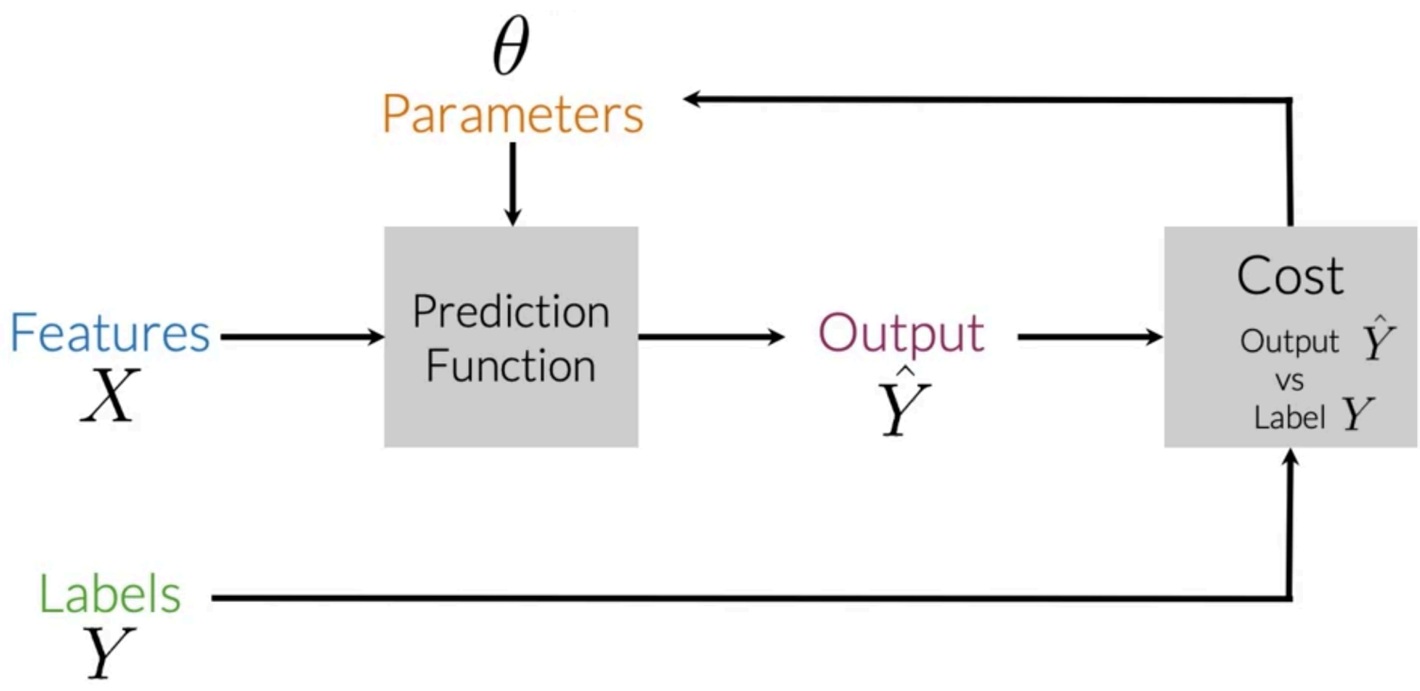
Natural Language Processing with Classification and Vector Spaces

Sentiment Analysis with Logistic Regression

2021.01.28

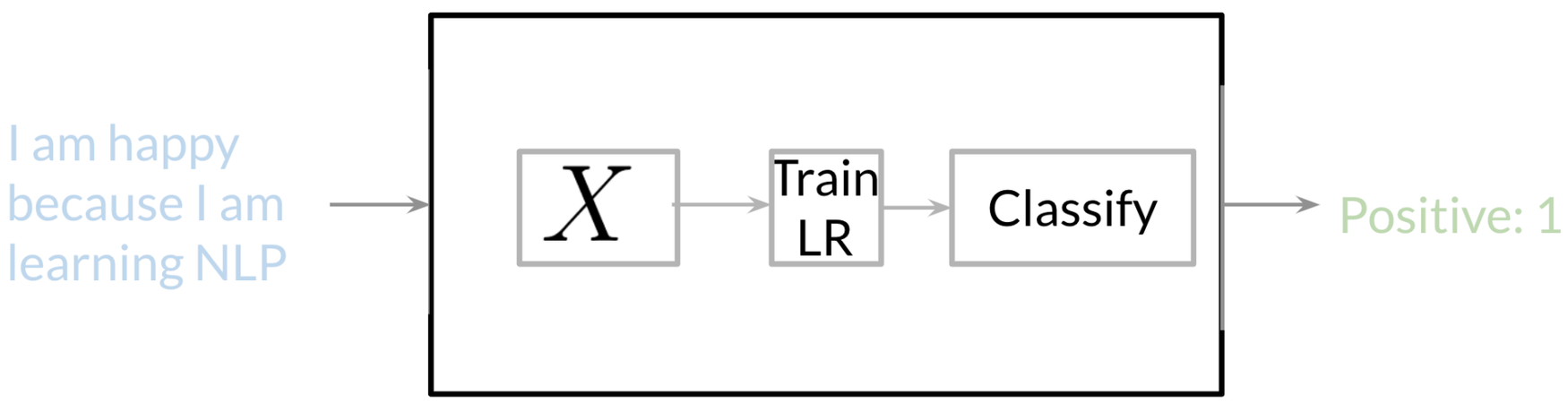
Supervised ML & Sentiment Analysis

- Supervised machine learning process:
  - Compare the predicted values of the inputs with the correct answers.
  - The difference represents the cost and updates the parameters of the prediction function.



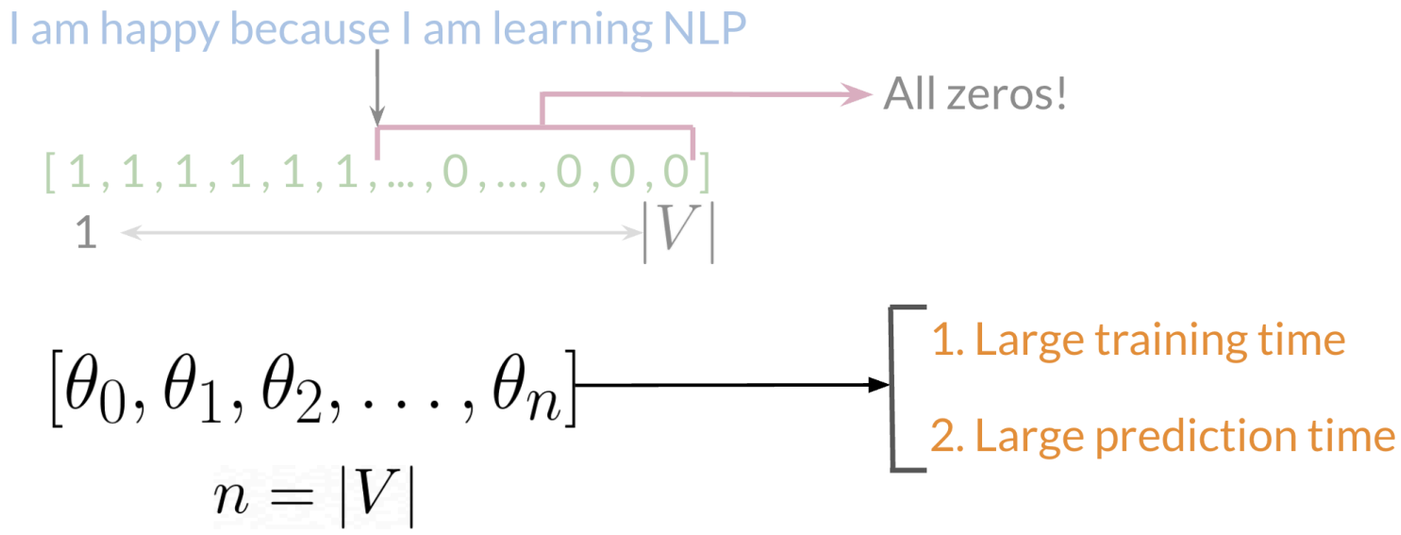
Supervised ML & Sentiment Analysis

- Sentiment Analysis
  - Represent the text as features, then train logistic regression classifier.
  - Then use it to classify the text.
  - Classify 1, for a positive sentiment, or 0, for a negative sentiment.



Vocabulary & Feature Extraction

- Represent text as a vector of dimension V (vocabulary size).
- Put a 1 in the corresponding index for any word, and a 0 otherwise.



- As V gets larger, the vector becomes more sparse.
  - Having many more features, training parameters → larger training/prediction time.

Feature Extraction with Frequencies

- Use as features into logistic regression classifier.
  - Keep track of the number of times, that's where it shows up as positive/negative class.

Positive tweets	Negative tweets
I am happy because I am learning NLP	I am sad, I am not learning NLP
I am happy	I am sad

- Encode each tweet as a vector.

Vocabulary	PosFreq (1)	NegFreq (0)
I	3	3
am	3	3
happy	2	0
because	1	0
learning	1	1
NLP	1	1
sad	0	2
not	0	1

freqs: dictionary mapping from (word, class) to frequency

Feature Extraction with Frequencies

- In the table → How words like happy and sad tend to take clear sides.
  - Represent it with a vector of dimension 3 = Feature.

Vocabulary	PosFreq (1)
I	3
am	3
happy	2
because	1
learning	1
NLP	1
sad	0
not	0

I am sad, I am not learning NLP

$$X_m = [1, \sum_w freqs(w, 1), \sum_w freqs(w, 0)]$$

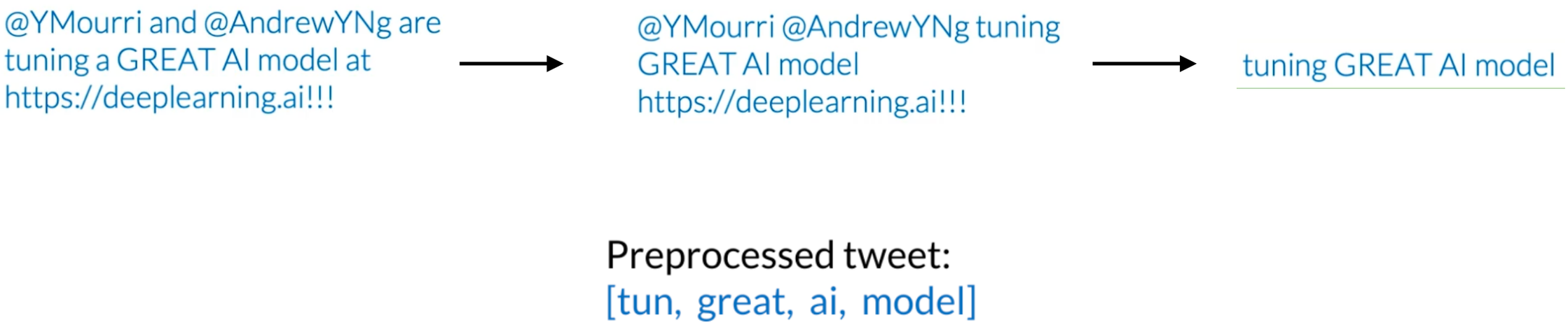
8

11

- End up getting the following feature vector [1, 8, 11].

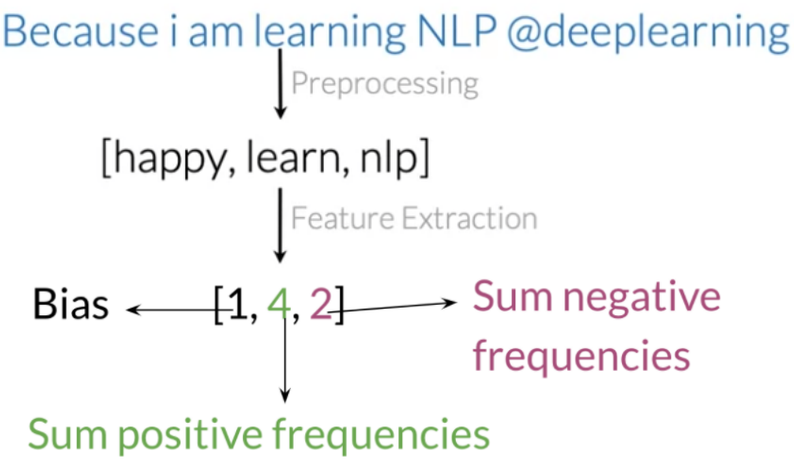
Preprocessing

- Perform the following:
    1. Eliminate handles and URLs.
    2. Tokenize the string into words.
    3. Remove stop words like ‘and, is, a, on, etc.’.
    4. Stemming or convert every word to its stem. (dancer, dancing, danced → ‘danc’)
    5. Convert all your words to lower case.



Putting it all together

- Text → Preprocessing → Feature extraction → Numerical reps.

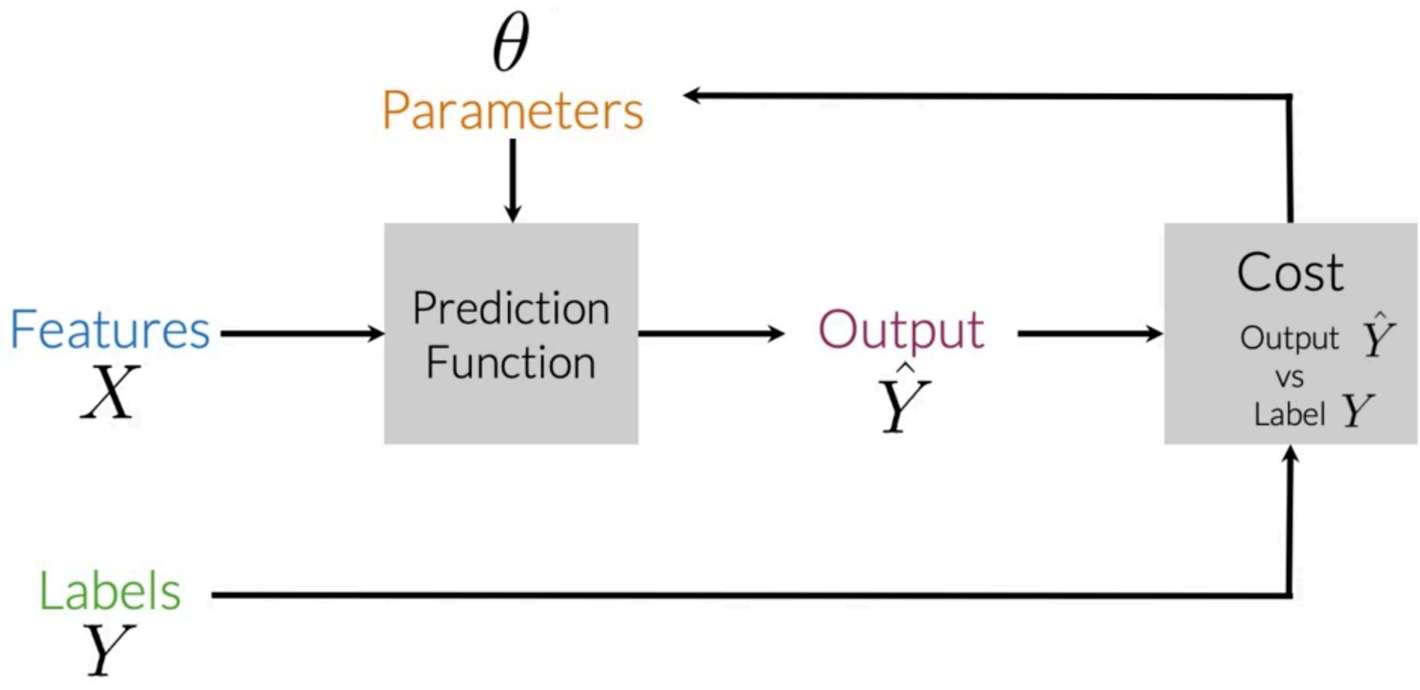


- Your input X becomes of dimension $(m,3)$ as follows:

$$\boldsymbol{X} = \begin{bmatrix} 1 & X_1^{(1)} & X_2^{(1)} \\ 1 & X_1^{(2)} & X_2^{(2)} \\ \vdots & \vdots & \vdots \\ 1 & X_1^{(m)} & X_2^{(m)} \end{bmatrix}$$

Logistic Regression Overview

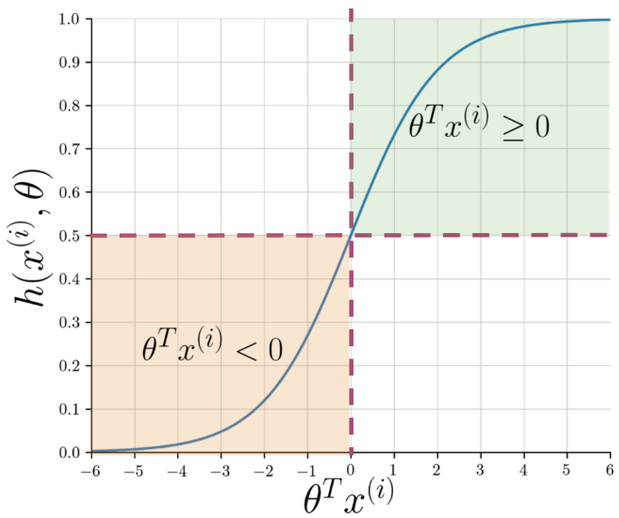
- Same as supervised learning process
  - For logistic regression, prediction function F is equal to the sigmoid function.



Logistic Regression Overview

- Use the sigmoid function which outputs a probability between 0 and 1.
  - With some weight parameter θ and some input x^i is defined as follows:

$$h(x^{(i)}, \theta) = \frac{1}{1 + e^{-\theta^T x^{(i)}}}$$



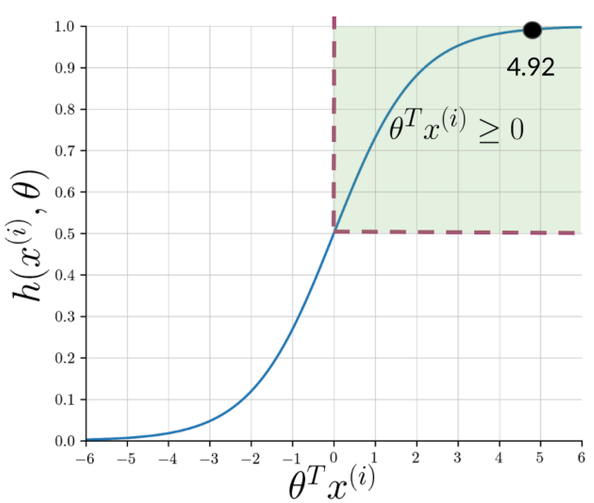
@YMourri and @AndrewYNg are tuning a GREAT AI model

↓

[tun, ai, great, model]

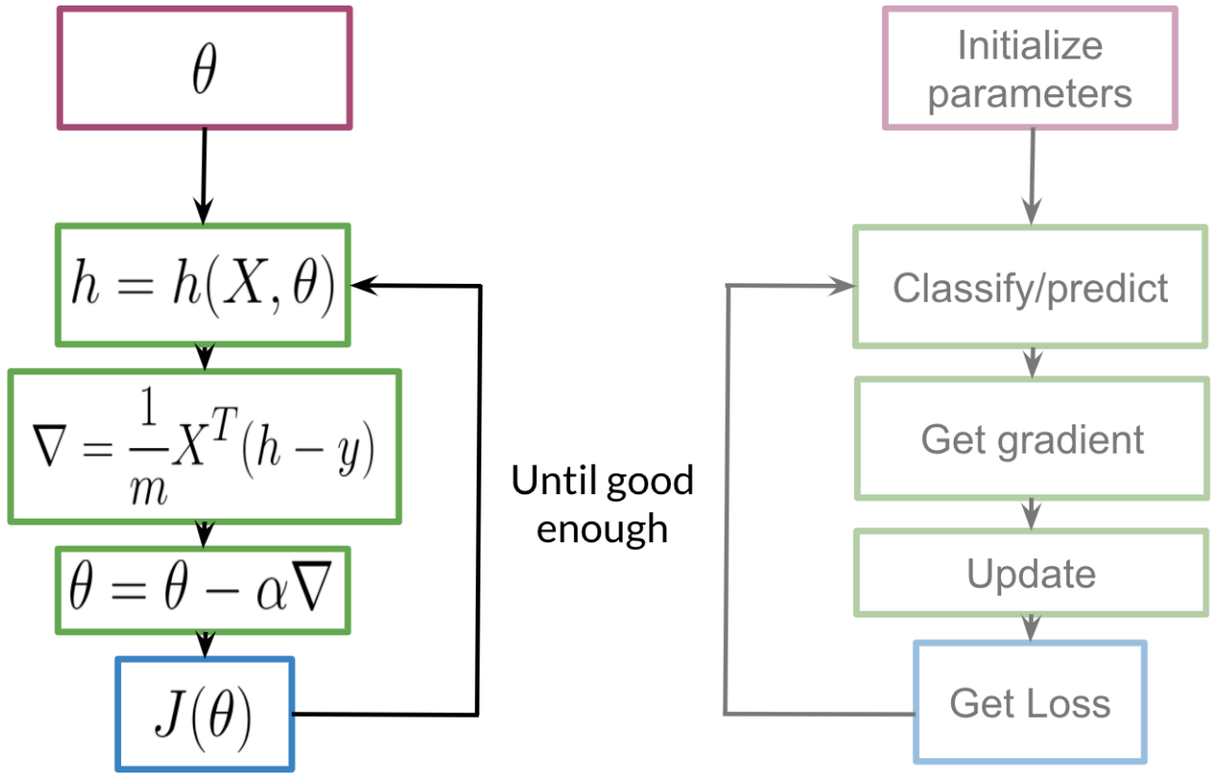
↓

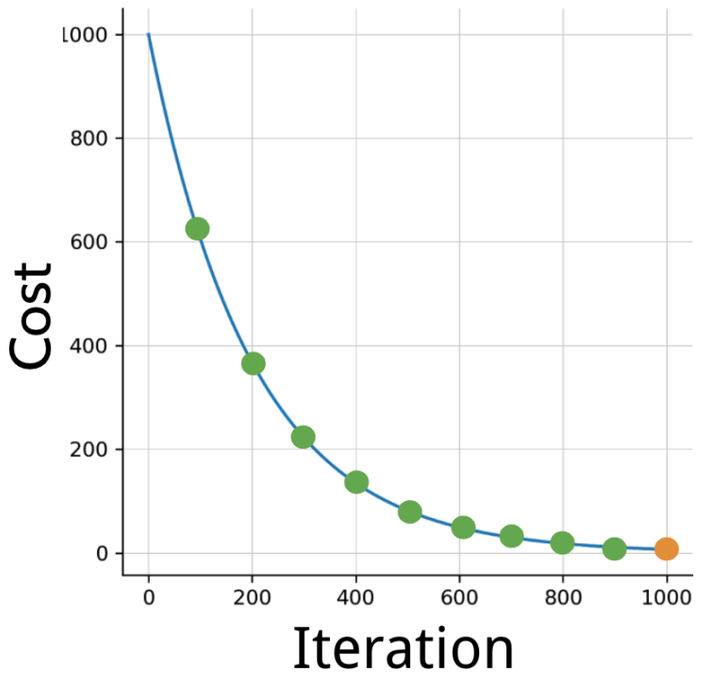
$$x^{(i)} = \begin{bmatrix} 1 \\ 3476 \\ 245 \end{bmatrix} \quad \theta = \begin{bmatrix} 0.00003 \\ 0.00150 \\ -0.00120 \end{bmatrix}$$



Logistic Regression: Training

- The process:
  - Initialize θ, use in sigmoid, compute the gradient, update θ, then calculate the cost until good enough.
  - = Keep training until the cost converges.





Logistic Regression: Testing

- Test the validation set on model to get predictions.
  - The predictions are the outputs of the sigmoid function:
  - Over 0.5, assign it to a positive class, otherwise, a negative class.

$$X_{val} \; Y_{val} \; \theta$$

$$h(X_{val}, \theta)$$

$$pred = h(X_{val}, \theta) \geq 0.5$$

$$\begin{bmatrix} 0.3 \\ 0.8 \\ 0.5 \\ \vdots \\ h_m \end{bmatrix} \geq 0.5 = \begin{bmatrix} 0.3 \geq 0.5 \\ 0.8 \geq 0.5 \\ 0.5 > 0.5 \\ \vdots \\ pred_m \geq 0.5 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ pred_m \end{bmatrix}$$

- To compute accuracy,

Accuracy $\longrightarrow$ $$\sum_{i=1}^{m} \frac{(pred^{(i)} == y_{val}^{(i)})}{m}$$

Logistic Regression: Cost Function

- Logistic regression cost function:
  - $$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log h\left(x^{(i)},\theta\right) + \left(1-y^{(i)}\right)\log\left(1-h\left(x^{(i)},\theta\right)\right)\right]$$

$y^{(i)}$	$h(x^{(i)},\theta)$	
0	any	0
1	0.99	~0
1	~0	-inf

$y^{(i)}$	$h(x^{(i)},\theta)$	
1	any	0
0	0.01	~0
0	~1	-Inf