# Lecture 5:
# Logistic Regression Continued

*Instructor: Swabha Swayamdipta*
*USC CSCI 499 LMs in NLP*
*Jan 31, Spring 2024*

Slides mostly adapted from Dan Jurafsky, some from Mohit Iyyer

## Logistics / Announcements

- HW1 due tonight by 12pm
- Graded Quiz 1 will be distributed next Monday
- Project Proposal (1 page) due next Wednesday

- What makes for a good project proposal?
  - Problem / Task Definition: Input and expected output
    - With examples
    - Motivation: Why is this an important problem?
  - Sources of data for training a model for your task (very important!)
    - How many instances for train / test / validation?
  - Models to be compared
    - Ideally, those taught in class
  - Evaluation Framework:
    - What test set?
    - What metric?
    - What baselines for comparison?

## Lecture Outline

- Recap: Supervised Classification / Logistic Regression
  - Data (Features)
  - Model (Logistic Regression)
  - Loss (Cross Entropy)
- Today: Logistic Regression Contd.
  - Optimization
  - Regularization
  - Multinomial Logistic Regression
- Word Embeddings (Teaser)

## Recap:
## Supervised Machine Learning

# Ingredients of Supervised Machine Learning

I. **Data** as pairs $(x^{(i)}, y^{(i)})$ s.t $i \in \{1 \ldots N\}$
   - $x^{(i)}$ usually represented by a feature vector $\mathbf{x}^{(i)} = [x_1, x_2, \ldots, x_d]$,
     - e.g. word embeddings

II. **Model**
   - A classification function that computes $\hat{y}$, the estimated class, via $p(y \mid x)$
     - e.g. sigmoid function: $\sigma(z) = 1/(1 + \exp(-z))$

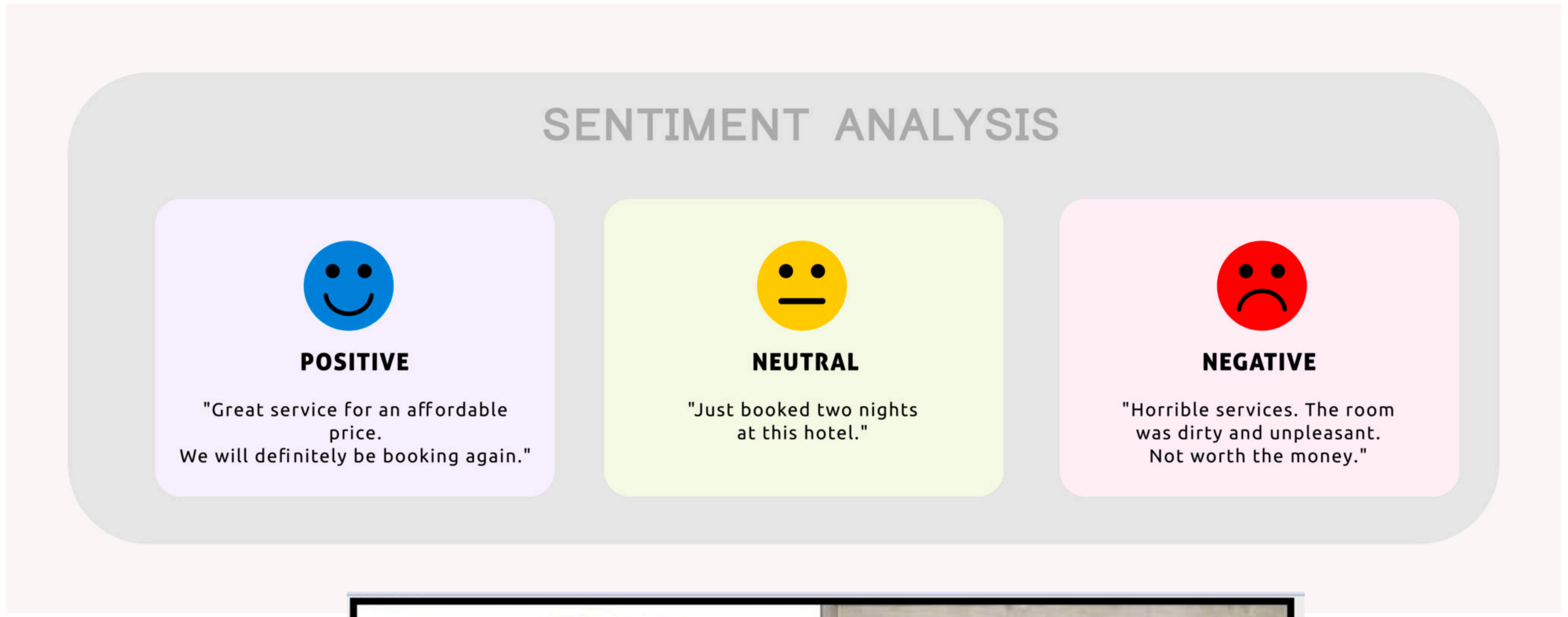
III. **Loss**
   - An objective function for learning
     - e.g. cross-entropy loss, $L_{CE}$

IV. **Optimization**
   - An algorithm for optimizing the objective function
     - e.g. stochastic gradient descent

V. **Inference** / Evaluation

Learning Phase

# Text Classification Tasks

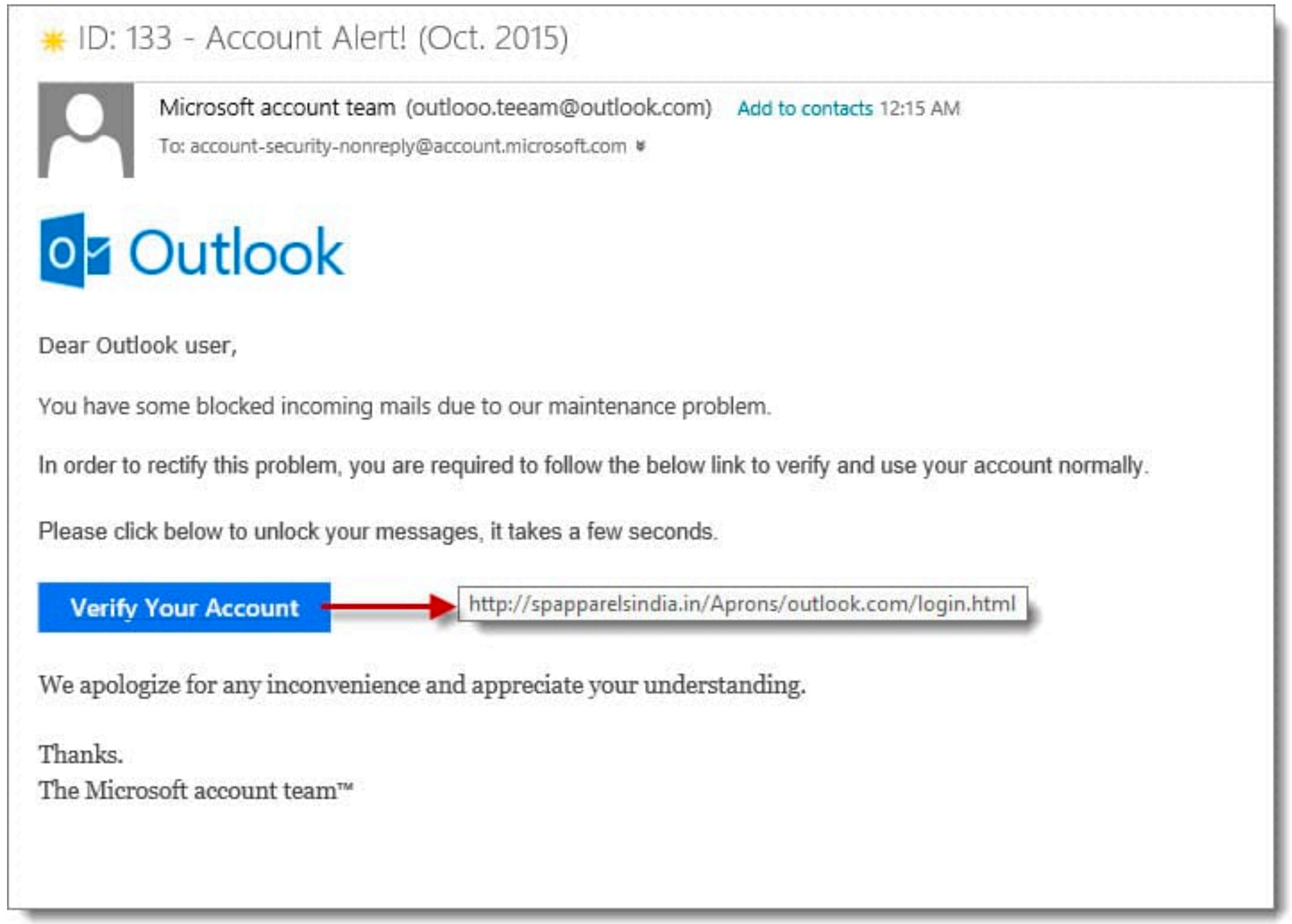


SENTIMENT ANALYSIS

POSITIVE
"Great service for an affordable price.
We will definitely be booking again."

NEUTRAL
"Just booked two nights at this hotel."

NEGATIVE
"Horrible services. The room was dirty and unpleasant.
Not worth the money."



ID: 133 - Account Alert! (Oct. 2015)

Microsoft account team (outlooo.teeam@outlook.com) Add to contacts 12:15 AM
To: account-security-nonreply@account.microsoft.com

Outlook

Dear Outlook user,

You have some blocked incoming mails due to our maintenance problem.

In order to rectify this problem, you are required to follow the below link to verify and use your account normally.

Please click below to unlock your messages, it takes a few seconds.

Verify Your Account → http://spapparelsindia.in/Aprons/outlook.com/login.html

We apologize for any inconvenience and appreciate your understanding.

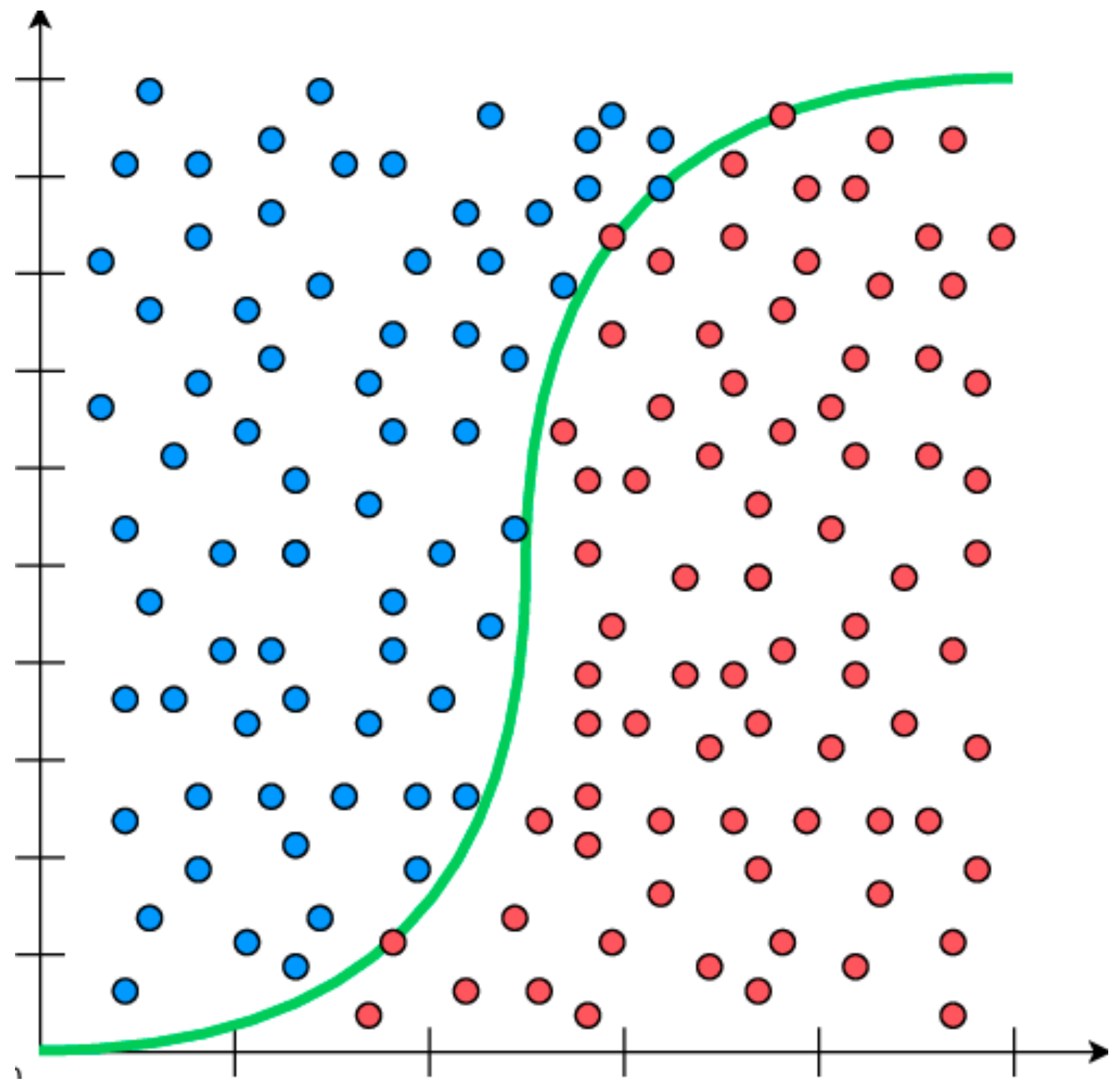
Thanks.
The Microsoft account team™

Mr. WILLIAM SHAKESPEARES COMEDIES, HISTORIES, & TRAGEDIES.
Published according to the True Originall Copies.
THE FEDERALIST: A COLLECTION OF ESSAYS, WRITTEN IN FAVOUR OF THE
Signature

Not just NLP, classification is a general ML technique often applied across a wide variety of prediction tasks!

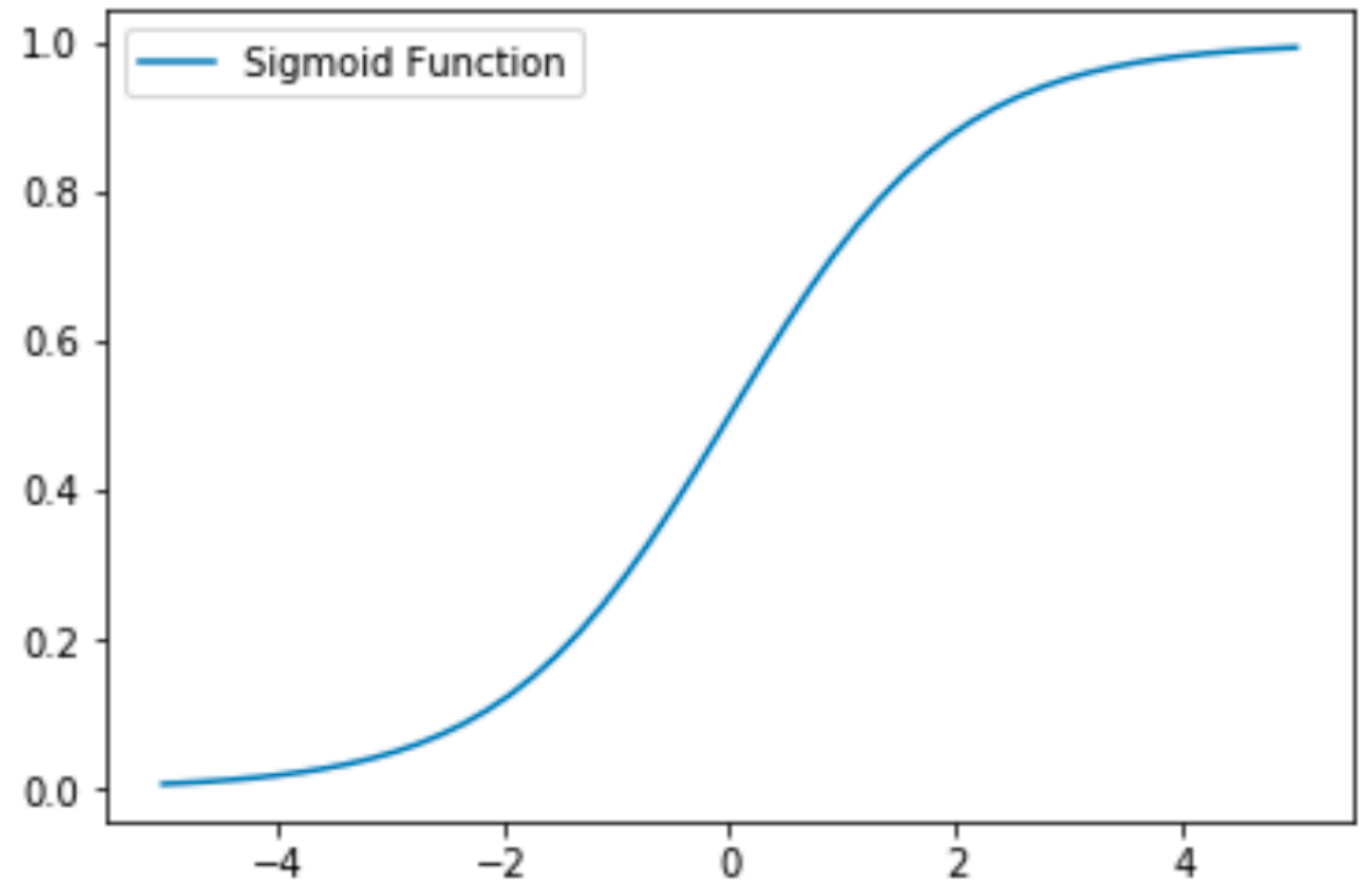
## Text Classification Setup

- Input:
  - a document $x$
    - Each observation $x^{(i)}$ is represented by a feature vector $\mathbf{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_d^{(i)}]$
  - a label $y$ from a fixed set of classes $C = c_1, c_2, \ldots, c_J$
    - Sometimes denoted by $y*$ for gold-standard (ground truth) label
- Output: a predicted class $\hat{y} \in C$
- Setting for Binary Classification: given a series of input / output pairs:
  - $(x^{(i)}, y^{(i)})$ where label $y^{(i)} \in C = \{0,1\}$
- Goal of Binary Classification
  - At test time, for input $x^{test}$, compute an output: a predicted class $\hat{y}^{test} \in \{0,1\}$

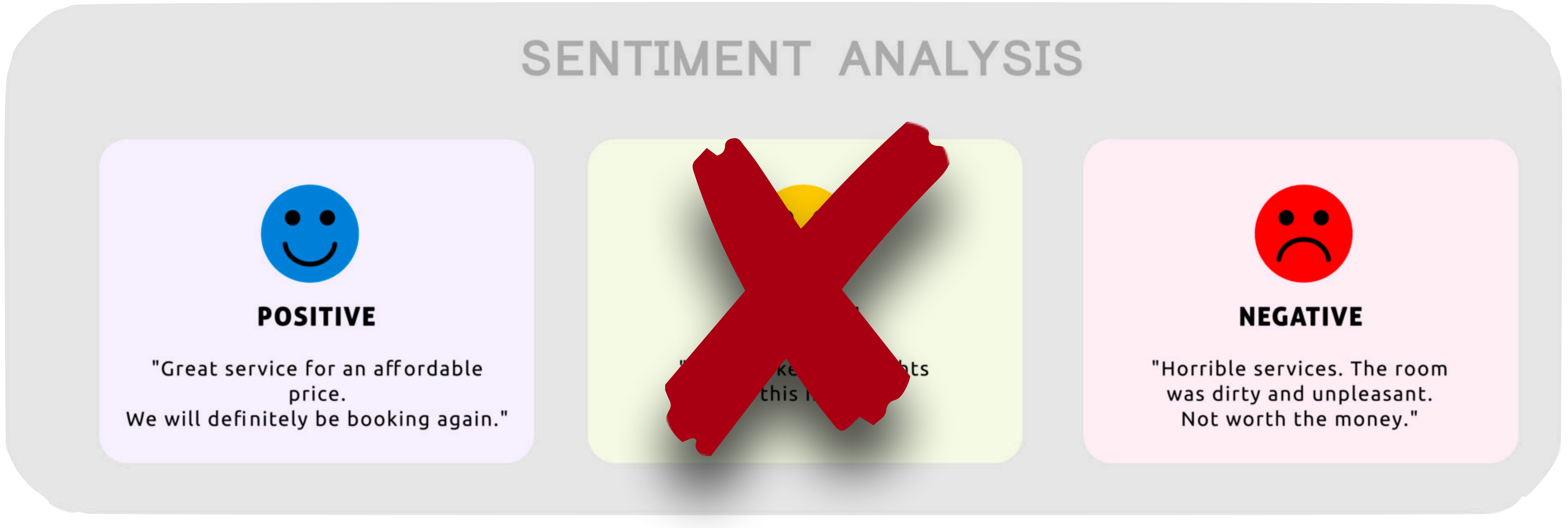


## One flavor of Classification: Logistic Regression

- Baseline supervised machine learning tool for classification; the foundation of neural networks
- Other classification algorithms: Naïve Bayes, K-Nearest Neighbors, Decision Trees, SVMs
- Input observation: vector of features, $\mathbf{x} = [x_1, x_2, \ldots, x_d]$
- Model: One weight per feature: $\mathbf{w} = [w_1, w_2, \ldots, w_d]$
- Output: a predicted class
    - Binary logistic regression $\hat{y} \in \{0,1\}$
    - Multinomial logistic regression (e.g. 5 classes): $\hat{y} \in \{0,1,2,3,4\}$



# Case Study: Sentiment Classification


SENTIMENT ANALYSIS
POSITIVE
"Great service for an affordable price. We will definitely be booking again."
NEGATIVE
"Horrible services. The room was dirty and unpleasant. Not worth the money."

- Data: Features $x_i$ of the input feature
  - $x_i =$ "review contains 'awesome'"; $w_i = +10$
  - $x_j =$ "review contains 'abysmal'"; $w_j = -10$
  - $x_k =$ "review contains 'mediocre'"; $w_k = -2$
- Features can be manually designed (classic) or learned (modern / neural architectures)
- Each $x_i$ is associated with a weight $w_i$ which determines how important $x_i$ is
  - (For predicting the positive class)
- Labels $y \in \{0,1\}$
  - $y = 0$ represents negative sentiment and
  - $y = 1$ represents positive sentiment

## How to get the right $y$?

- For each feature $x_i$, introduce a weight $w_i$, which determines the importance of $x_i$
  - Sometimes we have a bias term, $b$ or $w_0$, which is just another weight not associated to any feature
  - Together, all parameters can be termed as $\theta = [w; b]$
- We consider the weighted sum of all features and the bias

$$z = \left( \sum_d w_d x_d + b \right)$$

$$= \mathbf{w} \cdot \mathbf{x} + b$$

If high, $\hat{y} = 1$ If low, $\hat{y} = 0$

But how to determine the threshold?

We need probabilistic models!

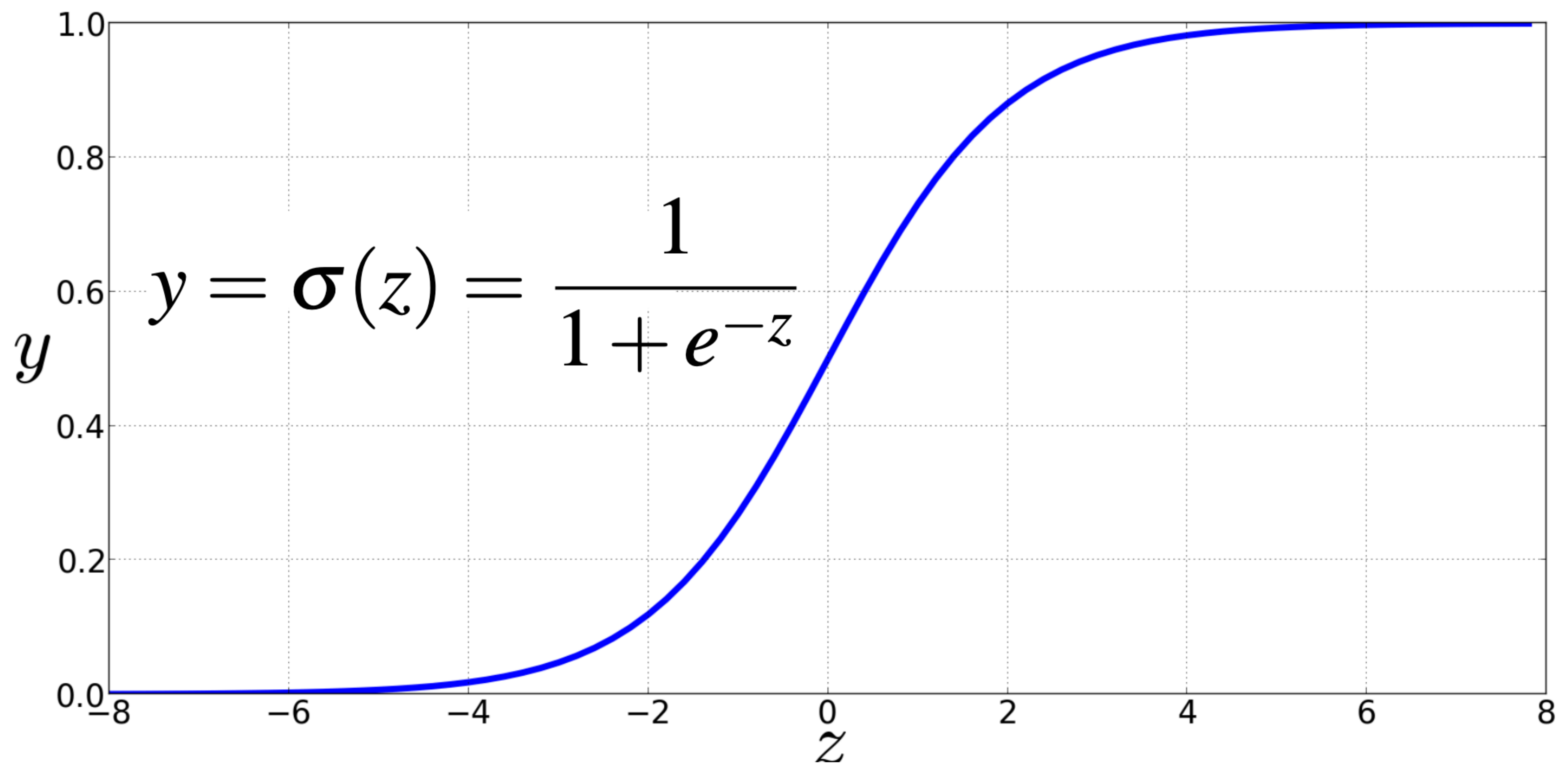
$$P(y = 1 \mid \mathbf{x}; \theta)$$

$$P(y = 0 \mid \mathbf{x}; \theta)$$

## Solution: Squish it into the 0-1 range

$$z = \mathbf{w} \cdot \mathbf{x} + b \qquad z \in \mathbb{R}$$

- Sigmoid Function, $\sigma(\cdot)$
  - Non-linear!
- Compute $z$ and then pass it through the sigmoid function
- Treat it as a probability!
- Also, a differentiable function, which makes it a good candidate for optimization (more on this later!)



## Sigmoids and Probabilities

$$P(y = 1 \mid \mathbf{x}; \theta) = \sigma(\mathbf{w} \cdot \mathbf{x} + b) = \frac{1}{1 + \exp(-(\mathbf{w} \cdot \mathbf{x} + b))}$$

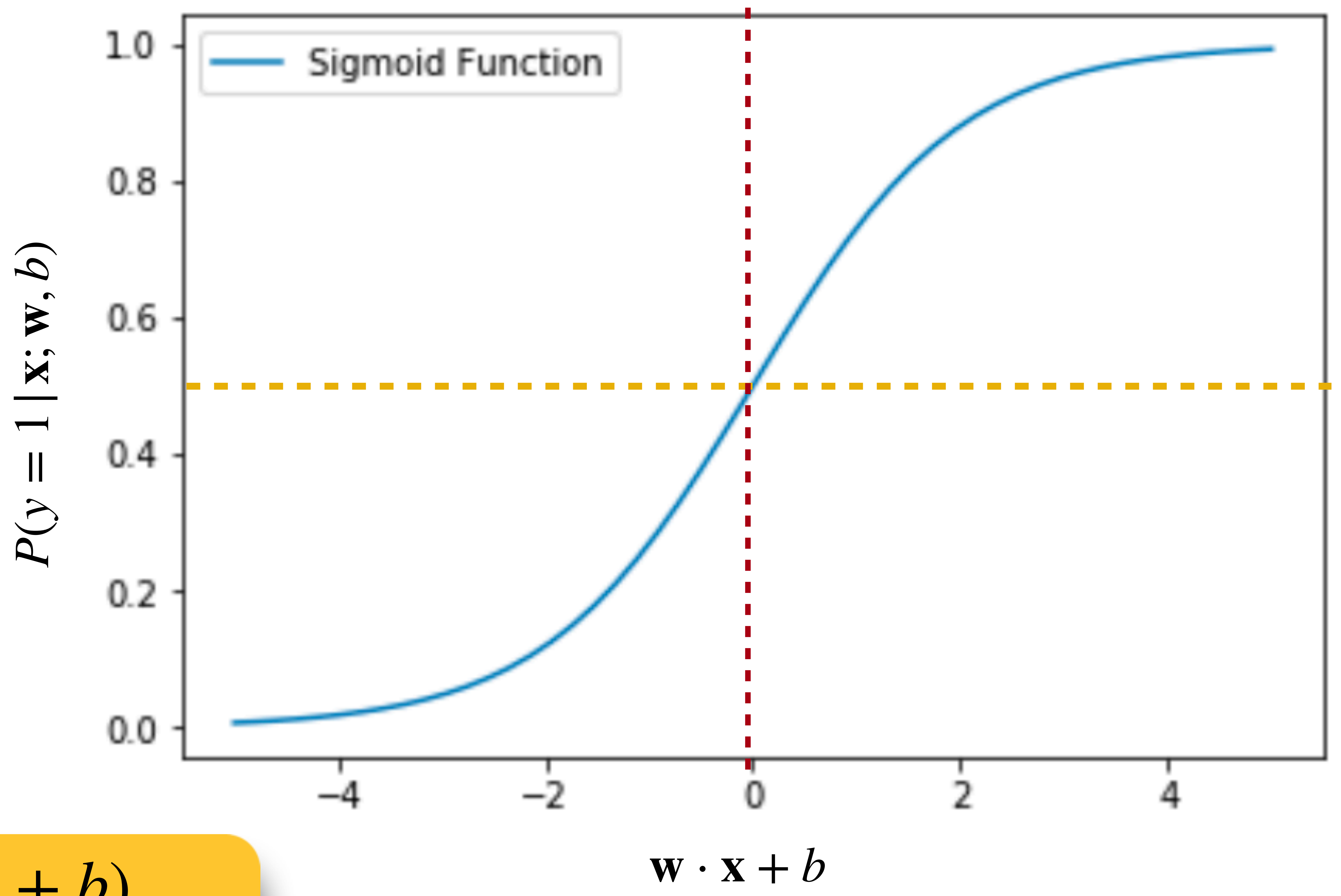
$$\begin{aligned} P(y = 0 \mid \mathbf{x}; \theta) &= 1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b) \\ &= 1 - \frac{1}{1 + \exp(-(\mathbf{w} \cdot \mathbf{x} + b))} \\ &= \frac{\exp(-(\mathbf{w} \cdot \mathbf{x} + b))}{1 + \exp(-(\mathbf{w} \cdot \mathbf{x} + b))} \\ &= \frac{1}{1 + \exp(\mathbf{w} \cdot \mathbf{x} + b)} \\ &= \sigma(-(\mathbf{w} \cdot \mathbf{x} + b)) \end{aligned}$$

# Classification Decision

$$\hat{y} = \begin{cases} 1 & \text{if } p(y = 1 \mid x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Decision Boundary

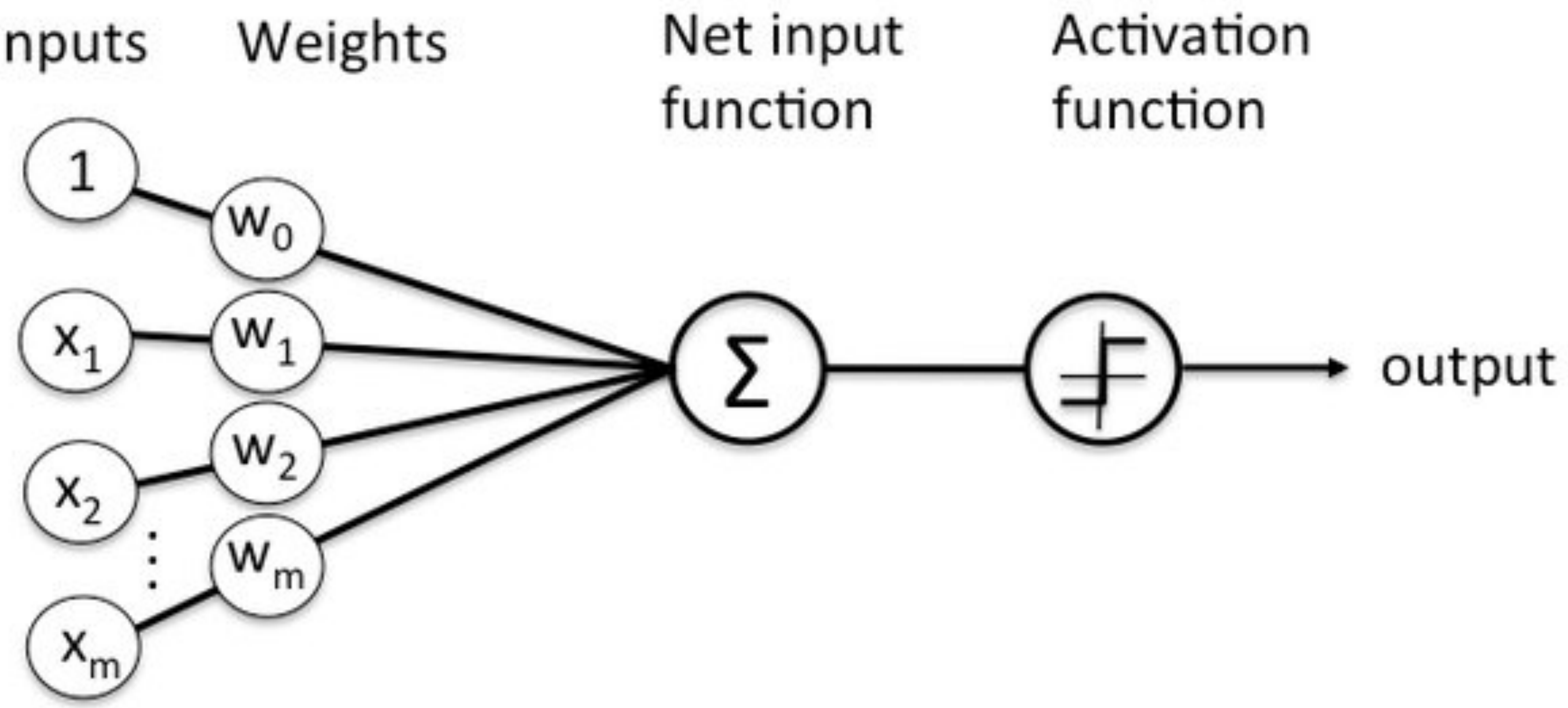
$$\hat{y} = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0 & \text{if } \mathbf{w} \cdot \mathbf{x} + b \leq 0 \end{cases}$$



Often we will denote $\hat{y}$ directly as $\sigma(\mathbf{w} \cdot \mathbf{x} + b)$

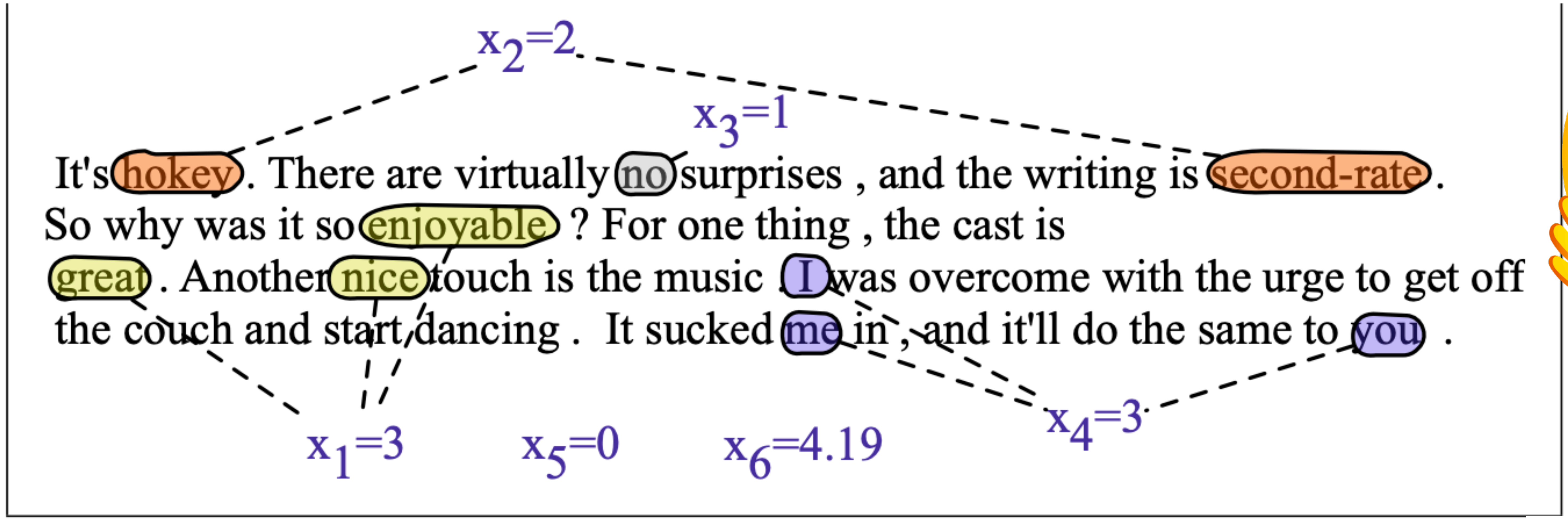
# Another notation

$$z = \left( \sum_m w_m x_m + w_0 \right)$$
$$= \mathbf{w} \cdot \mathbf{x}$$



$$\hat{y} = P(y = 1 \mid \mathbf{x}; \mathbf{w}) = \sigma(z)$$
$$= \frac{1}{1 + \exp(-z)}$$

# Example: Sentiment Classification


x2=2
x3=1
It's hokey . There are virtually no surprises , and the writing is second-rate .
So why was it so enjoyable ? For one thing , the cast is
great . Another nice touch is the music . I was overcome with the urge to get off
the couch and start dancing . It sucked me in , and it'll do the same to you .
x1=3
x5=0
x6=4.19
x4=3

$$\hat{y} = \begin{cases} 1 & \textbf{if } p(y=+\,|\,x) > 0.5 \\ 0 & \textbf{otherwise} \end{cases}$$

| Var | Definition | Value in Fig. 5.2 |
|---|---|---|
| $x_1$ | count(positive lexicon) $\in$ doc) | 3 |
| $x_2$ | count(negative lexicon) $\in$ doc) | 2 |
| $x_3$ | $\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$ | 1 |
| $x_4$ | count(1st and 2nd pronouns $\in$ doc) | 3 |
| $x_5$ | $\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$ | 0 |
| $x_6$ | log(word count of doc) | $\ln(66) = 4.19$ |

$$p(+|x) = P(Y=1|x)$$

$$\begin{aligned} &= \sigma(w \cdot x + b) \\ &= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\ &= \sigma(.833) \\ &= 0.70 \end{aligned}$$

$$\begin{aligned} p(-|x) = P(Y=0|x) &= 1 - \sigma(w \cdot x + b) \\ &= 0.30 \end{aligned}$$

## Model = the $\mathbf{w}$'s and the $b$'s

- Supervised Classification:
  - We know the correct label $y$ (either 0 or 1) for each $x$
  - But what the system produces is an estimate, $\hat{y}$
- Set $\mathbf{w}$ and $b$ to minimize the **distance** between our estimate $\hat{y}^{(i)}$ and the true $y^{(i)}$
  - We need a distance estimator: a **loss function** or a **cost function**
  - We need an **optimization algorithm** to update $\mathbf{w}$ and $b$ to minimize the loss.

Loss function

Optimization Algorithm

# Loss: the distance between $\hat{y}$ and $y$

- We want to know how far is the classifier output $\hat{y} = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$ is from the true (ground truth / gold standard) label, $y \in \{0,1\}$

- This difference is called the loss or cost, $L(\hat{y}, y)$ = how much $\hat{y}$ differs from $y$
  - Loss indicates how much would you lose if you mispredicted or how much would it cost you to mispredict

- We want the distance to be as small as possible
  - Hence, we choose the parameters $\mathbf{w}, b$ that maximize the log probability of the true labels $y$ in the training data, **given** the observations $x$
  - Maximum likelihood estimate for $p(y \mid \mathbf{x}; \mathbf{w}, b)$

# Maximizing conditional likelihood

**Goal**: maximize probability of the correct label $p(y|\mathbf{x})$

Maximize: $p(y|x) = \hat{y}^y(1-\hat{y})^{1-y}$

Or, maximize: $\log p(y|x) = \log(\hat{y}^y(1-\hat{y})^{1-y})$

$$= y\log\hat{y} + (1-y)\log(1-\hat{y})$$

Whatever values maximize $\log p(y|x)$ will also maximize $p(y|x)$

Now flip the sign for something to minimize (we minimize the loss / cost)

Minimize: $L_{CE}(y,\hat{y}) = -\log p(y|x) = -[y\log\hat{y} + (1-y)\log(1-\hat{y})]$

$$= -[y\log\sigma(\mathbf{w}\cdot\mathbf{x}+b) + (1-y)\log\sigma(-(\mathbf{w}\cdot\mathbf{x}+b))]$$

Cross-Entropy Loss

# Case Study: Sentiment Classification

We want loss to be:

- smaller if the model estimate is close to correct
- bigger if model is confused

It's hokey. There are virtually no surprises , and the writing is second-rate. So why was it so enjoyable? For one thing, the cast is great. Another nice touch is the music. I was overcome with the urge to get off the couch and start dancing. It sucked me in, and it'll do the same to you.

The loss when the model is right (if true $y = 1$):

$$\begin{aligned} L_{\mathrm{CE}}(\hat{y},y) &= -[y\log\sigma(w\cdot x+b)+(1-y)\log(1-\sigma(w\cdot x+b))] \\ &= -[\log\sigma(w\cdot x+b)] \\ &= -\log(.70) \\ &= .36 \end{aligned}$$

..is lower than the loss when the model was wrong (if true $y = 0$)

$$\begin{aligned} L_{\mathrm{CE}}(\hat{y},y) &= -[y\log\sigma(w\cdot x+b)+(1-y)\log(1-\sigma(w\cdot x+b))] \\ &= -[\log(1-\sigma(w\cdot x+b))] \\ &= -\log(.30) \\ &= 1.2 \end{aligned}$$

Loss is bigger when the model is wrong!

## Lecture Outline

- Recap: Supervised Classification / Logistic Regression
  - Data (Features)
  - Model (Logistic Regression)
  - Loss (Cross Entropy)
- Today: Logistic Regression Contd.
  - Optimization
  - Regularization
  - Multinomial Logistic Regression
- Word Embeddings (Teaser)

## IV. Optimization: Stochastic Gradient Descent

# Our goal: minimize the loss

- Loss function is parameterized by weights: $\theta = [\mathbf{w}; b]$
- We will represent $\hat{y}$ as $f(x; \theta)$ to make the dependence on $\theta$ more obvious

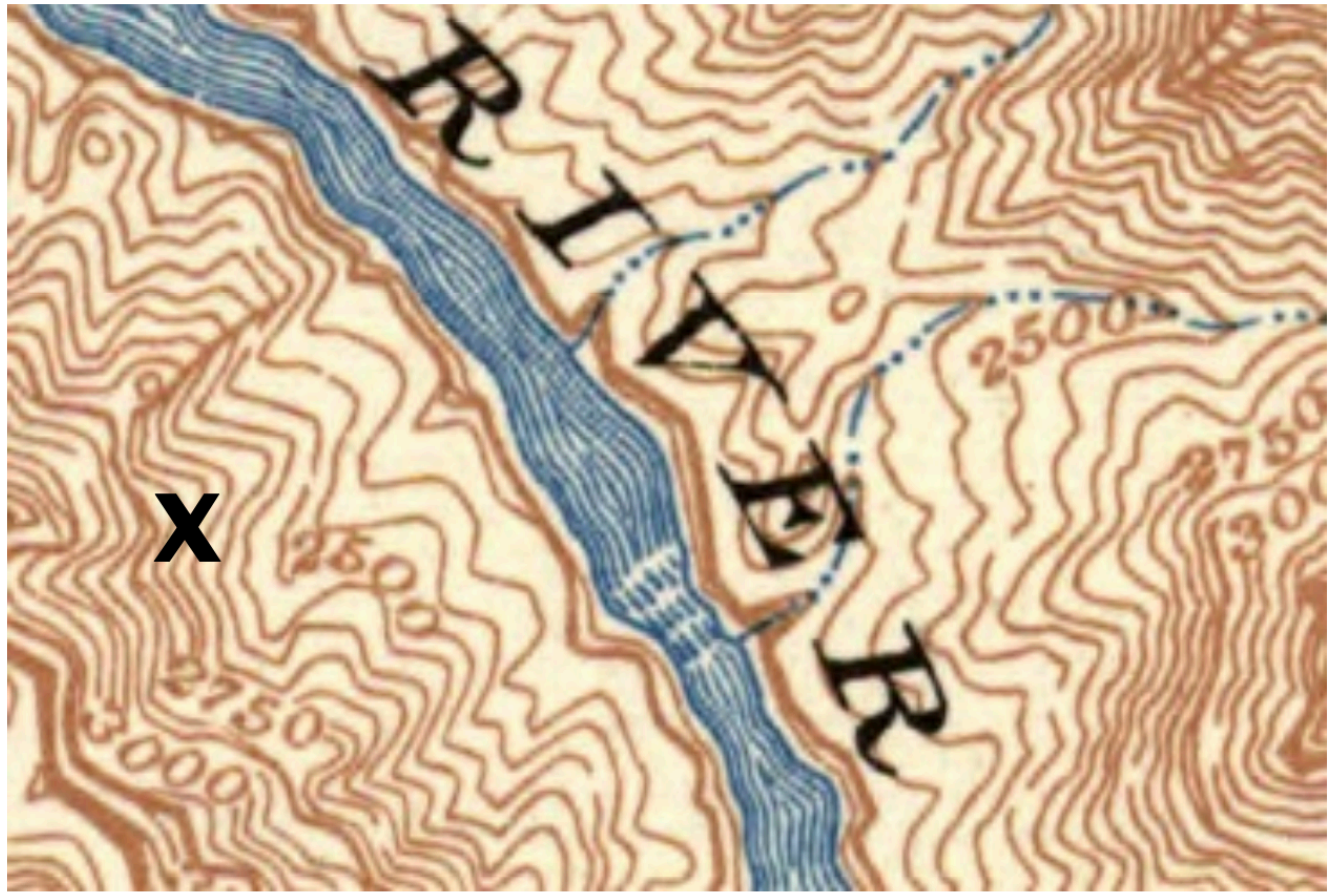
We want the weights that minimize the loss, averaged over all examples:

$$L_{CE}(f(\mathbf{x}^{(i)}; \theta), y^{(i)})$$

# Intuition for gradient descent

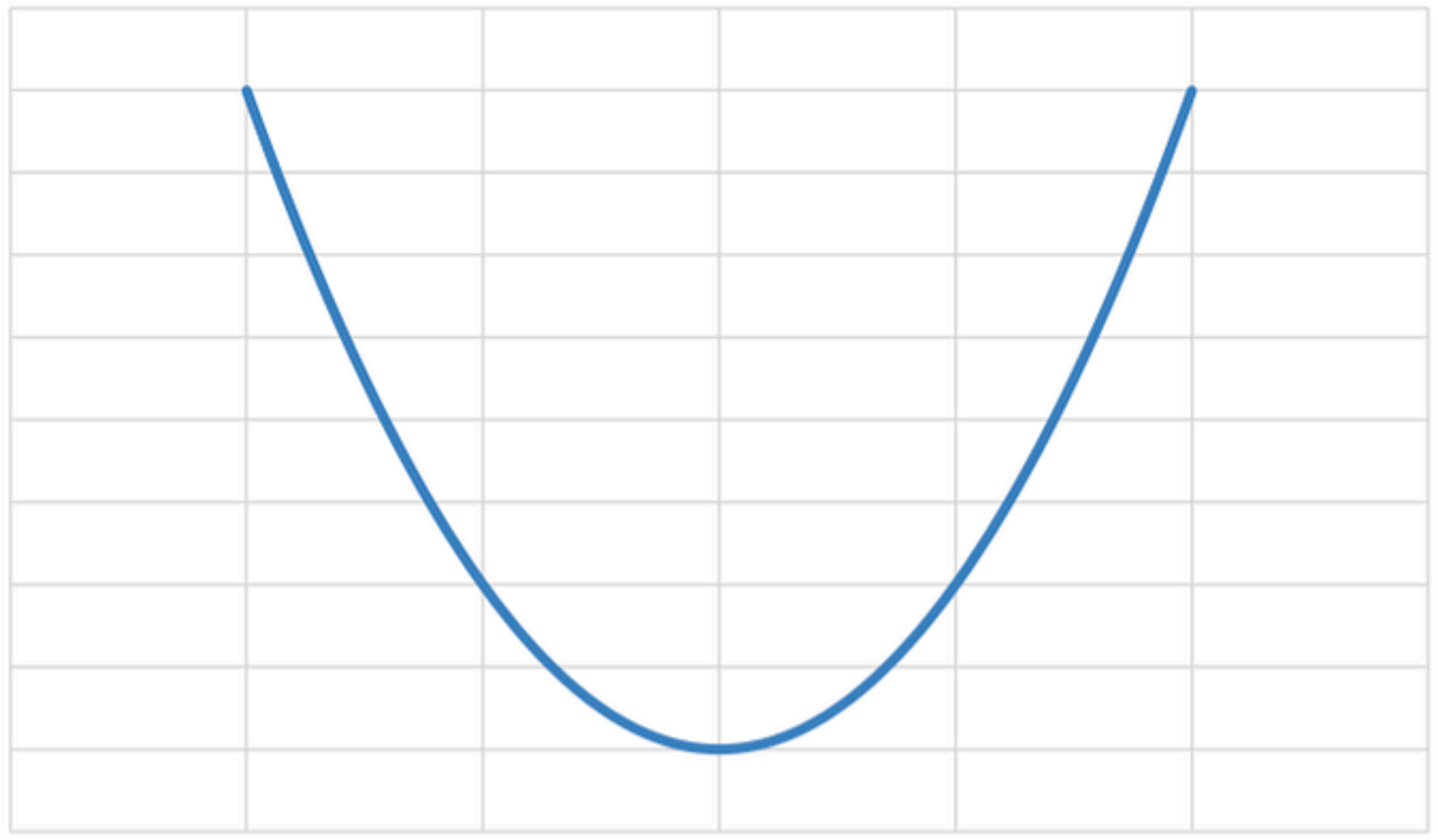
How to get to the bottom of the river canyon?

- Look around 360°
- Find the direction of steepest slope down
- Go that way



What if multiple equally good alternatives?

# Logistic Regression: Loss



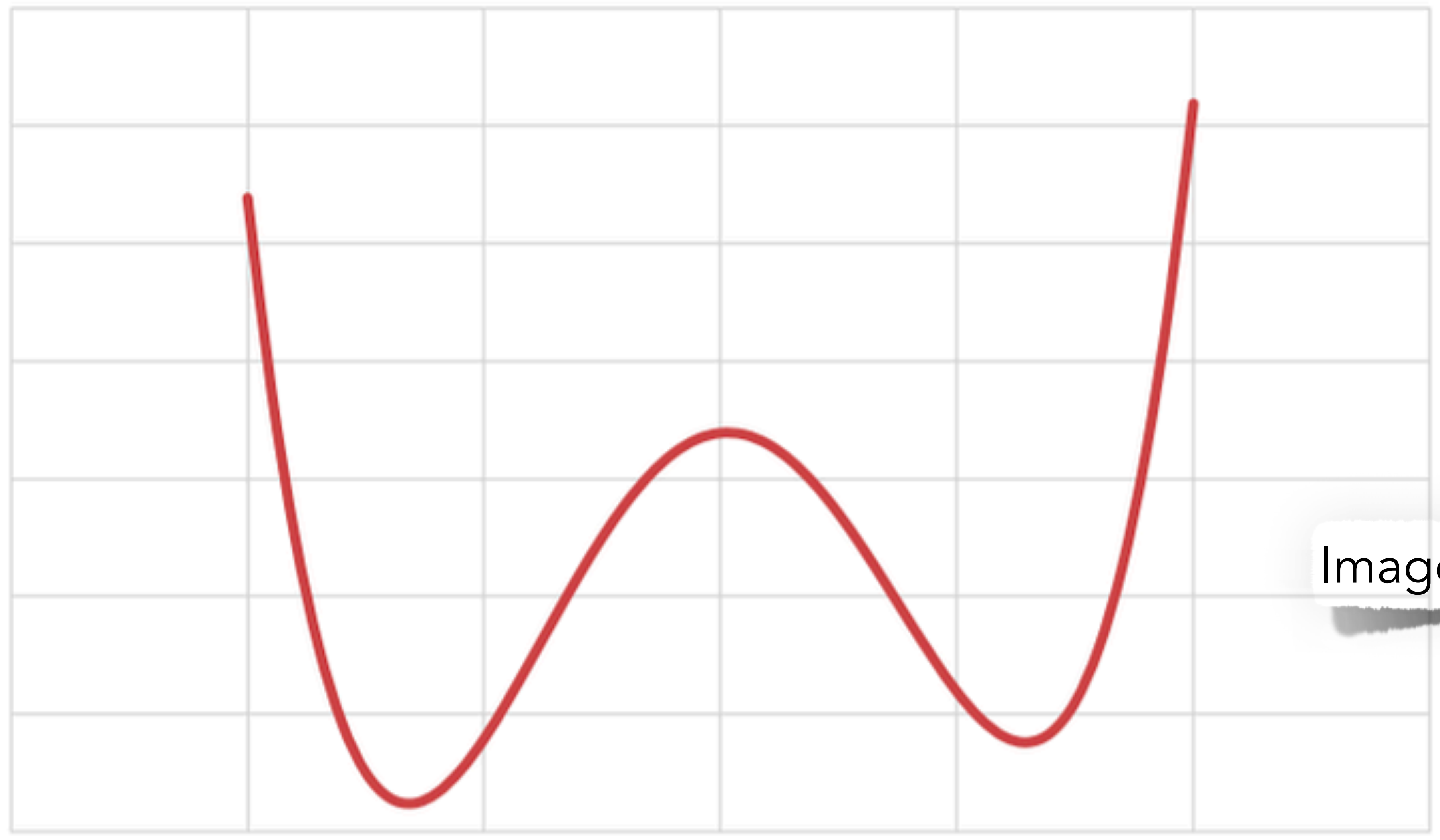


Image Credit: Medium

Convex function

Non-convex function

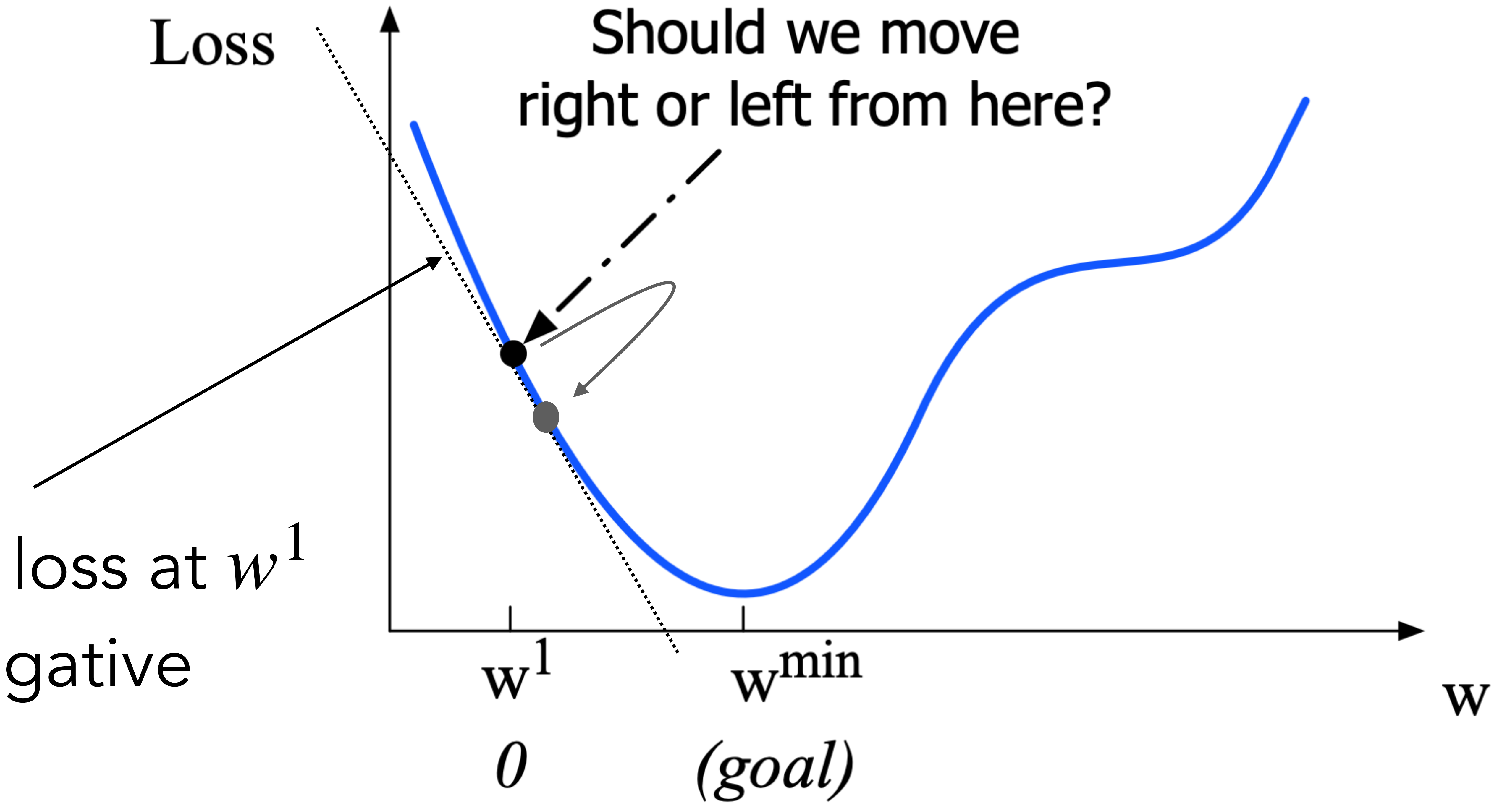
- Has only one option for steepest gradient
  - Or one minimum
- Gradient descent starting from any point is guaranteed to find the minimum

Neural Networks - multiple alternatives

# Consider: a single scalar $w$

Given current $w$, should we make it bigger or smaller?

Move $w$ in the reverse direction from the slope of the function



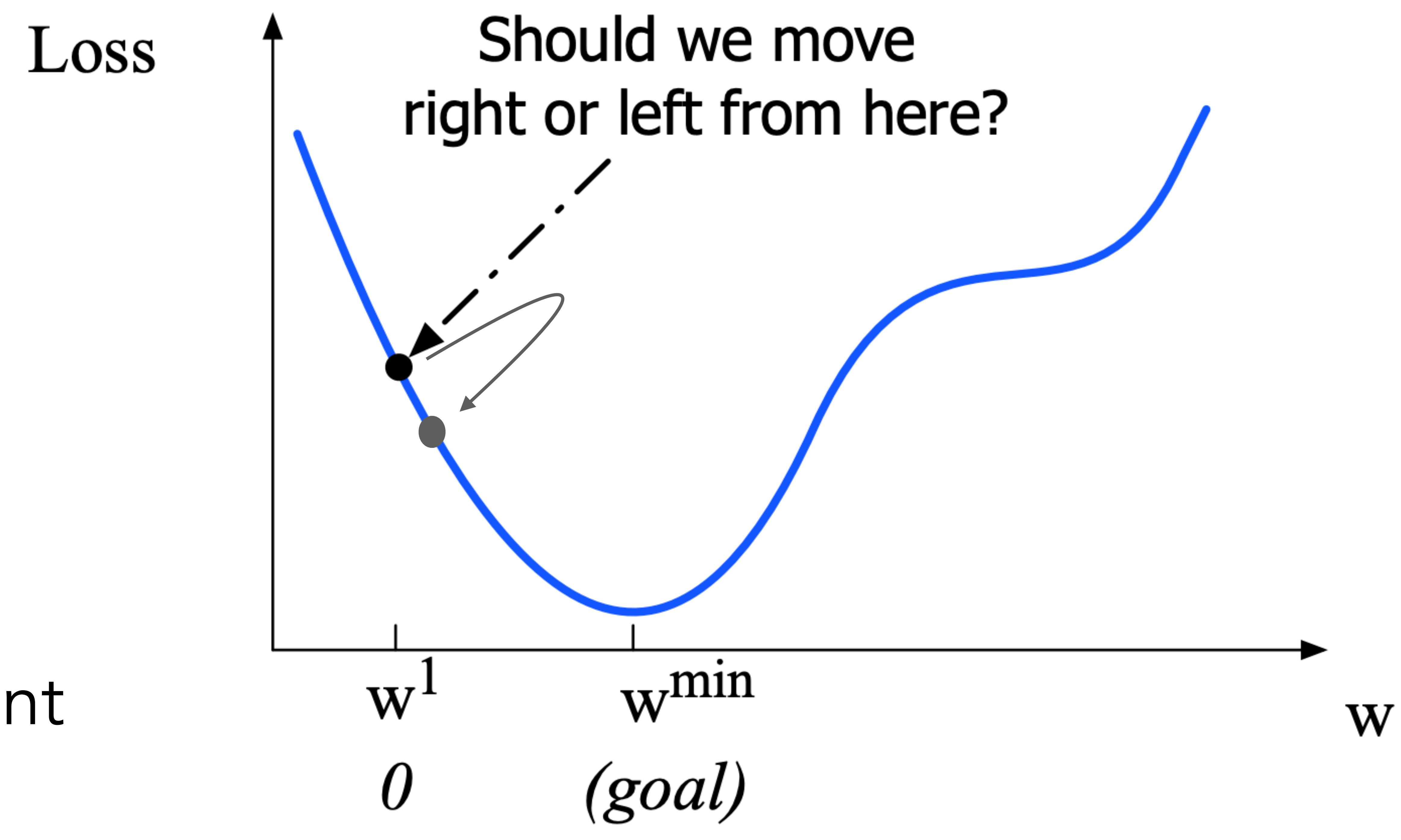
slope of loss at $w^1$ is negative

need to move positive

# Gradients

The **gradient** of a function of many variables is a vector pointing in the direction of the greatest increase in a function.

Find the gradient of the loss function at the current point and move in the **opposite** direction.



Gradient Descent

But by how much?

## Gradient Updates

- Move the value of the gradient $\frac{\partial}{\partial w} L(f(x; w), y^*)$, weighted by a learning rate $\eta$
- Higher learning rate means move $w$ faster

**$\eta$ Too high: the learner will take big steps and overshoot**

$$w_{t+1} = w_t - \eta \frac{\partial}{\partial w} L(f(x; w), y^*)$$

**$\eta$ Too low: the learner will take too long**

If parameter $\theta$ is a vector of $d$ dimensions:

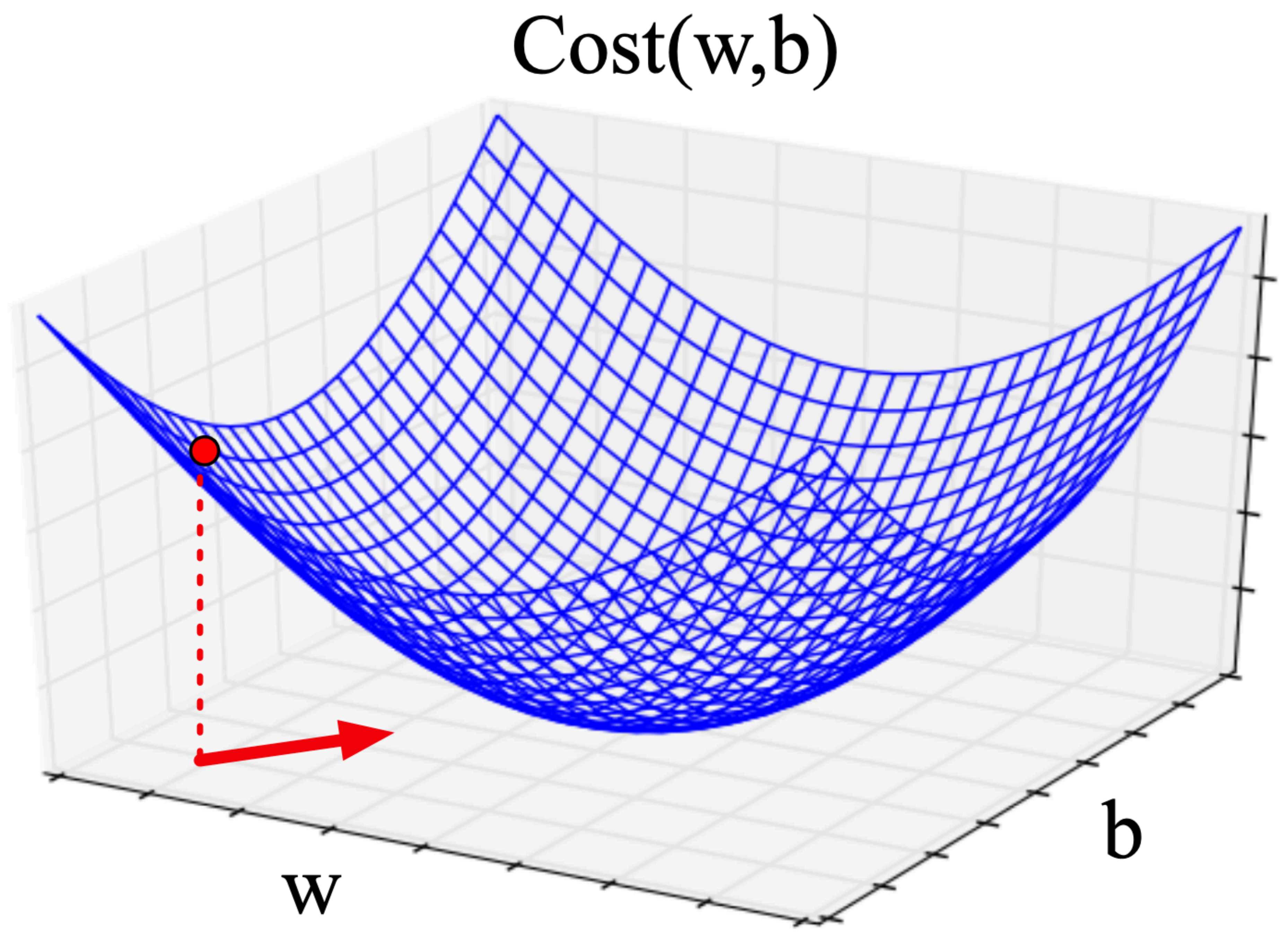
The gradient is just such a vector; it expresses the directional components of the sharpest slope along each of the $d$ dimensions.

# Under 2 dimensions

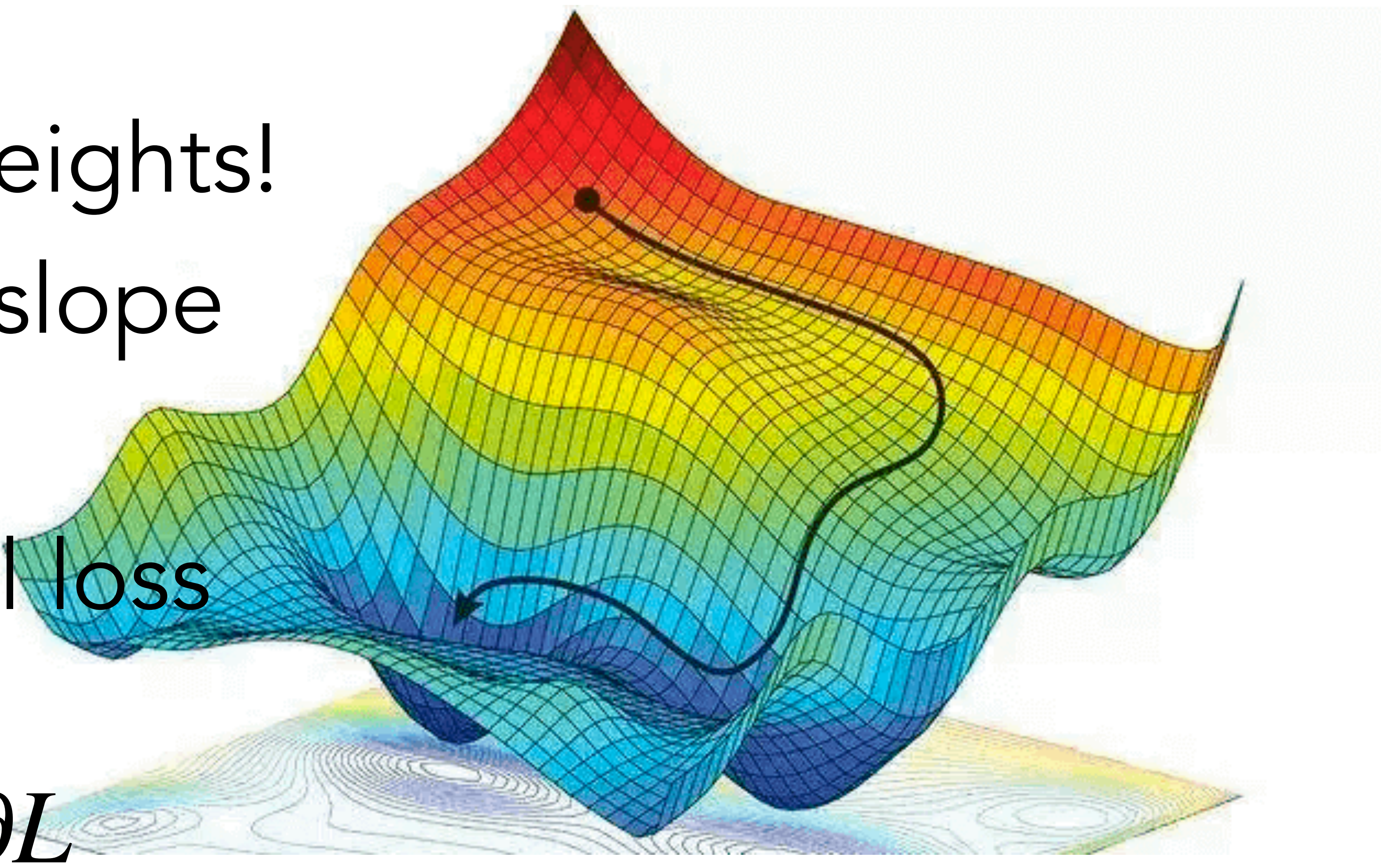
Consider 2 dimensions, $w$ and $b$:

Visualizing the gradient vector at the red point

It has two dimensions shown in the x-y plane



# Real-life gradients, however…



- …are much longer; models usually contain lots and lots of weights!
- For each dimension $\theta_i$ the gradient component $i$ tells us the slope with respect to that variable
  - "How much would a small change in $\theta_i$ influence the total loss function $L$?"
- We express the slope as a partial derivative $\frac{\partial}{\partial \theta_i}$ of the loss, $\frac{\partial L}{\partial \theta_i}$
  - The gradient is then defined as a vector of these partials

# Real-life gradients

We will represent $\hat{y}$ as $f(x; \theta)$ to make the dependence on $\theta$ more obvious

$$\nabla_{\theta} L(f(x;\theta), y)) = \begin{bmatrix} \frac{\partial}{\partial w_1} L(f(x;\theta), y) \\ \frac{\partial}{\partial w_2} L(f(x;\theta), y) \\ \vdots \\ \frac{\partial}{\partial w_n} L(f(x;\theta), y) \end{bmatrix}$$

The final equation for updating $\theta$ at time step $t+1$ based on the gradient is thus:

$$\theta_{t+1} = \theta_t - \eta \frac{\partial}{\partial \theta} L(f(x;\theta), y)$$

# Gradients for Logistic Regression

Case: Sentiment Analysis

Recall: the cross-entropy loss for logistic regression

$$L_{CE}(y, \hat{y}) = -\left[y \log \sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1 - y)\log(\sigma(-\mathbf{w} \cdot \mathbf{x} + b))\right]$$

Derivatives have a closed form solution:

$$\frac{\partial L_{CE}(y, \hat{y})}{\partial w_j} = [\sigma(\mathbf{w} \cdot \mathbf{x} + b) - y]x_j$$

## Pseudocode

```
function STOCHASTIC GRADIENT DESCENT ($L(), f(), x, y$) returns $\theta$
  # where: $L$ is the loss function
  #        $f$ is a function parameterized by $\theta$
  #        $\mathbf{x}$ is the set of training inputs $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots \mathbf{x}^{(N)}$
  #        $y$ is the set of training outputs (labels) $y^{(1)}, y^{(2)}, \ldots y^{(N)}$
$\theta \leftarrow 0$  (or randomly initialized)
repeat till done
  for each training tuple$(x^{(i)}, y^{(i)})$: (in random order)
        1. Compute $\hat{y}^{(i)} = f(\mathbf{x}^{(i)}; \theta)$          # What is our estimated output $\hat{y}^{(i)}$?
        2. Compute the loss $L(\hat{y}^{(i)}, y^{(i)})$     # How far off is $\hat{y}^{(i)}$ from the true output $y^{(i)}$ ?
        3. $g \leftarrow \nabla L(f(\mathbf{x}^{(i)}; \theta), y^{(i)})$       # How should we move $\theta$ to maximize loss?
        4. $\theta \leftarrow \theta - \eta g$                  # Go the other way instead
return $\theta$
```

Stochastic Gradient Descent

## Mini-Batching

```
function STOCHASTIC GRADIENT DESCENT $(L(), f(), x, y, m)$ returns $\theta$
  # where: $L$ is the loss function
  #        $f$ is a function parameterized by $\theta$
  #        $\mathbf{x}$ is the set of training inputs $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots \mathbf{x}^{(N)}$
  #        $y$ is the set of training outputs (labels) $y^{(1)}, y^{(2)}, \ldots y^{(N)}$ and $m$ is the mini-batch size
$\theta \leftarrow 0$  (or randomly initialized)
repeat till done
  for each randomly sampled minibatch of size $m$:
      1. for each training tuple$(\mathbf{x}^{(i)}, y^{(i)})$ in the minibatch: (in random order)
          i.  Compute $\hat{y}^{(i)} = f(\mathbf{x}^{(i)}; \theta)$                    # What is our estimated output $\hat{y}^{(i)}$?
          ii. Compute the loss $L_{mini} \leftarrow L_{mini} + L(\hat{y}^{(i)}, y^{(i)})$   # How far off is $\hat{y}^{(i)}$ from the true output $y^{(i)}$ ?
      2. $g \leftarrow \frac{1}{m} \nabla L_{mini}(f(\mathbf{x}^{(i)}; \theta), y^{(i)})$              # How should we move $\theta$ to maximize loss?
      3. $\theta \leftarrow \theta - \eta g$                                  # Go the other way instead
return $\theta$
```

## Lecture Outline

- Recap: Supervised Classification / Logistic Regression
  - Data (Features)
  - Model (Logistic Regression)
  - Loss (Cross Entropy)
- Today: Logistic Regression Contd.
  - Optimization
  - Regularization
  - Multinomial Logistic Regression
- Word Embeddings (Teaser)

## Regularization

# Overfitting

- A model that perfectly match the training data has a problem
- It will also overfit to the data, modeling noise
  - A random word that perfectly predicts $y$ (it happens to only occur in one class) will get a very high weight
  - Failing to generalize to a test set without this word

Why?

A good model should be able to generalize

What happens when a feature only occurs with one class?

e.g. word "wow" for positive reviews

## Overfitting: Features

This movie drew me in, and it'll do the same to you.

Useful or harmless features

$x_1$ = **"this"**
$x_2$ = **"movie**
$x_3$ = **“hated”**
$x_4$ = **"drew me in"**

I can't tell you how much I hated this movie. It sucked.

4-gram features that just "memorize" training set and might cause problems

$x_5$ = **"the same to you"**
$x_6$ = **"tell you how much”**

## Overfitting

- 4-gram model on tiny data will just memorize the data
  - 100% accuracy on the training set
- But it will be surprised by the novel 4-grams in the test data
  - Low accuracy on test set
- Models that are too powerful can overfit the data
  - Fitting the details of the training data so exactly that the model doesn't generalize well to the test set

How to avoid overfitting?

Regularization in logistic regression

Dropout in neural networks

## Regularization

- A solution for overfitting: Add a regularization term $R(\theta)$ to the loss function
  - (for now written as maximizing logprob rather than minimizing loss)

- Idea: choose an $R(\theta)$ that penalizes large weights
  - fitting the data well with lots of big weights not as good as
  - fitting the data a little less well, with small weights

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log P(y^{(i)} | \mathbf{x}^{(i)}) - \alpha R(\theta)$$

## L2 / Ridge Regularization

- The sum of the squares of the weights
- The name is because this is the (square of the) L2 norm $||\theta||_2^2$, = Euclidean distance of $\theta$ to the origin.

$$R(\theta) = ||\theta||_2^2 = \sum_{j=1}^{d} \theta_j^2$$

L2 regularized objective function:

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log P(y^{(i)} \,|\, x^{(i)}) - \alpha \sum_{j=1}^{d} \theta_j^2$$

# L1 / Lasso Regularization

- The sum of the (absolute value of the) weights
- Named after the L1 norm $\|\theta\|_1$ = sum of the absolute values of the weights = Manhattan distance

$$R(\theta) = \|\theta\|_1 = \sum_{j=1}^{d} |\theta_j|$$

L1 regularized objective function:

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log P(y^{(i)} | x^{(i)}) - \alpha \sum_{j=1}^{d} |\theta_j|$$

## Lecture Outline

- Recap: Supervised Classification / Logistic Regression
  - Data (Features)
  - Model (Logistic Regression)
  - Loss (Cross Entropy)
- Today: Logistic Regression Contd.
  - Optimization
  - Regularization
  - Multinomial Logistic Regression
- Word Embeddings (Teaser)

## Multinomial Logistic Regression

## Multinomial Logistic Regression

- Often we need more than 2 classes
  - Positive / negative / neutral sentiment of a document
  - Parts of speech of a word (noun, verb, adjective, adverb, preposition, etc.)
  - Actionable classes for emergency SMSs
- If >2 classes we use multinomial logistic regression
  - = Softmax regression
  - = Multinomial logit
  - = (defunct names : Maximum entropy modeling or MaxEnt)
- So "logistic regression" will just mean binary (2 output classes)

## Multinomial Logistic Regression

The probability of everything must still sum to 1

$$P(+|x) + P(-|x) + P(\sim|x) = 1$$

- Need a generalization of the sigmoid!
- Introducing the softmax function, which
  - Takes a vector $\mathbf{z} = [z_1, z_2, \ldots, z_K]$ of $K$ arbitrary values
    - Each $z_i$ corresponds to weighted sum of features for the $K$th class
  - Outputs a probability distribution
    - each value in the range $[0,1]$
    - all the values summing to $1$

Softmax

# The Softmax Function

Turns a vector $\mathbf{z} = [z_1, z_2, \ldots, z_K]$ of $K$ arbitrary values into probabilities

$$\textbf{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{K} \exp(z_j)} \qquad 1 \leq i \leq K$$

The denominator $\displaystyle\sum_{i=1}^{K} \exp(z_i)$ is used to normalize all the values into probabilities.

$$\textbf{softmax}(\mathbf{z}) = \left[ \frac{\exp(z_1)}{\sum_{i=1}^{K} \exp(z_i)}, \frac{\exp(z_2)}{\sum_{i=1}^{K} \exp(z_i)}, \ldots, \frac{\exp(z_K)}{\sum_{i=1}^{K} \exp(z_i)} \right]$$

## Softmax: Example

Turns a vector $\mathbf{z} = [z_1, z_2, \ldots, z_K]$ of $K$ arbitrary values into probabilities

$$\mathbf{z} = [0.6,\ 1.1,\ 1.5,\ 1.2,\ 3.2,\ 1.1]$$

$$\textbf{softmax}(\mathbf{z}) = \left[\frac{\exp(z_1)}{\sum_{i=1}^{K}\exp(z_i)}, \frac{\exp(z_2)}{\sum_{i=1}^{K}\exp(z_i)}, \ldots, \frac{\exp(z_K)}{\sum_{i=1}^{K}\exp(z_i)}\right]$$

$$\textbf{softmax}(\mathbf{z}) = [0.055,\ 0.090,\ 0.0067,\ 0.10,\ 0.74,\ 0.010]$$

## Binary versus Multinomial

**Binary Logistic Regression**

$$\begin{tikzcd} y = 1 \arrow[dr, "w_5 > 0"'] & & y = 0 \arrow[dl, "w_5 \leq 0"] \\ & w_5 & \end{tikzcd}$$

$$x_5 = \begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases} \qquad w_5 = 3.0$$

Why do we NOT need a different weight for each class in binary logistic regression?

**Multinomial Logistic Regression**

$$\begin{tikzcd} y = + \arrow[d] & y = \sim \arrow[d] & y = - \arrow[d] \\ w_{5,+} & w_{5,\sim} & w_{5,-} \end{tikzcd}$$

**Separate weights for each class**

| Feature | Definition | $w_{5,+}$ | $w_{5,-}$ | $w_{5,0}$ |
|---|---|---|---|---|
| $f_5(x)$ | $\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$ | 3.5 | 3.1 | −5.3 |

# Softmax in multinomial logistic regression

Parameters are now a matrix $\mathbf{W} \in \mathbb{R}^{d \times K}$ and $b \in \mathbb{R}^1$

$$P(y = c \mid \mathbf{x}; \theta) = \frac{\exp(\boxed{\mathbf{w}_c} \cdot \mathbf{x} + b)}{\sum_{j=1}^{K} \exp(\boxed{\mathbf{w}_j} \cdot \mathbf{x} + b)}$$

- Input is still the dot product between weight vector $\mathbf{w}_c$ and input vector $\mathbf{x}$, offset by $b$
- But separate weight vectors for each of the $K$ classes, each of dimension $d$

Multinomial LR Loss:

$$L_{CE} = -\log P(y = c \mid \mathbf{x}; \theta) = -(\mathbf{w}_c \cdot \mathbf{x} + b) + \boxed{\log \left[ \sum_{j=1}^{K} \exp(\mathbf{w}_j \cdot \mathbf{x} + b) \right]}$$

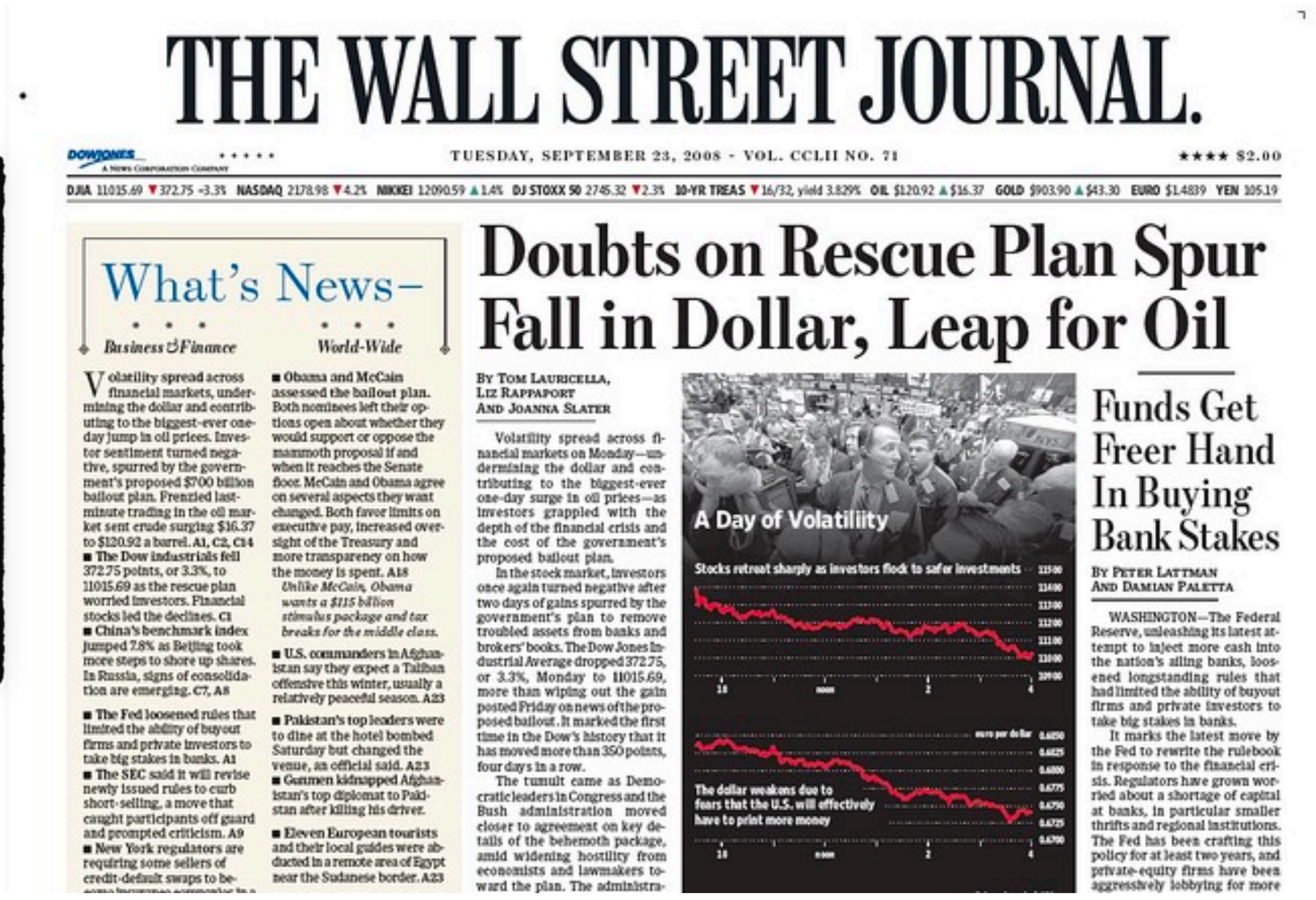
## Lecture Outline

- Recap: Supervised Classification / Logistic Regression
  - Data (Features)
  - Model (Logistic Regression)
  - Loss (Cross Entropy)
- Today: Logistic Regression Contd.
  - Optimization
  - Regularization
  - Multinomial Logistic Regression
- Word Embeddings (Teaser)

## Case Study: Word Neighborhood Predictor

- Yet do I fear thy nature. It is too full o'the milk of human kindness To catch the nearest way.
- If it were done when 'tis done, then 'twere well It were done quickly.
- Is this a dagger which I see before me, The handle toward my hand?
- My hands are of ;your color, but I shame To wear a heart so white.
- Knock, knock, knock! Who's there,

Where is it more likely to find the word "doth"?



THE WALL STREET JOURNAL.

DOWJONES A News Corporation Company ★★★★★ TUESDAY, SEPTEMBER 23, 2008 - VOL. CCLII NO. 71 ★★★★ $2.00

DJIA 11015.69 ▼372.75 -3.3% NASDAQ 2178.98 ▼4.2% NIKKEI 12090.59 ▲1.4% DJ STOXX 50 2745.32 ▼2.3% 10-YR TREAS ▼16/32, yield 3.829% OIL $120.92 ▲$16.37 GOLD $903.90 ▲$43.30 EURO $1.4839 YEN 105.19

What's News–

*Business & Finance*

Volatility spread across financial markets, undermining the dollar and contributing to the biggest-ever one-day jump in oil prices. Investor sentiment turned negative, spurred by the government's proposed $700 billion bailout plan. Frenzied last-minute trading in the oil market sent crude surging $16.37 to $120.92 a barrel. A1, C2, C14

■ The Dow industrials fell 372.75 points, or 3.3%, to 11015.69 as the rescue plan worried investors. Financial stocks led the declines. C1

■ China's benchmark index jumped 7.8% as Beijing took more steps to shore up shares. In Russia, signs of consolidation are emerging. C7, A8

■ The Fed loosened rules that limited the ability of buyout firms and private investors to take big stakes in banks. A1

■ The SEC said it will revise newly issued rules to curb short-selling, a move that caught participants off guard and prompted criticism. A9

■ New York regulators are requiring some sellers of credit-default swaps to be-

*World-Wide*

■ Obama and McCain assessed the bailout plan. Both nominees left their options open about whether they would support or oppose the mammoth proposal if and when it reaches the Senate floor. McCain and Obama agree on several aspects they want changed. Both favor limits on executive pay, increased oversight of the Treasury and more transparency on how the money is spent. A18

*Unlike McCain, Obama wants a $115 billion stimulus package and tax breaks for the middle class.*

■ U.S. commanders in Afghanistan say they expect a Taliban offensive this winter, usually a relatively peaceful season. A23

■ Pakistan's top leaders were to dine at the hotel bombed Saturday but changed the venue, an official said. A23

■ Gunmen kidnapped Afghanistan's top diplomat to Pakistan after killing his driver.

■ Eleven European tourists and their local guides were abducted in a remote area of Egypt near the Sudanese border. A23

Doubts on Rescue Plan Spur Fall in Dollar, Leap for Oil

BY TOM LAURICELLA, LIZ RAPPAPORT AND JOANNA SLATER

Volatility spread across financial markets on Monday—undermining the dollar and contributing to the biggest-ever one-day surge in oil prices—as investors grappled with the depth of the financial crisis and the cost of the government's proposed bailout plan.

In the stock market, investors once again turned negative after two days of gains spurred by the government's plan to remove troubled assets from banks and brokers' books. The Dow Jones Industrial Average dropped 372.75, or 3.3%, Monday to 11015.69, more than wiping out the gain posted Friday on news of the proposed bailout. It marked the first time in the Dow's history that it has moved more than 350 points, four days in a row.

The tumult came as Democratic leaders in Congress and the Bush administration moved closer to agreement on key details of the behemoth package, amid widening hostility from economists and lawmakers toward the plan. The administra-

A Day of Volatiliity
Stocks retreat sharply as investors flock to safer investments
11500
11400
11300
11200
11100
11000
10900
10
noon
2
4
euro per dollar
0.6850
0.6825
0.6800
0.6775
0.6750
0.6725
0.6700
The dollar weakens due to fears that the U.S. will effectively have to print more money
10
noon
2
4

Funds Get Freer Hand In Buying Bank Stakes

BY PETER LATTMAN AND DAMIAN PALETTA

WASHINGTON—The Federal Reserve, unleashing its latest attempt to inject more cash into the nation's ailing banks, loosened longstanding rules that had limited the ability of buyout firms and private investors to take big stakes in banks.

It marks the latest move by the Fed to rewrite the rulebook in response to the financial crisis. Regulators have grown worried about a shortage of capital at banks, in particular smaller thrifts and regional institutions. The Fed has been crafting this policy for at least two years, and private-equity firms have been aggressively lobbying for more

# Case Study: Word Neighborhood Predictor

What words are likely to co-occur with the word "garnish"?

**Ingredients**

12 Large Eggs, Hard-boiled And Peeled (See Note)

1/2 c. Mayonnaise

2 tbsp. Prepared Yellow Mustard

1/4 tsp. Kosher Salt

See Nutritional Information

**Directions**

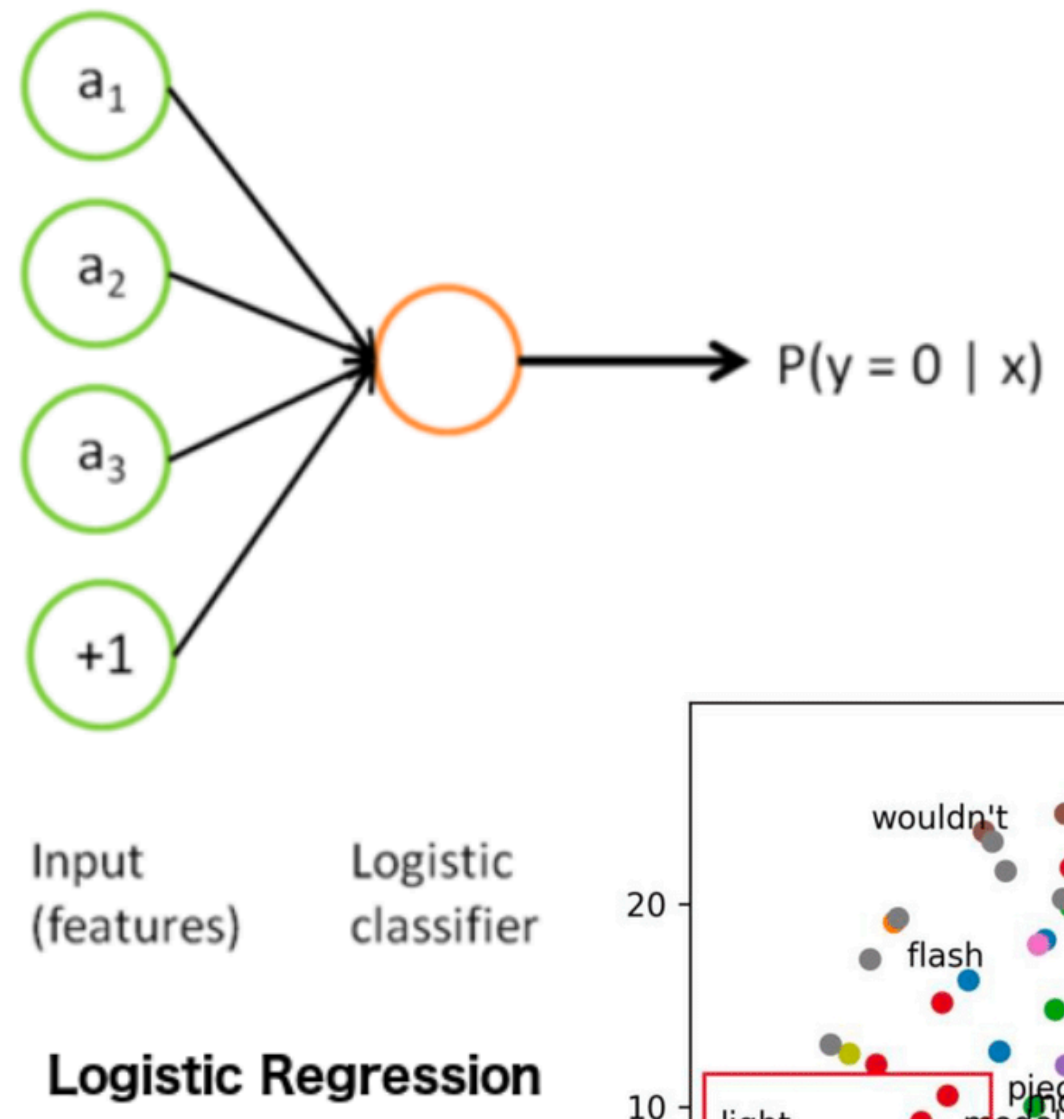
Slice eggs in half lengthwise. Pop out yolks and place them in a medium-sized mixing bowl. Mash yolks with a fork and then add mayonnaise, mustard, and salt. Mix until smooth. To achieve maximum smoothness, blitz the yolk mixture a few times with an immersion blender. If you are planning to use a pastry bag and piping tip to add the yolk mixture to the egg whites, it helps to get the yolk mixture nice and smooth.

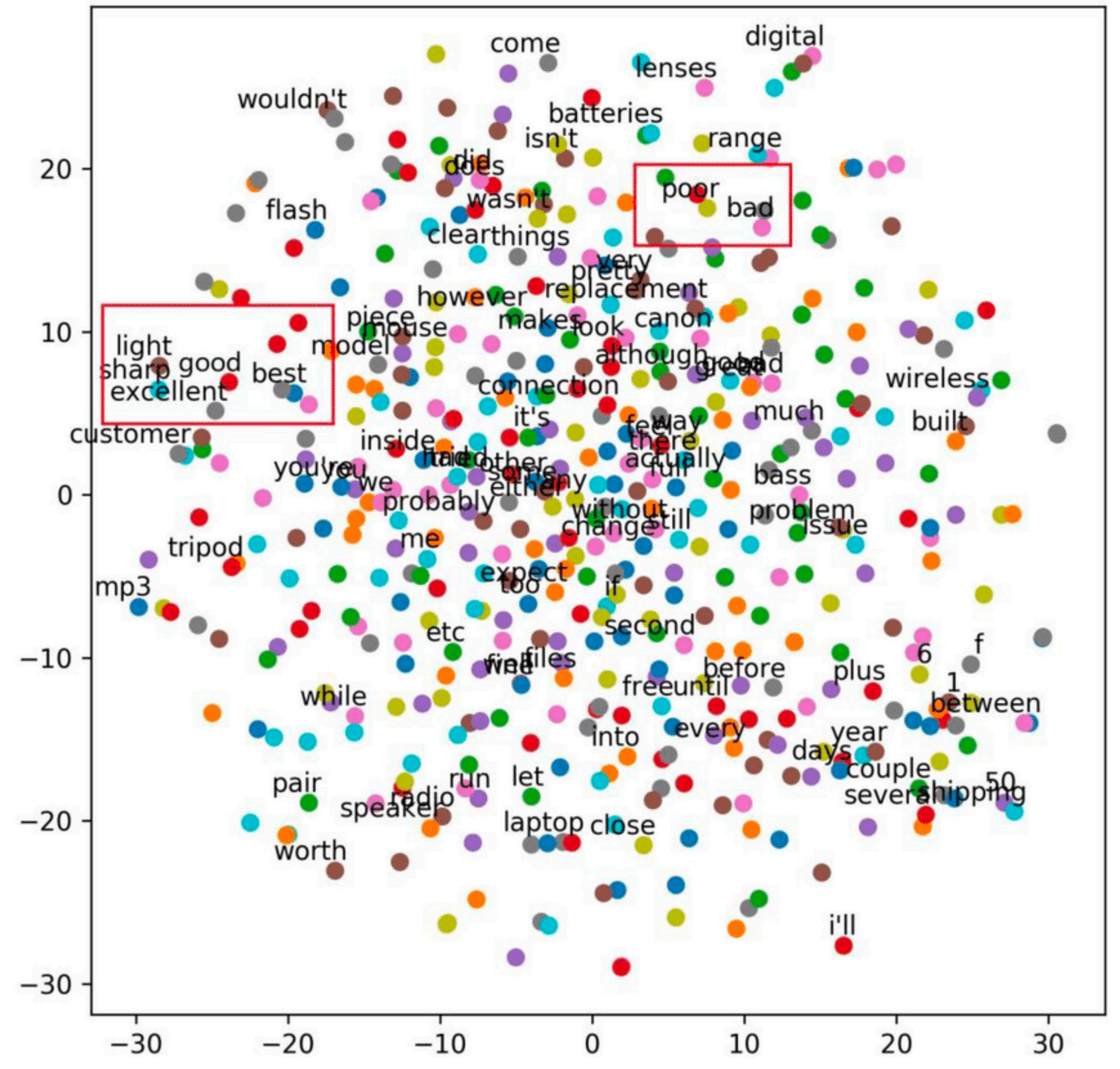
Divide the yolk mixture evenly between the egg white halves. Garnish as desired. Serve immediately or refrigerate until ready to serve.

Next Class: This is the text classification task we will use logistic regression for to learn word embeddings!

# Concluding Thoughts

- Logistic Regression
  - Case Study: Sentiment Classification
  - A very simple neural network
- Next Class: Let's learn some automatic features
  - Word Embeddings


$a_1$
$a_2$
$a_3$
+1
P(y = 0 | x)
Input (features)
Logistic classifier
Logistic Regression


come
digital
lenses
wouldn't
batteries
isn't
range
did
does
wasn't
poor
bad
flash
clear
things
very
pretty
however
replacement
piece
house
makes
look
canon
light
sharp
good
best
excellent
model
although
connection
wireless
built
customer
it's
much
inside
there
actually
you
we
probably
bass
problem
issue
without
me
change
still
tripod
expect
too
mp3
etc
second
files
before
plus
while
free
until
every
year
days
couple
into
between
pair
run
let
radio
speaker
laptop
close
worth
i'll
shipping
several