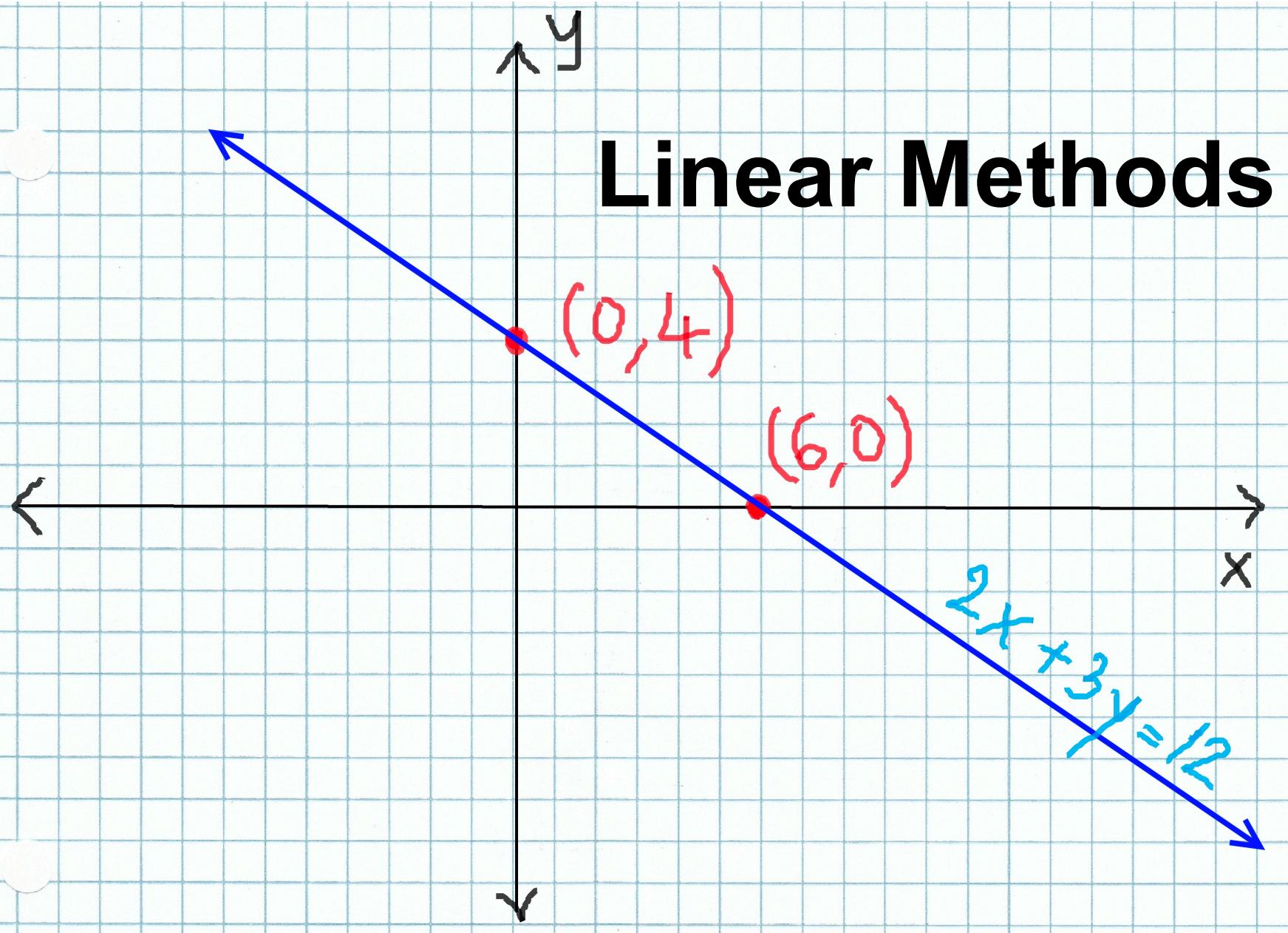


Introduction to (Deep) Learning

4. Linear Regression, Basic Optimization

Linear Methods



House Buying 101

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

Listing price from agent

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

Predicted sale price



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

Interior Features

Bedroom Information

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

Multi-Unit Information

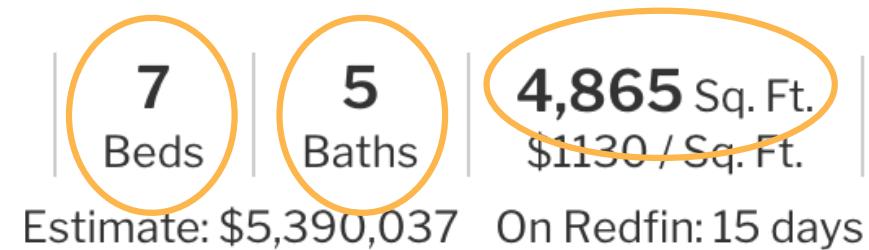
- # of Stories: 2

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

- Kitchen Description: Countertop Dishwasher, Garbage Disposal, Hood Island with Sink, Microwave, Oven

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

$$y = w_1x_1 + w_2x_2 + w_3x_3 + b$$

Weights and bias are determined later

Linear Model

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

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

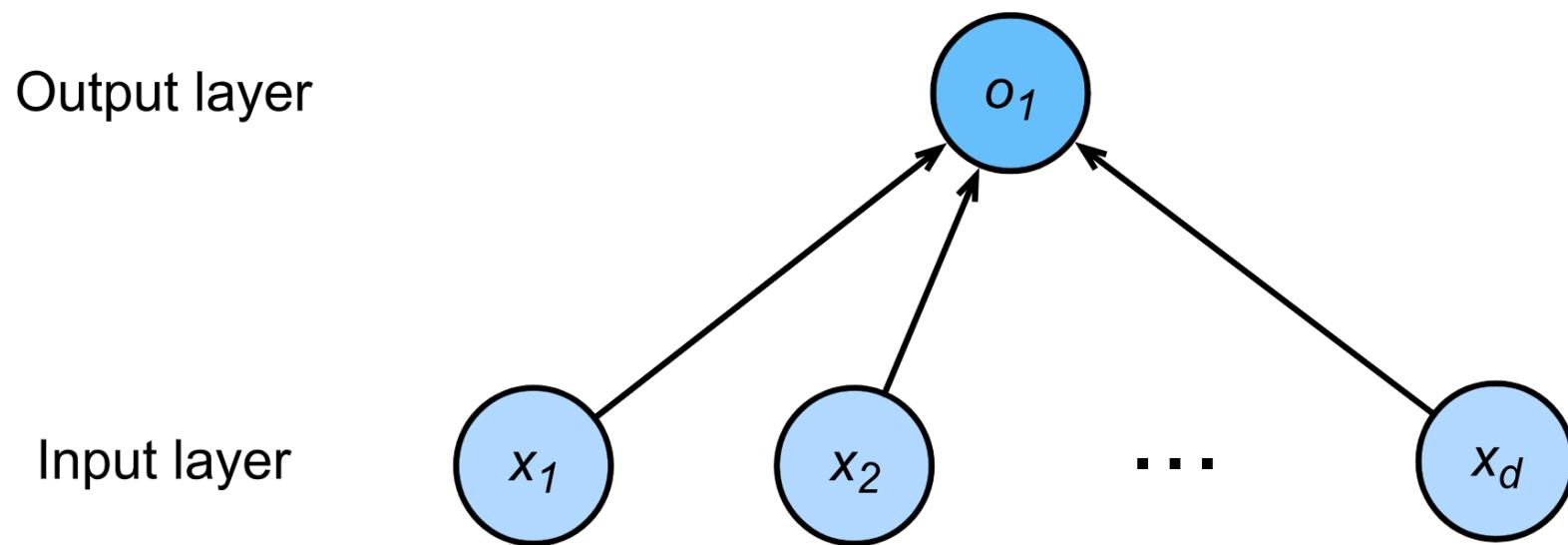
- The output is a weighted sum of the inputs

$$y = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$$

Vectorized version

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

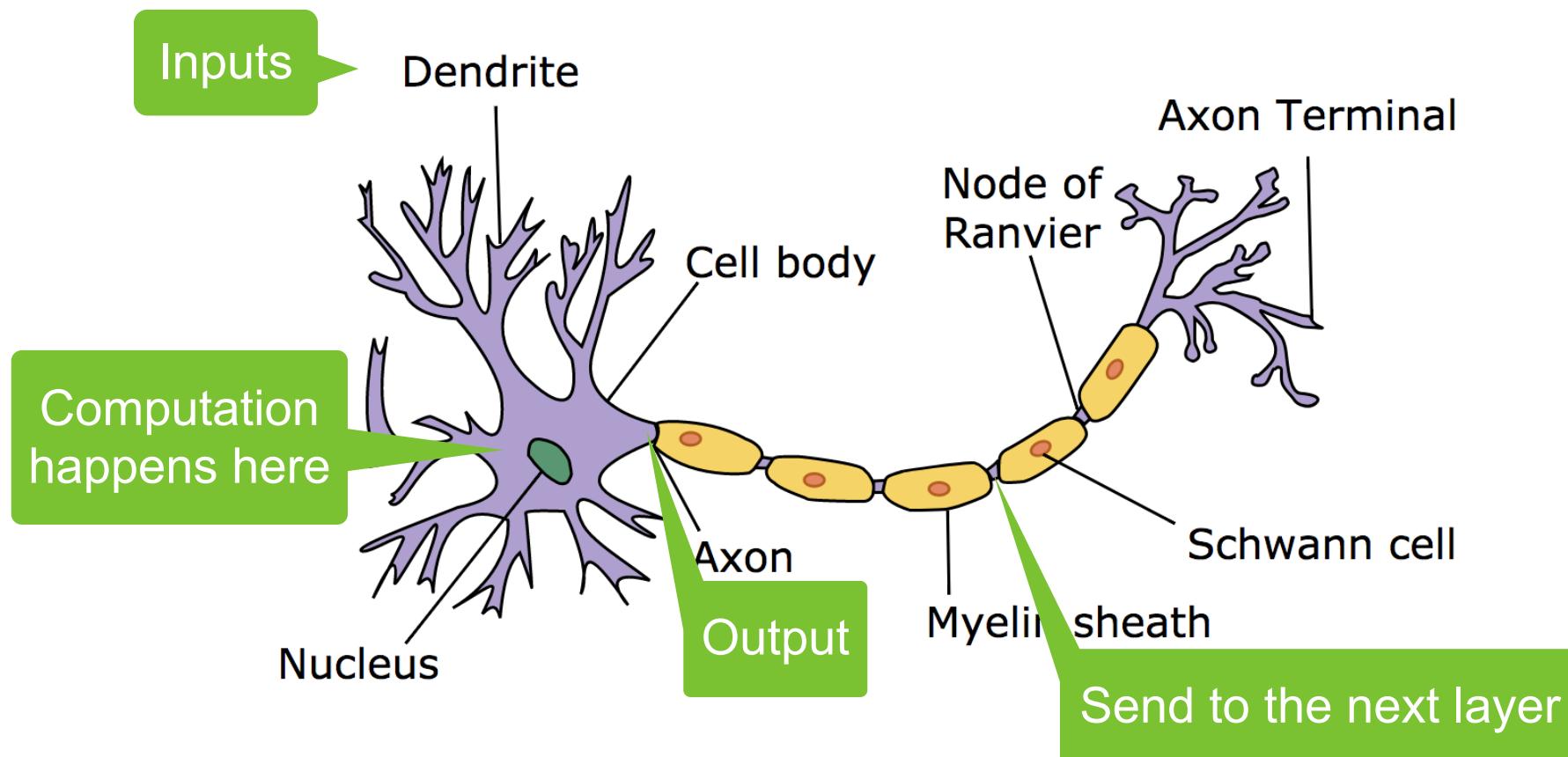
Linear Model as a Single-layer Neural Network



We can stack multiple layers to get deep neural networks

Neural Networks Derive from Neuroscience

The real neuron



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

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

It is called squared loss

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

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

Learn Parameters

- Training loss

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

- Minimize loss to learn parameters

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

Closed-form Solution

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

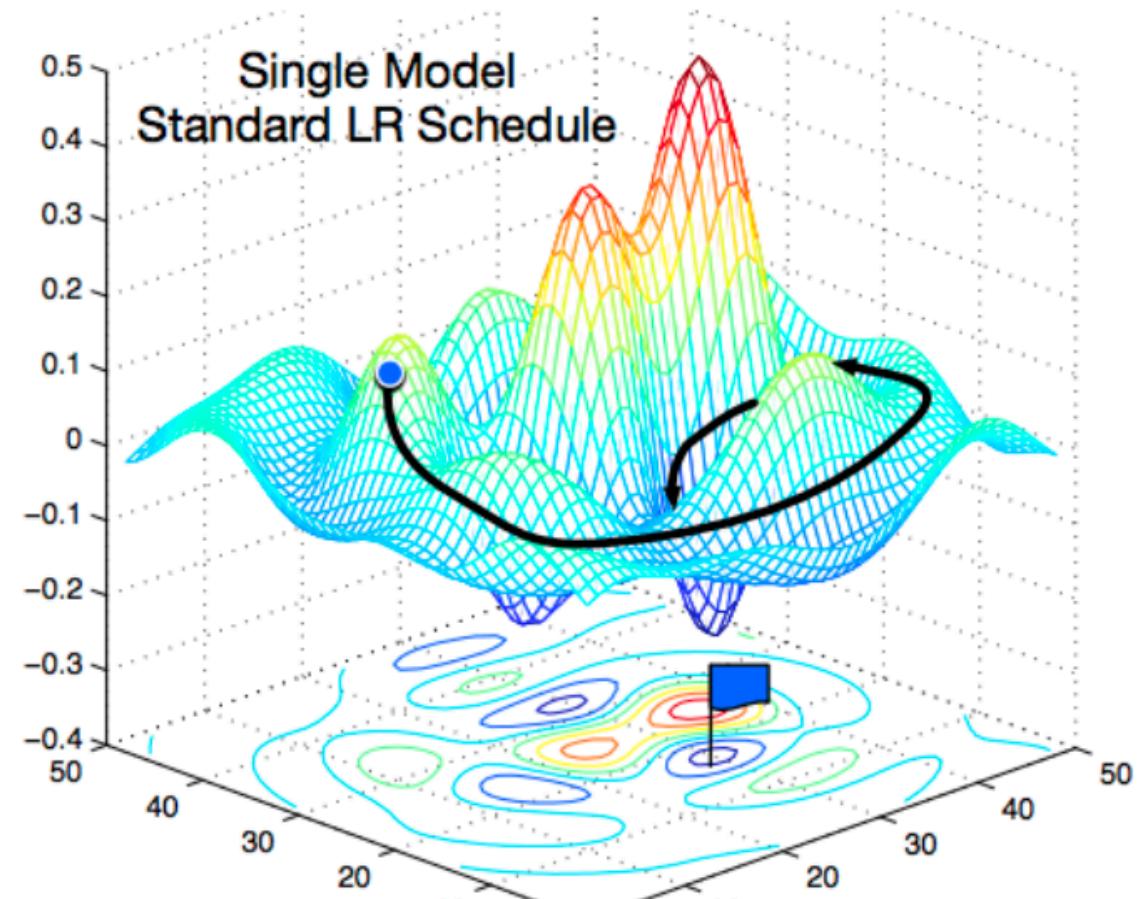
$$\ell(\mathbf{X}, \mathbf{y}, \mathbf{w}) = \frac{1}{n} \parallel \mathbf{y} - \mathbf{X}\mathbf{w} \parallel^2 \quad \frac{\partial}{\partial \mathbf{w}} \ell(\mathbf{X}, \mathbf{y}, \mathbf{w}) = \frac{2}{n} (\mathbf{y} - \mathbf{X}\mathbf{w})^T \mathbf{X}$$

- Loss is convex, so the optimal solutions satisfies

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} \ell(\mathbf{X}, \mathbf{y}, \mathbf{w}) &= 0 \\ \Leftrightarrow \frac{2}{n} (\mathbf{y} - \mathbf{X}\mathbf{w})^T \mathbf{X} &= 0 \\ \Leftrightarrow \mathbf{w}^* &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X} \mathbf{y} \end{aligned}$$

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

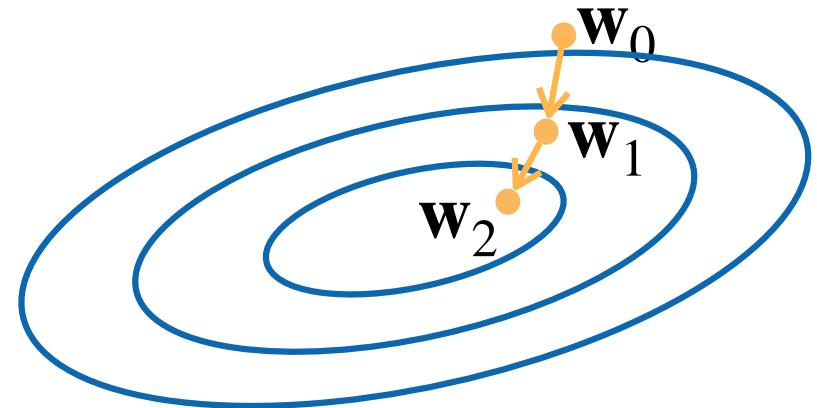
Basic Optimization



Gradient Descent

- Choose a starting point w_0
- Repeat to update the weight $t=1,2,3$

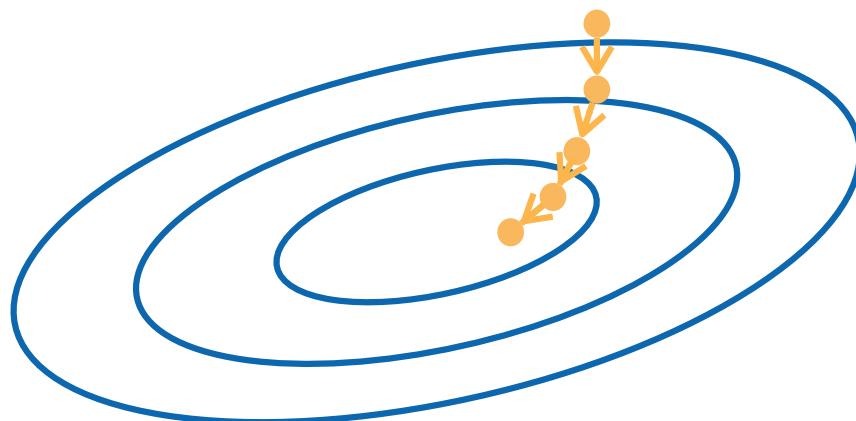
$$w_t = w_{t-1} - \eta \frac{\partial \ell}{\partial w_{t-1}}$$



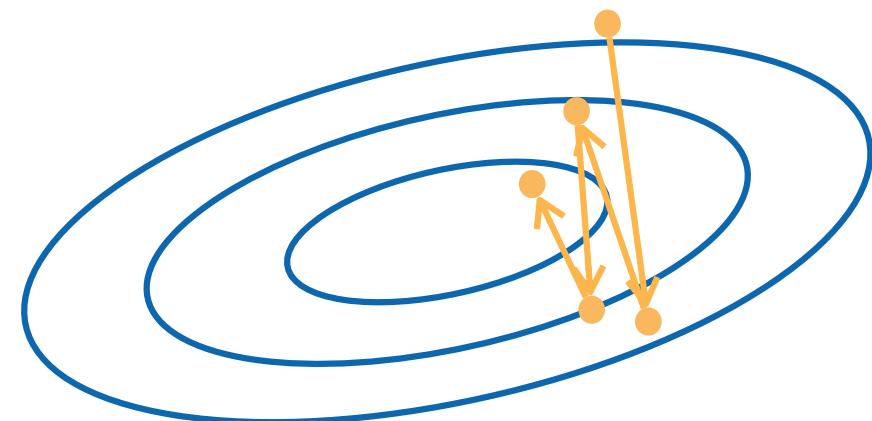
- Gradient: a direction that increases the value
- Learning rate: a hyper-parameter specifies the step length

Choose a Learning Rate

Not too small



Not too big



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, \dots, 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

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



MAGIC Etch A Sketch[®] SCREEN

LOSS FUNCTIONS



Horizontal
Dial



Vertical
Dial

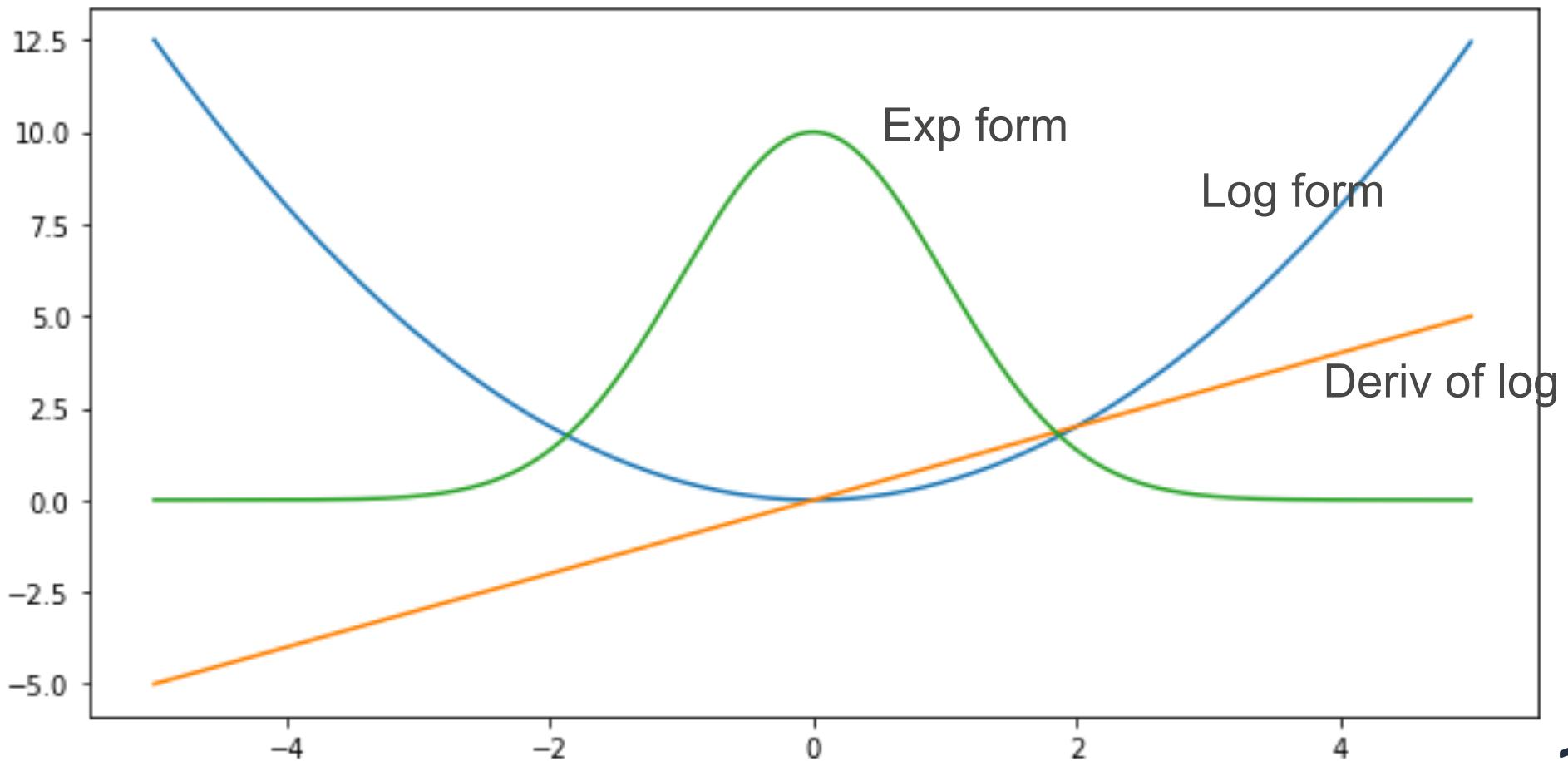
OHIO ART The World of Toys[®]

MAGIC SCREEN IS GLASS SET IN STURDY PLASTIC FRAME
USE WITH CARE

adapted

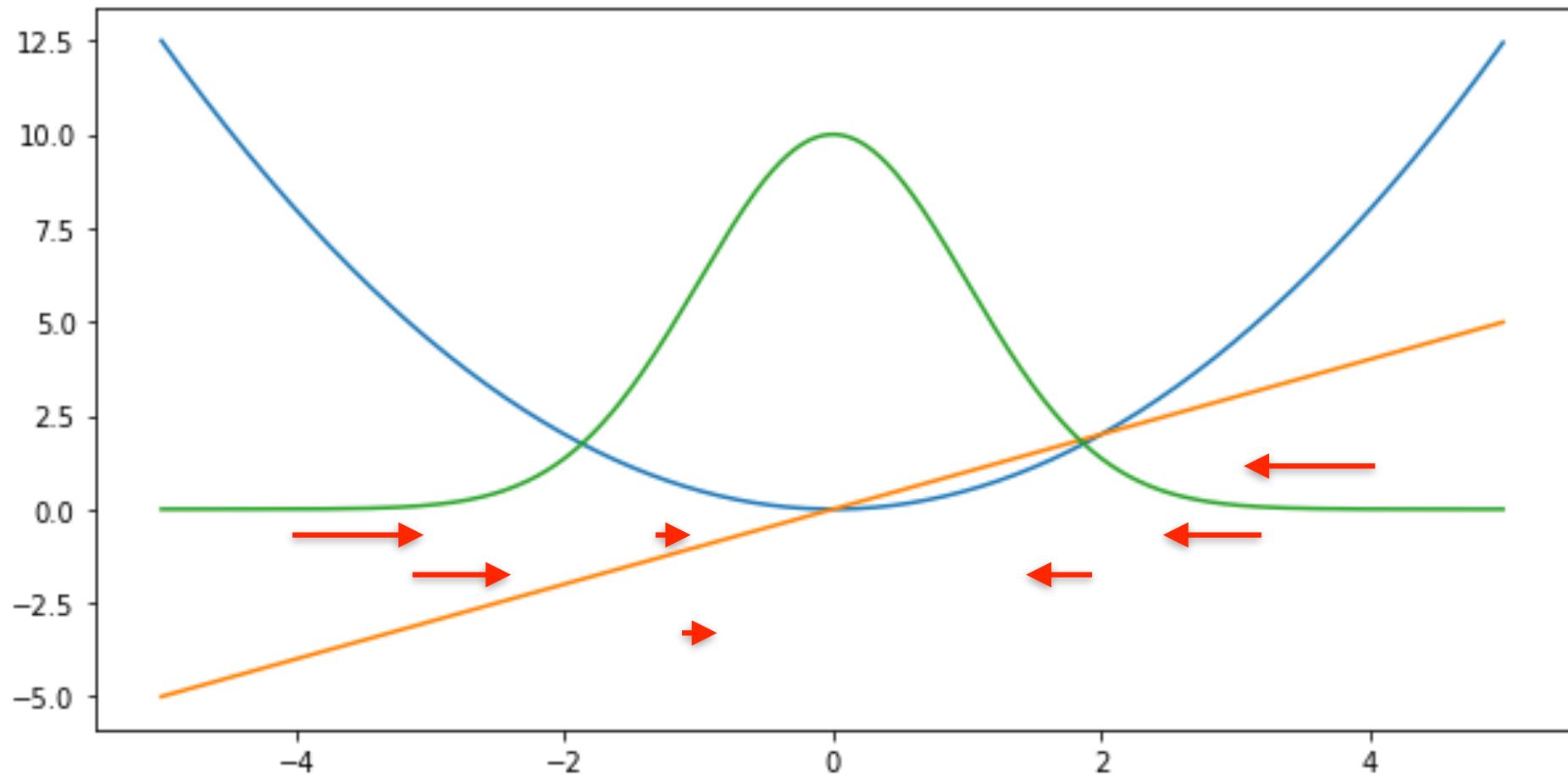
L2 Loss

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



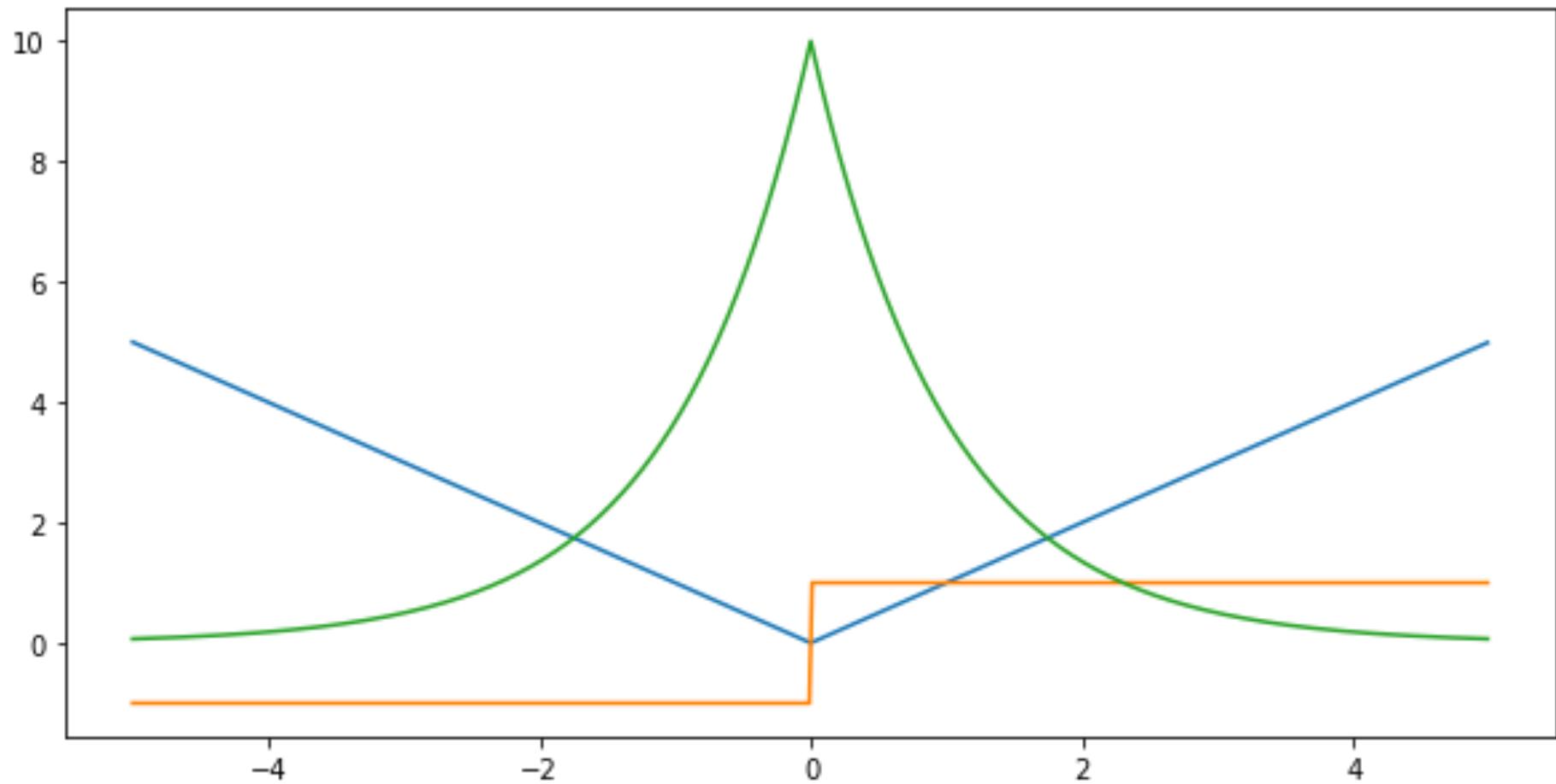
L2 Loss - mean

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

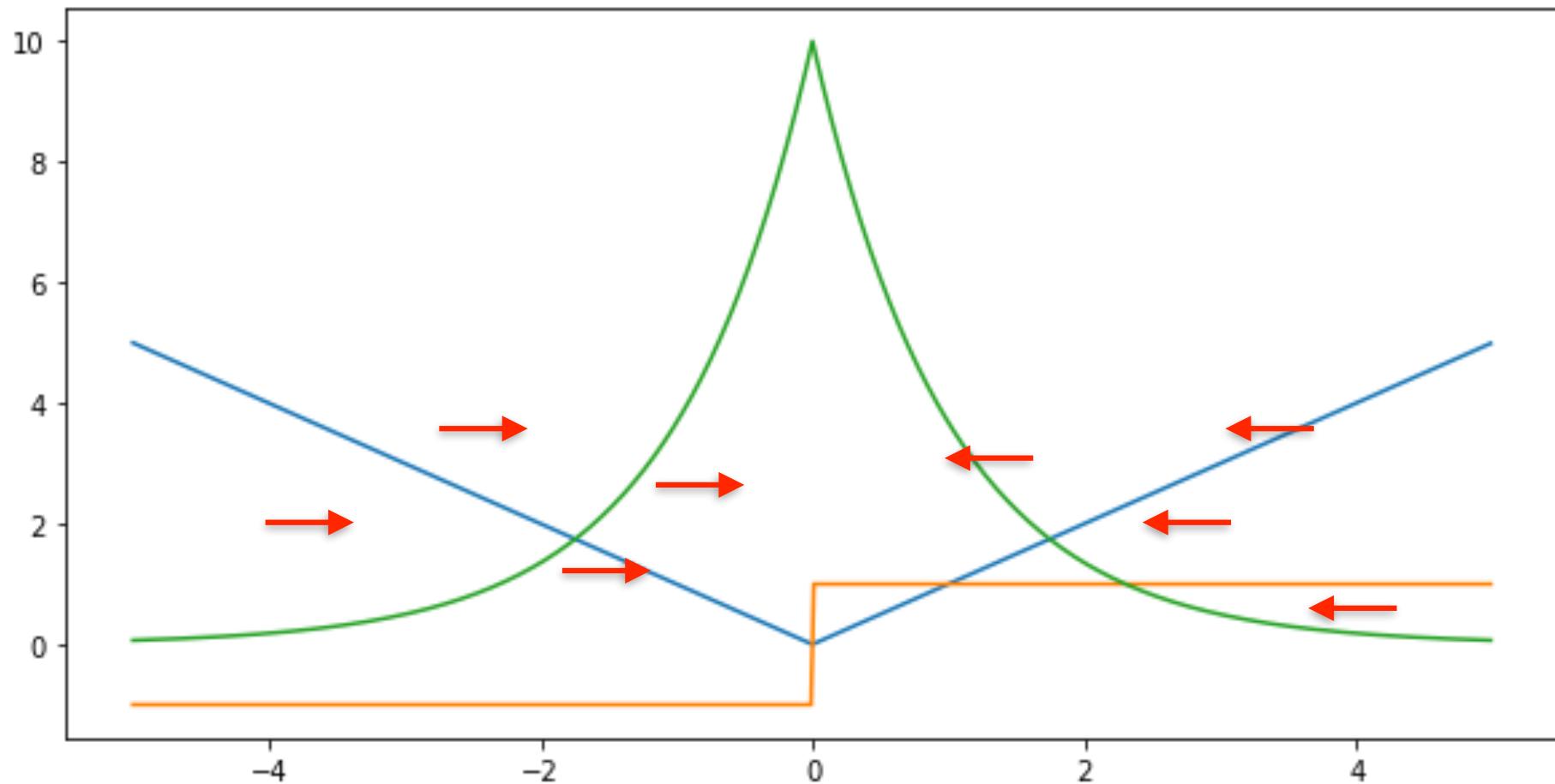


L1 Loss

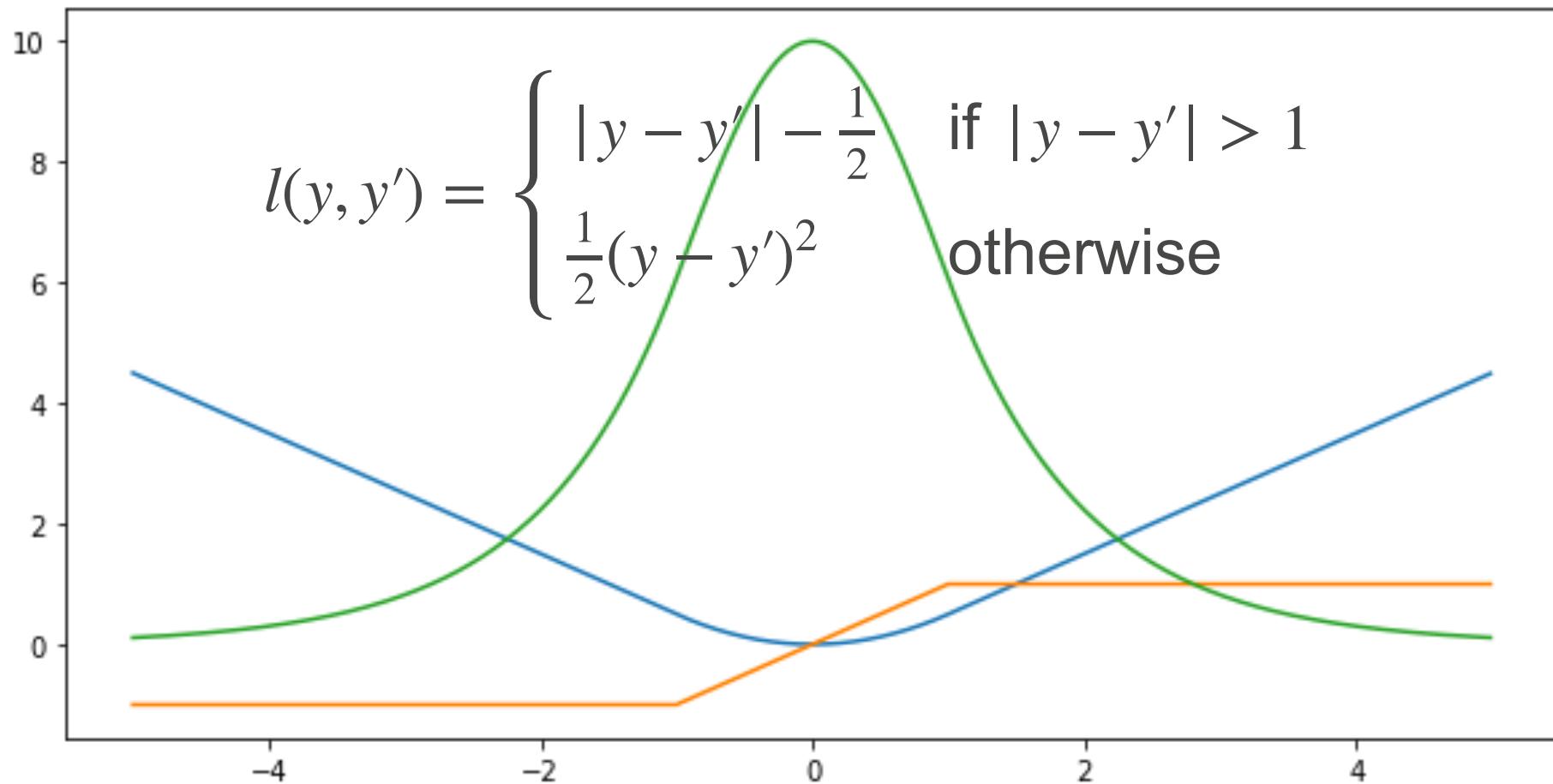
$$l(y, y') = |y - y'|$$



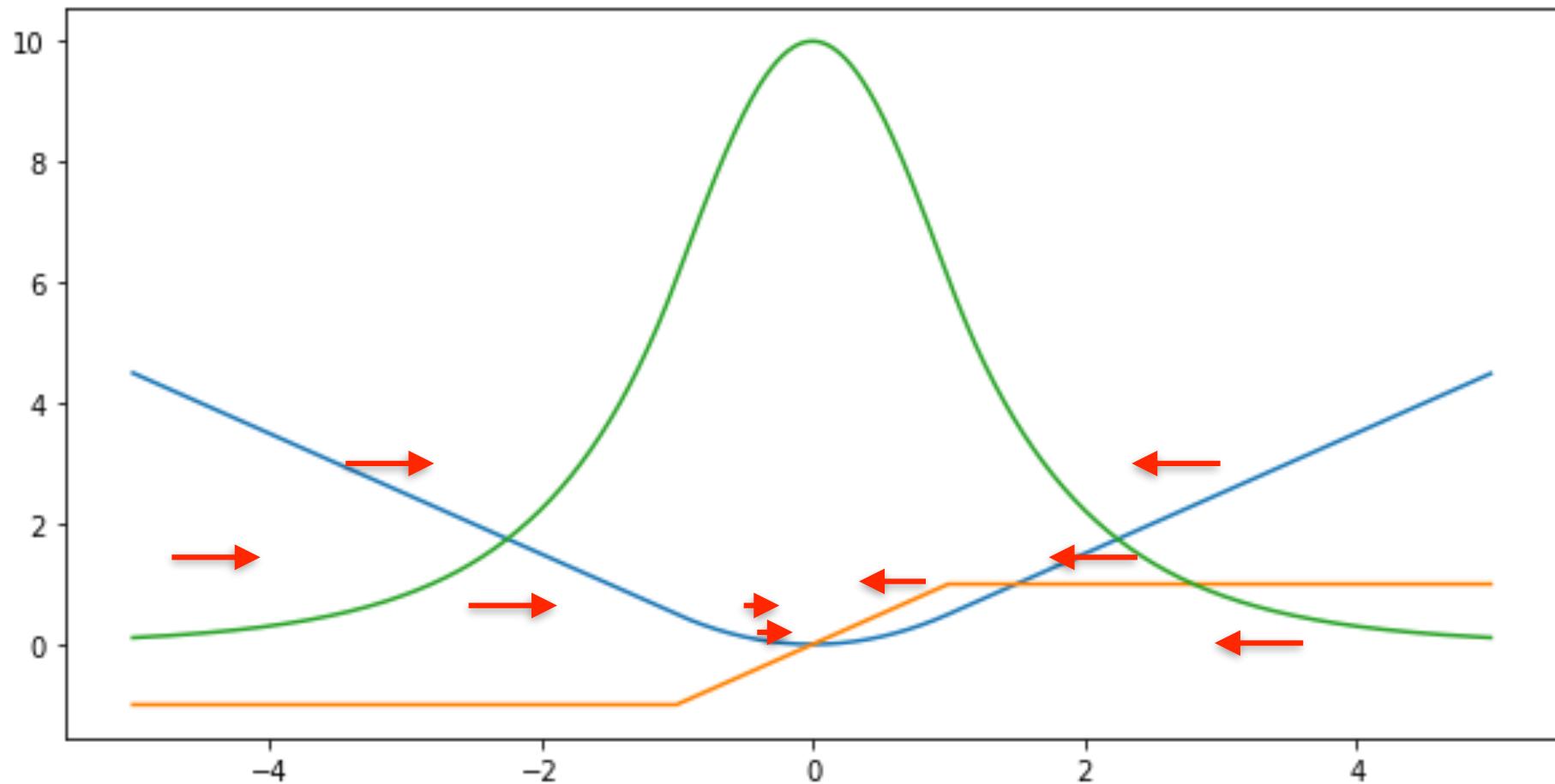
L1 Loss - median $l(y, y') = |y - y'|$



Huber's Robust Loss



Huber's Robust Loss - trimmed mean





MAGIC Etch A Sketch[®] SCREEN

Logistic Regression



Horizontal
Dial



Vertical
Dial

OHIO ART The World of Toys[®]

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)

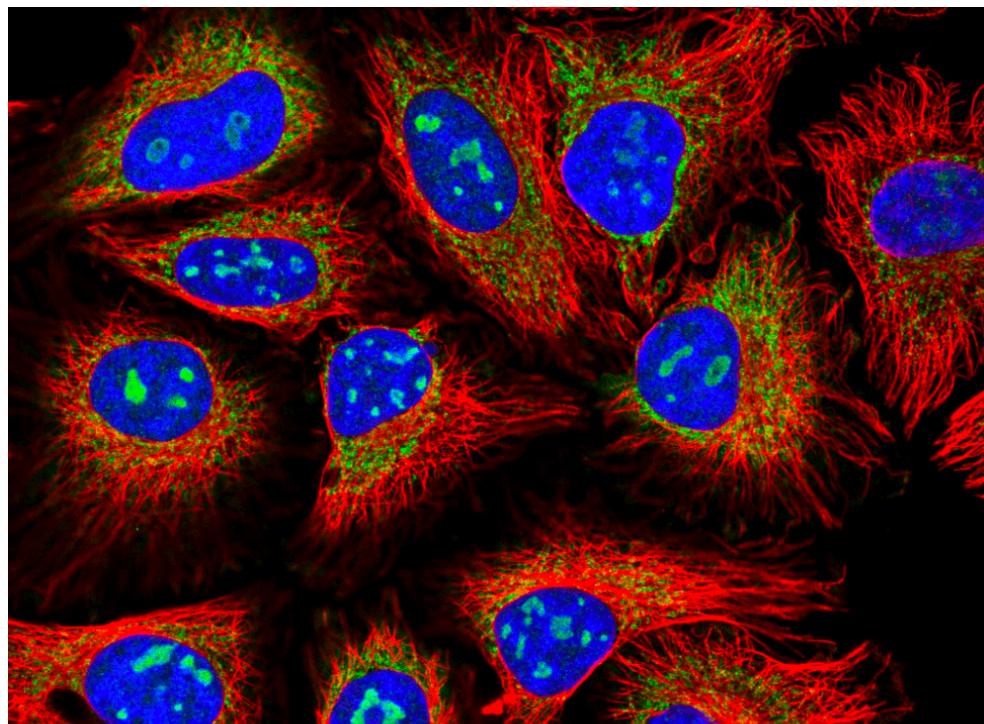
A 10x10 grid of handwritten digits from 0 to 9. Each digit is enclosed in a small square box. The digits are arranged in a pattern: row 1 has all zeros; row 2 has alternating ones and twos; row 3 has alternating threes and fours; row 4 has alternating fours and fives; row 5 has alternating fives and sixes; row 6 has alternating sixes and sevens; row 7 has alternating sevens and eights; row 8 has alternating eights and nines; and row 9 has alternating nines and zeros.

ImageNet: classify nature objects (1000 classes)



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 reticulu
- 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	obscene
Explanation\nWhy the edits made under my user...	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

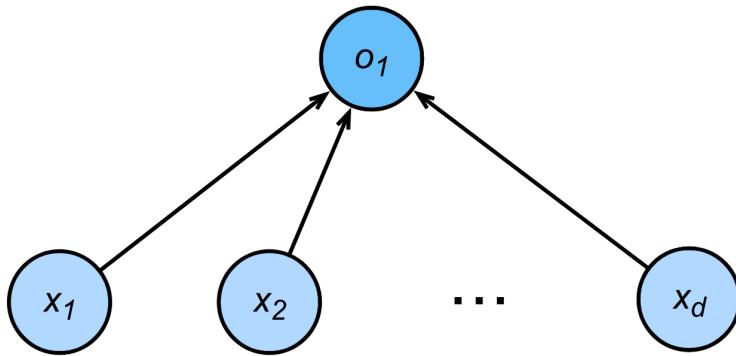
<https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge>

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

From Regression to Multi-class Classification

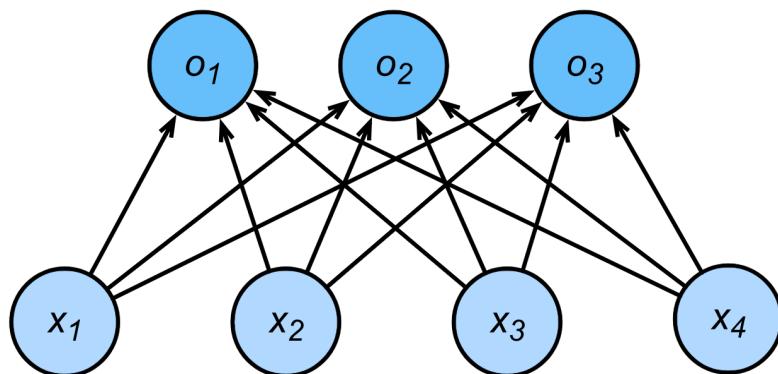
Regression

- 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

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

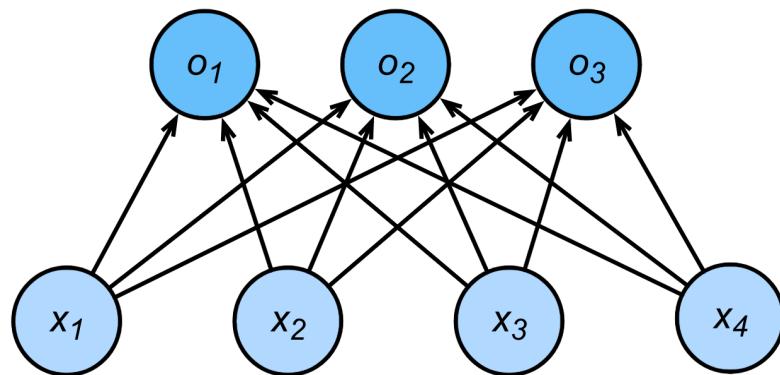
$$y_i = \begin{cases} 1 & \text{if } i = y \\ 0 & \text{otherwise} \end{cases}$$

- Train with squared loss
- Largest output wins

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

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

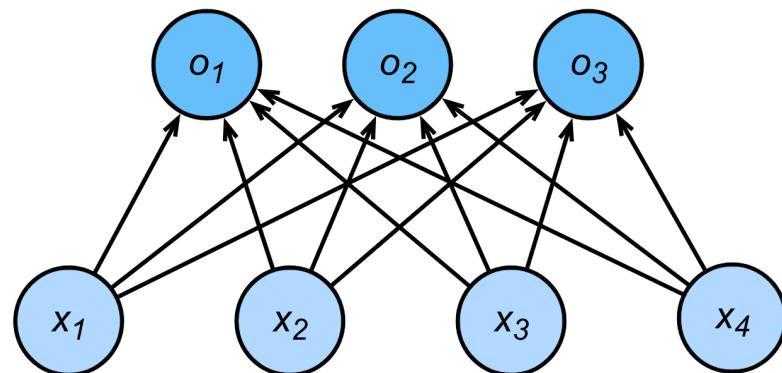
$$\hat{y} = \operatorname{argmax}_i o_i$$

- 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)

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

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

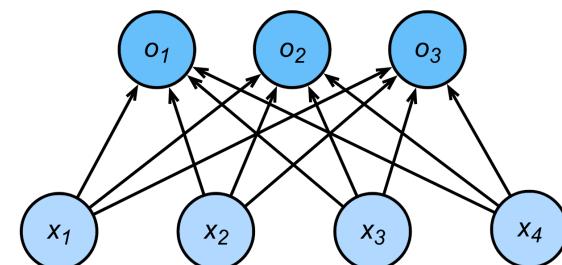
- Negative log-likelihood

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

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

Classification

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



Summary

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