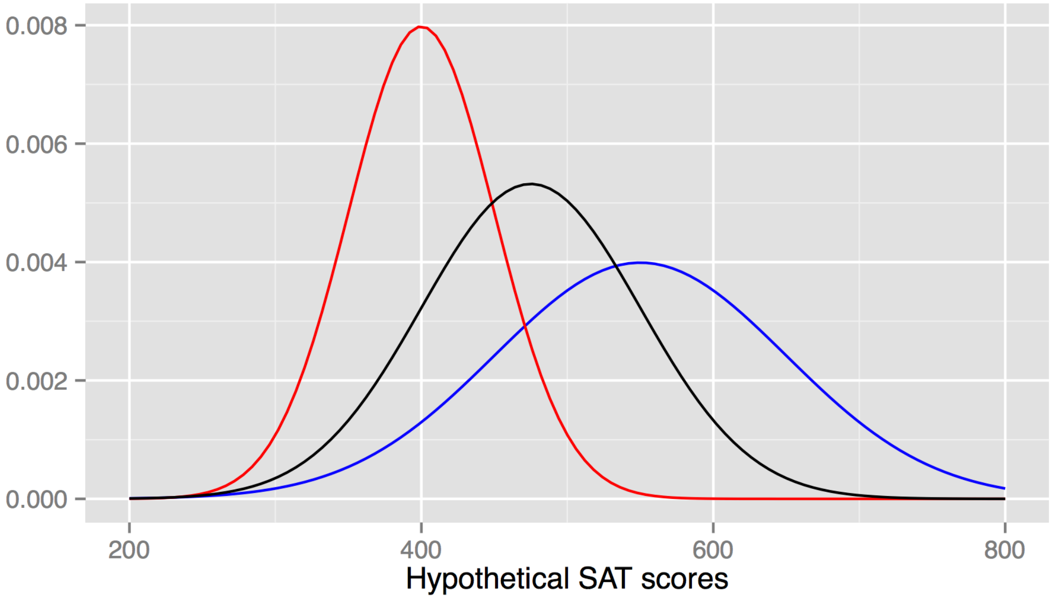


Figure 1: Consider the fake probability density functions shown here where the blue curve shows the distribution of SAT scores (Y) for $X =$ `female`, with $\mu = 550, \sigma = 100$, while the red curve shows the distribution of SAT scores for $X =$ `male`, with $\mu = 400, \sigma = 50$. The resulting fully repaired data is the distribution in black, with $\mu = 475, \sigma = 75$. Male students who originally had scores in the 95th percentile, i.e., had scores of 500, are given scores of 625 in the 95th percentile of the new distribution in $\bar{Y}$, while women with scores of 625 in $\bar{Y}$ originally had scores of 750.

“Certifying and Removing Disparate Impact” Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, Suresh Venkatasubramanian

Example of repair

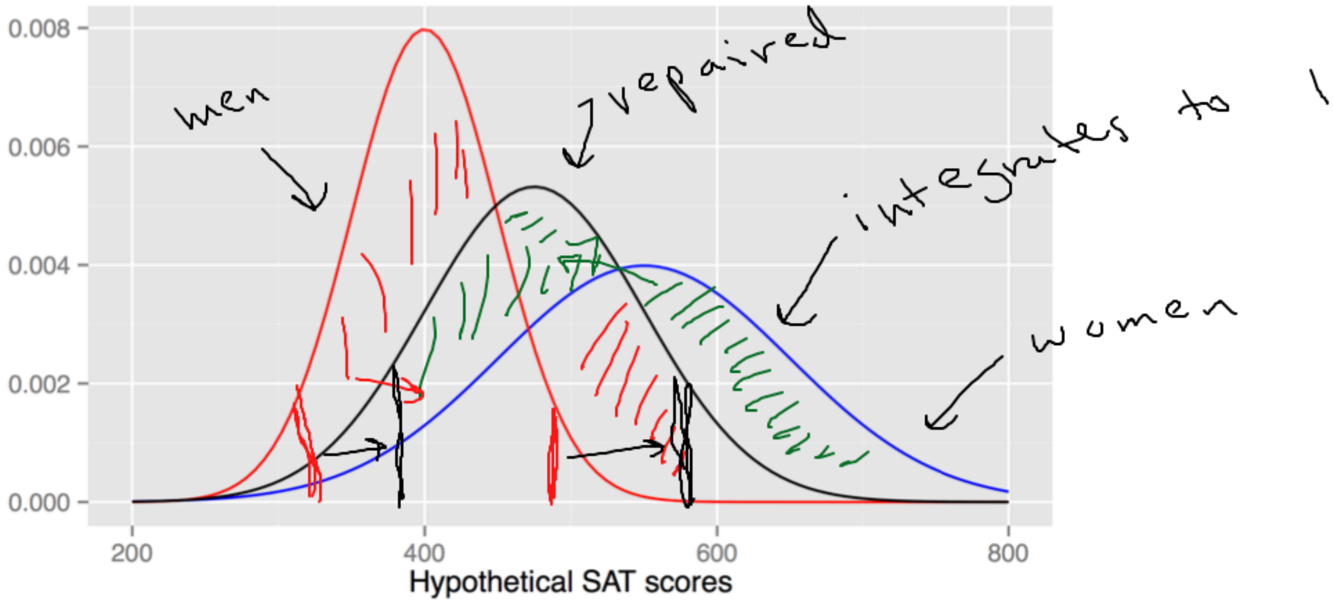


Figure 1: Consider the fake probability density functions shown here where the blue curve shows the distribution of SAT scores (Y) for $X = \texttt{female}$, with $\mu = 550, \sigma = 100$, while the red curve shows the distribution of SAT scores for $X = \texttt{male}$, with $\mu = 400, \sigma = 50$. The resulting fully repaired data is the distribution in black, with $\mu = 475, \sigma = 75$. Male students who originally had scores in the 95th percentile, i.e., had scores of 500, are given scores of 625 in the 95th percentile of the new distribution in $\bar{Y}$, while women with scores of 625 in $\bar{Y}$ originally had scores of 750.

"Certifying and Removing Disparate Impact" Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, Suresh Venkatasubramanian

Discussion Questions

1) What are our responsibilities as engineers to ensure that our algorithms are fair?

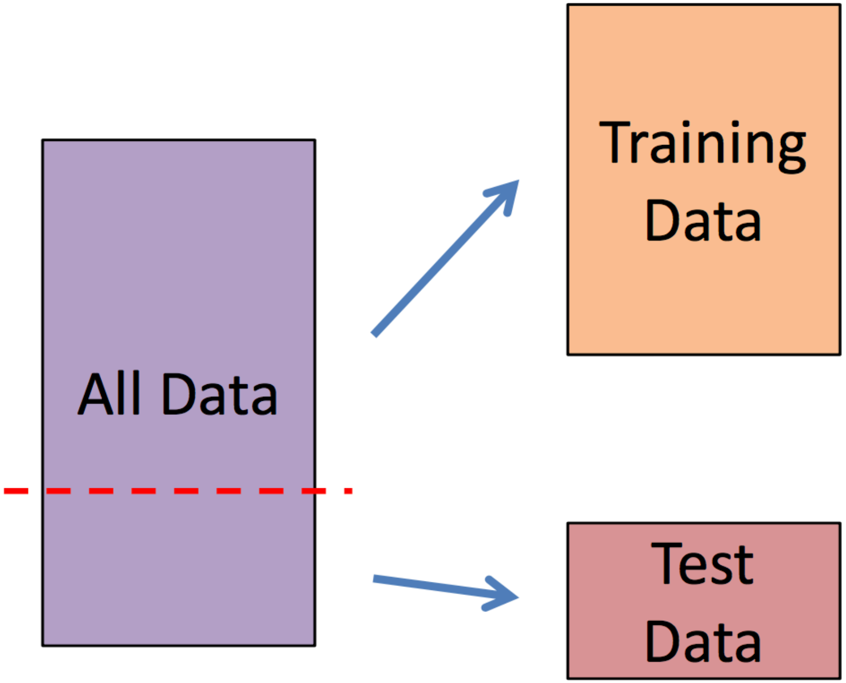
2) How would you handle a situation where you felt you didn’t have enough data (or the right data) necessary to build your algorithm?

3) How would you try to detect if your algorithm was making biased decisions during deployment?

Outline for November 10

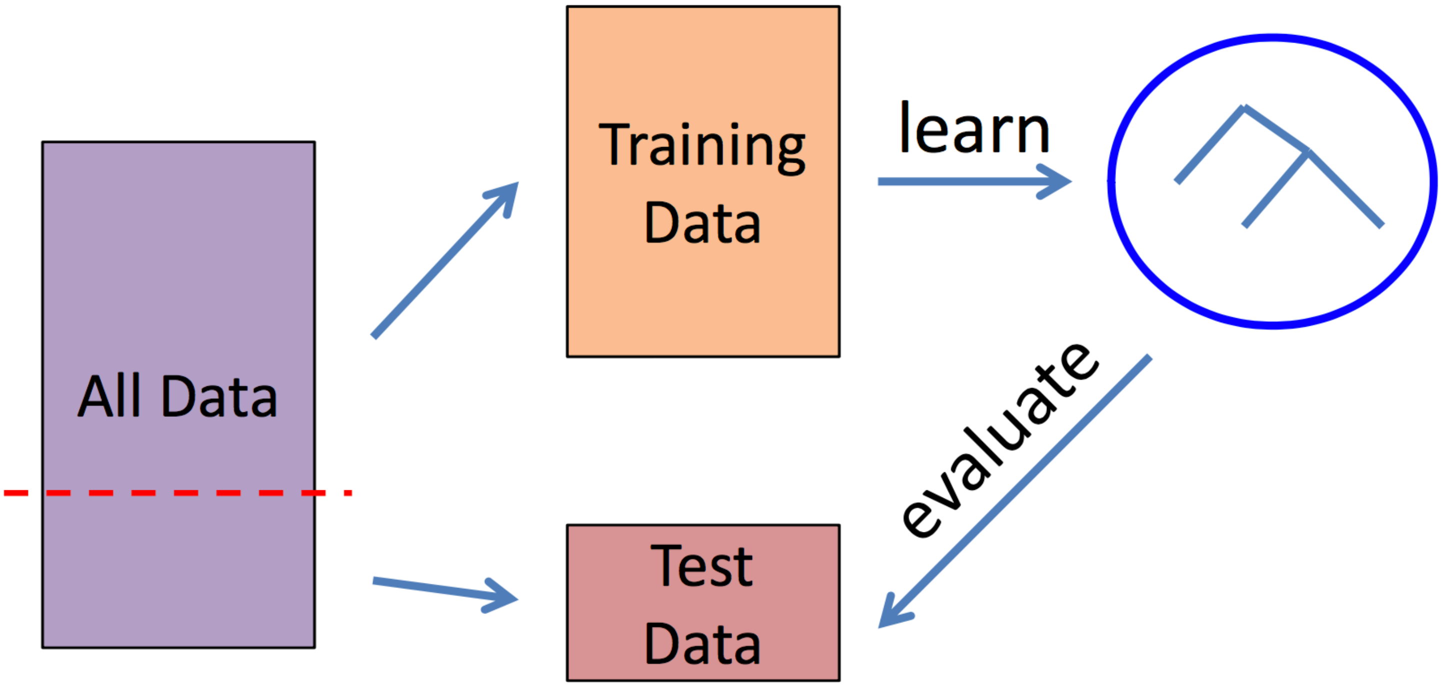
- Ways to repair biased data algorithmically
- Validation best practices
- Introduction to neural networks

Evaluation in Practice



Modified from Jessica Wu

Evaluation in Practice


All Data

Training Data

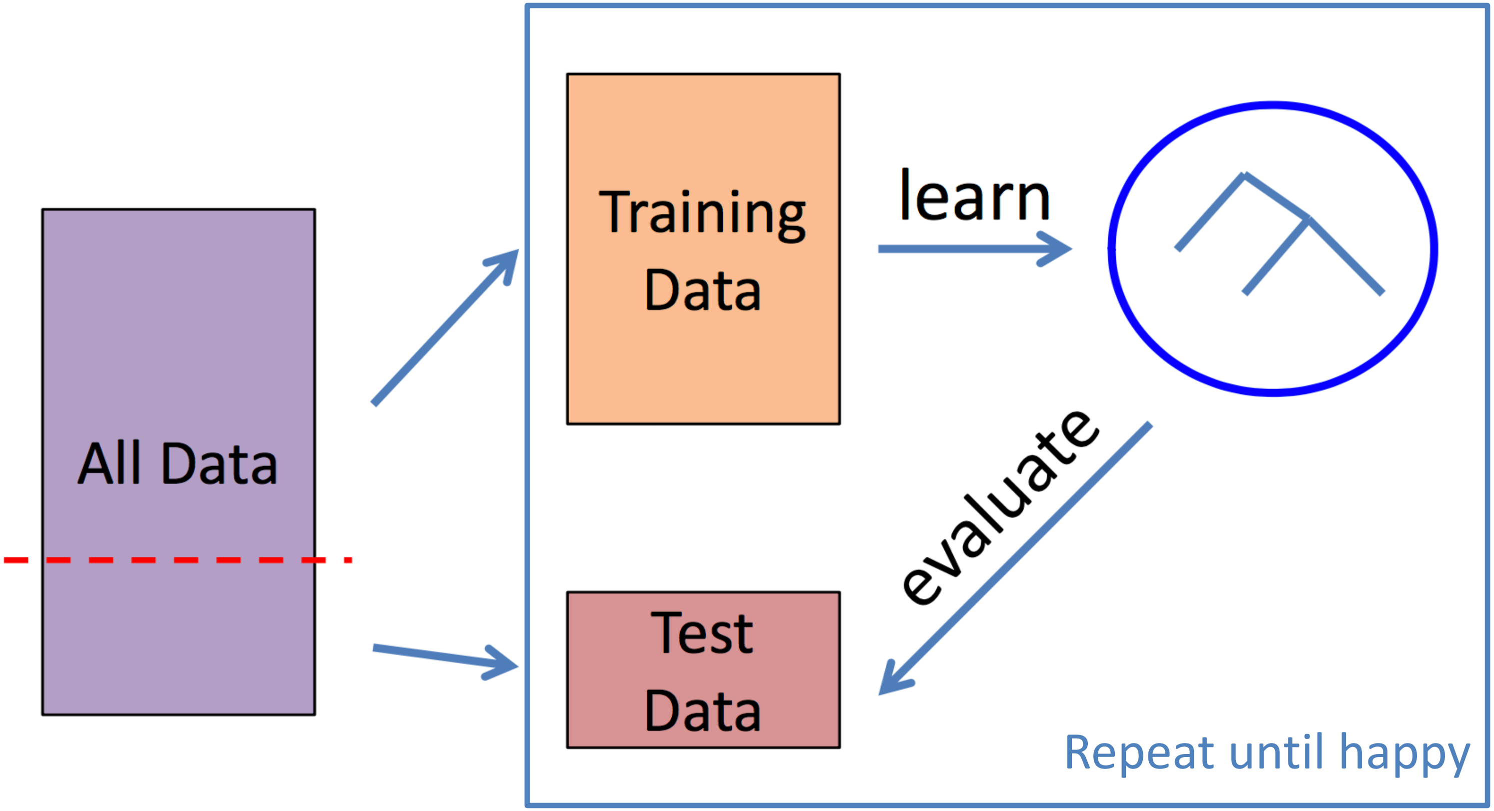
Test Data

learn

evaluate

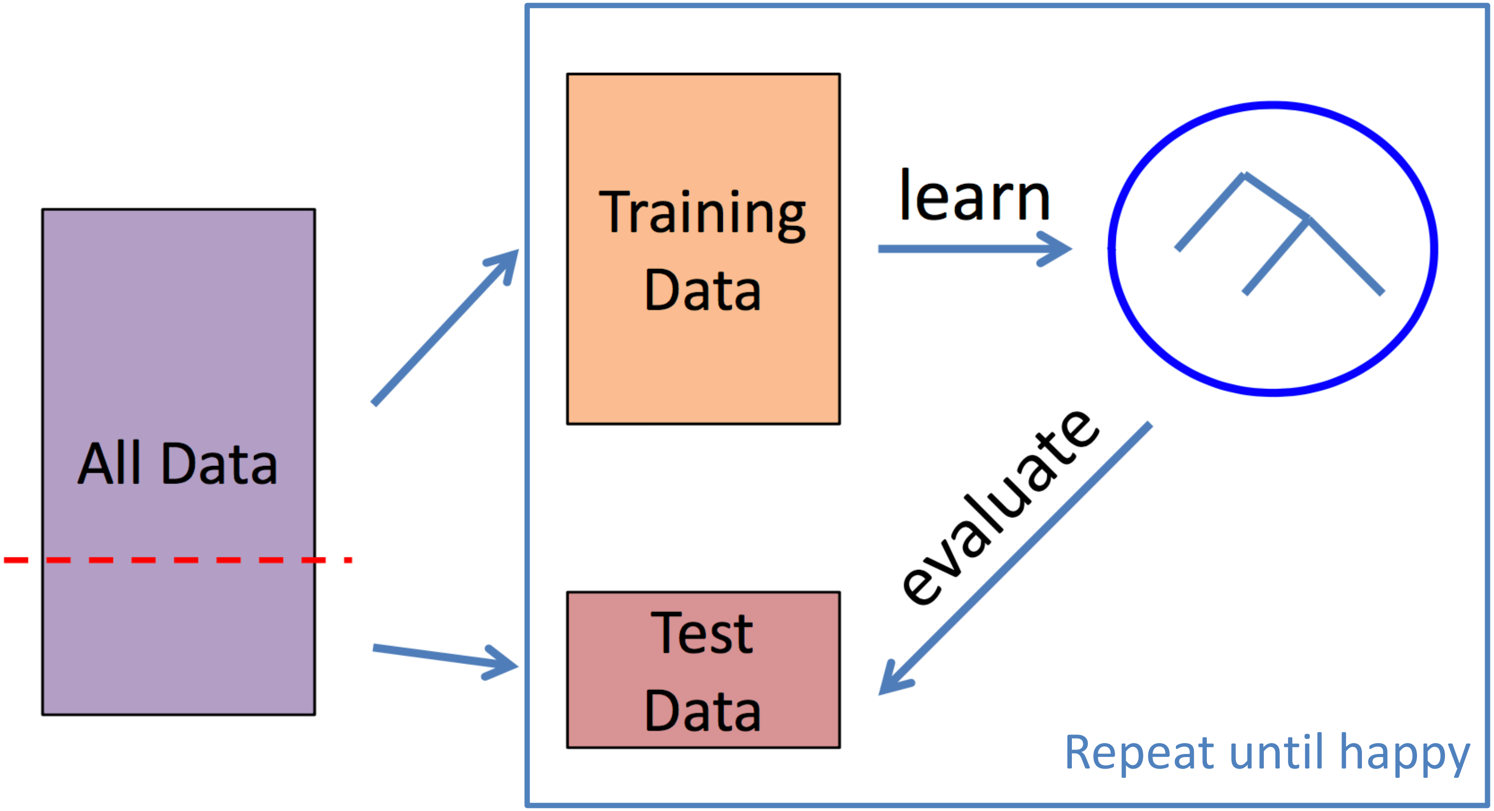
Modified from Jessica Wu

Evaluation in Practice



Modified from Jessica Wu

Evaluation in Practice

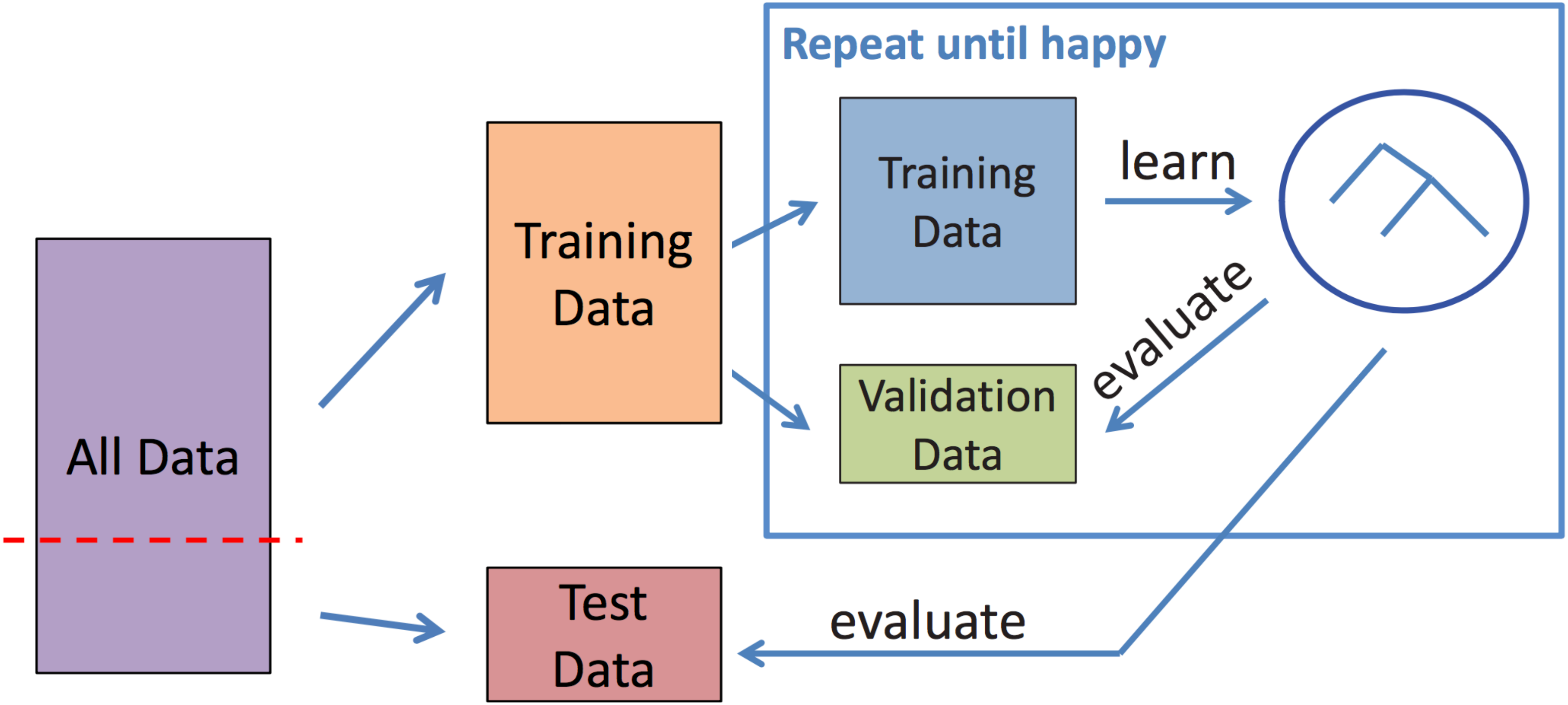


Repeat until happy

NO! Using test data as part of the model selection process

Modified from Jessica Wu

Better: use a *validation* dataset



Repeat until happy

All Data

Training Data

Test Data

Training Data

Validation Data

learn

evaluate

evaluate

Modified from Jessica Wu

Outline for November 10

- Ways to repair biased data algorithmically

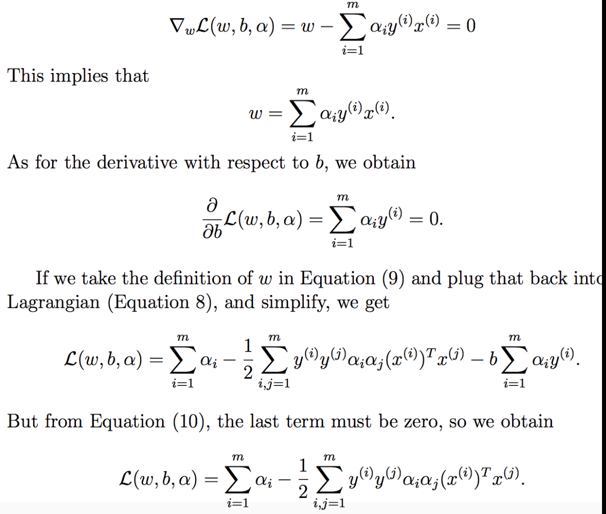
- Validation best practices

- Introduction to neural networks

MACHINE LEARNING

Adapted from: “Know Your Meme”

What society thinks I do



$$\nabla_w \mathcal{L}(w,b,\alpha) = w - \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} = 0$$

This implies that

$$w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}.$$

As for the derivative with respect to b, we obtain

$$\frac{\partial}{\partial b}\mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m} \alpha_i y^{(i)} = 0.$$

If we take the definition of w in Equation (9) and plug that back int Lagrangian (Equation 8), and simplify, we get

$$\mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j (x^{(i)})^T x^{(j)} - b\sum_{i=1}^{m} \alpha_i y^{(i)}.$$

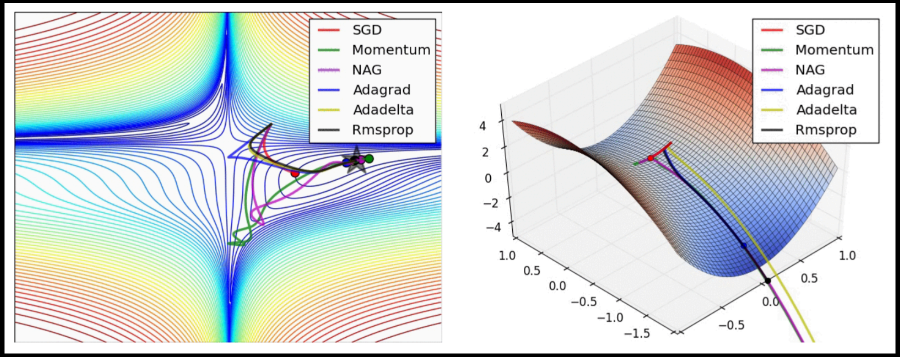
But from Equation (10), the last term must be zero, so we obtain

$$\mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j (x^{(i)})^T x^{(j)}.$$

What my boss thinks I do

What other computer scientists think I do

What mathematicians think I do



What I think I do

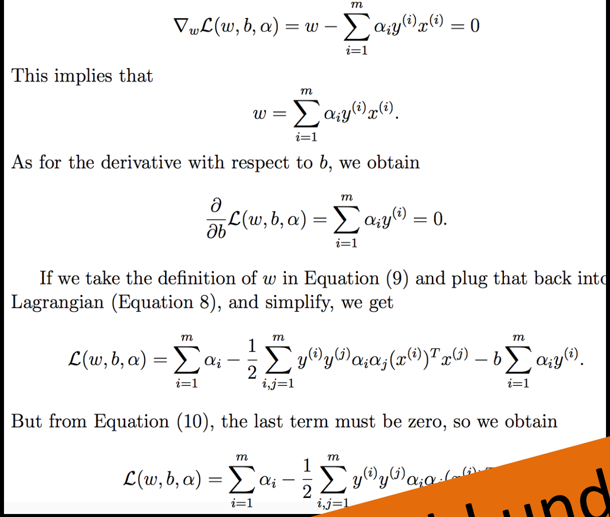
```
>>> from sklearn import svm
>>> import tensorflow as tf
```

What I really do

Adapted from: “Know Your Meme”

MACHINE LEARNING

What society thinks I do



$$\nabla_w \mathcal{L}(w,b,\alpha) = w - \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} = 0$$

This implies that

$$w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}.$$

As for the derivative with respect to b, we obtain

$$\frac{\partial}{\partial b}\mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m} \alpha_i y^{(i)} = 0.$$

If we take the definition of w in Equation (9) and plug that back into Lagrangian (Equation 8), and simplify, we get

$$\mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j (x^{(i)})^T x^{(j)} - b\sum_{i=1}^{m} \alpha_i y^{(i)}.$$

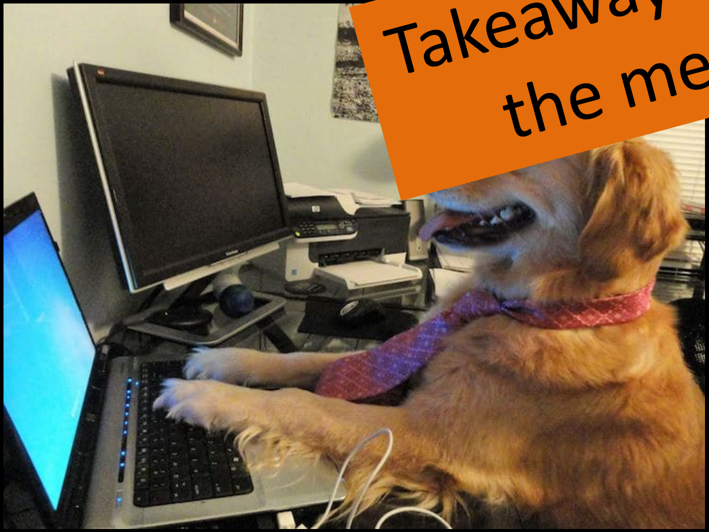
But from Equation (10), the last term must be zero, so we obtain

$$\mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j$$

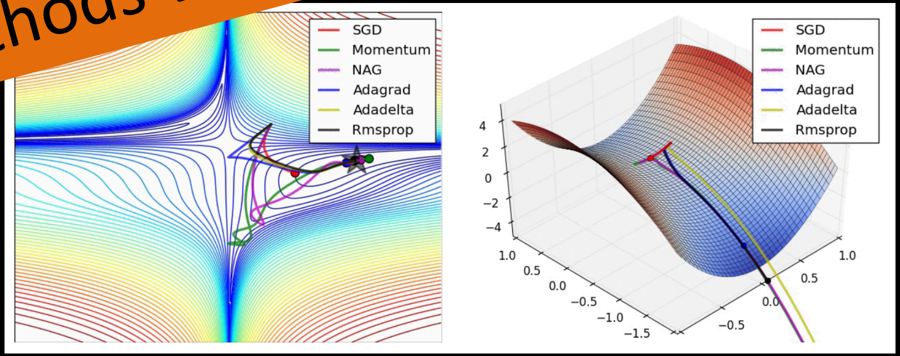
Wh

t other computer scientists think I do

Takeaway: we should understand the methods we are using!



What mathematicians think I do



What I think I do

```
>>> from sklearn import svm
>>> import tensorflow as tf
```

What I really do

Biological Inspiration

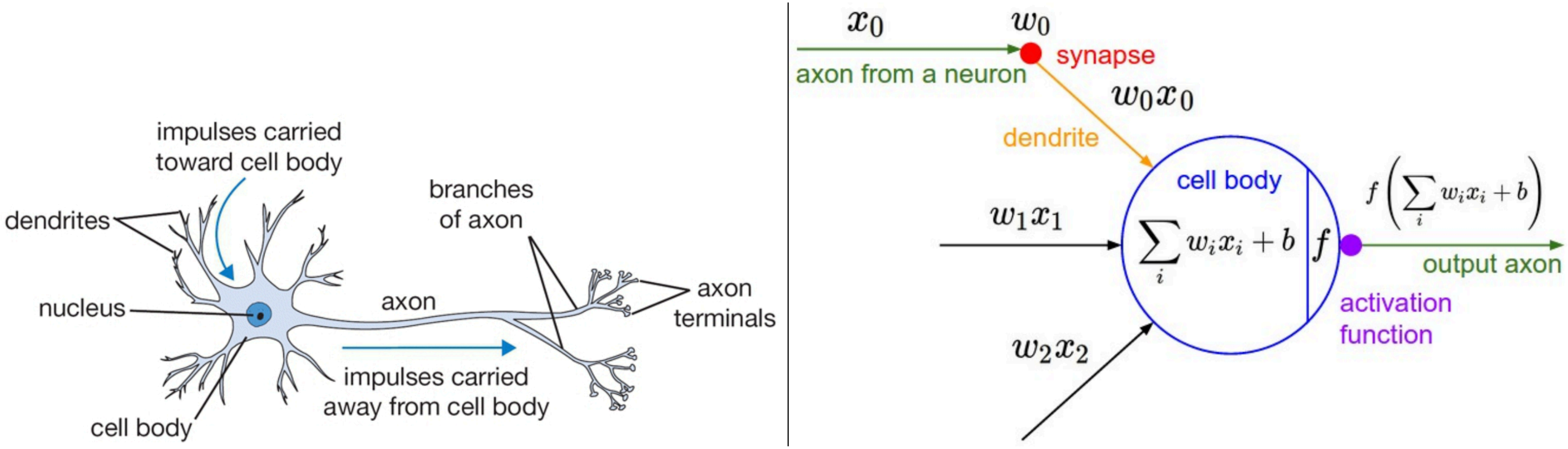
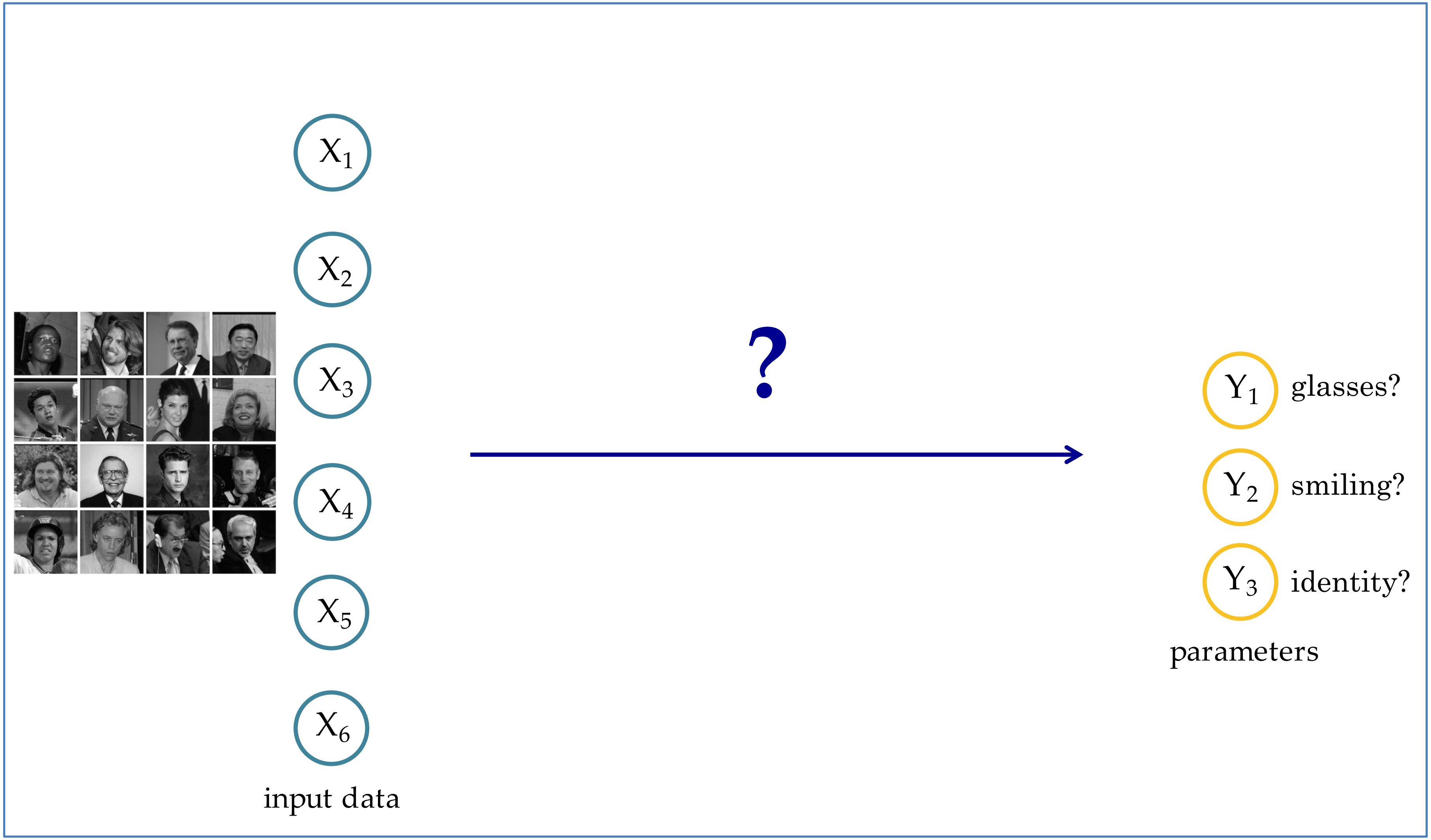


Figure: Stanford CS231n http://cs231n.github.io/neural-networks-1/

Goal: learn from complicated inputs



X_1

X_2

X_3

X_4

X_5

X_6

input data

?

Y_1 glasses?

Y_2 smiling?

Y_3 identity?

parameters

Image: Labeled Faces in the Wild (UMass)

Idea: transform data into lower dimension

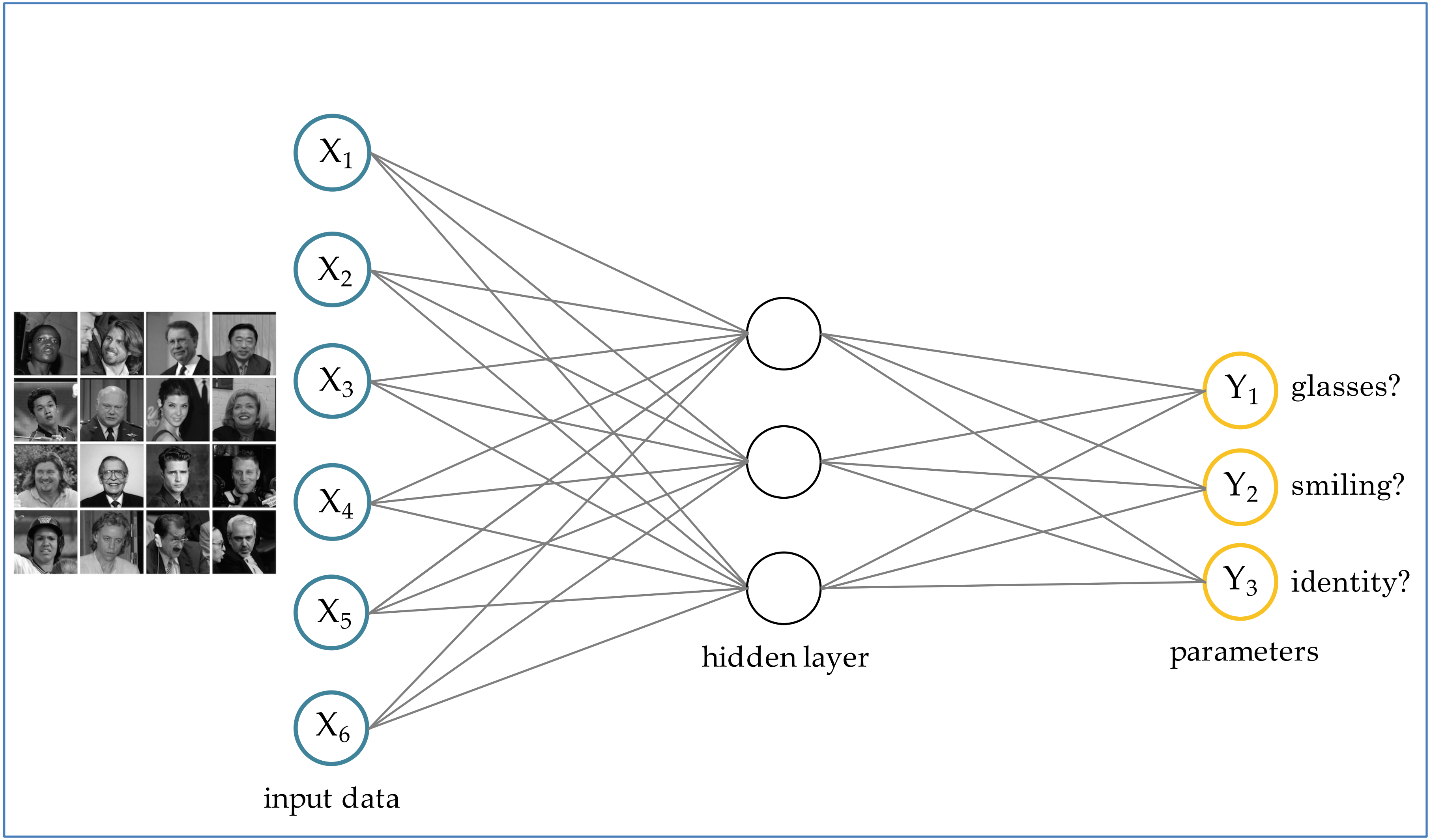


Image: Labeled Faces in the Wild (UMass)

Multi-layer networks = “deep learning”

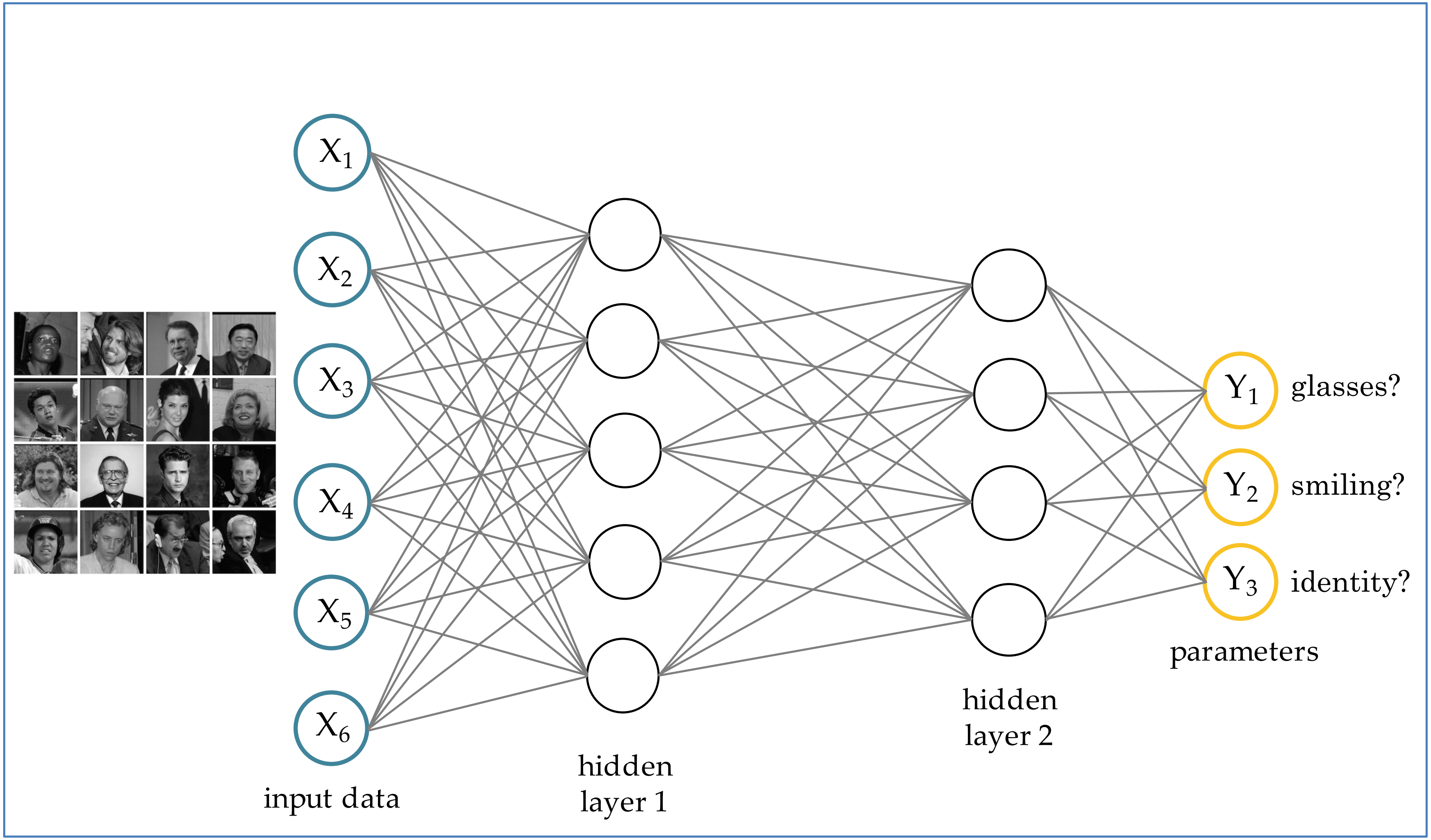
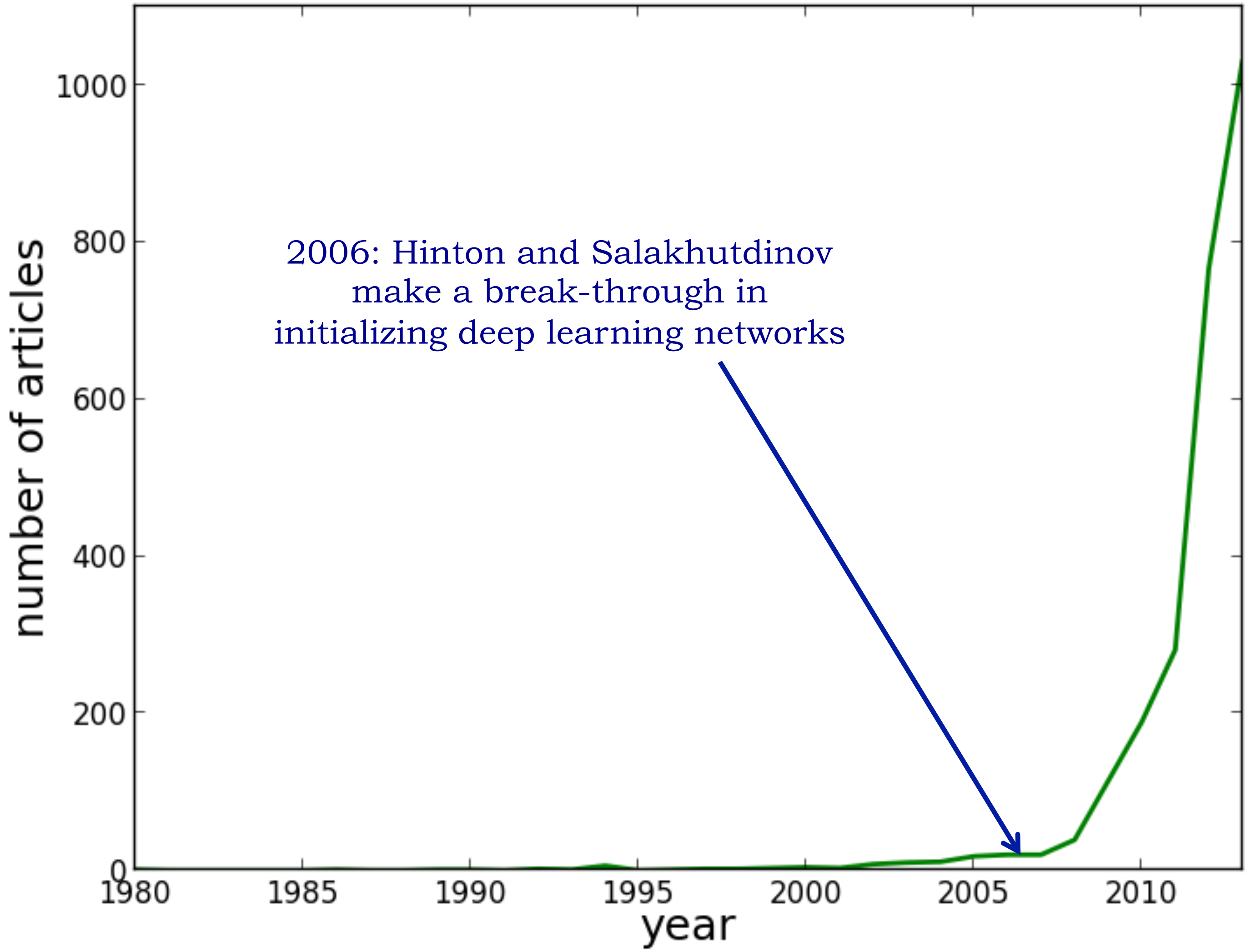


Image: Labeled Faces in the Wild (UMass)

History of Neural Networks

- Perceptron can be interpreted as a simple neural network
- Misconceptions about the weaknesses of perceptrons contributed to declining funding for NN research
- Difficulty of training multi-layer NNs contributed to second setback
- Mid 2000’s: breakthroughs in NN training contribute to rise of “deep learning”

Number of papers that mention “deep learning” over time



2006: Hinton and Salakhutdinov make a break-through in initializing deep learning networks

Big picture for today

- Neural networks can approximate any function!

Big picture for today

- Neural networks can approximate any function!
- For our purposes in ML, we want to use them to approximate a function from our inputs to our outputs

Big picture for today

- Neural networks can approximate any function!

- For our purposes in ML, we want to use them to approximate a function from our inputs to our outputs

- We will train our network by asking it to minimize the loss between its output and the true output

Big picture for today

- Neural networks can approximate any function!
- For our purposes in ML, we want to use them to approximate a function from our inputs to our outputs
- We will train our network by asking it to minimize the loss between its output and the true output
- We will use SGD-like approaches to minimize loss