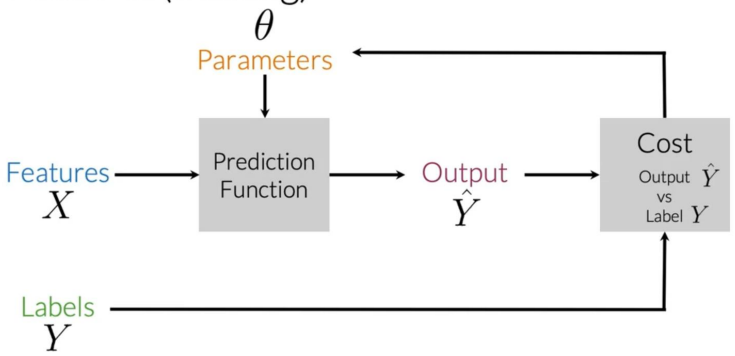
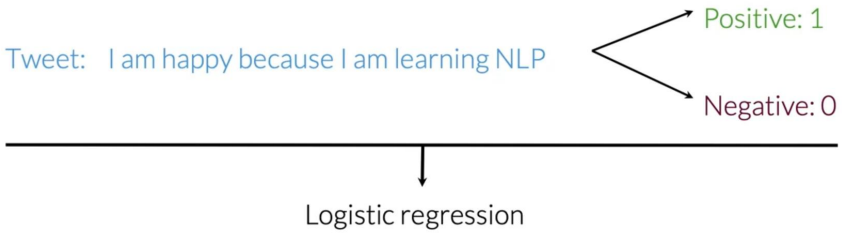
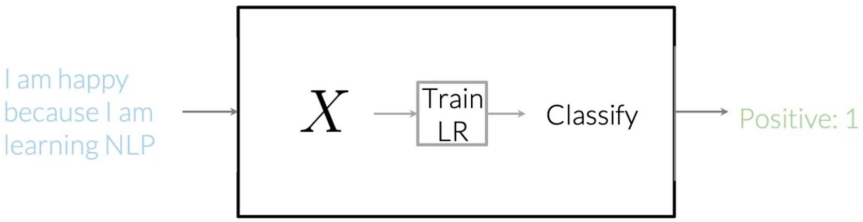
Supervised ML (training)



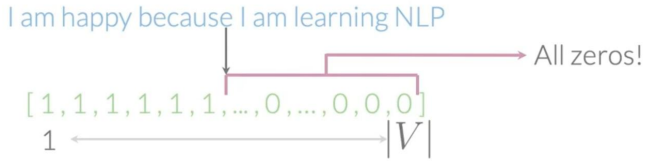
Sentiment analysis



Sentiment analysis



Problems with sparse representations



$$[\theta_0, \theta_1, \theta_2, \ldots, \theta_n]$$

$$n = |V|$$

1. Large training time
2. Large prediction time

Vocabulary

Tweets:
[tweet_1, tweet_2, ..., tweet_m]

I am happy because I am learning NLP
...
...
...
I hated the movie

$V =$

[I, am, happy, because, learning, NLP, ... hated, the, movie]

Feature extraction

I am happy because I am learning NLP

[I , am, happy, because, learning, NLP, ... hated, the, movie]

[1, 1, 1, 1, 1, 1, ... 0, 0, 0]

A lot of zeros! That's a sparse representation.

Positive and negative counts

Corpus

I am happy because I am learning NLP
I am happy
I am sad, I am not learning NLP
I am sad

Vocabulary
I
am
happy
because
learning
NLP
sad
not

Positive and negative counts

Positive tweets

I am happy because I am learning NLP
I am happy

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

Positive and negative counts

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

Negative tweets

I am sad, I am not learning NLP

I am sad

Word frequency in classes

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	1
not	0	1

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

Feature extraction

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

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

Features of tweet m

Bias

Sum Pos. Frequencies

Sum Neg. Frequencies

Feature extraction

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

I am sad, I am not learning NLP

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

11

Feature extraction

I am sad, I am not learning NLP

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

$$\downarrow$$

$$X_m = [1, 8, 11]$$

Preprocessing: stop words and punctuation

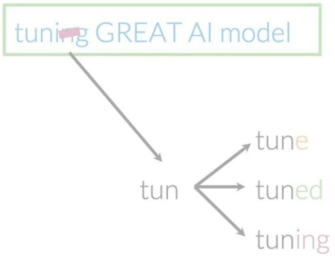
@YMourri @AndrewYNg tuning GREAT AI model https://deeplearning.ai!!!

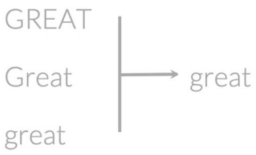
@YMourri @AndrewYNg tuning GREAT AI model https://deeplearning.ai

Stop words
and
is
a
at
has
for
of

Punctuation
,
.
:
!
"
'

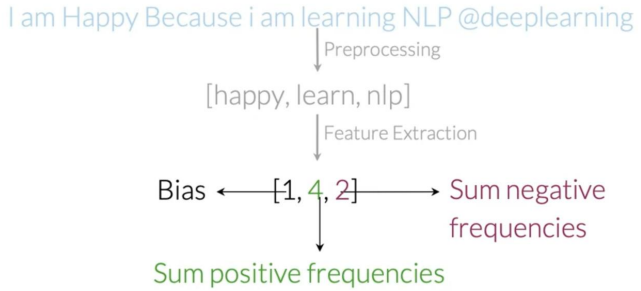
Preprocessing: Stemming and lowercasing





Preprocessed tweet:
[tun, great, ai, model]

General overview



General overview

I am Happy Because i am learning NLP @deeplearning

I am sad not learning NLP

...

I am sad :(

→ [happy, learn, nlp]

[sad, not, learn, nlp]

...

[sad]

→ [[1, 40, 20],

[1, 20, 50],

...

[1, 5, 35]]

General overview

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

[1, 40, 20]

[1, 20, 50],

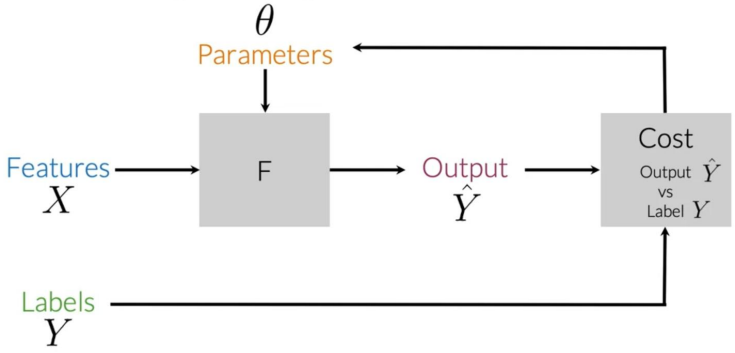
...

[1, 5, 35]]

General Implementation

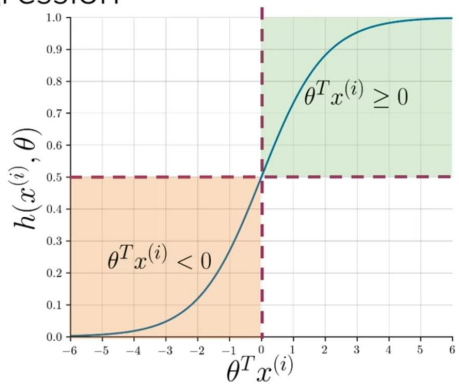
```
freqs = build_freqs(tweets,labels) #Build frequencies dictionary
X = np.zeros((m,3)) #Initialize matrix X
for i in range(m): #For every tweet
        p_tweet = process_tweet(tweets[i]) #Process tweet
    X[i,:] = extract_features(p_tweet,freqs) #Extract Features
```


Overview of logistic regression



Overview of logistic regression

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



Overview of logistic regression

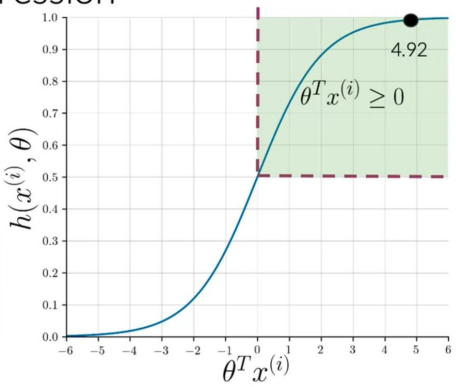
@YMourri and
@AndrewYNg are tuning a
GREAT AI model

↓

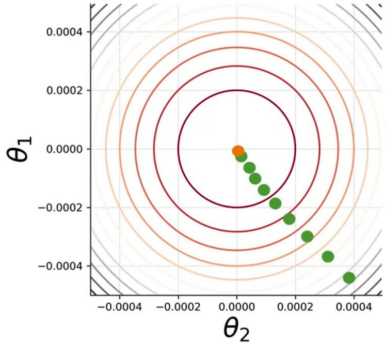
[tun, ai, great, model]

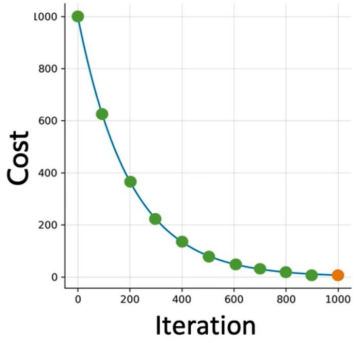
↓

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

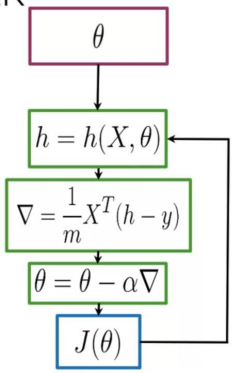


Training LR

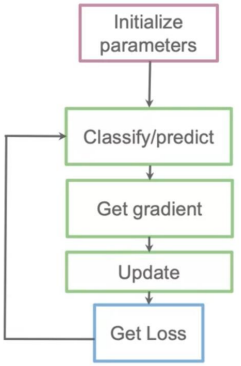




Training LR



Until good enough



Testing logistic regression

- X_{val} Y_{val} θ

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

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

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

Testing logistic regression

- X_{val} Y_{val} θ

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

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

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

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ pred_m \end{bmatrix} == \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ Y_{val_m} \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ pred_m == Y_{val_m} \end{bmatrix}$$

Testing logistic regression

$$Y_{val} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad pred = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (Y_{val} == pred) = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$\text{accuracy} = \frac{4}{5} = 0.8$$

Cost function for logistic regression

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} \log h(x^{(i)}, \theta) + (1 - y^{(i)}) \log(1 - h(x^{(i)}, \theta))]$$