Live session
Methods in AI research

Dong Nguyen

17 Sept 2020

Utrecht University

Practicalities

- This session won't be recorded
- Please mute your mic.
- One hour. Afterwards there's an virtual "office hour".
- Structure
    - Discussion topics/questions related to the quiz
    - Remaining topics/questions

- Next week
    - Chris Janssen is taking over

- Q: *Will there be math questions in the exam? Can we bring a calculator? Can we have a cheat sheet?*
- Q: *In what detail do we need to know RNNs? A: Only the high-level idea.*

I posted additional exercises on blackboard (folder for this week)

Features

You like to train a machine learning system to predict whether a book will be come a "bestseller". You've collected a large dataset, and for each book you have the following information:

- The author: You have 1000 unique authors in your dataset 1000
- Has the author written a bestseller before? Yes or no 1
- Genre: {Crime, Fantasy, Historical Fiction, Science Fiction, Thriller} 5
- The number of pages of the book 1

Each book is one instance in your dataset. You first need to represent each book as a vector before training your machine learning model.

really, a vector of more than 1000?

Each book will be represented as a [?]-dimensional vector. (Fill in the correct number.)

Representing the author

Suppose we use the first dimension to encode the author

A = [4, …, …]

B = [6, …, …]

C = [1, …, …]

1 = Hemingway
2 = Shakespeare
3 = Kafka
4 = Austen
5 = Galman
6 = King
7 = Grisham
…

k-NN with Manhattan distance

$$\sum |a_i - b_i|$$

Representing the author

Suppose we use the first dimension to encode the author

A = [4, ..., ...]

B = [6, ..., ...]

C = [1, ..., ...]

1 = Hemingway
2 = Shakespeare
3 = Kafka
4 = Austen
5 = Galman
6 = King
7 = Grisham
...

Better (**one hot encoding**):

A = [0,0,0,1,0,0, ..., ...]

B = [0,0,0,0,0,1, ..., ...]

C = [1,0,0,0,0,0, ..., ...]

Having different authors increases the Manhattan distance with 2
Same author: 0

Come up with a task for which bag of words is probably sufficient.

- language identification
- topic classification
- spam classification
- finding keywords
- classification text types

And another task for which it is not.

- translation
- coreference resolution
- parsing
- checking if two essays come to the same conclusion
- text generation
- grammar checker

“document classification”
“sentiment analysis”

Come up with a task for which bag of words is probably sufficient.

- language identification
- topic classification
- spam classification
- finding keywords
- classification text types

"document classification"
"sentiment analysis"

And another task for which it is not.

- translation
- coreference resolution
- parsing
- checking if two essays come to the same conclusion
- text generation
- grammar checker

You work on a text classification problem and each document is represented by a vector with the frequency counts of words in your document.

You now train a **logistic regression model**. This is a bag of words representation [True/False]

Come up with a task for which bag of words is probably sufficient.

- language identification
- topic classification
- spam classification
- finding keywords
- classification text types

"document classification"
"sentiment analysis"

And another task for which it is not.

- translation
- coreference resolution
- parsing
- checking if two essays come to the same conclusion
- text generation
- grammar checker

You work on a text classification problem and each document is represented by a vector with the frequency counts of words in your document.

You now train **a feed forward neural network with 3 hidden layers**. This is a bag of words representation [True/False]

Jaccard similarity

`Bob and John just went to the grocery store`
`Bob bought the book at the auction`

union: {Bob, and, John, just, went, to, the, grocery, store, bought, book, at, auction} = 13
intersection = {Bob, the} = 2

Jaccard = 0.1538

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Cosine similarity

```
Bob and John just went to the grocery store
Bob bought the book at the auction
```

What would be the cosine similarity between these two sentences? (assume each sentence is represented by a vector with word frequencies)

Cosine similarity

```
Bob and John just went to the grocery store
Bob bought the book at the auction
```

```
Bob  and ....                 the
[1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
[1, 0, 0, 0, 0, 0, 2, 0, 0, 1, 1, 1, 1]
```

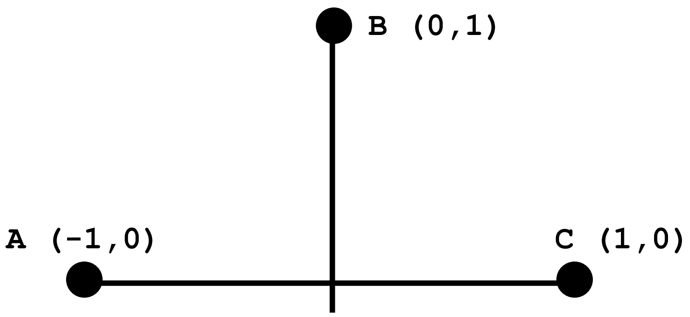
$\boldsymbol{a} \bullet \boldsymbol{b} = 3$
$\|\boldsymbol{a}\| = \text{sqrt}(9) = 3$
$\|\boldsymbol{b}\| = \text{sqrt}(9) = 3$

cosine similarity = 3/9 = 1/3

Cosine similarity

$$= \frac{\boldsymbol{a} \bullet \boldsymbol{b}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|} = \frac{\sum \boldsymbol{a}_i \boldsymbol{b}_i}{\sqrt{\sum \boldsymbol{a}_i^2}\sqrt{\sum \boldsymbol{b}_i^2}}$$

Cosine similarity



$$\text{cosine similarity}$$

$$\frac{\boldsymbol{a} \bullet \boldsymbol{b}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|} = \frac{\sum \boldsymbol{a}_i \boldsymbol{b}_i}{\sqrt{\sum \boldsymbol{a}_i^2} \sqrt{\sum \boldsymbol{b}_i^2}}$$

Cosine ranges from -1 (vectors pointing in opposite directions) to 0 (orthogonal) to 1 (vectors pointing in the same direction).

When documents are represented by word frequency counts, the elements are non-negative. The cosine similarity therefore has to lie in [0,1]

Noisy features

You have a dataset where each instance is represented by 100 features. However, a large fraction of these features are noisy and not useful signals for making the classifications. Which of these two classifiers do you think would perform better?

- **Logistic Regression**
- k-Nearest Neighbors
- I expect both to perform similarly

bias

weights

$$z = \underbrace{b}_{\text{bias}} + \underbrace{w_1}_{\text{weights}} x_1 + \ \ldots + w_d x_d$$

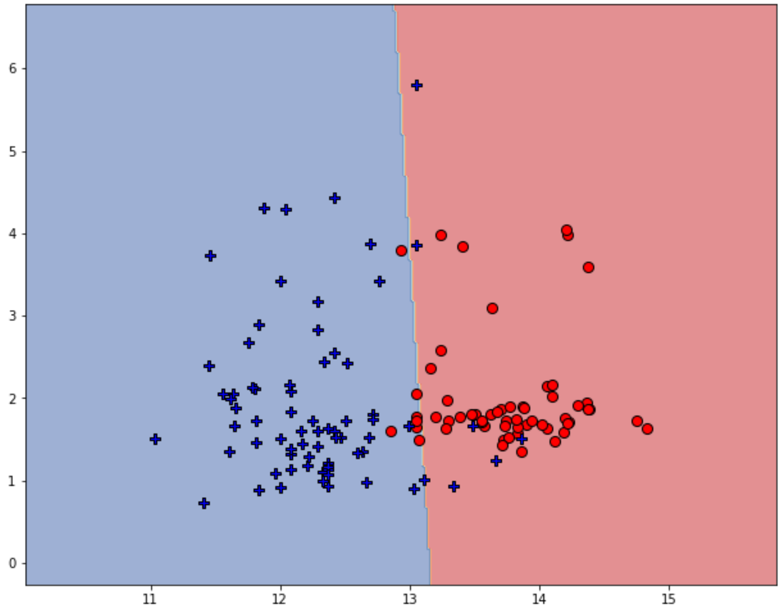
LR

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

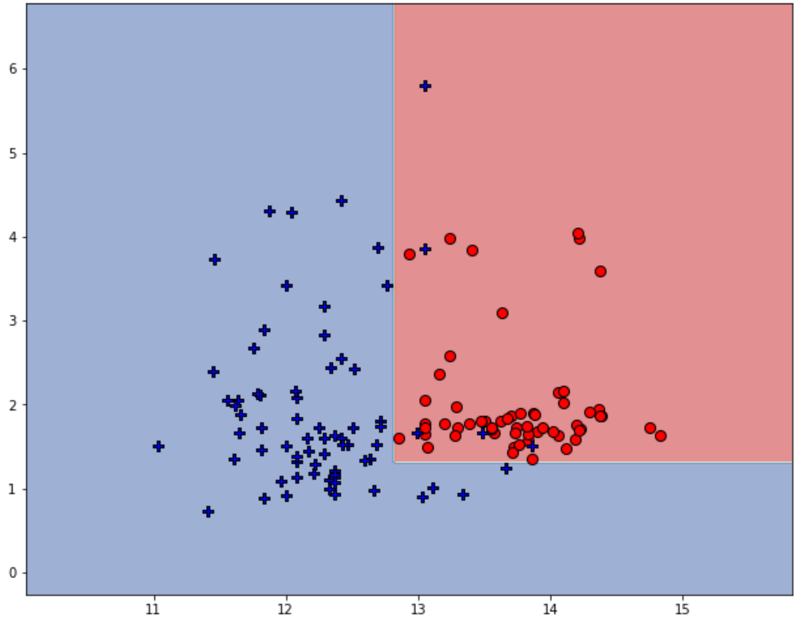
Manhattan distance $\sum |a_i - b_i|$

Decision boundaries

Logistic regression



Decision tree

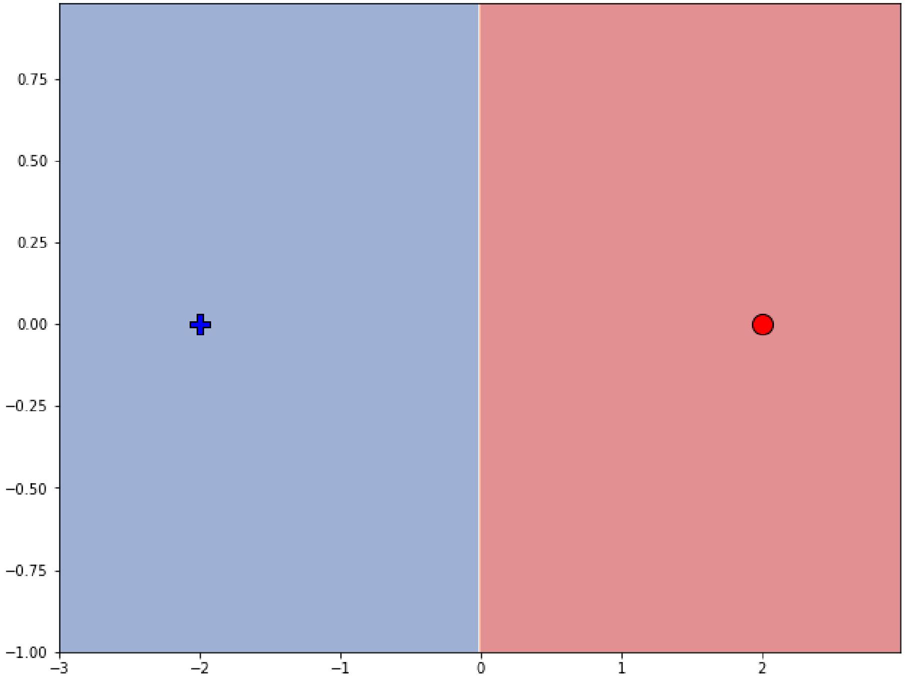


Can we have a training set for which logistic regression and a decision tree will learn the same decision boundary?

Decision boundaries

A decision tree
and logistic regression
will have the same
decision boundary

What about 1NN?

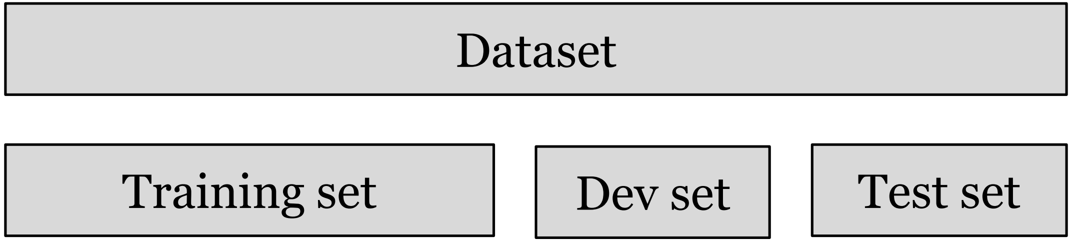


Your friend claims that her logistic regression classifier achieves better performance than a state-of-the-art system. Her system uses L2 regularization with lambda set to 0.75 as this performed best on the test set.

Explain:
(1) why you can't trust her claim and
(2) what she should have done instead

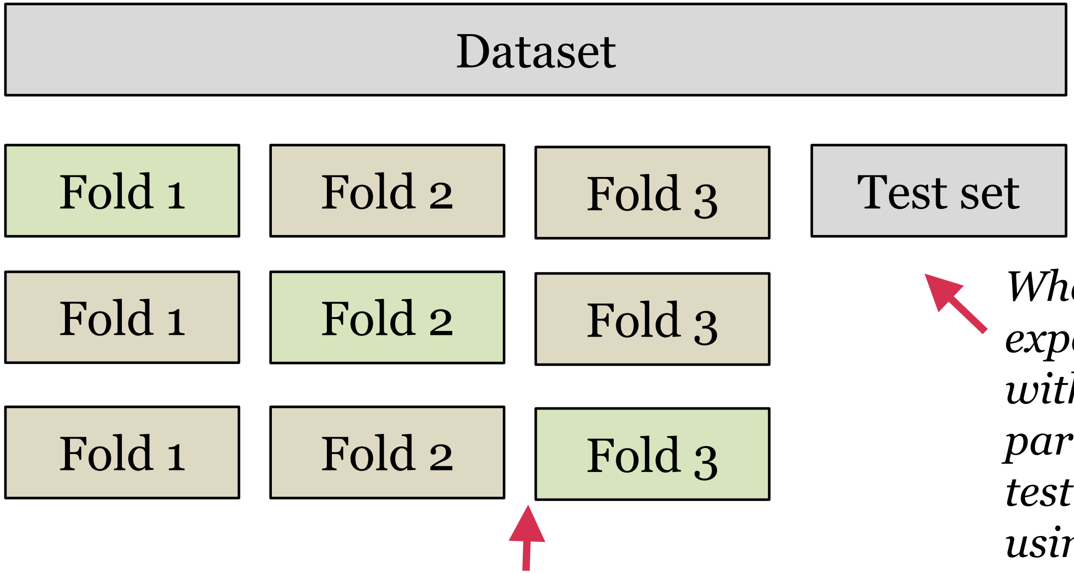
RECAP!

Cross validation



RECAP!

Cross validation



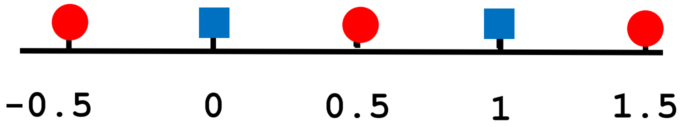
leave one out validation

When you're done with experimenting with features, tuning parameters etc. test your final model using this data.

Train and tune your parameters on folds 1-3

E.g. train on folds 2 and 3, test on fold 1.
Usually 10 folds (i.e. 10-fold cross validation), but depends on the data

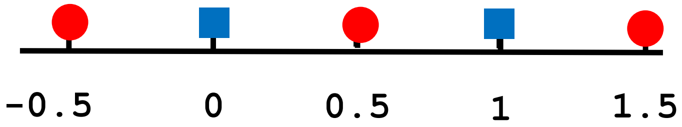
Leave one out validation and kNN



What would be the leave-one-out cross validation error on this dataset using a 1-NN? (provide the answer as the number of errors)

The 1-NN uses the Euclidian distance.

Leave one out validation and kNN



What would be the leave-one-out cross validation error on this dataset using a 1-NN? (provide the answer as the number of errors)

The 1-NN uses the Euclidian distance.

5

Learning the parameters

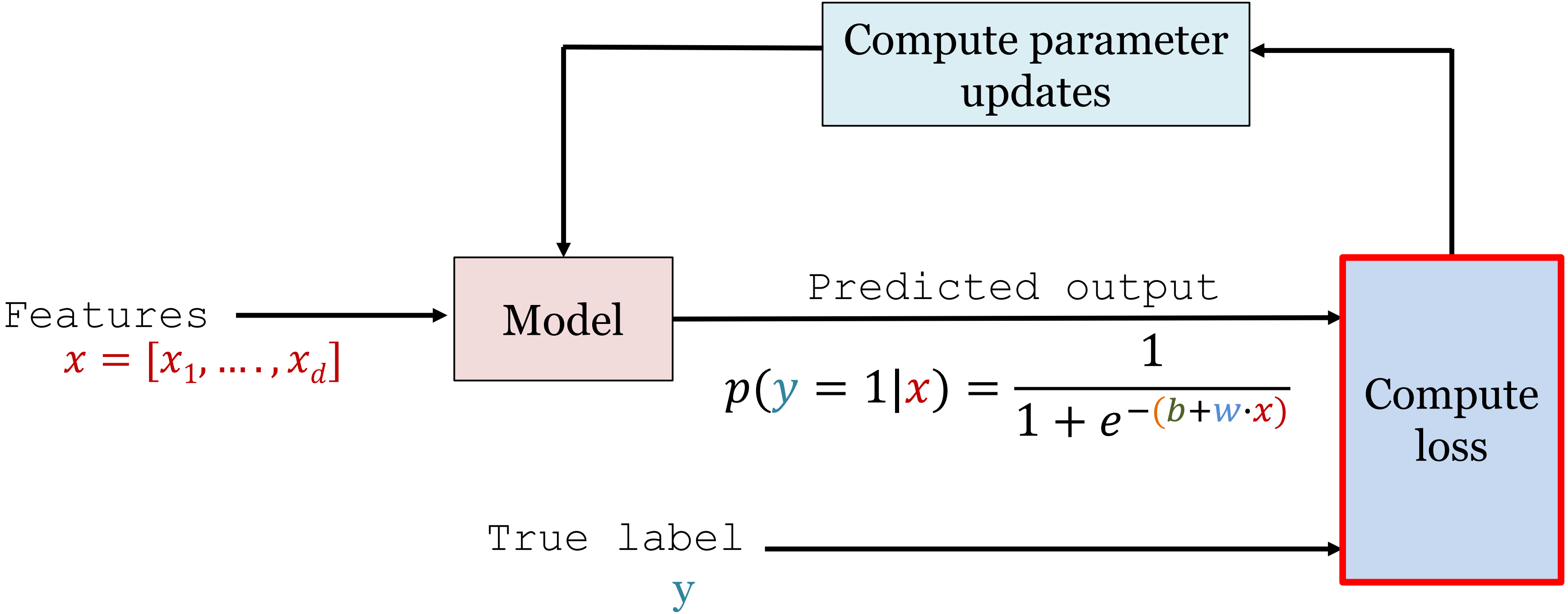


Figure based on https://developers.google.com/machine-learning/crash-course/reducing-loss/an-iterative-approach

RECAP!

Entropy & cross-entropy

	p
A	0.5
B	0.25
C	0.25

Entropy:

$$H(S) = -\sum_i p_i \log_2 p_i$$

"the amount of randomness"
"the average number of yes/no questions to guess a draw from S"

Heads? → A 1 * 0.5 = 0.5
Tails? → Heads? → B 2 * 0.25 = 0.5
Tails? → C 2 * 0.25 = 0.5

On average we need 1.5 questions

-0.5*log2(0.5)-0.25*log2(0.25)
-0.25*log2(0.25) = 1.5

Entropy & cross-entropy

Entropy:

$$H(S) = -\sum_i p_i \log_2 p_i$$

"the amount of randomness"
"the average number of yes/no questions to guess a draw from S"

Heads? → A 1 * 0.5 = 0.5
Tails? → Heads? → B 2 * 0.25 = 0.5
Tails? → C 2 * 0.25 = 0.5

On average we need 1.5 questions

-0.5*log2(0.5)-0.25*log2(0.25)
-0.25*log2(0.25) = 1.5

	q	p
A	0.5	1
B	0.25	0
C	0.25	0

p: true label distribution
q: predicted label distribution

$$H(p,q) = -\sum p(x)\log(q(x))$$

How many yes/no questions would you need to ask to guess a draw from p given the encoding for q? → 1 (-log2(0.5) = 1)

How are cross entropy and cross entropy loss related?

Entropy & cross-entropy

Entropy:

$$H(S) = -\sum_i p_i \log_2 p_i$$

"the amount of randomness"
"the average number of yes/no questions to guess a draw from S"

Heads? → A 1 * 0.5 = 0.5
Tails? → Heads? → B 2 * 0.25 = 0.5
Tails? → C 2 * 0.25 = 0.5

On average we need 1.5 questions

-0.5*log2(0.5)-0.25*log2(0.25)
-0.25*log2(0.25) = 1.5

	q	p
A	0.5	0
B	0.25	1
C	0.25	0

p: true label distribution
q: predicted label distribution

$$H(p,q) = -\sum p(x)\log(q(x))$$

How many yes/no questions would you need to ask to guess a draw from p given the encoding for q? → 2 (-log2(0.25) = 2)

Learning the parameters

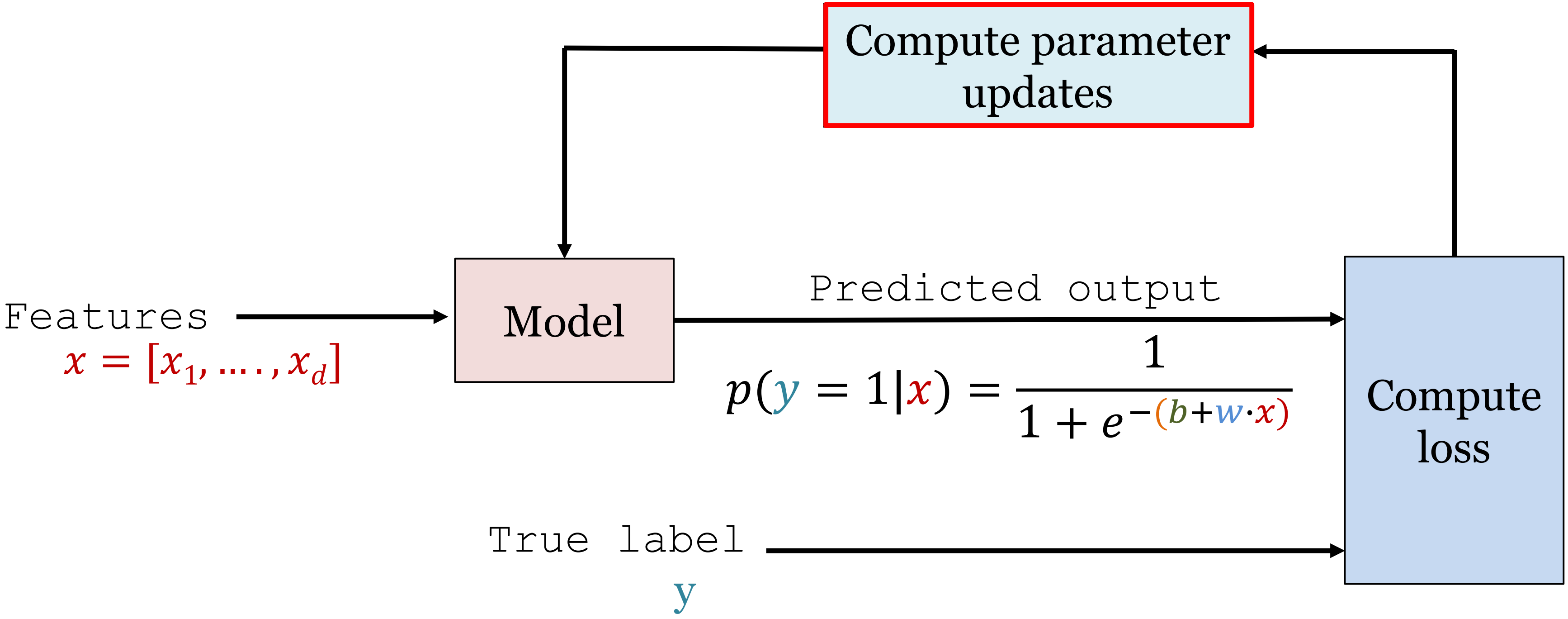


Figure based on https://developers.google.com/machine-learning/crash-course/reducing-loss/an-iterative-approach

RECAP!

Gradient descent example

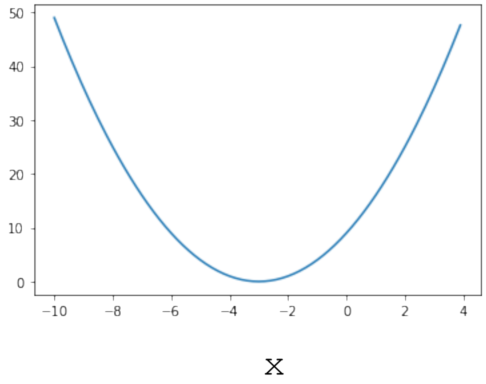
$$w^{t+1} = w^t - \eta \frac{d}{dx} f(x)$$

next step current step learning rate slope

Let's start at $x_0 = 4$,
learning rate = 0.25

$x_1 = 4 - 0.25 * (2 * (4 + 3)) = 0.5$

```
y = (x + 3)²
dy = 2 * (x + 3)
```




```
4
0.5
-1.25
-2.125
-2.5625
-2.78125
-2.890625
-2.9453125
-2.97265625
-2.986328125
-2.9931640625
```

Converges to -3!

Gradient descent example

$$w^{t+1} = w^{t} - \eta \nabla f(x)$$

next step current step learning rate gradient

```
y = (x1 + 3)² + (x2 + 5)²
dy/dx1 = 2 * (x1 + 3)
dy/dx2 = 2 * (x2 + 5)
```

Let's start at $x1_0 = 4$, $x2_0 = -2$
learning rate = 0.25

at each step: update both x1 and x2

```
t=0: [4 -2]
t=1: [0.5 -3.5]
t=2: [-1.25 -4.25]
t=3: [-2.125 -4.625]
...

[-2.99 -4.99]
```

a worked out example for LR is in the book (5.4.2)

Converges to -3 and -5!

Gradient descent example

$$w^{t+1} = w^{t} - \eta \nabla f(x)$$

next step current step learning rate gradient

```
y = (x1 + 3)^2 + (x2 + 5)^2
dy/dx1 = 2 * (x1 + 3)
dy/dx2 = 2 * (x2 + 5)
```

Let's start at $x1_0 = 4$, $x2_0 = -2$
learning rate = 0.25

at each step: update both x1 and x2

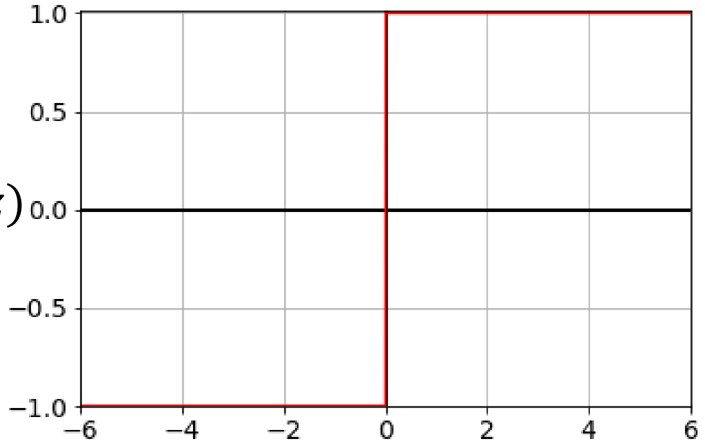
```
[ 4.  -2.]
[ 0.5 -3.5]
[-1.25 -4.25]
[-2.125 -4.625]
[-2.5625 -4.8125]
[-2.78125 -4.90625]
[-2.890625 -4.953125]
[-2.9453125 -4.9765625]
[-2.97265625 -4.98828125]
[-2.98632812 -4.99414062]
[-2.99316406 -4.99707031]
```

```python
import autograd.numpy as np
from autograd import grad

def f(weights):
    return (weights[0] + 3)**2 + (weights[1] + 5)**2

gradient_fun = grad(f)

weights = np.array([4., -2.])
print(weights)

for i in range(10):
    weights -= 0.25 * gradient_fun(weights)
    print(weights)
```

Choosing activation functions

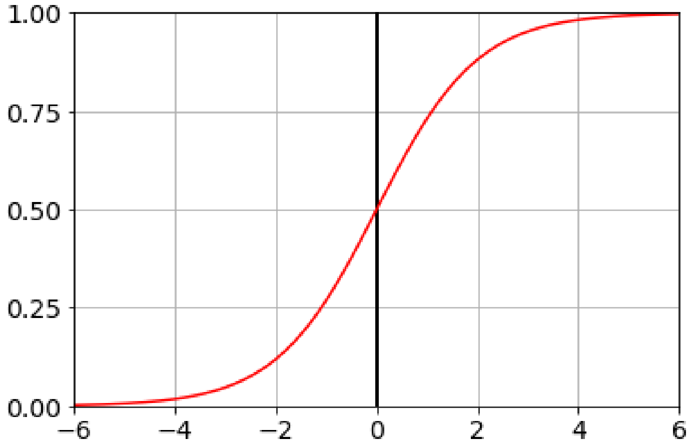
sign function

$sign(z)$



sigmoid function

$$\sigma(z) = \frac{1}{1+e^{-z}}$$



To be able to do gradient descent, we need to take the derivative!

Some other questions

On k-NN and using an odd K to prevent ties, does this only hold for binary classifation problems? → Yes!

On Levenhstein distance, isn't the cost for substitutions 2? → There's a variant with cost 1 and one with cost 2. See also the Speech and Language Processing book (chapter 2, page 22)

Is it possible to use Logistic Regression for something other than classification?