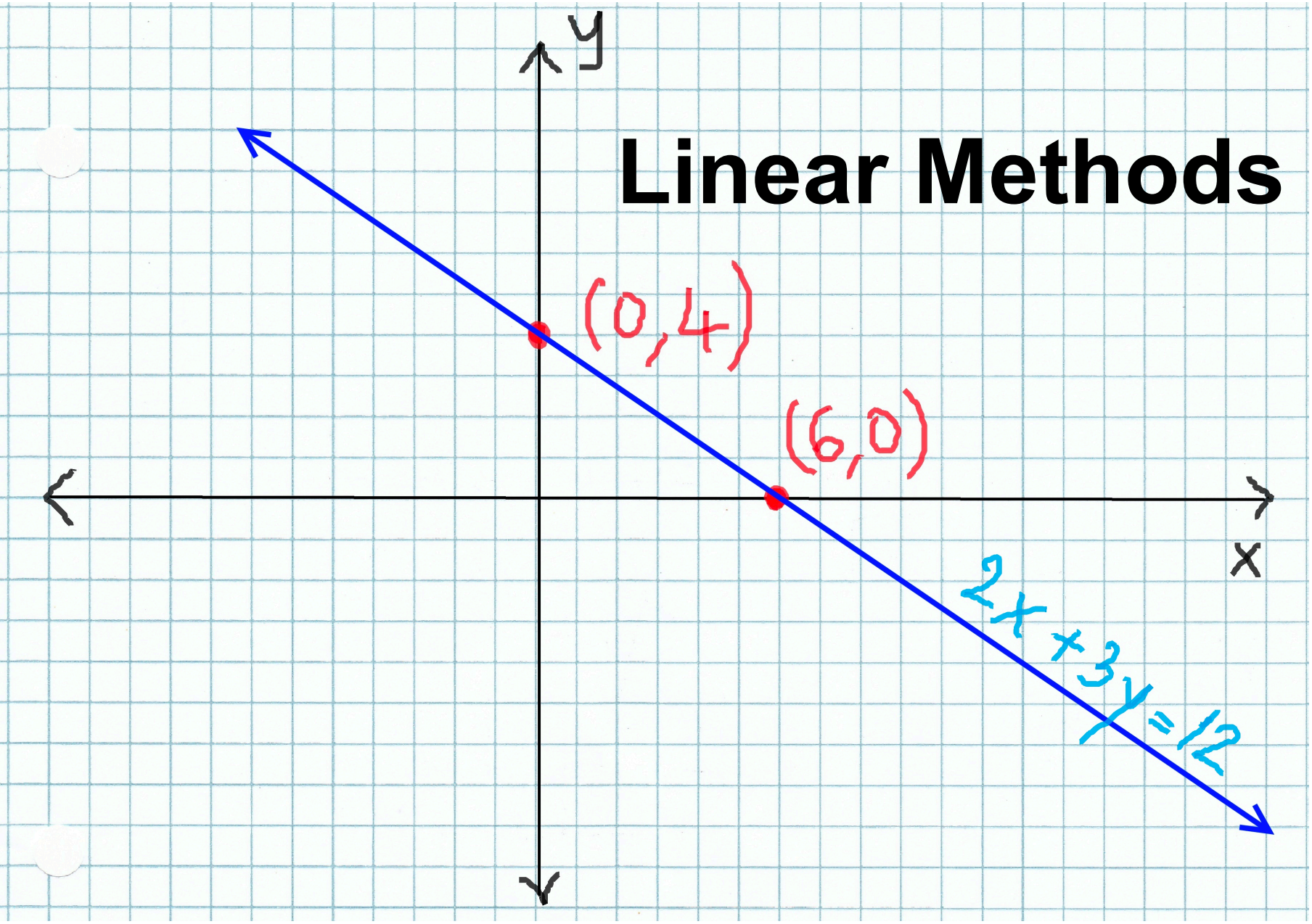
Introduction to (Deep) Learning

4. Linear Regression, Basic Optimization

slides adapted from courses.d2l.ai/berkeley-stat-157


Linear Methods
y
x
(0,4)
(6,0)
2x + 3y = 12

House Buying 101

- Pick a house, take a tour, and read facts
- Estimate its price, bid



Listing price from agent

$5,498,000 Price | **7** Beds | **5** Baths | **4,865** Sq. Ft. $1130 / Sq. Ft.

Redfin Estimate: $5,390,037 On Redfin: 15 days

Predicted sale price

slides adapted from courses [illegible] at-15

Virtual Tour
- Branded Virtual Tour
- Virtual Tour (External Link)

Parking Information
- Garage (Minimum): 2
- Garage (Maximum): 2
- Parking Description: Attached Garage, On Street
- Garage Spaces: 2

Multi-Unit Information
- # of Stories: 2

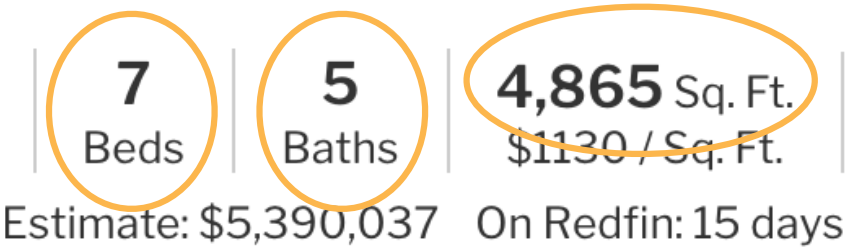
School Information
- Elementary School: El Carmelo El
- Elementary School District: Palo A
- Middle School: Jane Lathrop Stan
- High School: Palo Alto High
- High School District: Palo Alto Un

Interior Features

Bedroom Information
- # of Bedrooms (Minimum): 7
- # of Bedrooms (Maximum): 7

- Kitchen Description: Countertop (Dishwasher, Garbage Disposal, Ho Island with Sink, Microwave, Over

A Simplified Model



- Assumption 1
  The key factors impacting the prices are
  #Beds, #Baths, Living Sqft, denoted by x_1, x_2, x_3
- Assumption 2
  The sale price is a weighted sum over the key factors

$$y = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

  Weights and bias are determined later

slides adapted from courses.d2l.ai/berkeley-stat-157

Linear Model

- Given *n*-dimensional inputs $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$
- Linear model has a *n*-dimensional weight and a bias

$$\mathbf{w} = [w_1, w_2, \ldots, w_n]^T, \quad b$$

- The output is a weighted sum of the inputs

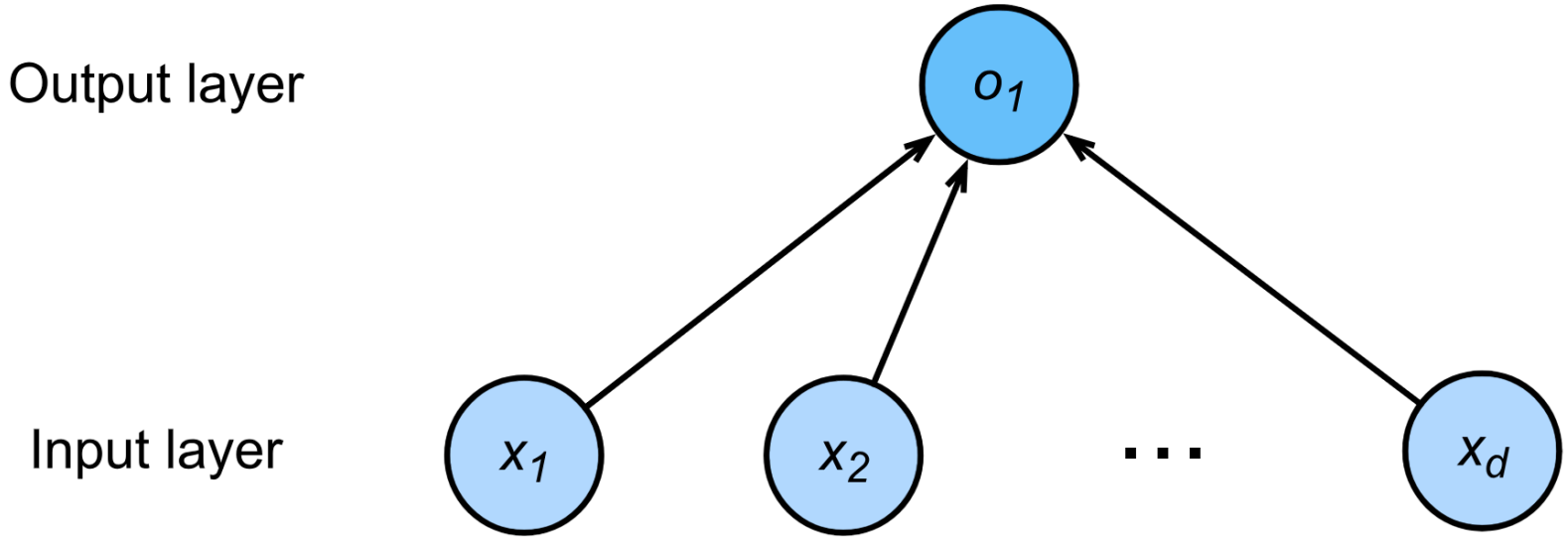
$$y = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b$$

Vectorized version

$$y = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

slides adapted from courses.d2l.ai/berkeley-stat-157

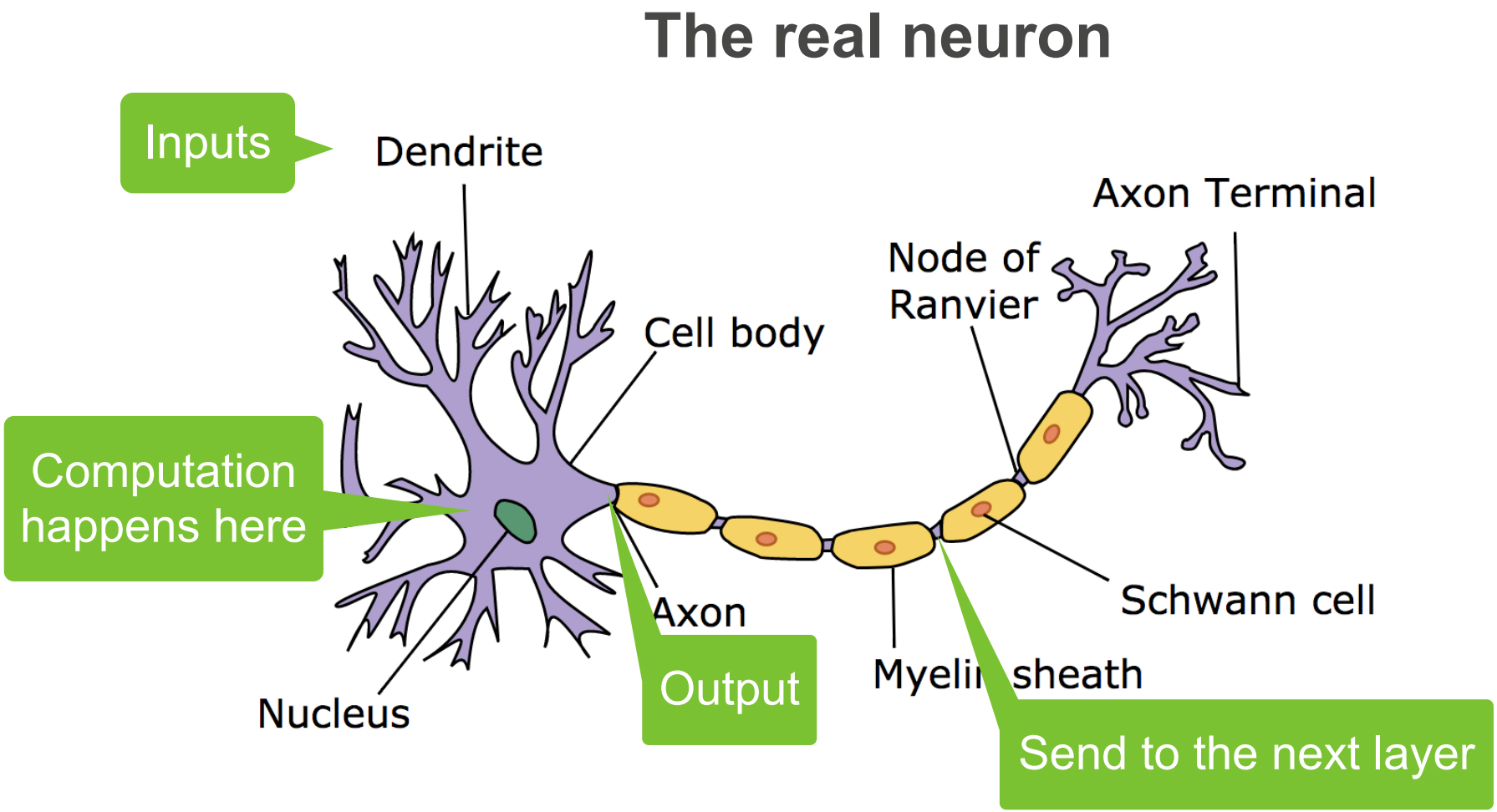
Linear Model as a Single-layer Neural Network



We can stack multiple layers to get deep neural networks

slides adapted from courses.d2l.ai/berkeley-stat-157

Neural Networks Derive from Neuroscience



slides adapted from courses.d2l.ai/berkeley-stat-157

Measure Estimation Quality

- Compare the true value vs the estimated value
  Real sale price vs estimated house price
- Let y the true value, and $\hat{y}$ the estimated value, we can compare the loss

$$\ell(y, \hat{y}) = \left(y - \hat{y}\right)^2$$

  It is called squared loss

slides adapted from courses.d2l.ai/berkeley-stat-157

Training Data

- Collect multiple data points to fit parameters
  Houses sold in the last 6 months
- It is called the training data
- The more the better
- Assume *n* examples

$$\mathbf{X} = \left[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\right]^T \quad \mathbf{y} = \left[y_0, y_1, \ldots, y_n\right]^T$$

slides adapted from courses.d2l.ai/berkeley-stat-157

Learn Parameters

- Training loss

$$\ell(\mathbf{X}, \mathbf{y}, \mathbf{w}, b) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \langle \mathbf{x}_i, \mathbf{w} \rangle - b\right)^2 = \frac{1}{n} \left\| \mathbf{y} - \mathbf{X}\mathbf{w} - b \right\|^2$$

- Minimize loss to learn parameters

$$\mathbf{w}^*, \mathbf{b}^* = \arg\min_{\mathbf{w}, b} \ell(\mathbf{X}, \mathbf{y}, \mathbf{w}, b)$$

slides adapted from courses.d2l.ai/berkeley-stat-157

Closed-form Solution

- Add bias into weights by $\mathbf{X} \leftarrow [\mathbf{X}, \mathbf{1}] \quad \mathbf{w} \leftarrow \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}$

$$\ell(\mathbf{X}, \mathbf{y}, \mathbf{w}) = \frac{1}{n} \left\| \mathbf{y} - \mathbf{X}\mathbf{w} \right\|^2 \qquad \frac{\partial}{\partial \mathbf{w}} \ell(\mathbf{X}, \mathbf{y}, \mathbf{w}) = \frac{2}{n} \left(\mathbf{y} - \mathbf{X}\mathbf{w} \right)^T \mathbf{X}$$

- Loss is convex, so the optimal solutions satisfies

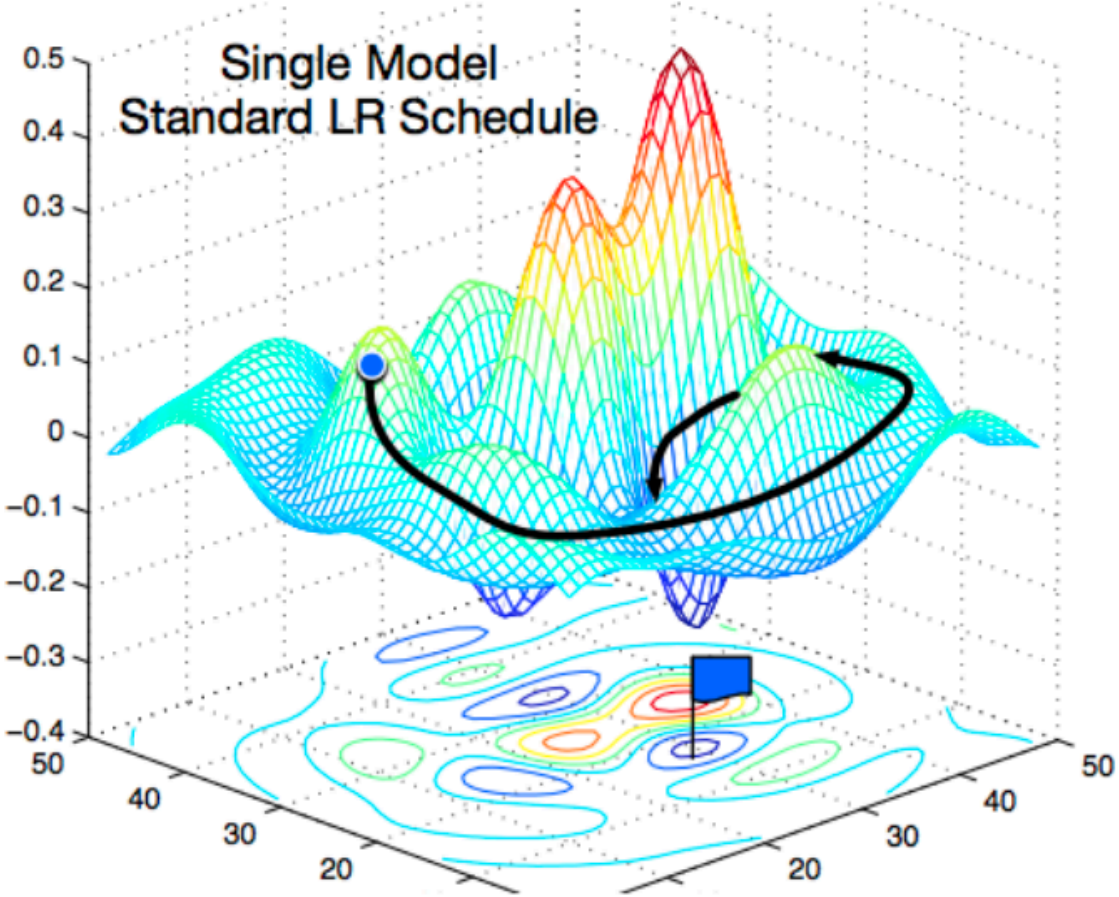
$$\frac{\partial}{\partial \mathbf{w}} \ell(\mathbf{X}, \mathbf{y}, \mathbf{w}) = 0$$

$$\Leftrightarrow \frac{2}{n} \left(\mathbf{y} - \mathbf{X}\mathbf{w} \right)^T \mathbf{X} = 0$$

$$\Leftrightarrow \mathbf{w}^* = \left(\mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{X}\mathbf{y}$$

slides adapted from courses.d2l.ai/berkeley-stat-157

Basic Optimization

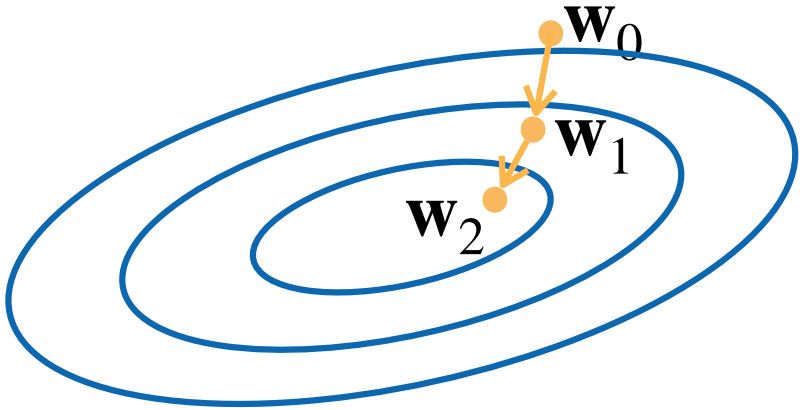


courses.d2l.ai/berkeley-stat-157

Gradient Descent

- Choose a staring point $\mathbf{w}_0$
- Repeat to update the weight *t=1,2,3*

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta \frac{\partial \ell}{\partial \mathbf{w}_{t-1}}$$

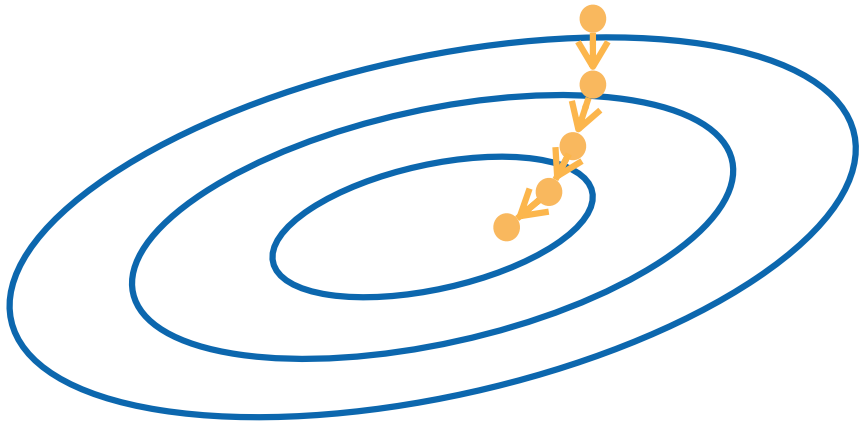


  - Gradient: a direction that increases the value
  - Learning rate: a hyper-parameter specifies the step length

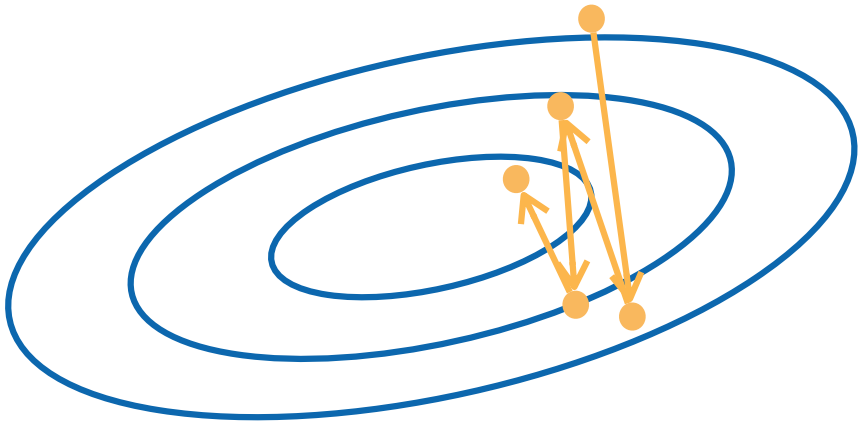
slides adapted from courses.d2l.ai/berkeley-stat-157

Choose a Learning Rate

Not too small



Not too big



slides adapted from courses.d2l.ai/berkeley-stat-157

Mini-batch Stochastic Gradient Descent (SGD)

- Computing the gradient over the whole training data is too expensive
  - Takes minutes to hours for DNN models
- Randomly sample *b* examples $i_1, i_2, \ldots, i_b$ to approximate the loss

$$\frac{1}{b} \sum_{i \in I_b} \ell(\mathbf{x}_i, y_i, \mathbf{w})$$

  - *b* is the batch size, another important hyper-parameters

slides adapted from courses.d2l.ai/berkeley-stat-157

Choose a Batch Size

Not too small

Workload is too small, hard to fully utilize computation resources

Not too big

Memory issue
Waste computation, e.g. when all x_i are identical

slides adapted from courses.d2l.ai/berkeley-stat-157

Introduction to Deep Learning

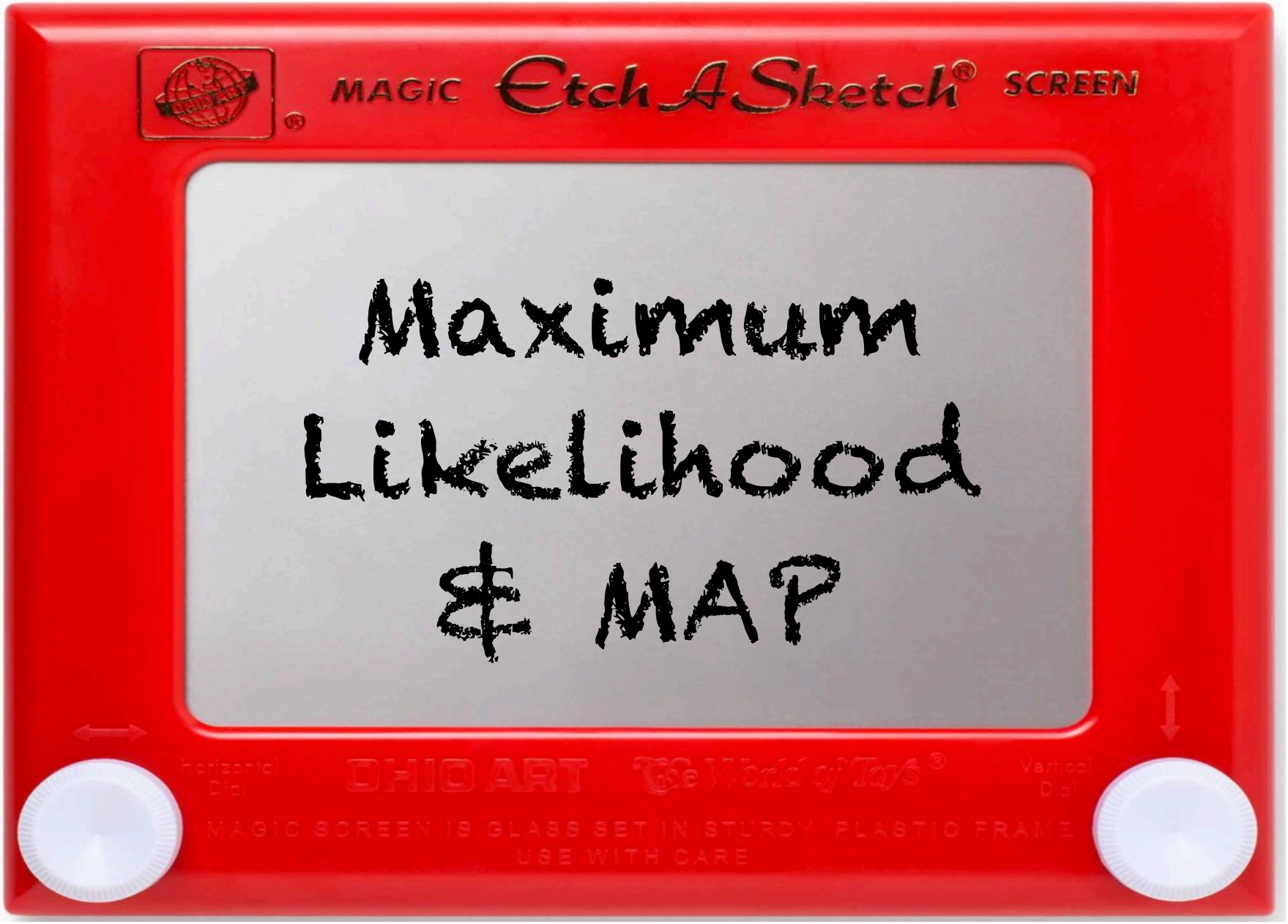
5. Maximum Likelihood and Logistic Regression

slides adapted from courses.d2l.ai/berkeley-stat-157

Outline

- **Maximum Likelihood**
  - Gauss and means
  - More loss functions (l_1 loss, trimmed mean)
  - Regression revisited
- **Classification**
  - Computing discrete probabilities
  - Likelihood and loss functions
- **Information Theory**

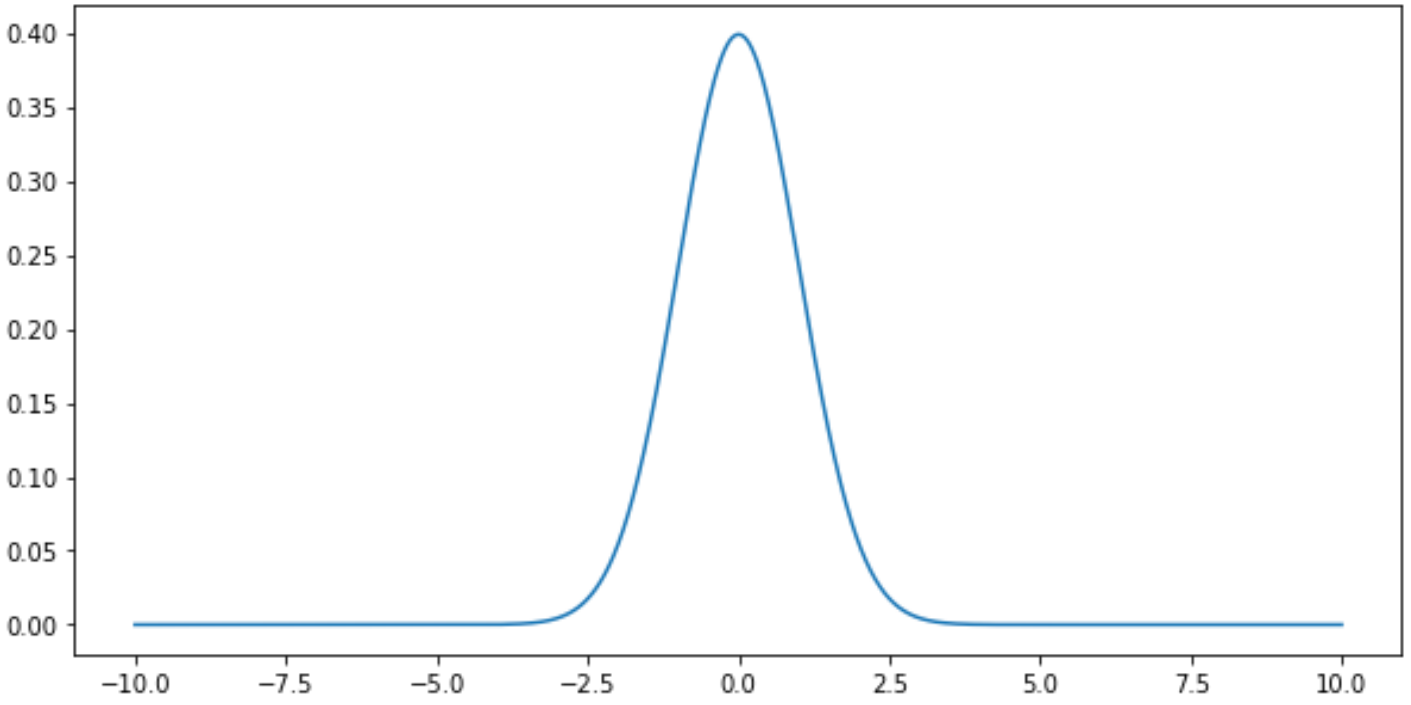
adapted from courses.d2l.ai/berkeley-stat-157


MAGIC Etch A Sketch® SCREEN
Maximum
Likelihood
& MAP
DHIO ART
MAGIC SCREEN IS GLASS SET IN STURDY PLASTIC FRAME
USE WITH CARE

adapted

Normal Distribution

Density $$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



adapted from courses.d2l.ai/berkeley-stat-157

Estimating the parameters in a Gaussian

- **Mean**

$$\mu = \mathbf{E}[x] \text{ hence } \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- **Variance**

$$\sigma^2 = \mathbf{E}\left[(x-\mu)^2\right] \text{ hence } \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \hat{\mu})^2$$

Why?

adapted from courses.d2l.ai/berkeley-stat-157

Likelihood

- Observe some data $X = \{x_1, \ldots x_n\}$
- Assume that the data is drawn from a Gaussian

$$p(X; \mu, \sigma^2) = \prod_{i=1}^{n} p(x_i; \mu, \sigma^2) = \prod_{i=1}^{n} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right)$$

- **Fitting parameters is maximizing** $p(X; \mu, \sigma^2)$ **wrt.** μ, σ^2 (maximize likelihood that data was generated by model)
- **Practical simplification**

$$\underset{\mu,\sigma^2}{\text{maximize}}\, p(X; \mu, \sigma^2) \iff \underset{\mu,\sigma^2}{\text{minimize}} - \log p(X; \mu, \sigma^2)$$

adapted from courses.d2l.ai/berkeley-stat-157

Maximum Likelihood

- Estimate parameters by finding ones that explain the data

$$\underset{\mu,\sigma^2}{\text{minimize}} \; -\log p(X;\mu,\sigma^2)$$

- **Decompose likelihood**

$$-\log p(X;\mu,\sigma^2) = \sum_{i=1}^{n} \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}(x_i-\mu)^2 = \frac{n}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

Minimized for $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$

adapted from courses.d2l.ai/berkeley-stat-157

Maximum Likelihood

- Estimating the variance

$$\frac{n}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

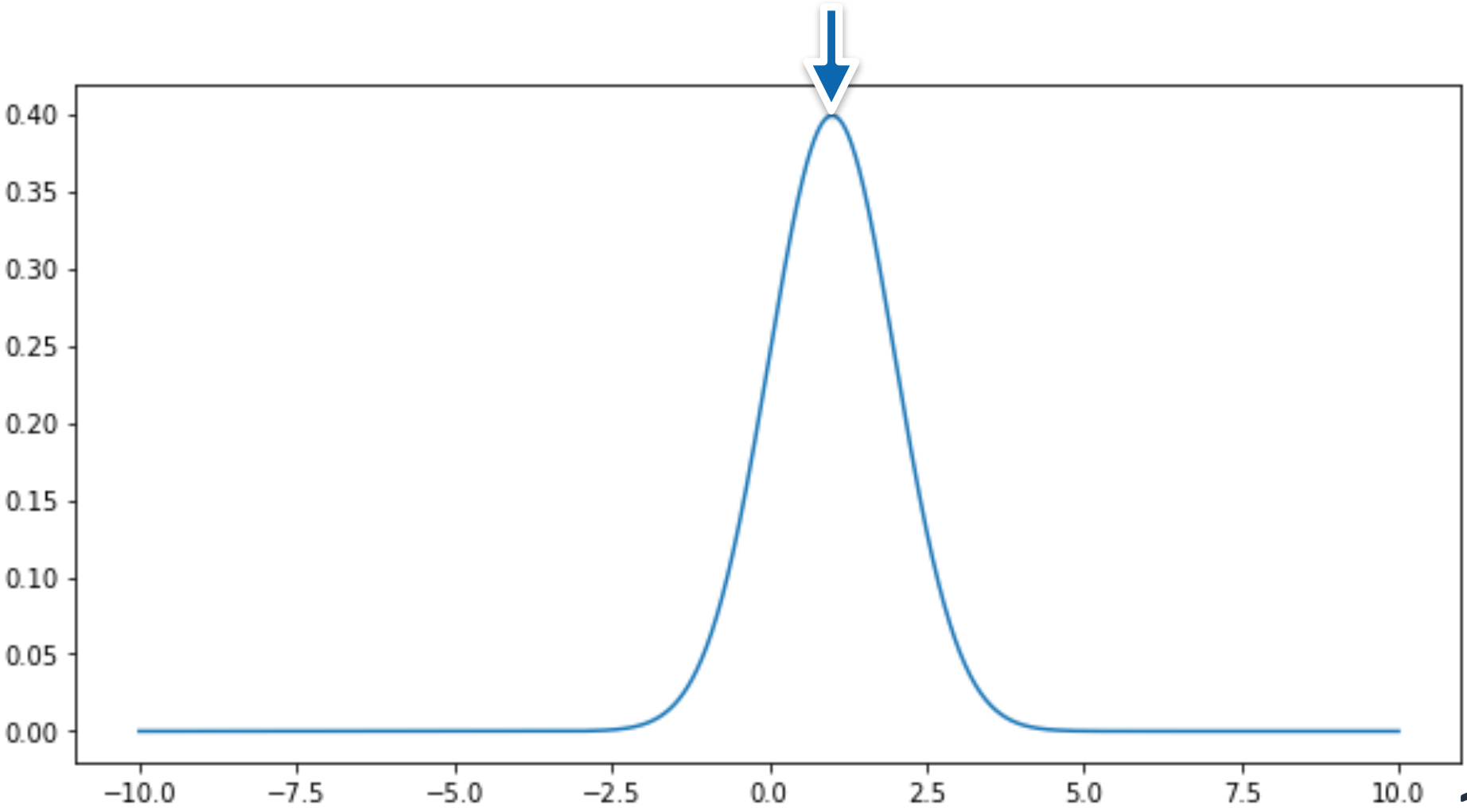
- Take derivatives with respect to it

$$\partial_{\sigma^2}[\cdot] = \frac{n}{2\sigma^2} - \frac{1}{2\sigma^4}\sum_{i=1}^{n}(x_i - \mu)^2 = 0$$

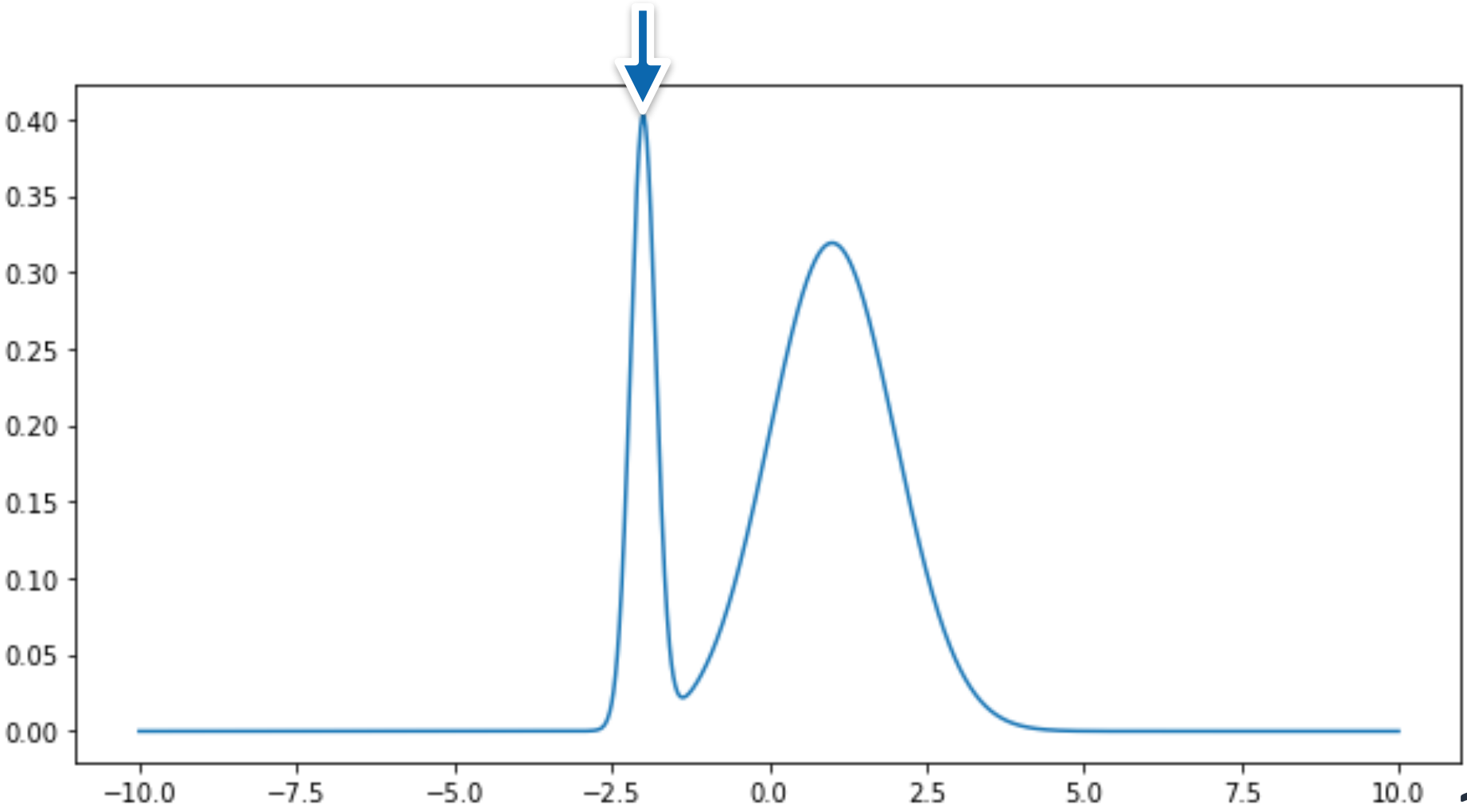
$$\implies \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

adapted from courses.d2l.ai/berkeley-stat-157

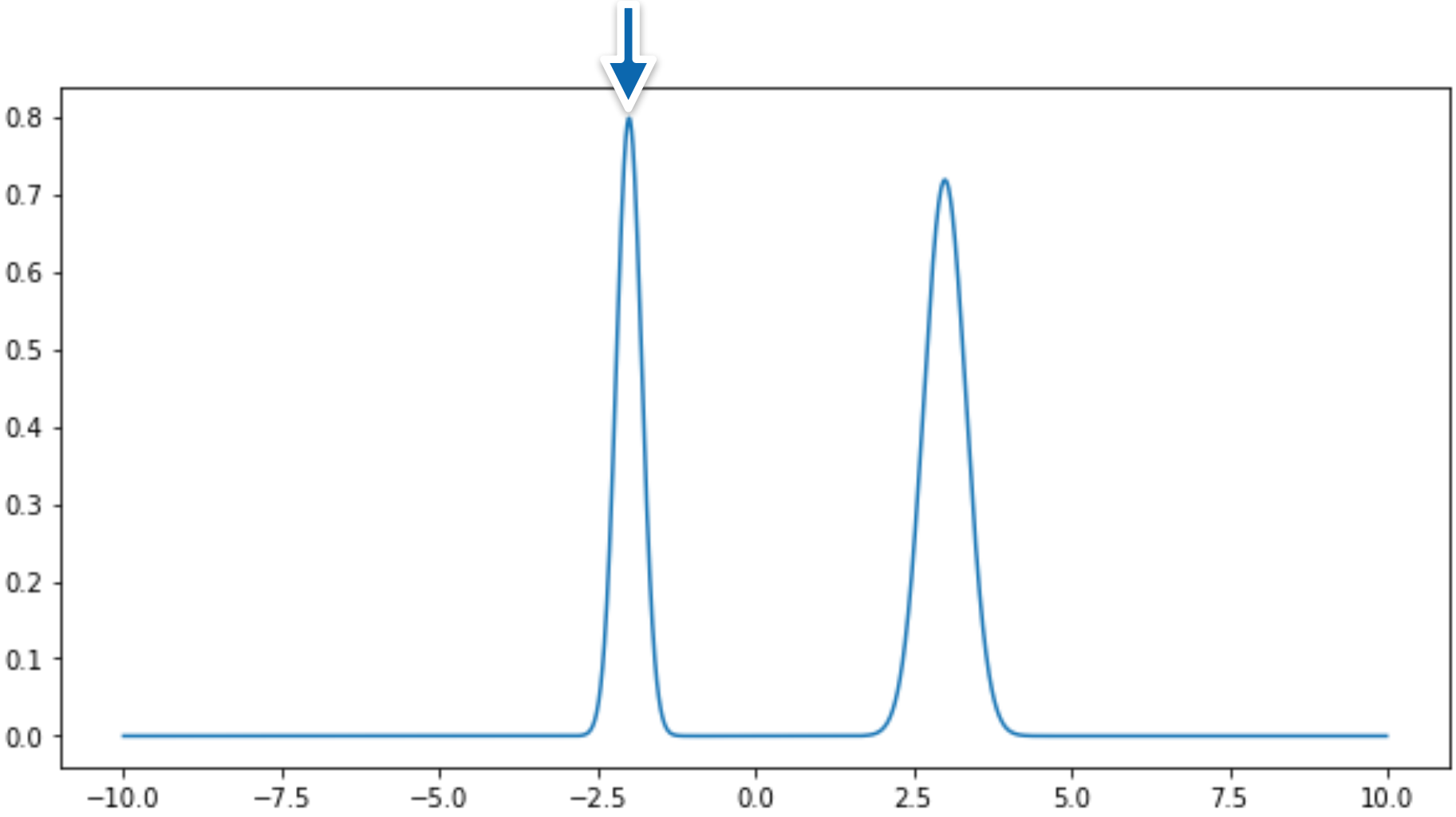
Maximum likelihood estimation



Maximum likelihood estimation



Maximum likelihood estimation



Maximum likelihood estimation

- Data - ‘student didn’t do homework’
- Possible parameters
  - **‘dog ate homework’**
  - **‘abducted by aliens’**
  - **‘too lazy’**
  - **‘sick grandmother’**

Should we believe all?

- **All parameters explain the data.**

adapted from courses.d2l.ai/berkeley-stat-157

Maximum a posteriori estimation

- Posterior Probability

$$p(w \mid X) \propto p(X \mid w) p(w)$$

penalty

$$\text{hence} \quad -\log p(w \mid X) = -\log p(X \mid w) - \log p(w) + c$$

- Maximum a Posteriori Estimation

$$\underset{w}{\text{minimize}} \; -\log p(X; w) - \log p(w)$$

- No homework example
  p('no homework'|explanation) = 1 (all explanations work)

lazy student	grandma sick	dog ate it	alien abduction
0.8	0.19	0.0099	0.0001

adapted from courses.d2l.ai/berkeley-stat-157

What does this have to do with regression?

adapted from courses.d2l.ai/berkeley-stat-157

Regression

- Recall optimization problem

$$\underset{w}{\text{minimize}} \sum_{i=1}^{n} (y_i - f(x_i, w))^2 + \text{penalty}(w)$$

$-\log p(w)$

Does the model work?

Additive Gaussian Noise

- Data generation model

$$y_i = f(x_i, w) + \epsilon_i \text{ where } \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

- Gaussian Prior $p(w)$ hence $-\log p(w) = \frac{1}{2\bar{\sigma}^2}\|w\|^2 + \text{const}.$

adapted from courses.d2l.ai/berkeley-stat-157

Regression

- Maximum a posteriori

$$\begin{aligned}
&\underset{w}{\text{minimize}} \; -\log p(w \mid X, Y) \\
\iff &\underset{w}{\text{minimize}} \; \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - f(x_i, w))^2 + \frac{1}{2\bar{\sigma}^2} \|w\|^2 + \text{const}. \\
\iff &\underset{w}{\text{minimize}} \; \frac{1}{2n} \sum_{i=1}^{n} (y_i - f(x_i, w))^2 + \frac{\lambda}{2} \|w\|^2
\end{aligned}$$

Implement this

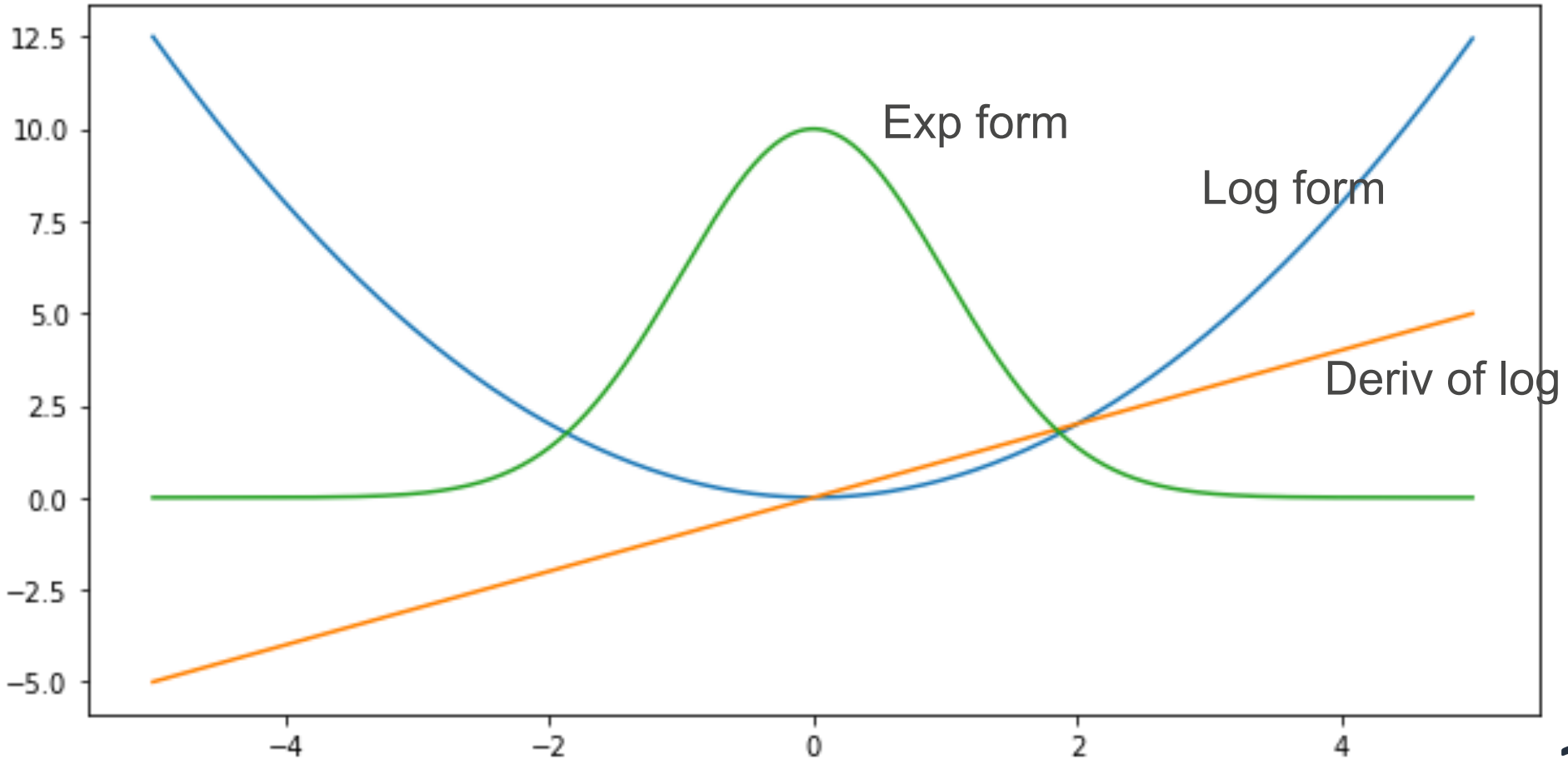
adapted from courses.d2l.ai/berkeley-stat-157


MAGIC Etch A Sketch® SCREEN
Loss
Functions
OHIO ART
MAGIC SCREEN IS GLASS SET IN STURDY PLASTIC FRAME
USE WITH CARE

adapted

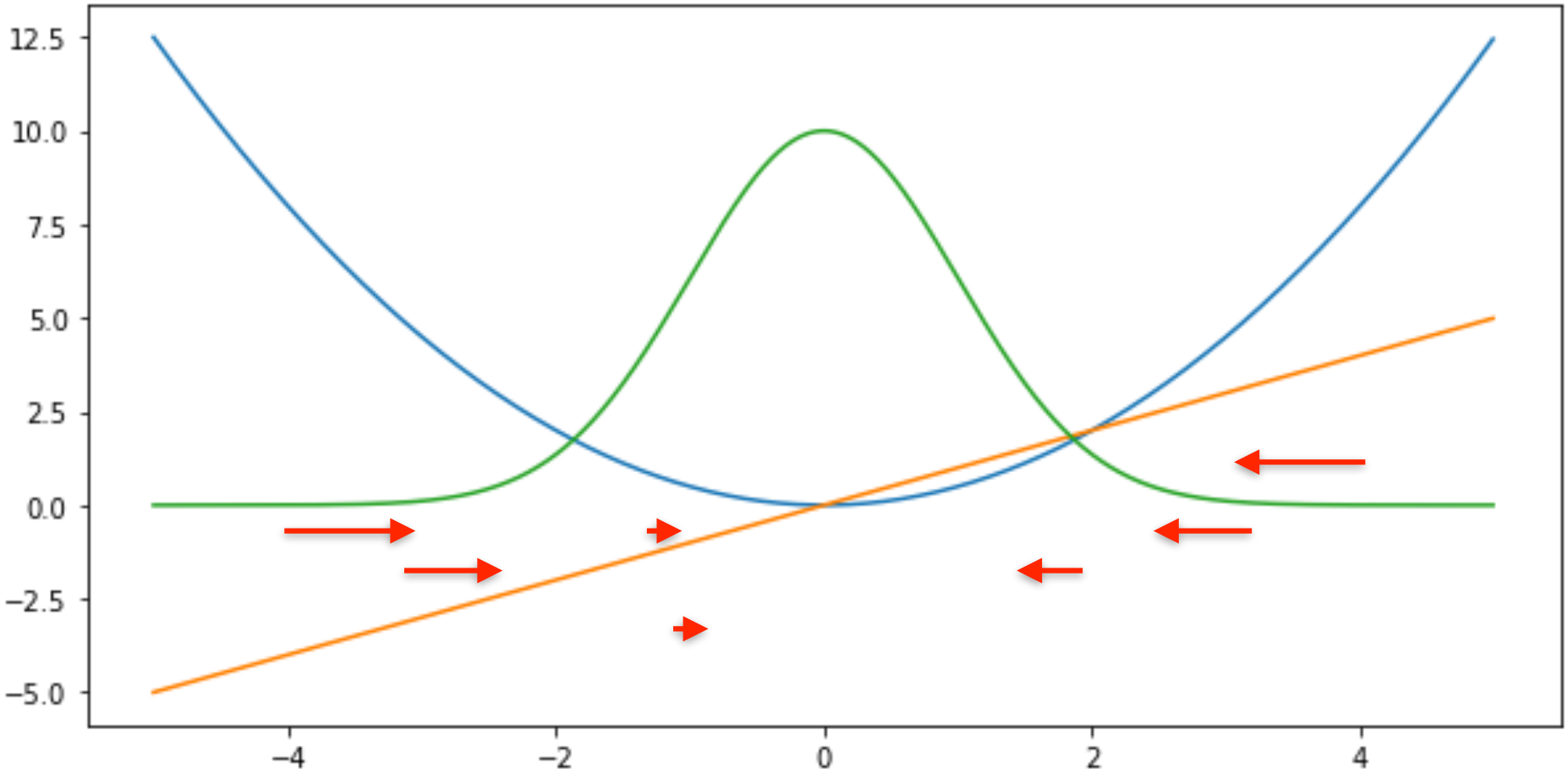
L2 Loss

$$l(y, y') = \frac{1}{2}(y - y')^2$$



L2 Loss - mean

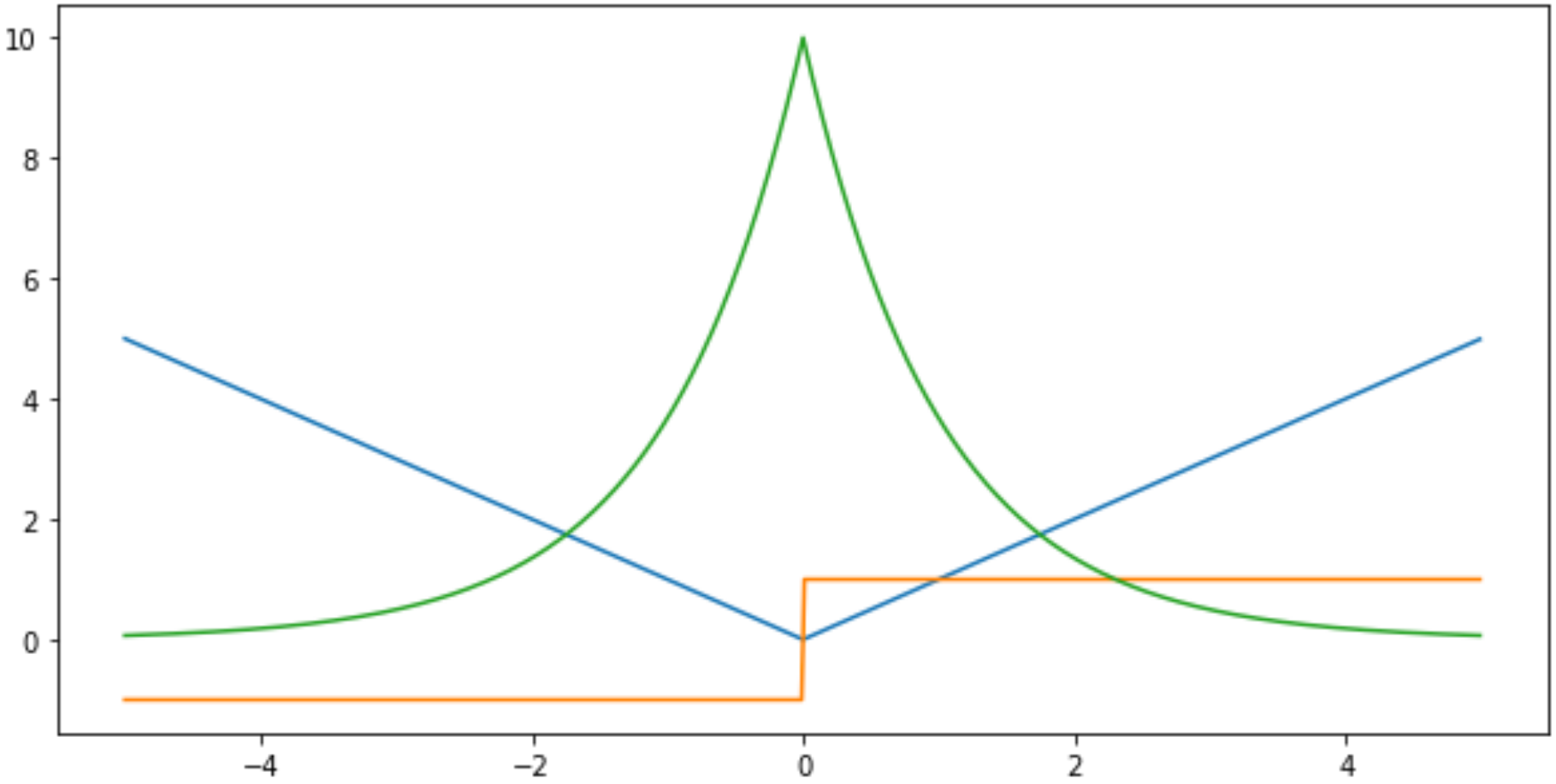
$$l(y, y') = \frac{1}{2}(y - y')^2$$



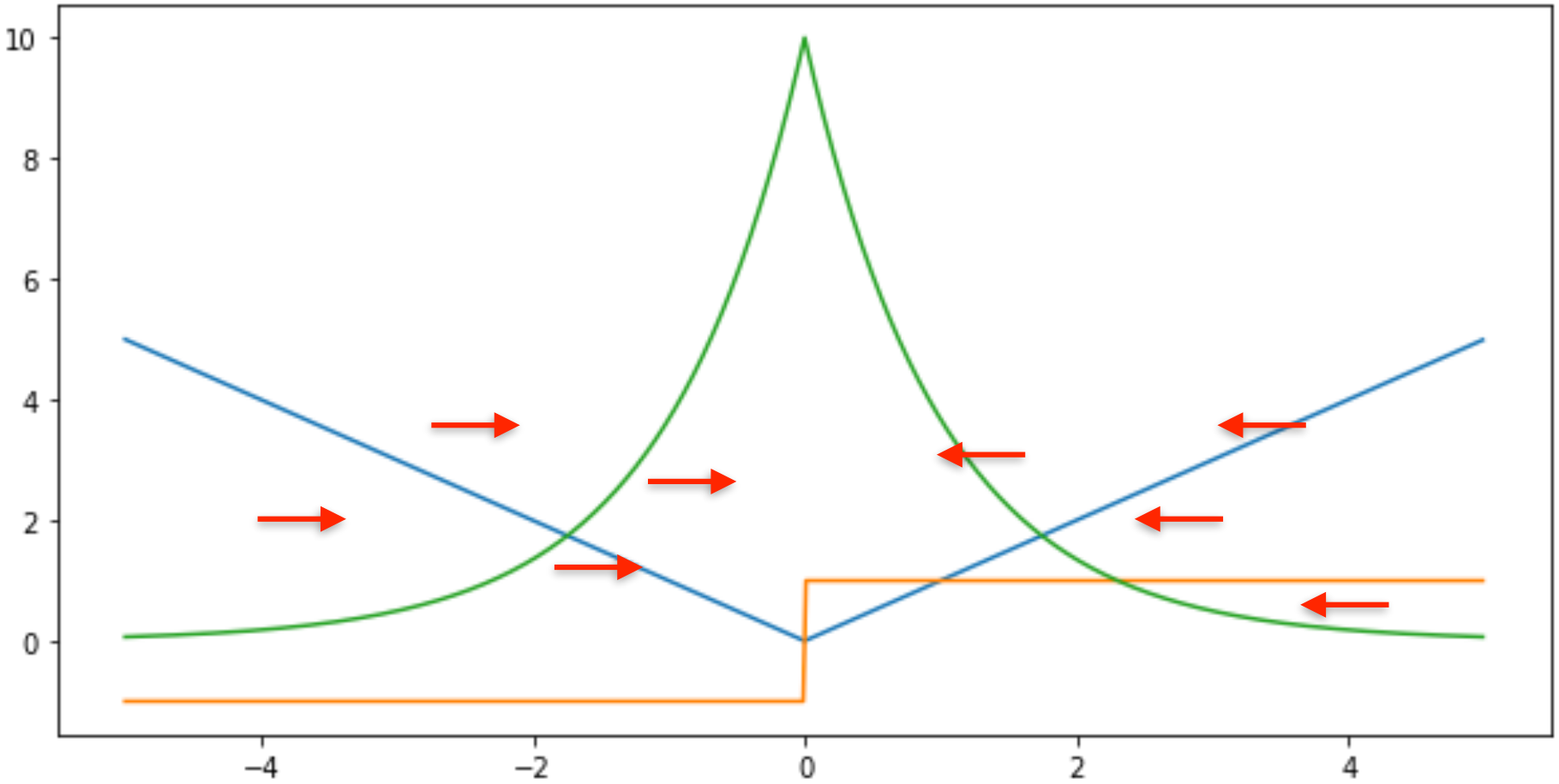
courses.d2l.ai/berkeley-stat-157

L1 Loss

$$l(y, y') = |y - y'|$$

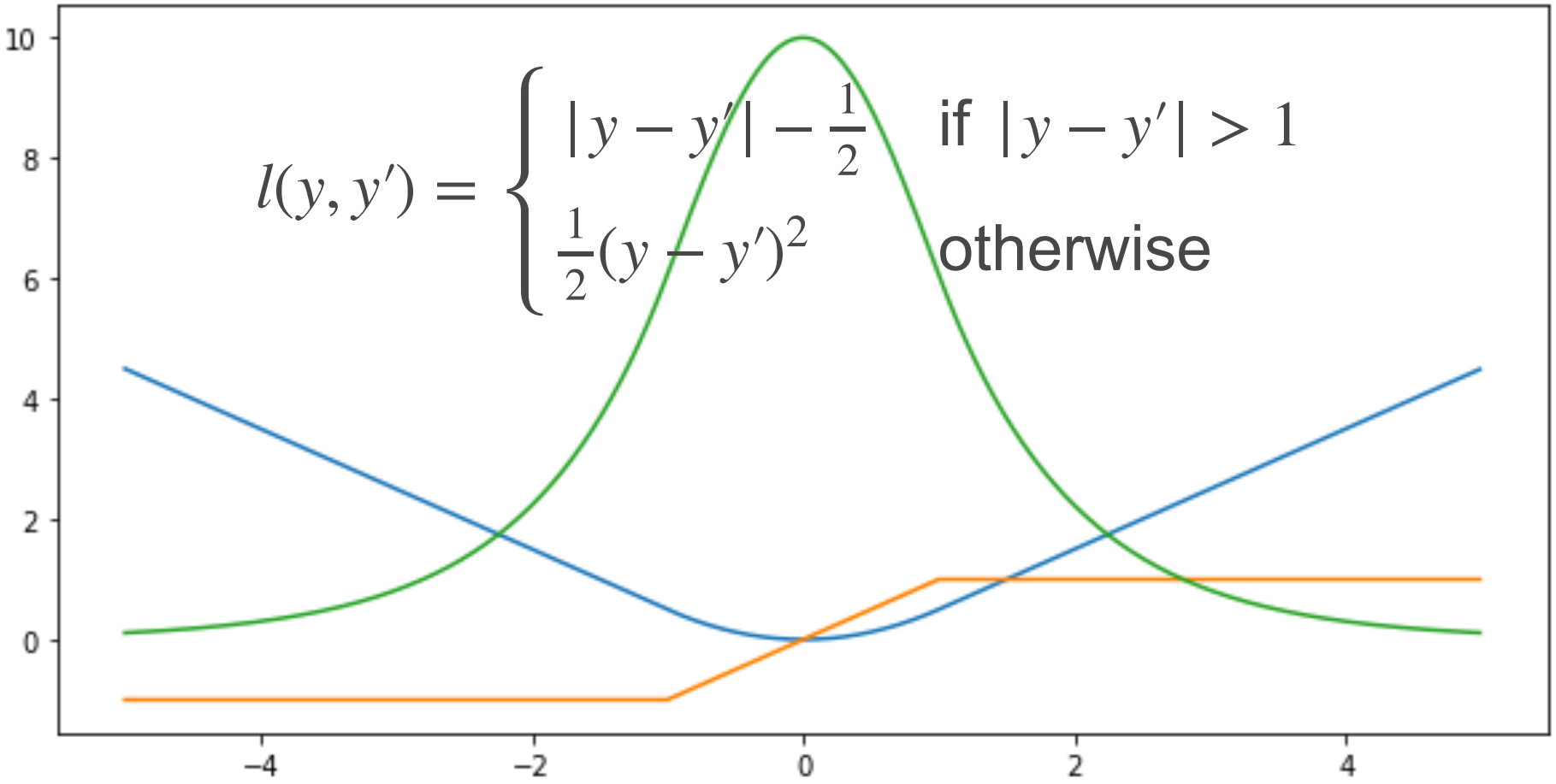


L1 Loss - median $l(y, y') = |y - y'|$

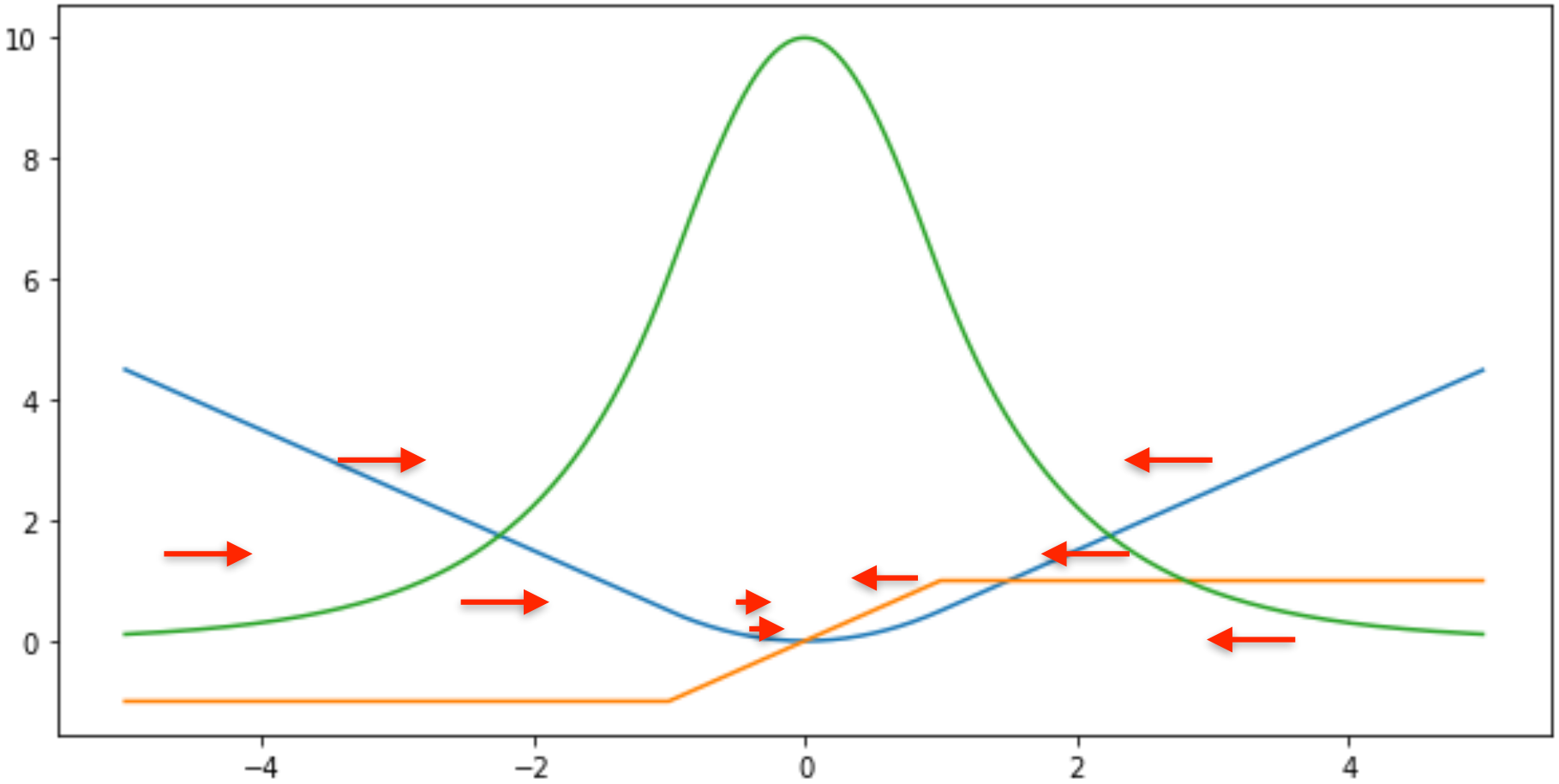


Huber's Robust Loss

$$l(y, y') = \begin{cases} |y - y'| - \frac{1}{2} & \text{if } |y - y'| > 1 \\ \frac{1}{2}(y - y')^2 & \text{otherwise} \end{cases}$$



Huber's Robust Loss - trimmed mean



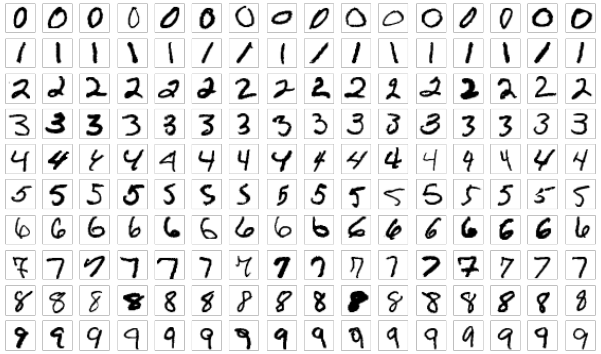

MAGIC Etch A Sketch® SCREEN
Logistic
Regression
DHIO ART
MAGIC SCREEN IS GLASS SET IN STURDY PLASTIC FRAME
USE WITH CARE

adapted

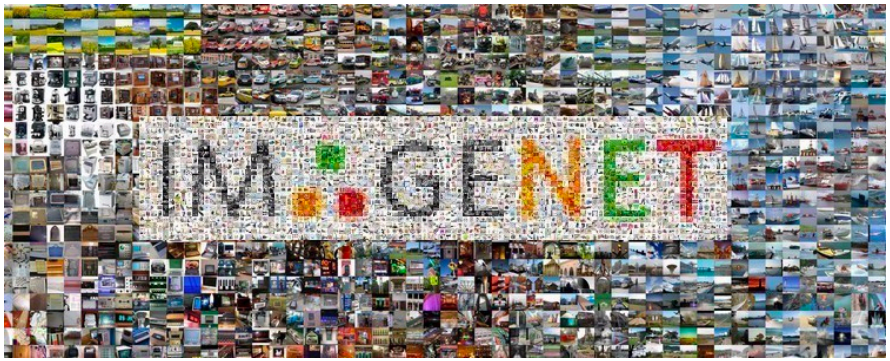
Regression vs. Classification

- Regression estimates a continuous value
- Classification predicts a discrete category

MNIST: classify hand-written digits (10 classes)



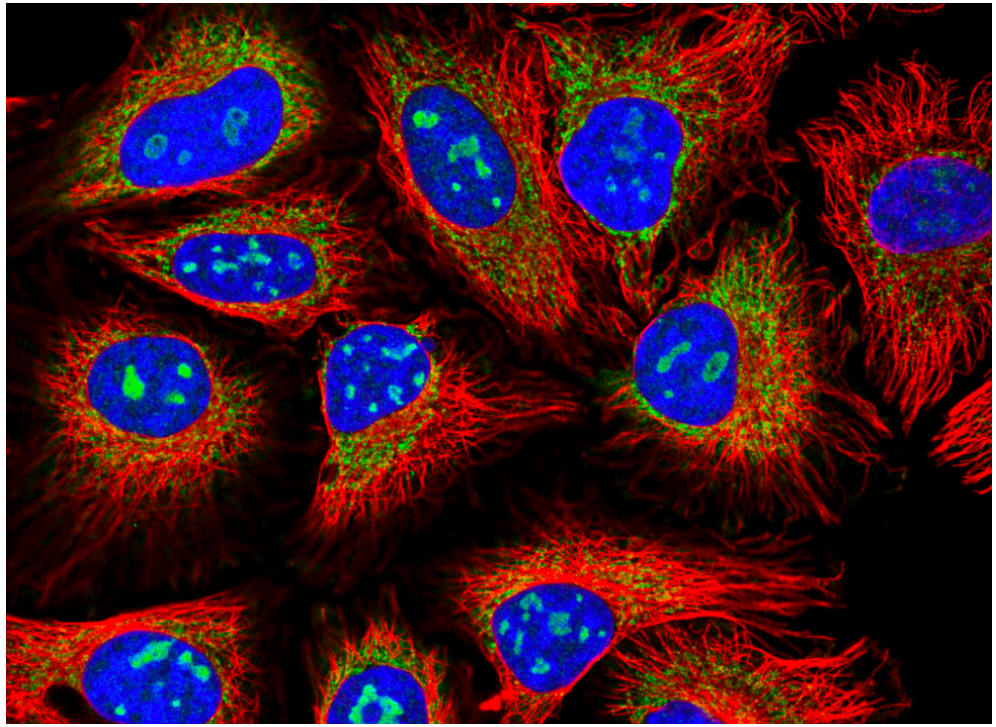
ImageNet: classify nature objects (1000 classes)



slides adapted from courses.d2l.ai/berkeley-stat-157

Classification Tasks at Kaggle

Classify human protein microscope images into 28 categories



```
0.  Nucleoplasm
1.  Nuclear membrane
2.  Nucleoli
3.  Nucleoli fibrillar
4.  Nuclear speckles
5.  Nuclear bodies
6.  Endoplasmic reticu
7.  Golgi apparatus
8.  Peroxisomes
9.  Endosomes
10.  Lysosomes
11.  Intermediate fila
12.  Actin filaments
13.  Focal adhesion si
14.  Microtubules
15.  Microtubule ends
16.  Cytokinetic bridg
```

https://www.kaggle.com/c/human-protein-atlas-image-classification

slides adapted from courses.d2l.ai/berkeley-stat-157

Classification Tasks at Kaggle

Classify malware into 9 categories

https://www.kaggle.com/c/malware-classification

slides adapted from courses.d2l.ai/berkeley-stat-157

Classification Tasks at Kaggle

Classify toxic Wikipedia comments into 7 categories

comment_text	toxic	severe_toxic	obsc
Explanation\nWhy the edits made under my usern...	0	0	0
D'aww! He matches this background colour I'm s...	0	0	0
Hey man, I'm really not trying to edit war. It...	0	0	0
"\nMore\nI can't make any real suggestions on ...	0	0	0
You, sir, are my hero. Any chance you remember...	0	0	0

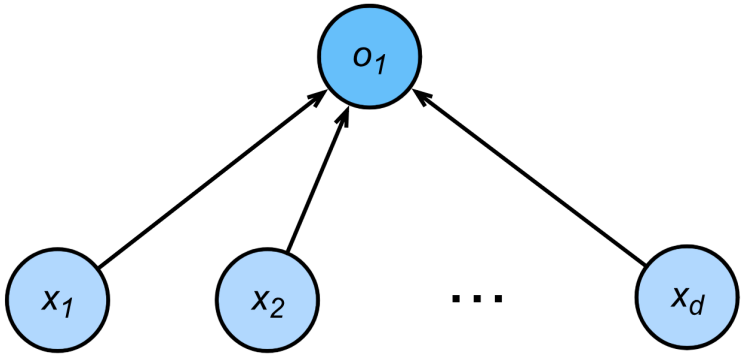
https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

slides adapted from courses.d2l.ai/berkeley-stat-157

From Regression to Multi-class Classification

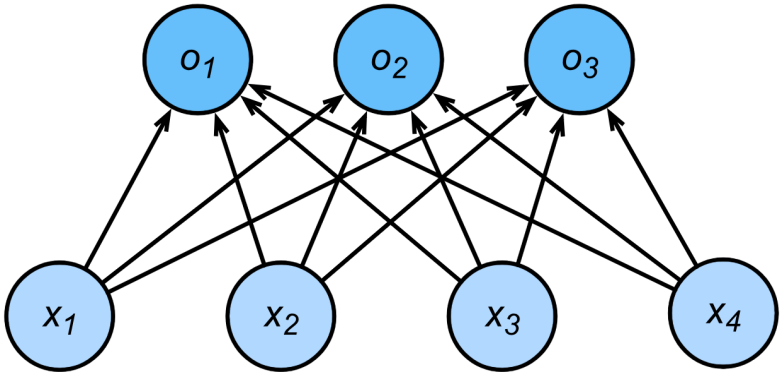
Regression

- Single continuous output
- Natural scale in $\mathbb{R}$
- Loss given e.g. in terms of difference $y - f(x)$



Classification

- Multiple classes, typically multiple outputs
- Score *should* reflect confidence …



slides adapted from courses.d2l.ai/berkeley-stat-157

From Regression to Multi-class Classification

Square Loss

- One hot encoding per class

$$\mathbf{y} = [y_1, y_2, \ldots, y_n]^\top$$

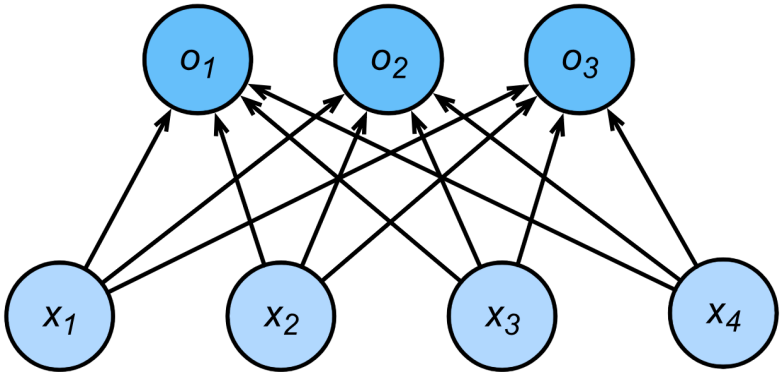
$$y_i = \begin{cases} 1 \text{ if } i = y \\ 0 \text{ otherwise} \end{cases}$$

- Train with squared loss
- Largest output wins

$$\hat{y} = \underset{i}{\operatorname{argmax}}\, o_i$$

Classification

- Multiple classes, typically multiple outputs
- Score *should* reflect confidence …



slides adapted from courses.d2l.ai/berkeley-stat-157

From Regression to Multi-class Classification

Uncalibrated Scale

- One output per class
- Largest output wins

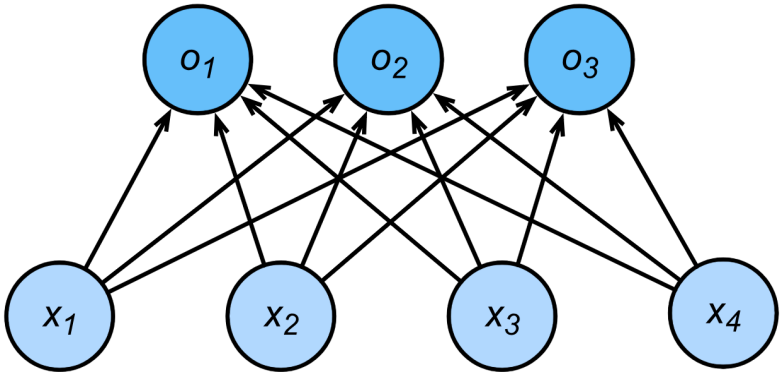
$$\hat{y} = \operatorname*{argmax}_i o_i$$

- Want that correct class is recognized confidently (**large margin**)

$$o_y - o_i \geq \Delta(y, i)$$

Classification

- Multiple classes, typically multiple outputs
- Score *should* reflect confidence …



slides adapted from courses.d2l.ai/berkeley-stat-157

From Regression to Multi-class Classification

Calibrated Scale

- Output matches probabilities (nonnegative, sums to 1)

$$p(y \mid o) = \text{softmax}(o)$$

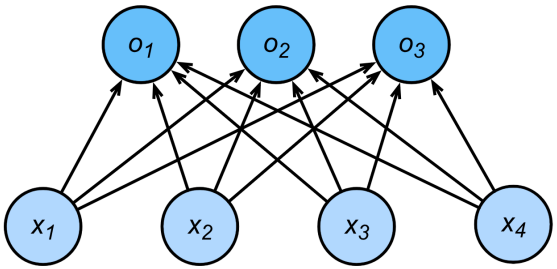
$$= \frac{\exp(o_y)}{\sum_i \exp(o_i)}$$

- Negative log-likelihood

$$-\log p(y \mid y) = \log \sum_i \exp(o_i) - o_y$$

Classification

- Multiple classes, typically multiple outputs
- Score *should* reflect confidence …



slides adapted from courses.d2l.ai/berkeley-stat-157

Summary

- **Maximum Likelihood**
  - Gauss and means
  - More loss functions (l_1 loss, trimmed mean)
  - Regression revisited
- **Classification**
  - Computing discrete probabilities
  - Likelihood and loss functions

adapted from courses.d2l.ai/berkeley-stat-157