CS21si: AI for Social Good

Lecture 1: Motivations and Basic Models

Instructors

Shubhang Desai

Karan Singhal

Swetha Revanur

Course Sponsor

Chris Piech

Course Staff

Rachel Gardner

Nidhi Manoj

Email us at

cs21si-staff@lists.stanford.edu!

Why AI for Social Good?

Why machine learning & deep learning?

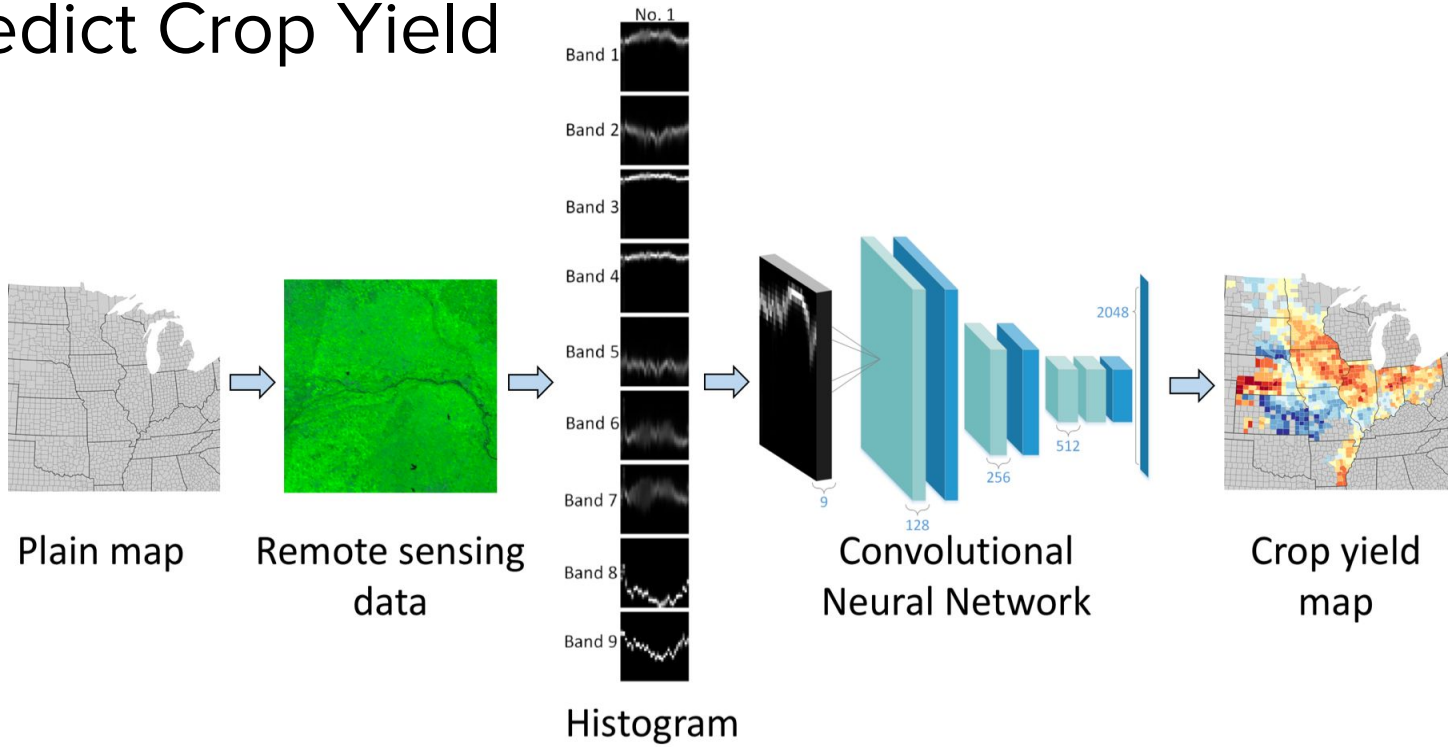
- Neural networks can simply model complex data
- More computational power
- More data
- Better models and training techniques

Predicting Poverty from Satellite Images

Neal Jean, Marshall Burke, Michael Xie, W. Matthew Davis, David B. Lobell, Stefano Ermon

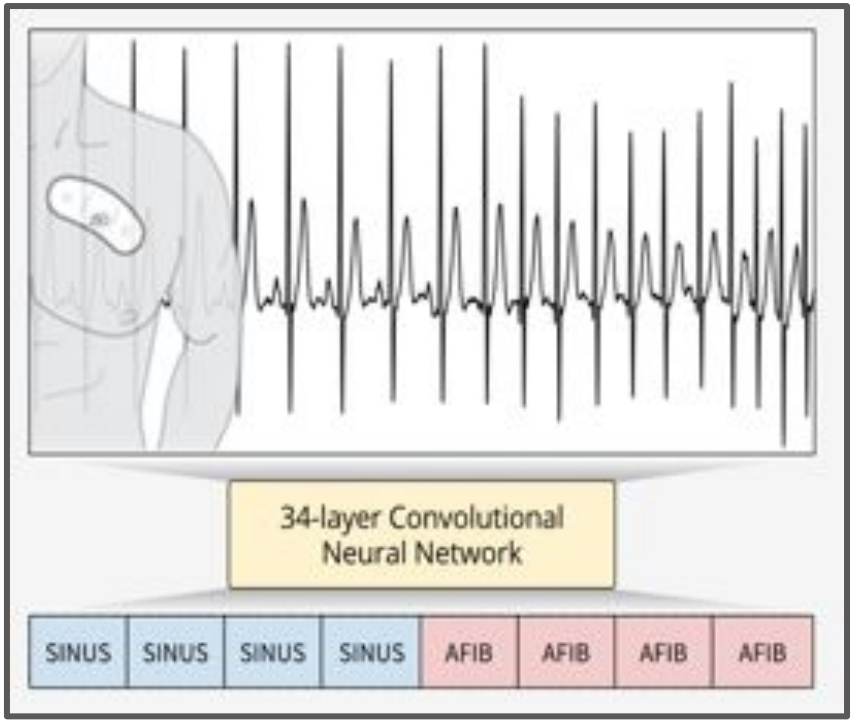
https://www.youtube.com/watch?v=DafZSeIGLNE

Combining Remote Sensing Data and Machine Learning to Predict Crop Yield

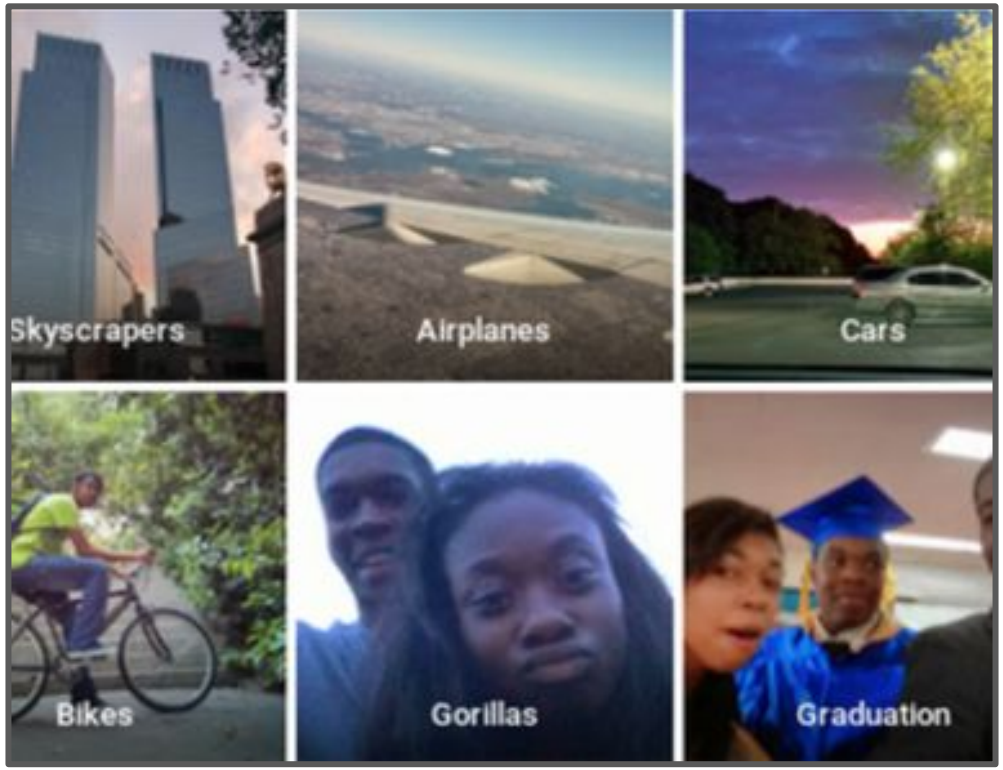


Jiaxuan You, Xiaocheng Li, Melvin Low, David B. Lobell, Stefano Ermon

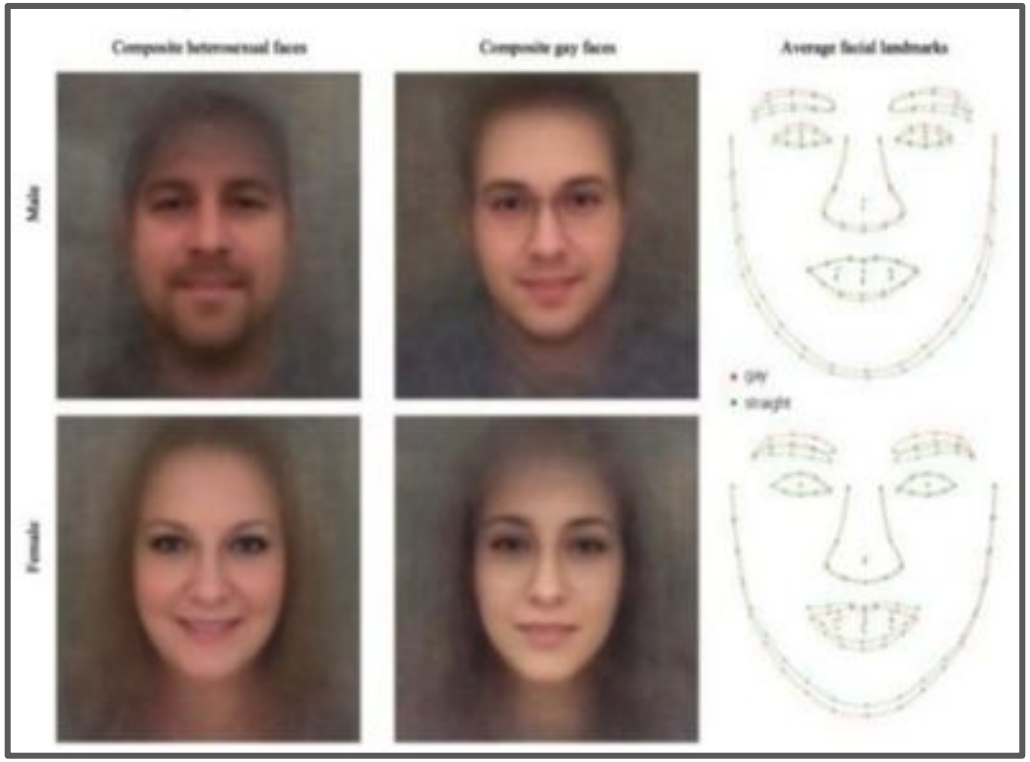
Using CNNs to Detect Arrhythmias



Labeling Pictures



Predicting Sexuality from Images



Problems with AI

- Bias in dataset--not enough diversity
- Engineers push cutting-edge, but not socially relevant bottom-line
- Automation could further widen wealth disparity

Class Goals

How to do AI

- Learn the techniques and some theory behind ML/DL
- Lectures and in-class exercises

How to do good with AI

- See and implement examples of AI being applied for good
- Even-week homework assignments

How to not do bad with AI

- Implement examples of AI systems with negative implications
- Odd-week homework assignments

Logistics

- Odd weeks are lectures, even weeks are speakers
  - Lectures will have in-class exercises!

- You are required to be at 9 out of 10 class sessions

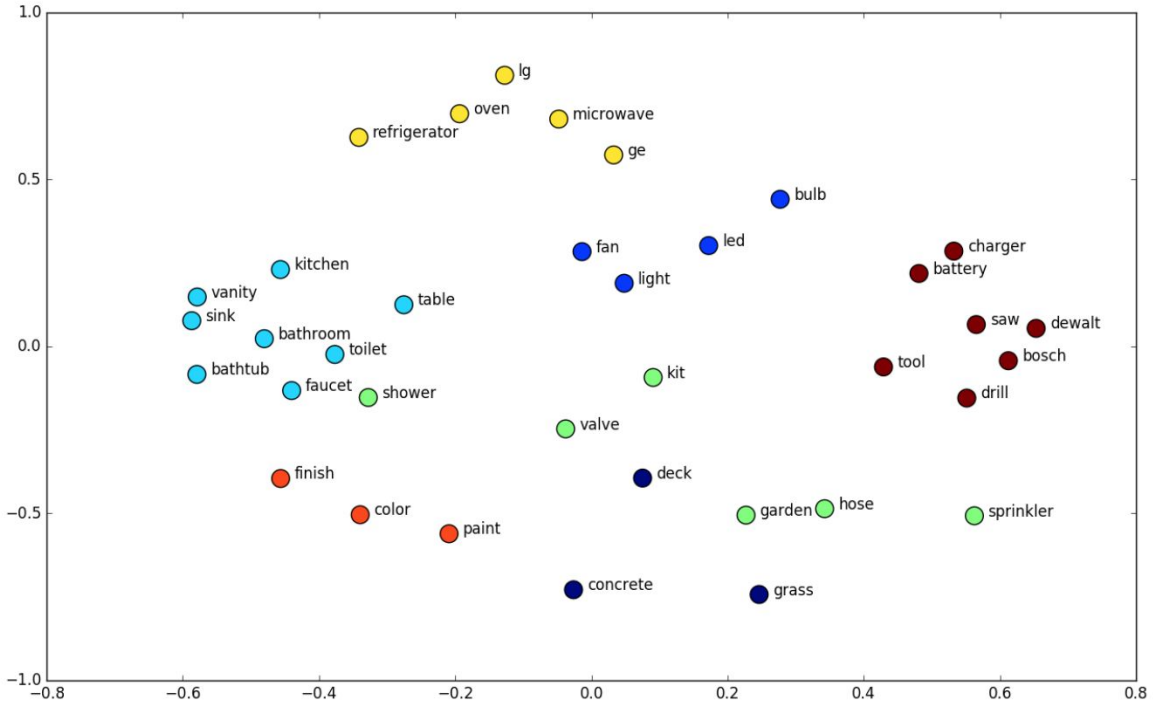
- Weekly homework assignments in the form of iPython notebooks
  - Turn in homework as a PDF of iPython notebook, emailed to CS 21si staff list
  - **Due 4:30 PM on Wednesday (before class)**

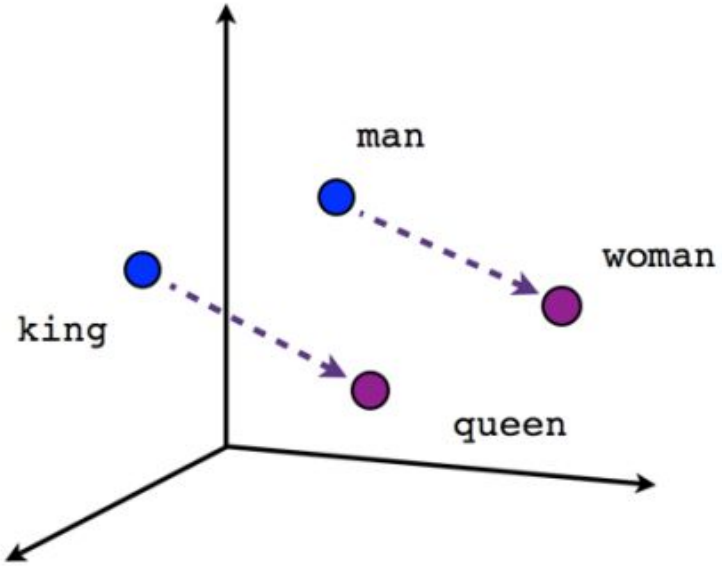
- Class enrollment codes will be handed out around Week 3

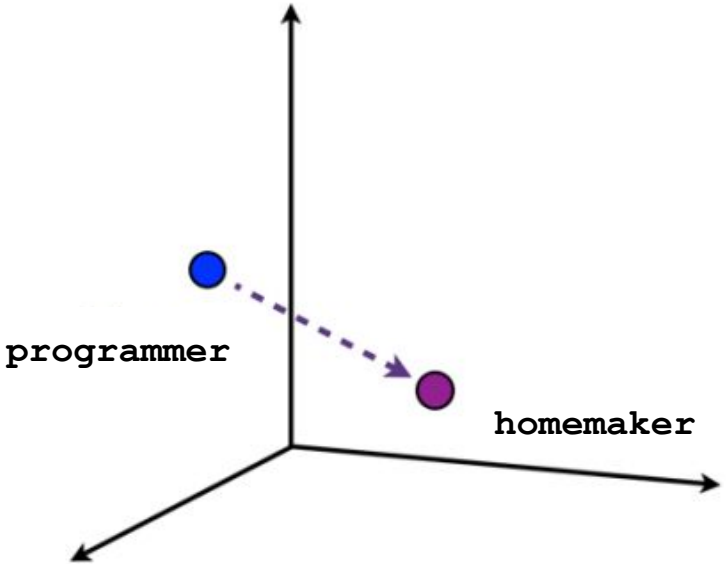
More Class Logistics

- 2 units, C/NC, meet once a week in this room (attendance required!)
- Class website: cs21si.stanford.edu
- Class GitHub: github.com/karan1149/cs21si
- Contact us: cs21si-staff@lists.stanford.edu

Word Vectors




man
woman
king
queen


programmer
homemaker

Why do we care about word vectors?


Google


NETFLIX


Spotify®

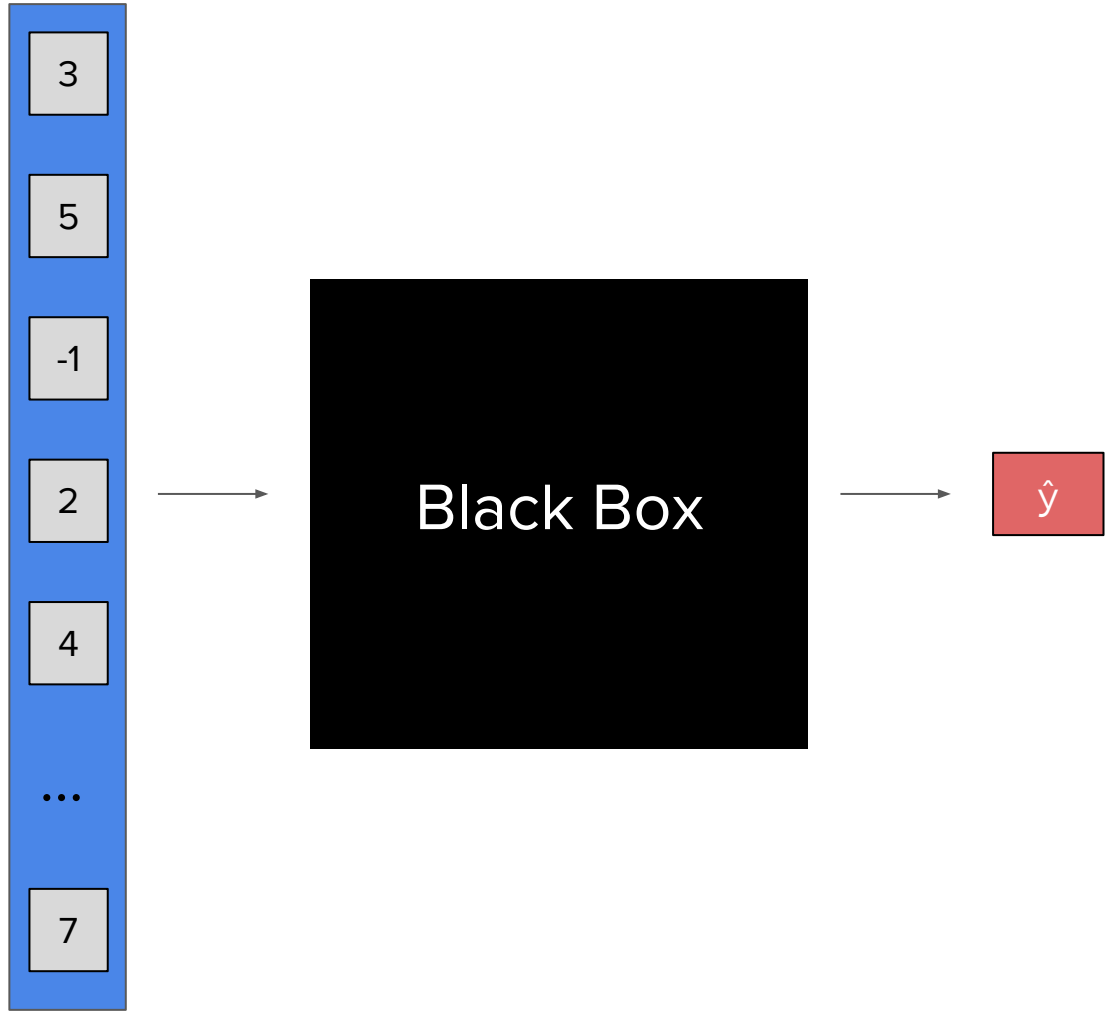

Google
Translate


YouTube

How bad is the problem?

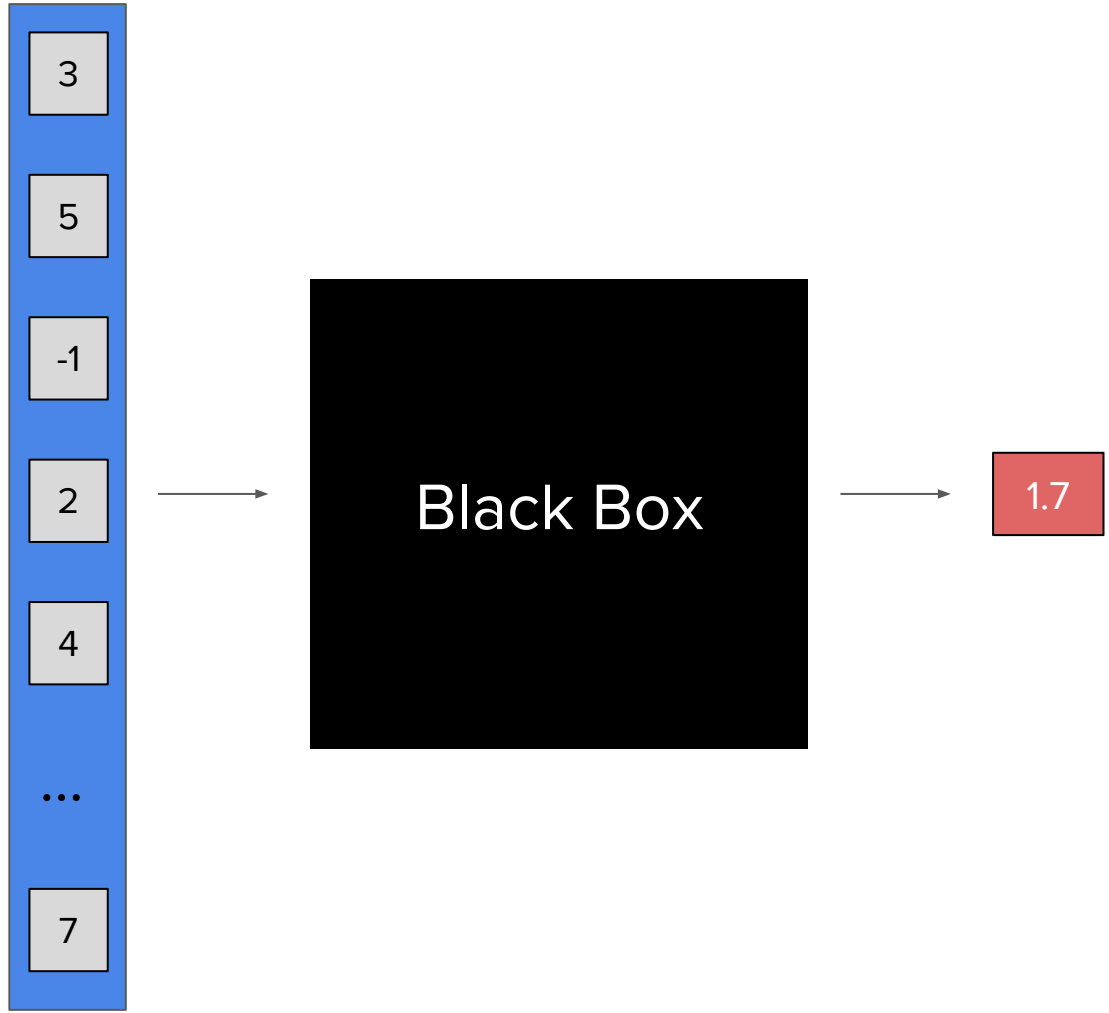
300-dimensional word vector

e.g. $v_{homemaker}$



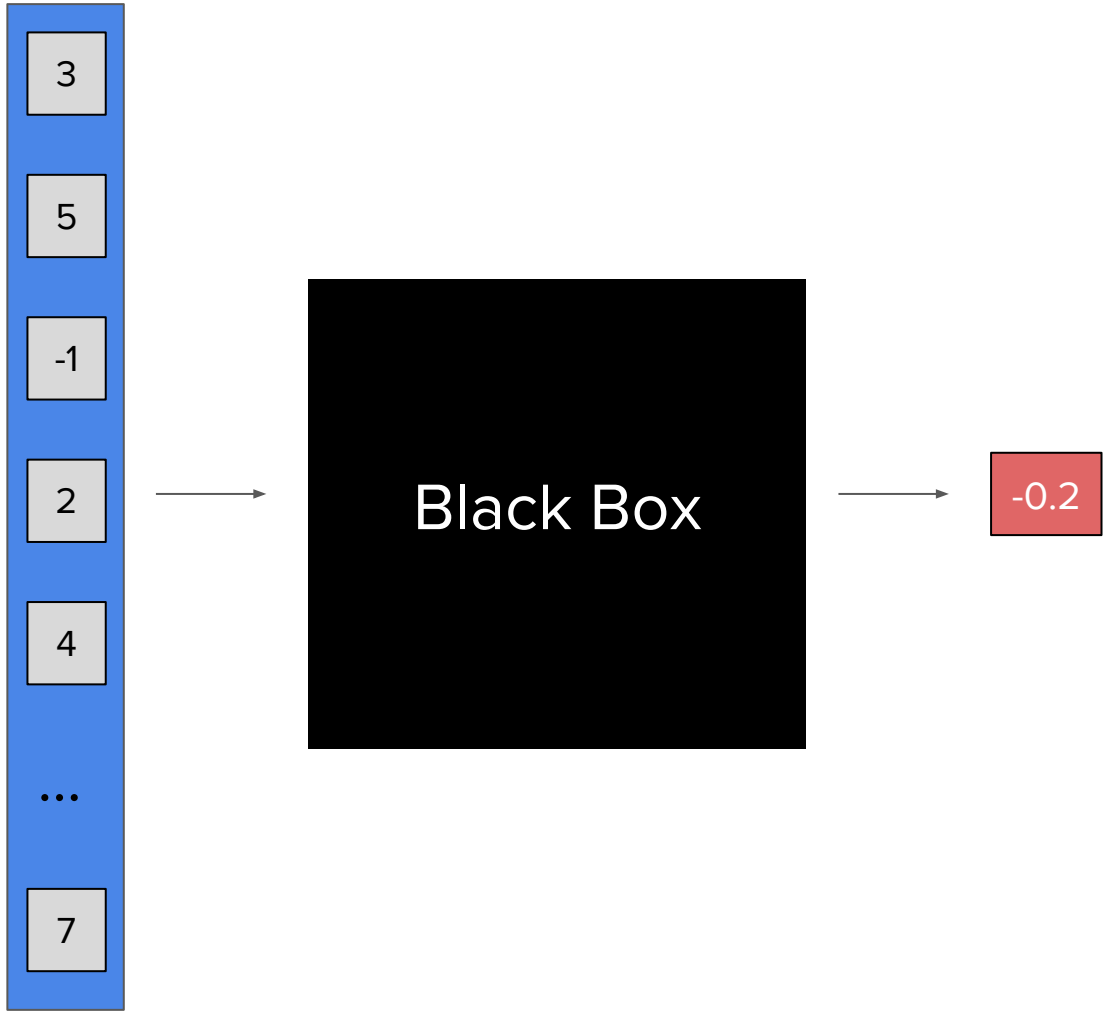
300-dimensional word vector

e.g. $v_{homemaker}$



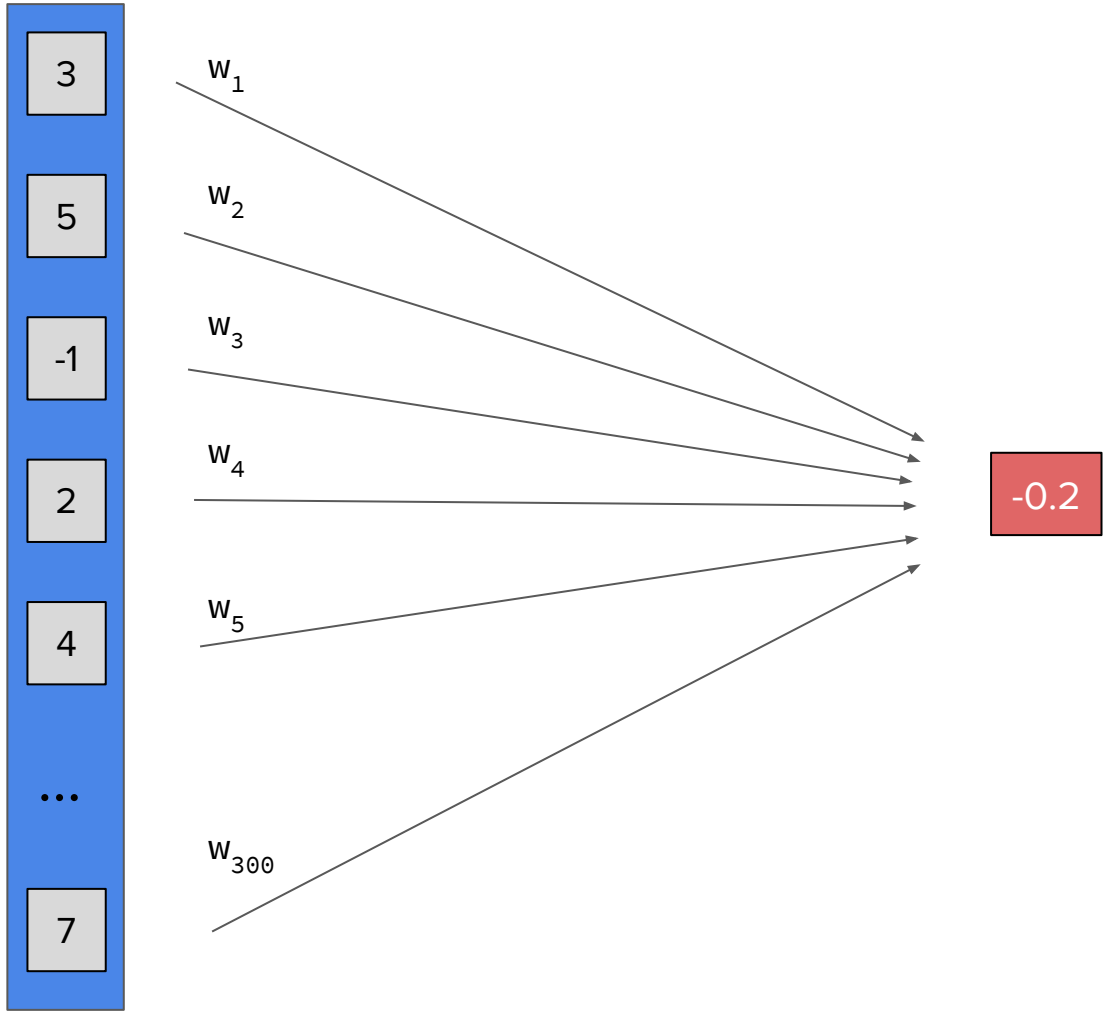
300-dimensional word vector

e.g. $v_{programmer}$



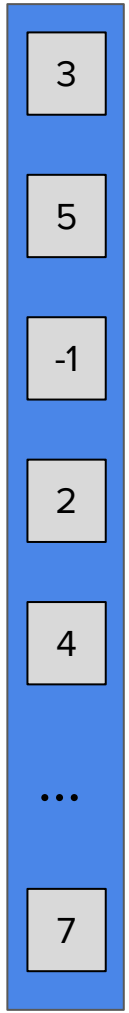
300-dimensional word vector

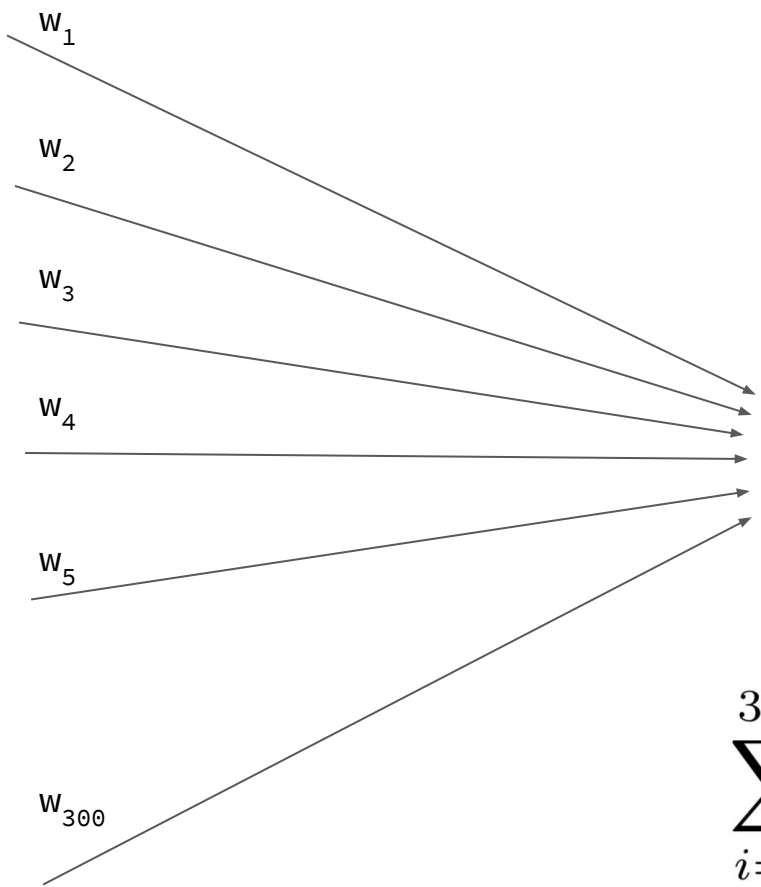
e.g. $v_{programmer}$



300-dimensional word vector

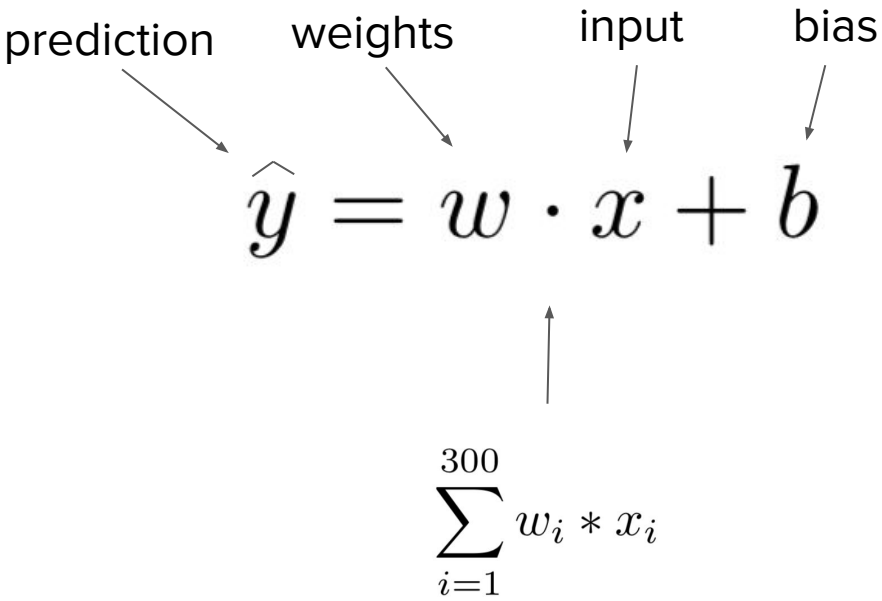
e.g. $v_{programmer}$





$$\sum_{i=1}^{300} w_i * x_i$$

Linear Model



How do we choose *w* and *b*?

$$\hat{y} = w \cdot x$$

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

$$\hat{y} = w \cdot x$$

This is a measure of the similarity between vectors *w* and *x*.

$$\hat{y} = w \cdot x$$

Choose $w = v_{woman} - v_{man}$!

Jupyter Notebook Exercises: Part 1

You'll need:

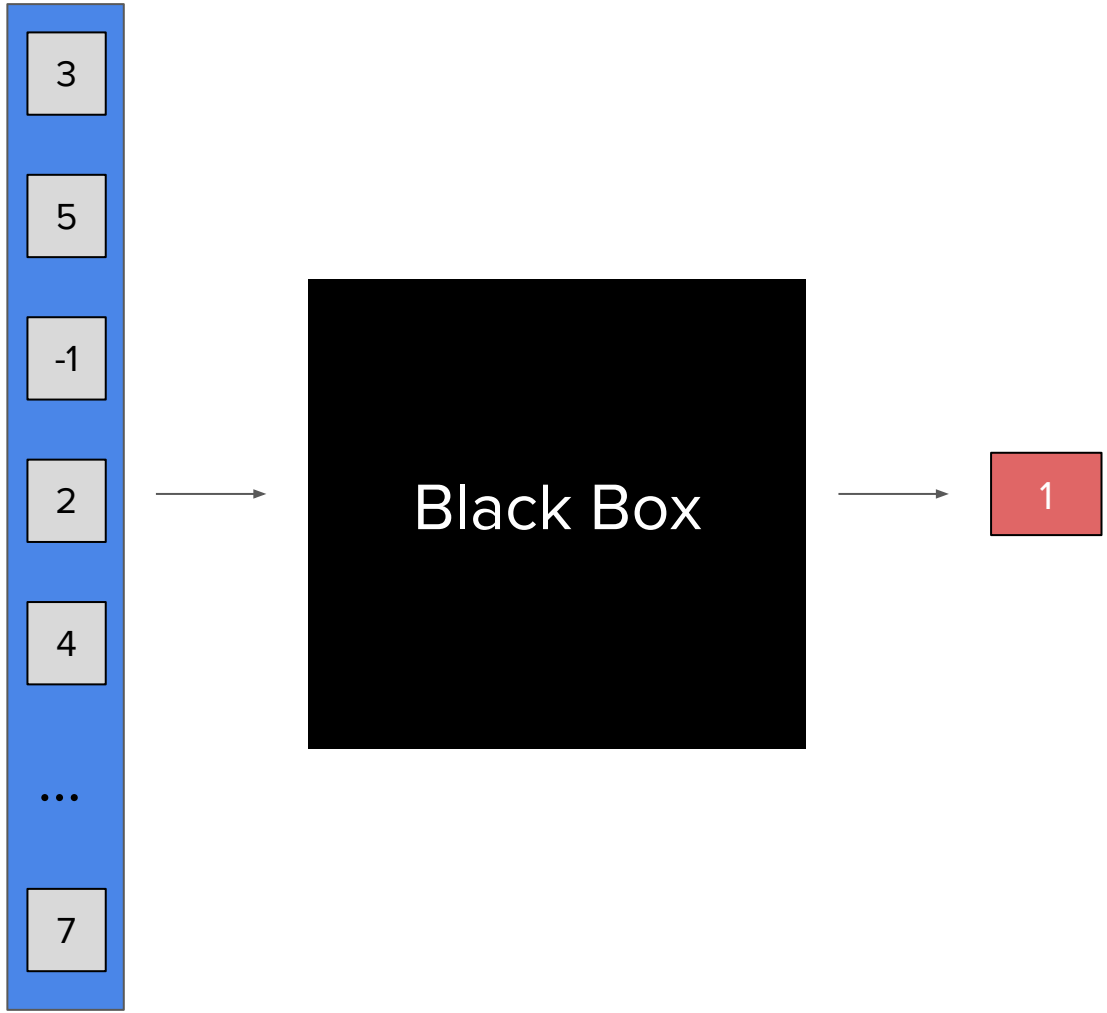
```
np.dot(a, b)

np.linalg.norm(a)
```

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Improving upon our model

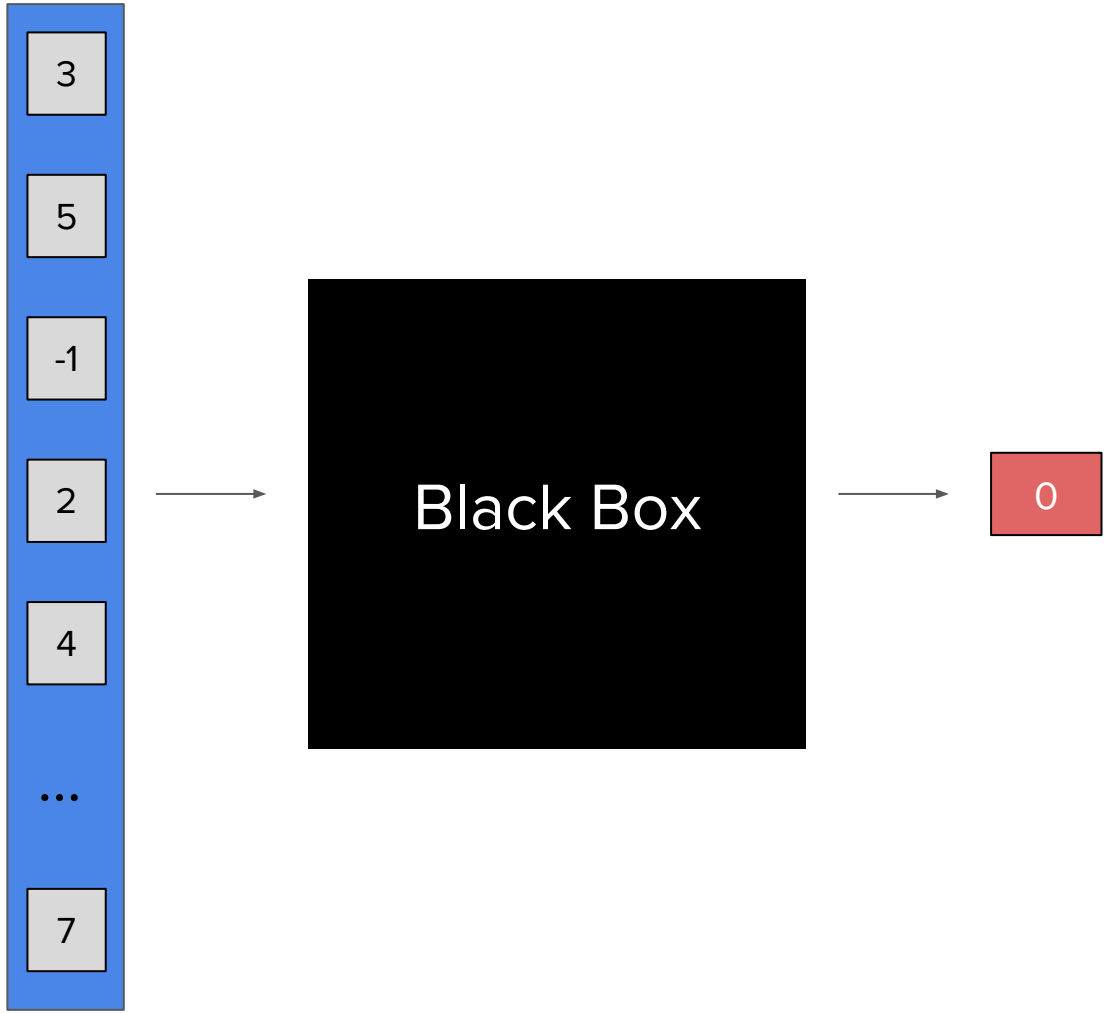
300-dimensional word vector

e.g. $v_{homemaker}$



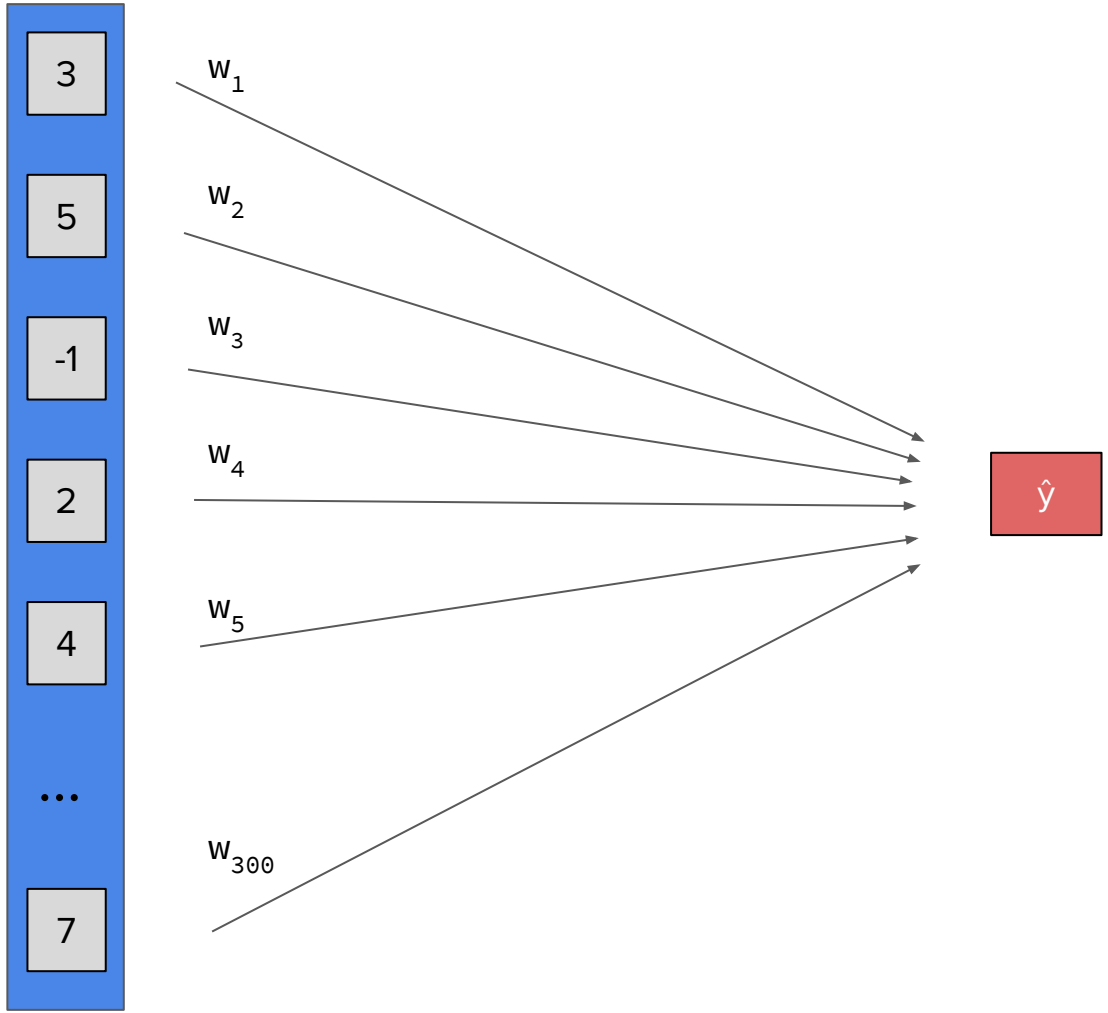
300-dimensional word vector

e.g. $v_{programmer}$



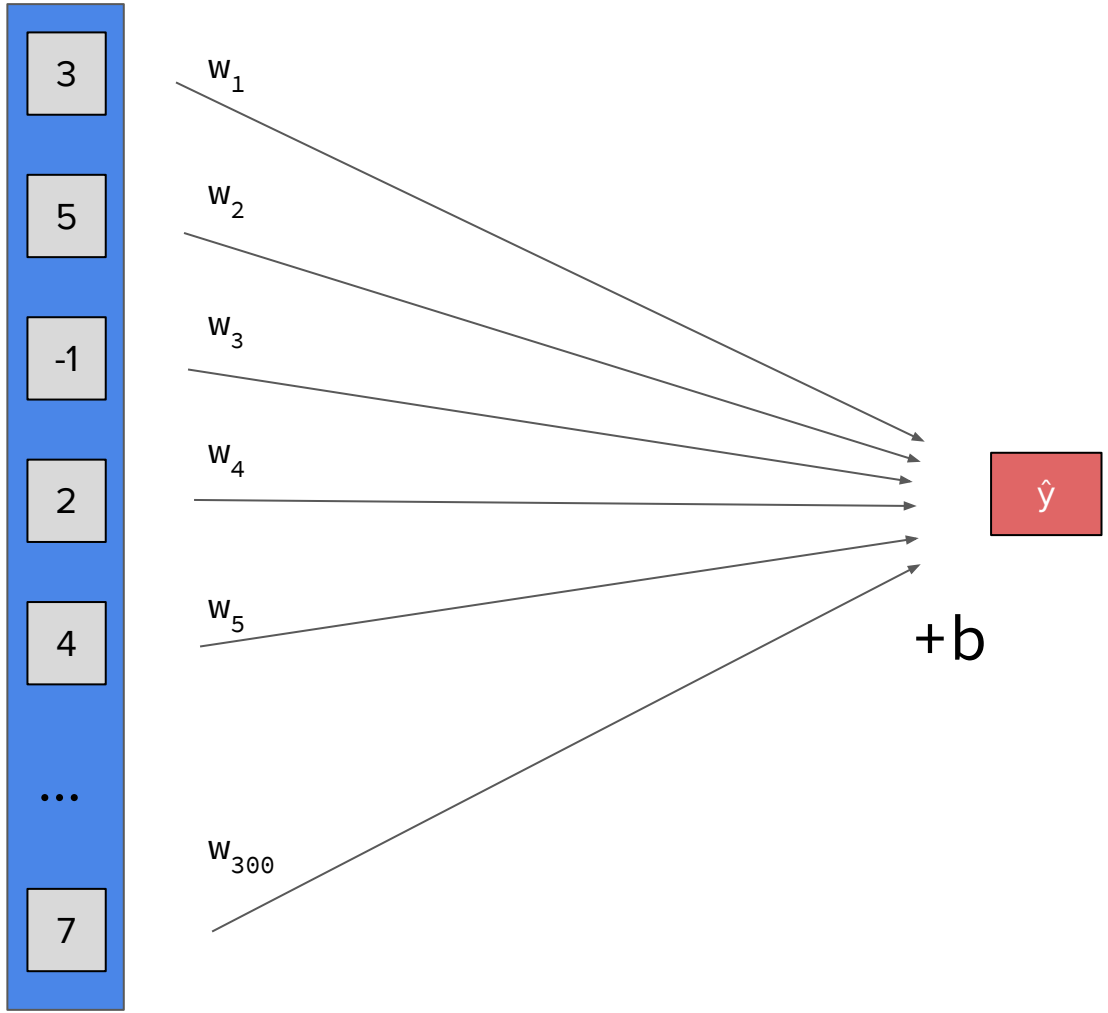
300-dimensional word vector

e.g. $v_{programmer}$

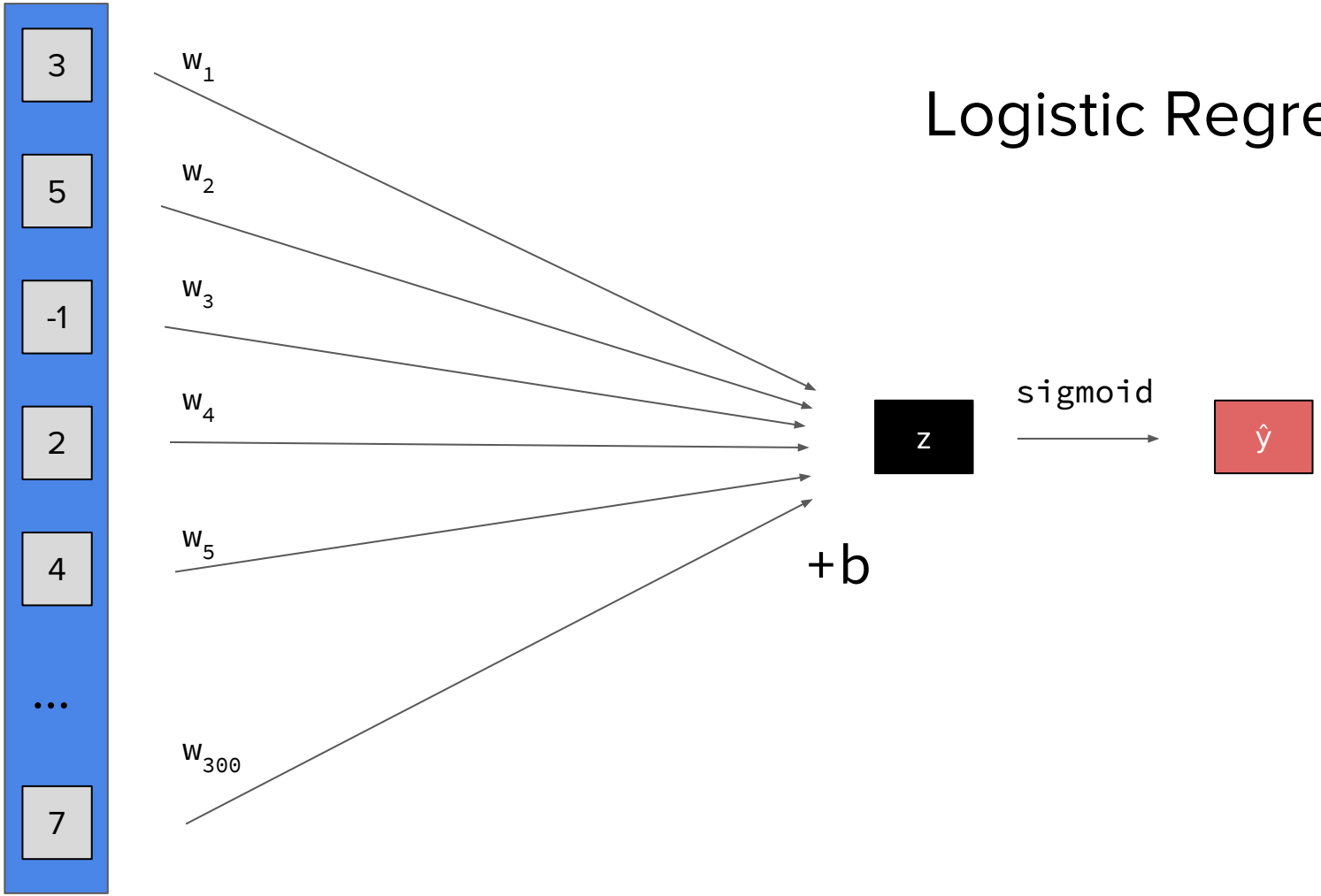


300-dimensional word vector

e.g. $v_{programmer}$

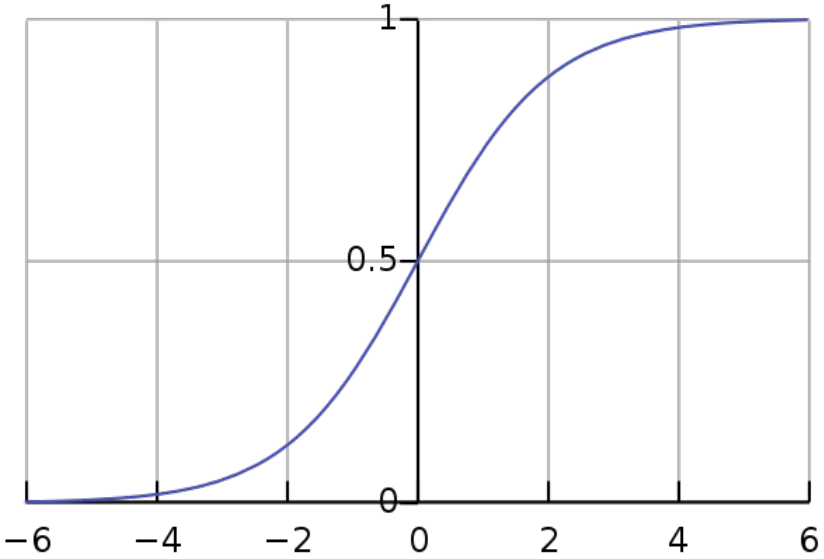


Logistic Regression

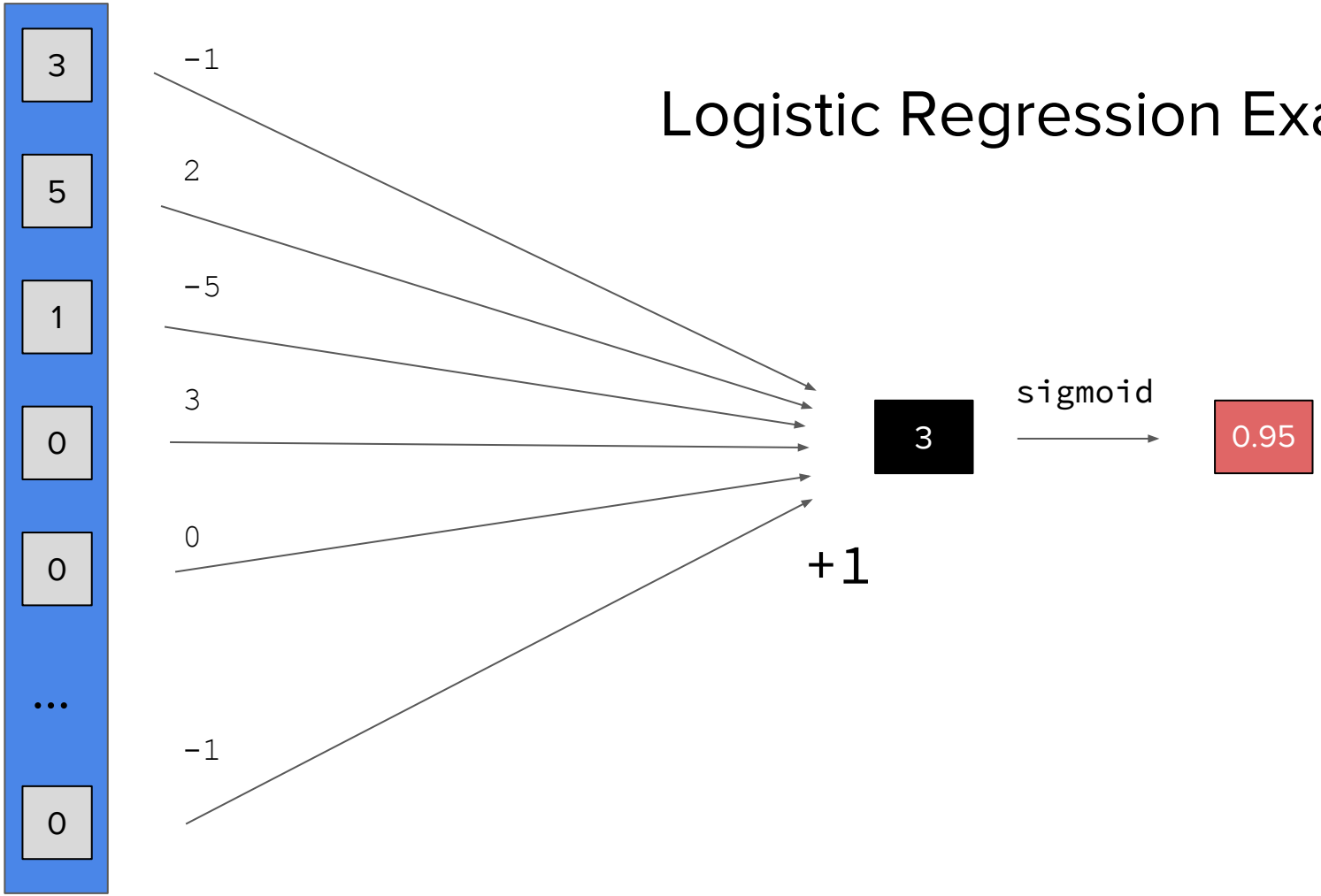


Sigmoid Function

$$\hat{y} = \frac{1}{1 + e^{-z}}$$



Logistic Regression Example



Questions?

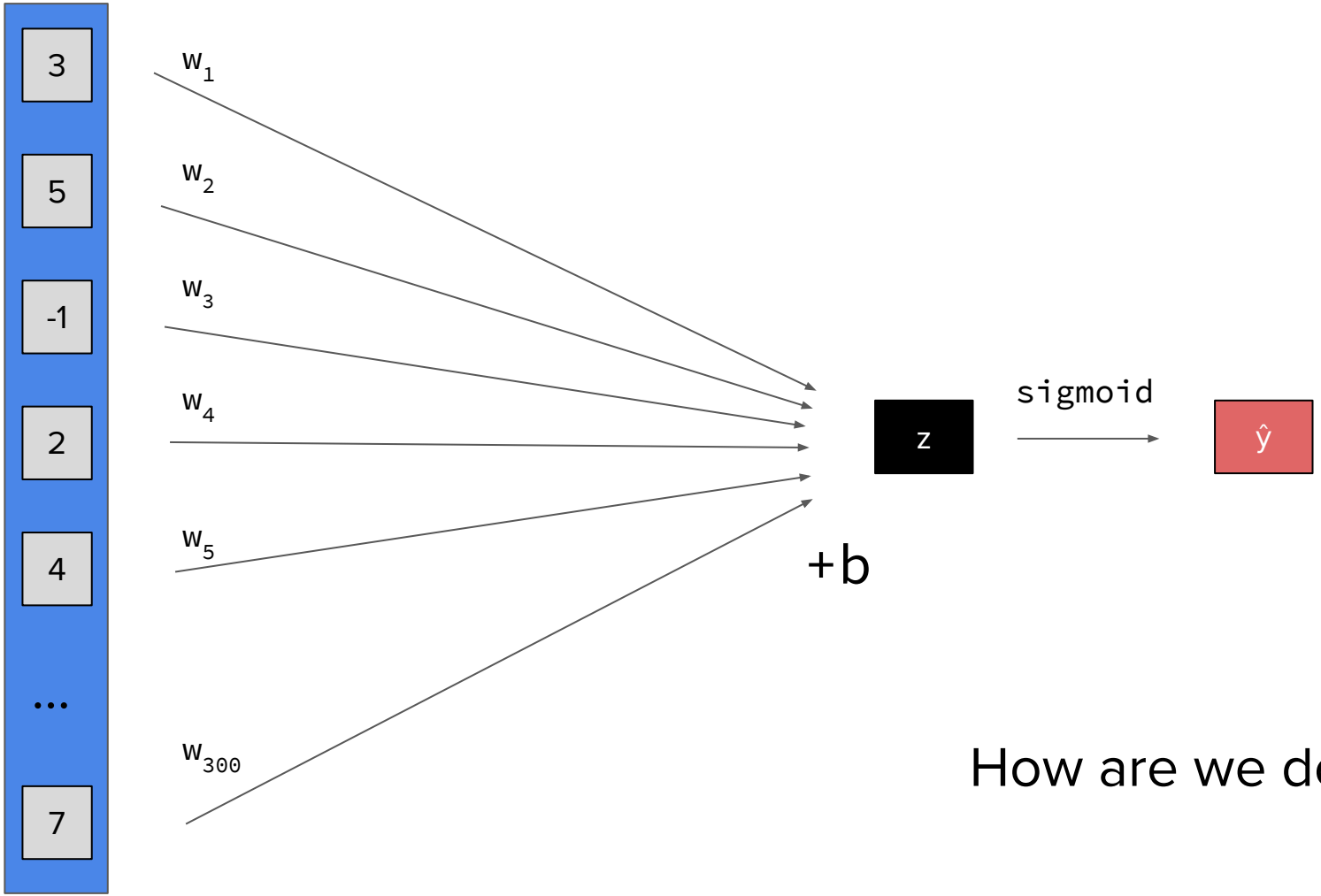
How do we choose *w* and *b*?

Why Train our Model?

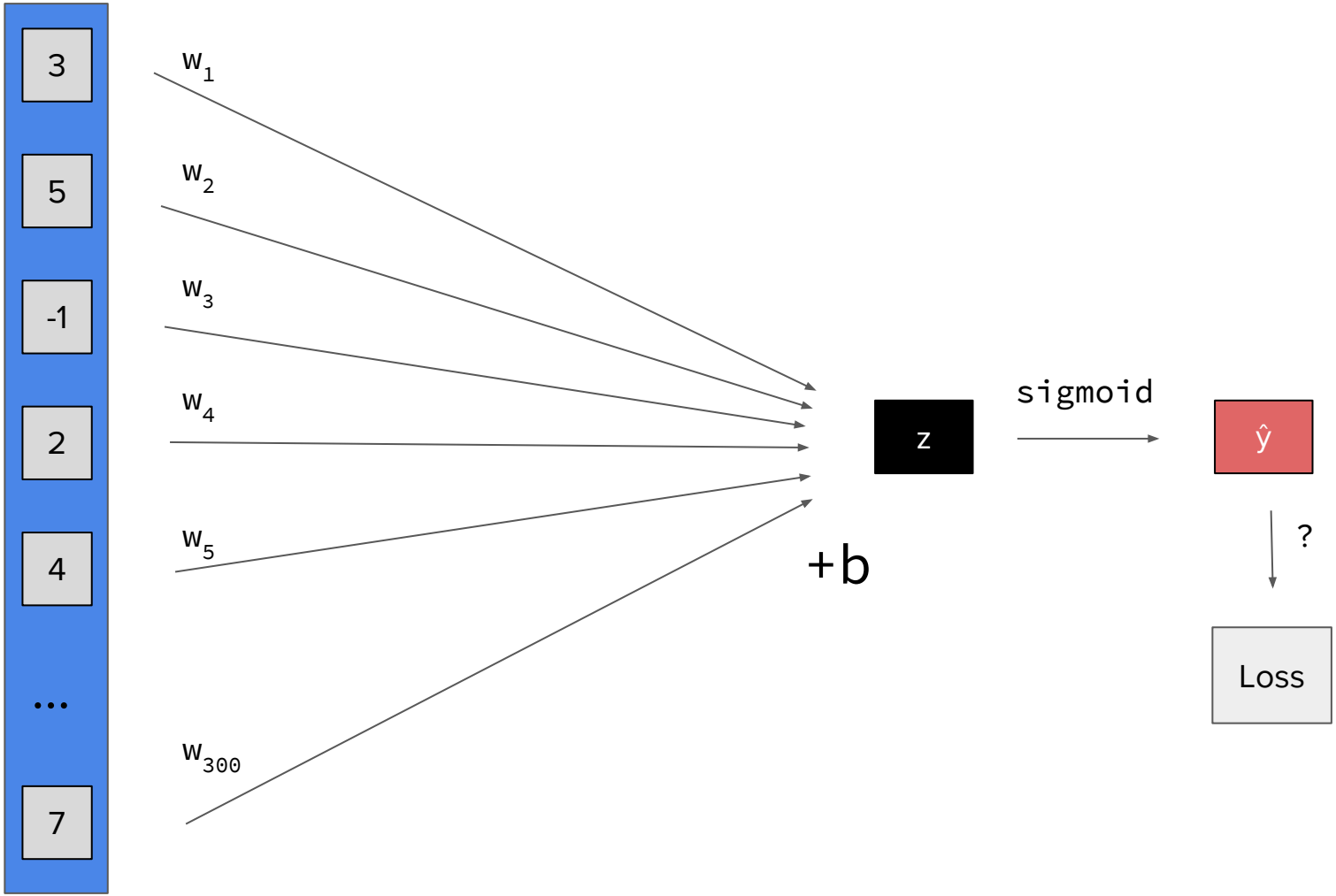
- For more complex models, you won't be able to guess the weights
- This is a more convincing demonstration of gender bias in word vectors

Training Data

Word (x)	**True Label (y)**
female	1
male	0
woman	1
man	0
...	...



How are we doing?

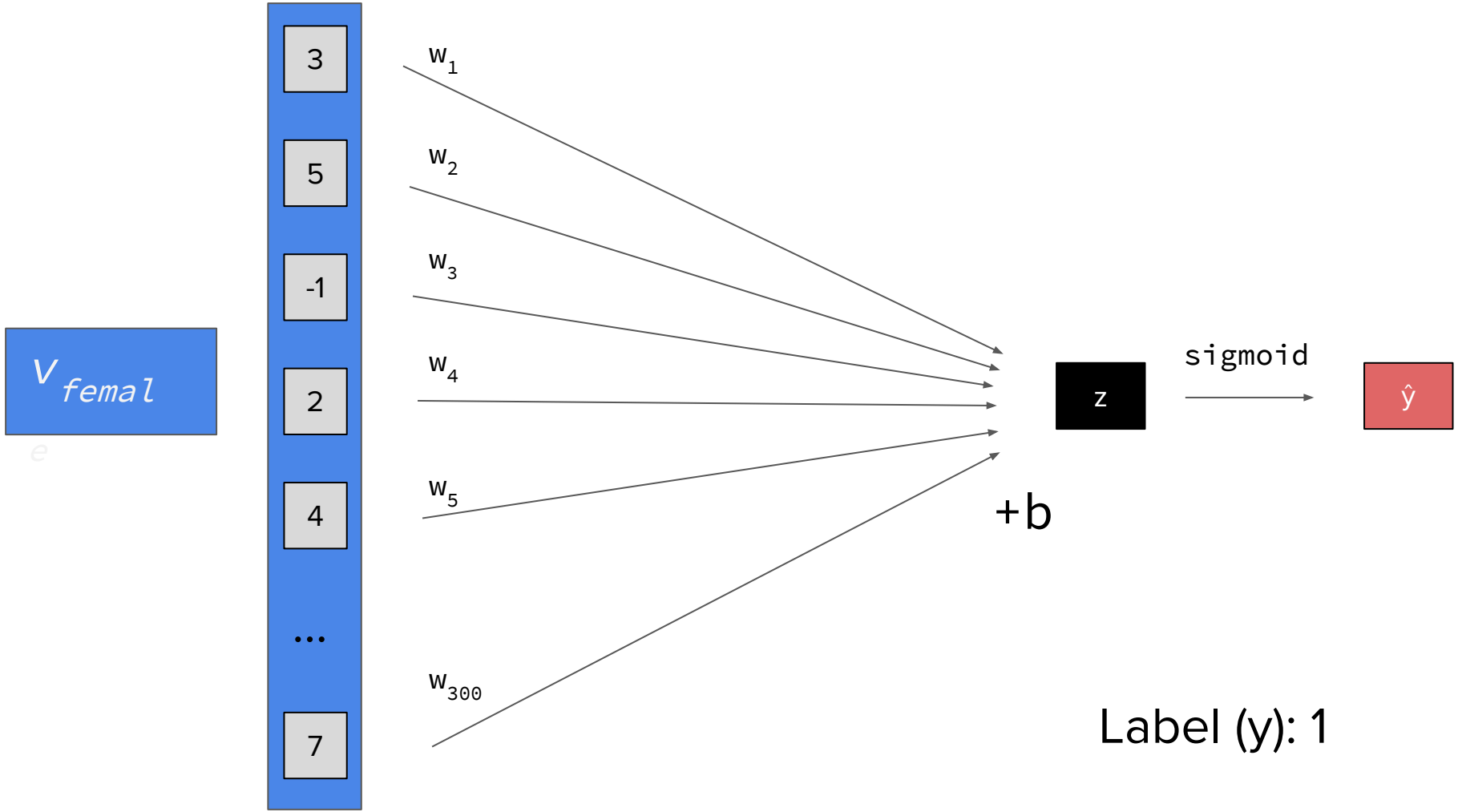

3
5
-1
2
4
...
7
w1
w2
w3
w4
w5
w300
+b
z
sigmoid
ŷ
?
Loss

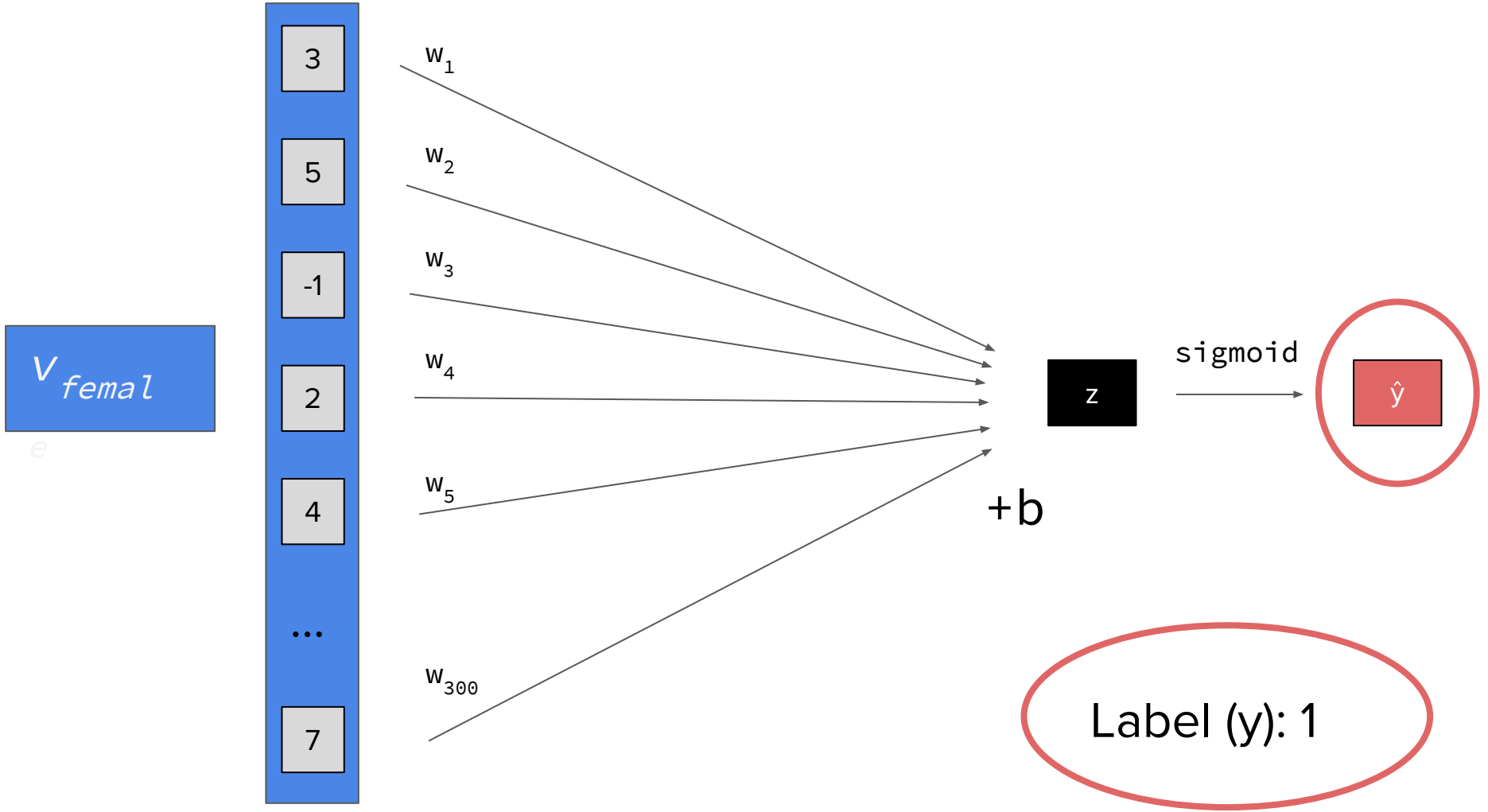
1. How do we calculate loss?

2. How do we use loss to get the right weights, bias?

1. How do we calculate loss?

2. How do we use loss to get the right weights, bias?


v_{femal}
3
5
-1
2
4
...
7
w_1
w_2
w_3
w_4
w_5
w_{300}
+b
z
sigmoid
ŷ
Label (y): 1


v_{femal}
3
5
-1
2
4
...
7
w_1
w_2
w_3
w_4
w_5
w_{300}
+b
z
sigmoid
ŷ
Label (y): 1

$$Loss(y, \hat{y})$$

Idea: maximize this!

$$\hat{y}^{y} * (1 - \hat{y})^{1-y}$$

Questions?

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

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

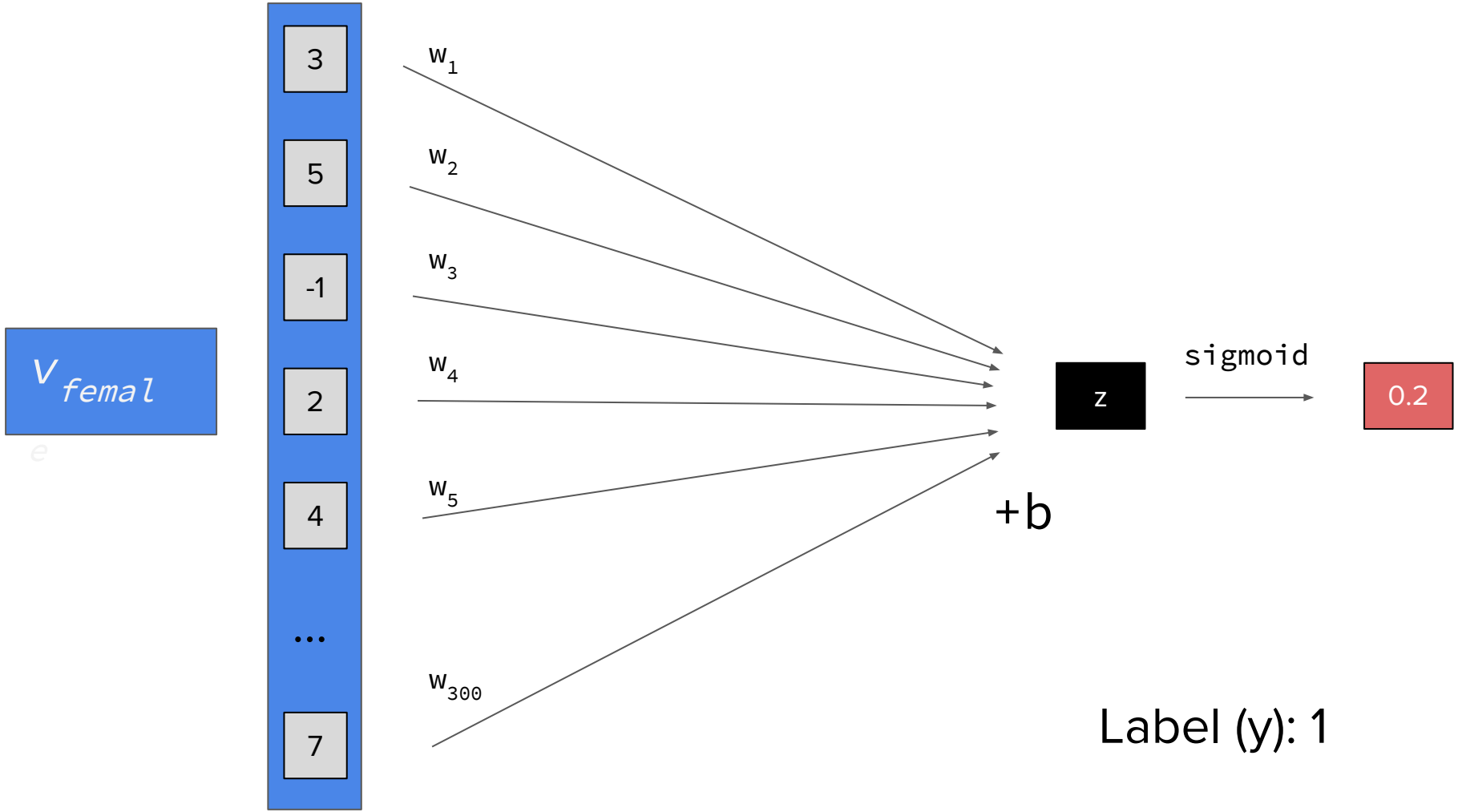
$$Cost = \sum_{i=1}^{m} Loss(y_i, \hat{y}_i)$$

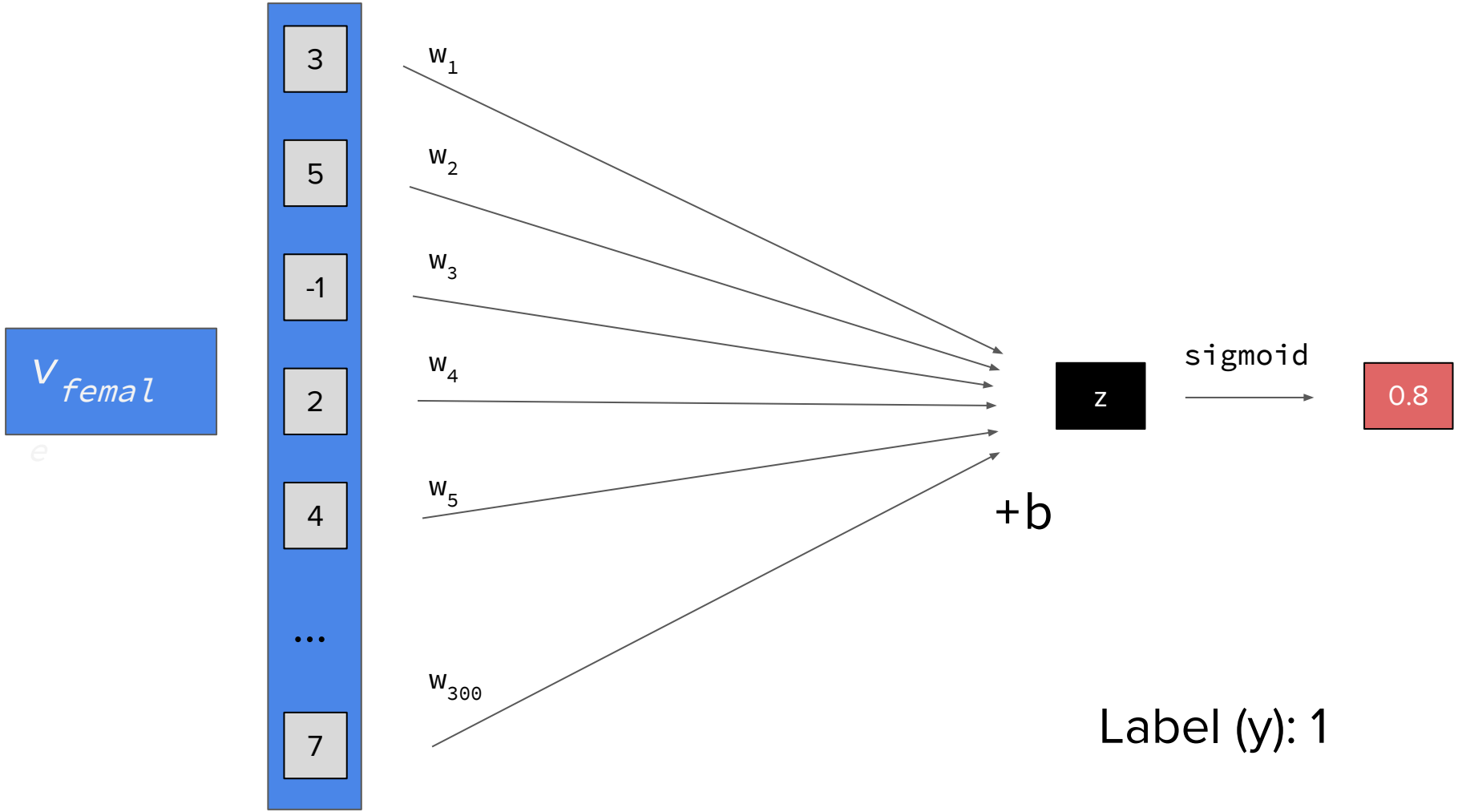
Questions?

1. How do we calculate loss?

2. How do we use loss to get the right weights, bias?

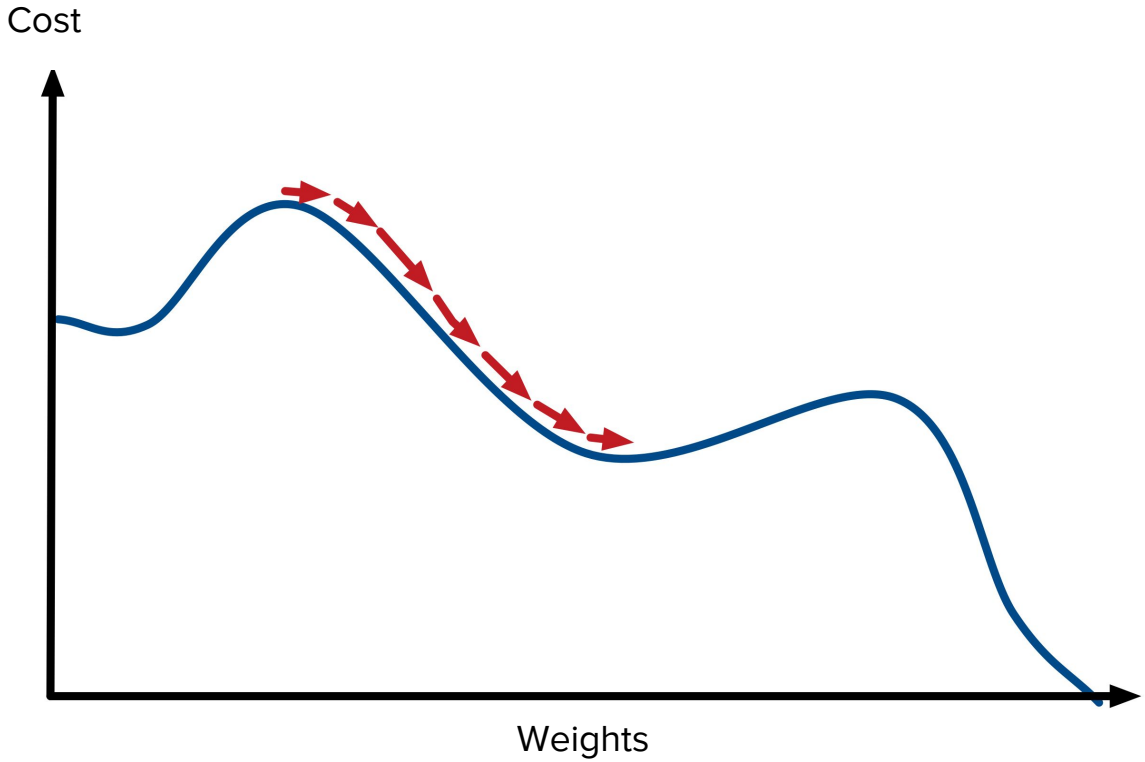
Can we randomly guess?


v_{femal}
3
5
-1
2
4
...
7
w_1
w_2
w_3
w_4
w_5
w_{300}
+b
z
sigmoid
0.2
Label (y): 1


v_{femal}
3
5
-1
2
4
...
7
w_1
w_2
w_3
w_4
w_5
w_{300}
+b
z
sigmoid
0.8
Label (y): 1

Can we do better?

Gradient Descent



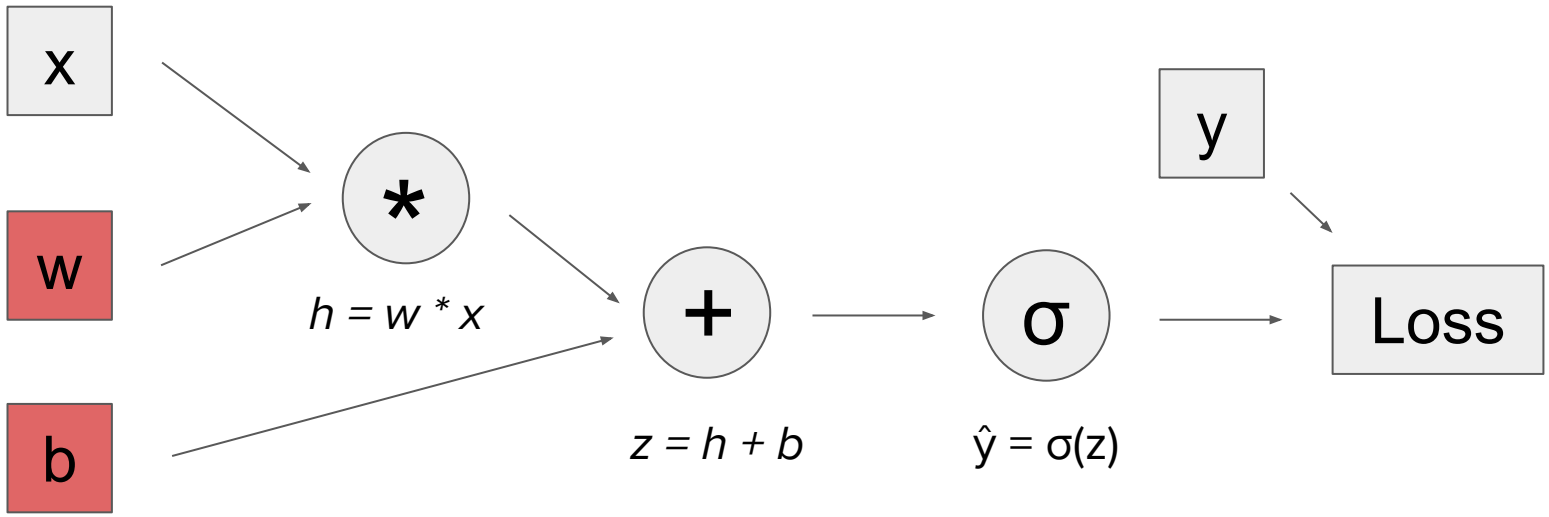
$$\frac{\partial Loss}{\partial w}$$

$$\frac{\partial Loss}{\partial b}$$

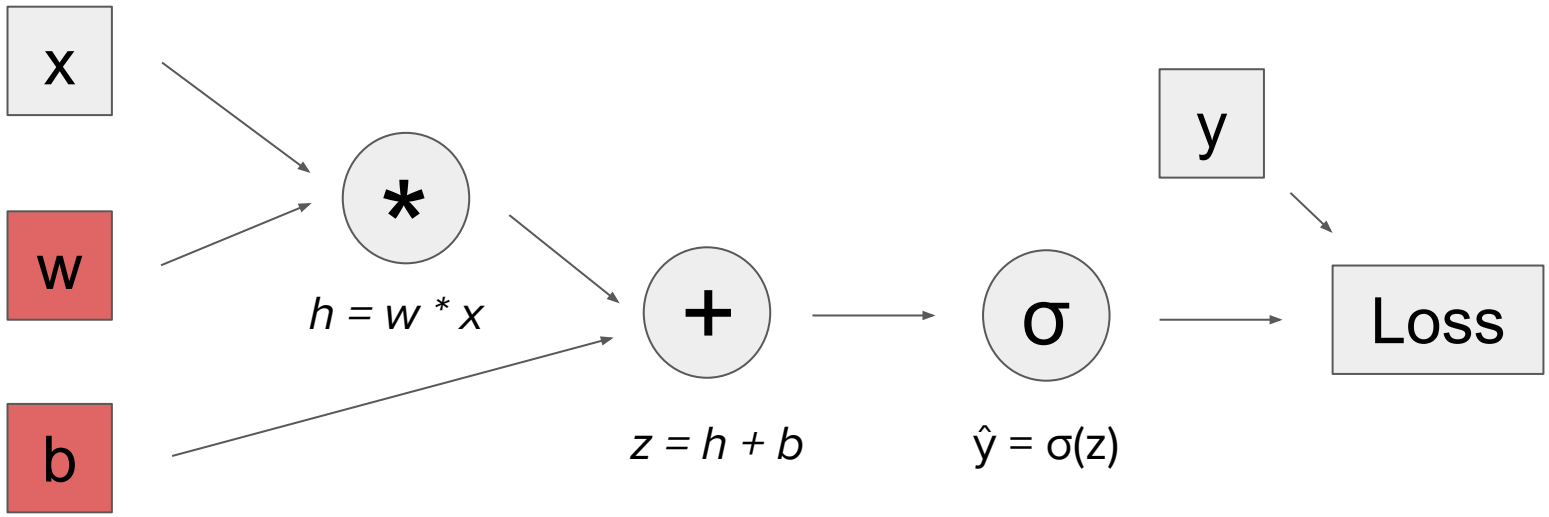
$$w = w - \eta * \frac{\partial Loss}{\partial w}$$

$$b = b - \eta * \frac{\partial Loss}{\partial b}$$

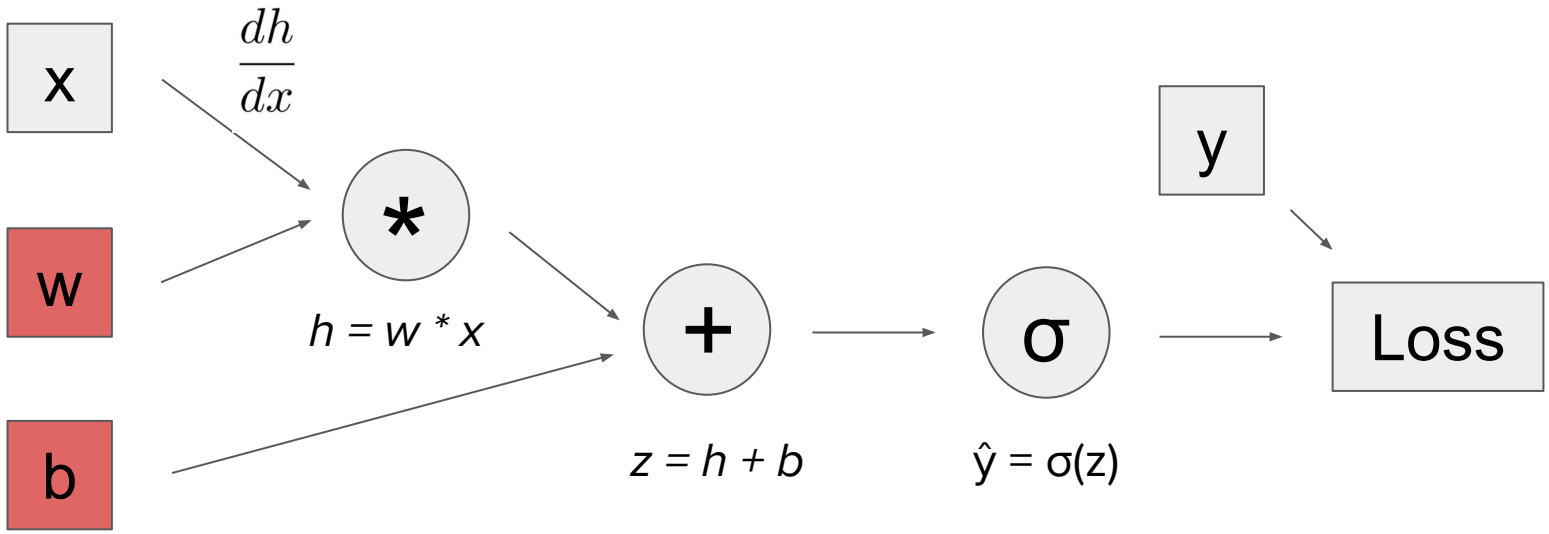
How do we calculate the gradients?


x
w
b
*
h = w * x
+
z = h + b
σ
ŷ = σ(z)
y
Loss

Backpropagation

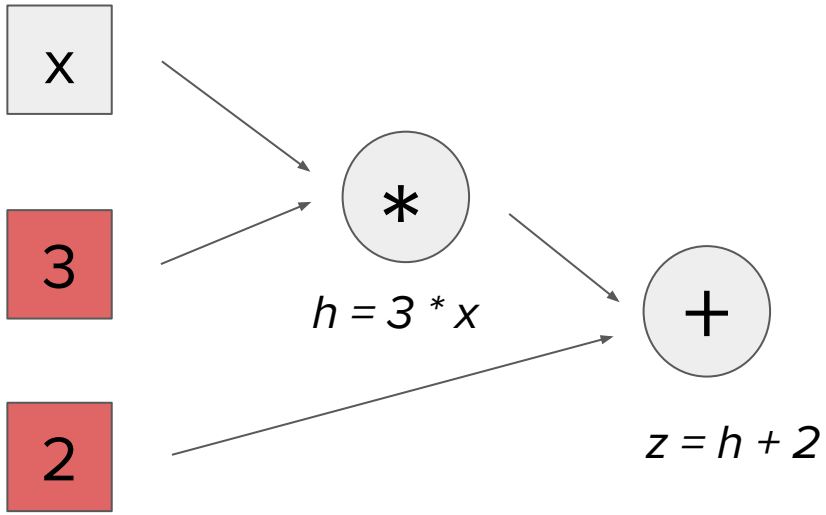


Backpropagation



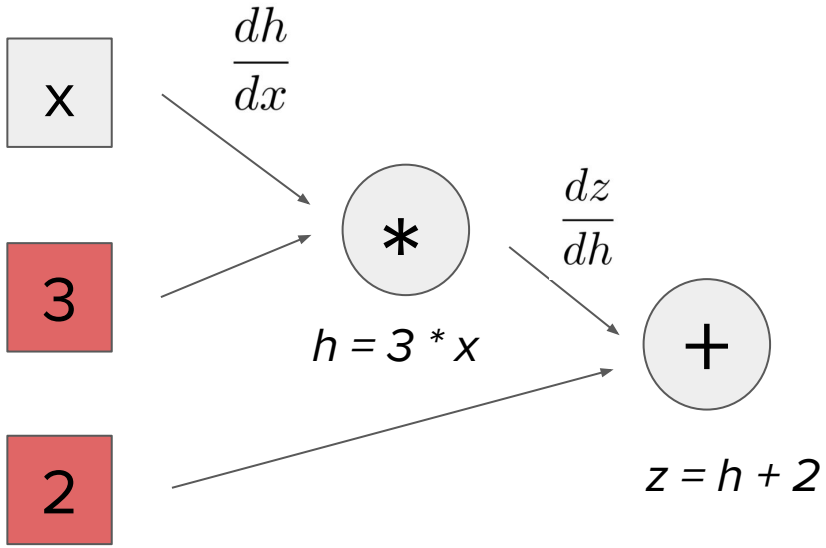
Backpropagation

$3x + 2$



Backpropagation

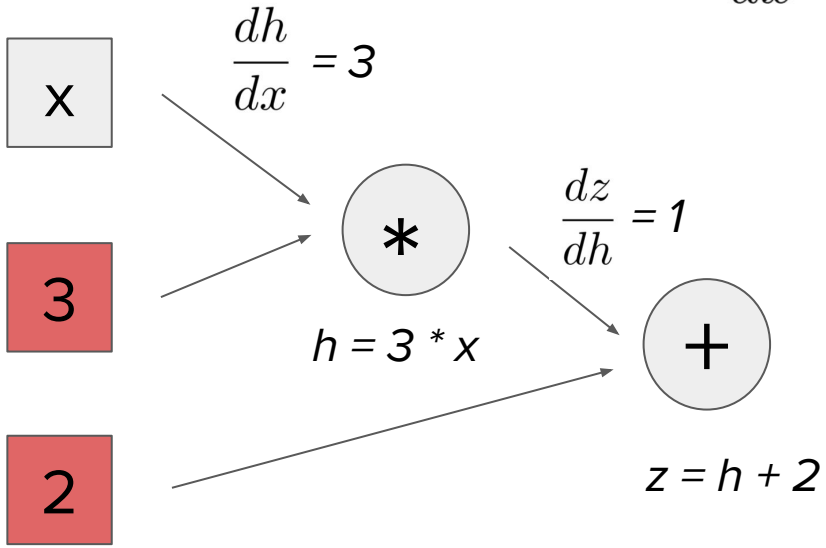
3x + 2



Backpropagation

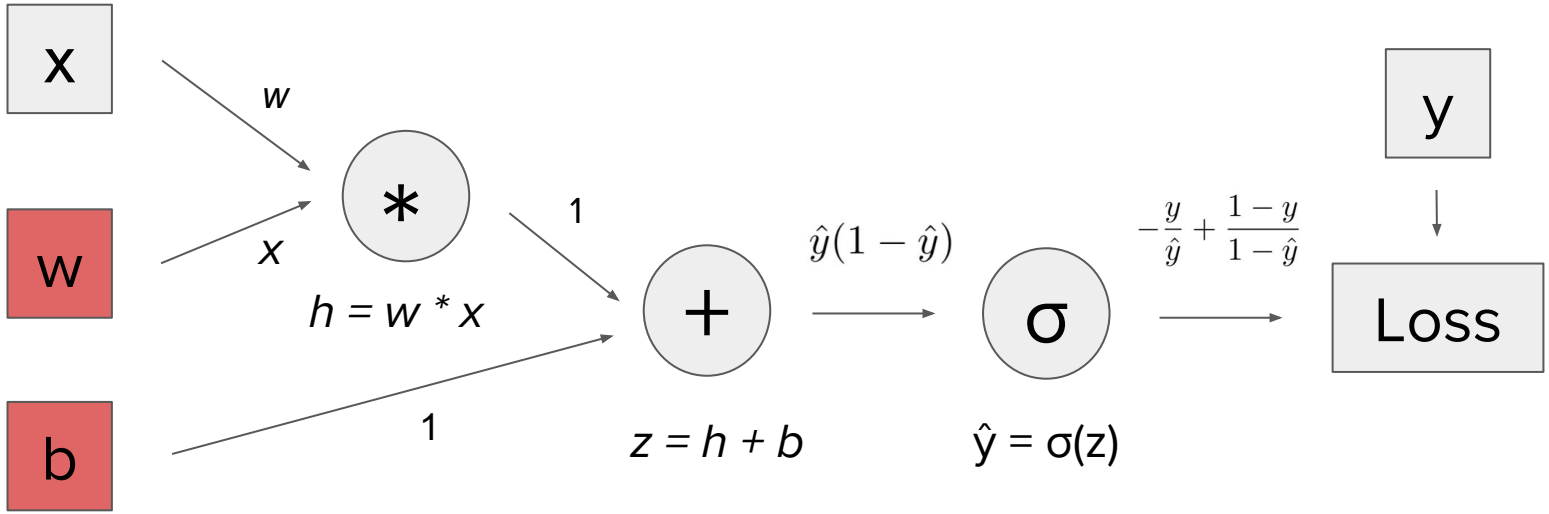
3x + 2

$$\frac{dz}{dx} = \frac{dz}{dh}\frac{dh}{dx} = 3$$

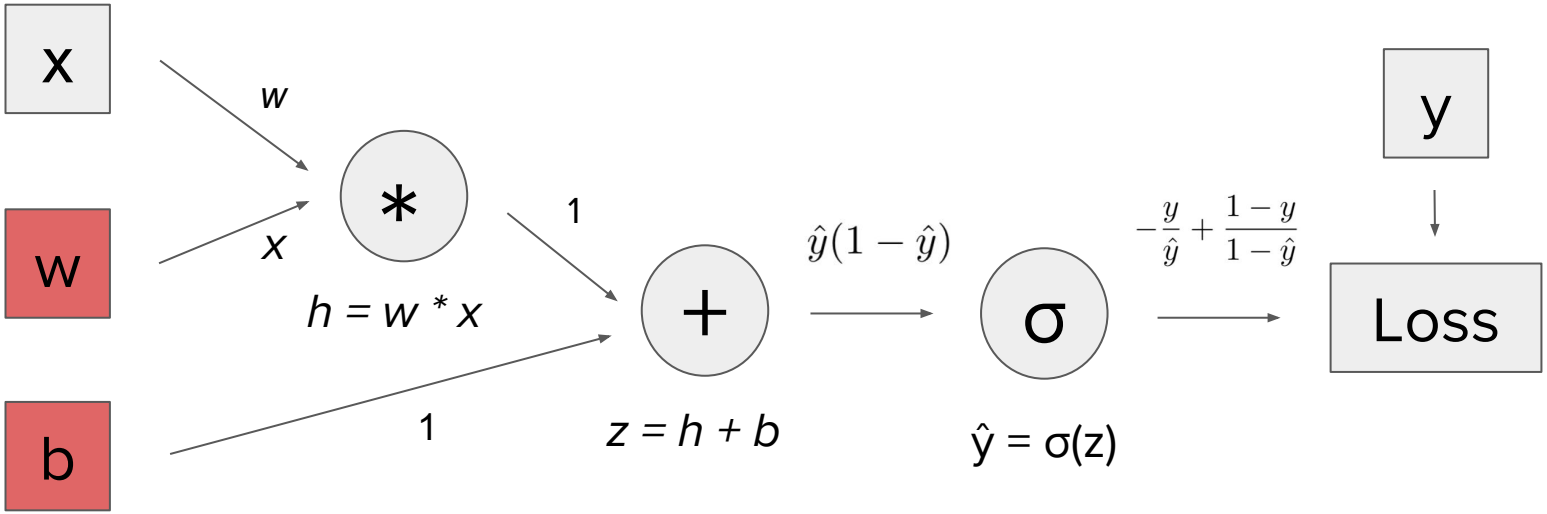


Questions?

Backpropagation



Backpropagation



$$\frac{\partial Loss}{\partial w} = (\hat{y} - y) * x \qquad \frac{\partial Loss}{\partial b} = \hat{y} - y$$

Training Loop

```
for each example:
    Calculate y_hat
    Calculate loss for this example
    Calculate gradients dw, db
    w -= learning_rate * dw
    b -= learning_rate * db
```

Jupyter Notebook Exercises: Part 2

You'll need:

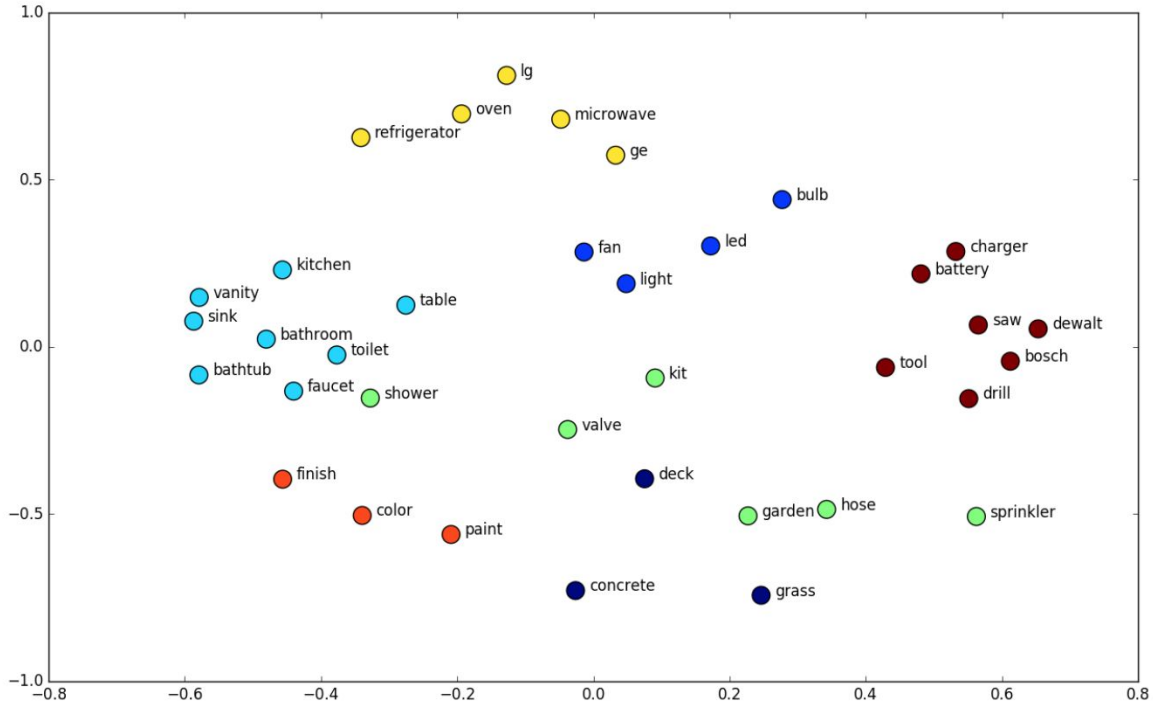
```
np.dot(a, b)
np.exp(x)
np.log(x)
np.random.randn(size)
np.random.rand()
```

$$\frac{\partial Loss}{\partial w} = (\hat{y} - y) * x \qquad \frac{\partial Loss}{\partial b} = \hat{y} - y$$

What we've learned...

- AI for social good is important!
- How to build a linear model
- How to build a logistic regression model
- The basics of machine learning
- Word vectors contain alarming gender biases

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings



Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, Adam Kalai