Announcement

Homework: HW6 due on 11/2 (after Midterm)

Midterm Evaluation: Received in email; complete by Saturday

Class roadmap

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Slides</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuesday, Oct 12</td>
<td>Machine Learning: Linear Regression</td>
<td></td>
<td>HW 4 Due, HW 5 Released</td>
</tr>
<tr>
<td>Thursday, Oct 14</td>
<td>Machine Learning: K-Nearest Neighbors &amp; Naive Bayes</td>
<td>Slides</td>
<td></td>
</tr>
<tr>
<td>Tuesday, Oct 19</td>
<td>Machine Learning: Neural Network I (Perceptron)</td>
<td></td>
<td>HW 5 Due, HW 6 Released</td>
</tr>
<tr>
<td>Thursday, Oct 21</td>
<td>Machine Learning: Neural Network II</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuesday, Oct 26</td>
<td>Machine Learning: Neural Network III</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MIDTERM EXAM October 28

Everything below here is tentative and subject to change.

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuesday, Nov 2</td>
<td>Machine Learning: Deep Learning I</td>
</tr>
</tbody>
</table>
Today’s outline

• Naive Bayes (cont.)
• Single-layer Neural Network (Perceptron)
Part I: Naïve Bayes (cont.)
Example 1: Play outside or not?

- If weather is sunny, would you likely to play outside?

**Posterior probability** \( p(Yes \mid ☀️) \) vs. \( p(No \mid ☀️) \)
Example 1: Play outside or not?

- If weather is sunny, would you likely to play outside?

**Posterior probability**  \( p(\text{Yes} \mid \text{Sunny} ) \) vs.  \( p(\text{No} \mid \text{Sunny} ) \)

- Weather = \{Sunny, Rainy, Overcast\}
- Play = \{Yes, No\}
- Observed data \{Weather, play on day \( m \)}, \( m=\{1,2,\ldots,N\} \)
Example 1: Play outside or not?

- If weather is sunny, would you likely to play outside?

**Posterior probability** $p(\text{Yes} \mid \text{Sunny})$ vs. $p(\text{No} \mid \text{Sunny})$

- Weather = {Sunny, Rainy, Overcast}
- Play = {Yes, No}
- Observed data {Weather, play on day $m$}, $m=\{1,2,\ldots,N\}$

$$p(\text{Play} \mid \text{Sunny}) = \frac{p(\text{Sunny} \mid \text{Play}) \cdot p(\text{Play})}{p(\text{Sunny})}$$

Bayes rule
Example 1: Play outside or not?

• **Step 1:** Convert the data to a frequency table of Weather and Play
Example 1: Play outside or not?

Step 1: Convert the data to a frequency table of Weather and Play

Step 2: Based on the frequency table, calculate **likelihoods** and **priors**

\[
p(\text{Play} = \text{Yes}) = 0.64 \\
p(\text{Sunny} | \text{Yes}) = 3/9 = 0.33
\]

https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/
Example 1: Play outside or not?

Step 3: Based on the likelihoods and priors, calculate posteriors

\[ P(\text{Yes} | ☀) = P(☀ | \text{Yes}) \times P(\text{Yes}) / P(☀) \]?

\[ P(\text{No} | ☀) = P(☀ | \text{No}) \times P(\text{No}) / P(☀) \]?
Example 1: Play outside or not?

Step 3: Based on the likelihoods and priors, calculate posteriors

\[
P(\text{Yes}|\odot) = P(\odot|\text{Yes}) \times P(\text{Yes}) / P(\odot)
\]
\[
= 0.33 \times 0.64 / 0.36
\]
\[
= 0.6
\]

\[
P(\text{No}|\odot) = P(\odot|\text{No}) \times P(\text{No}) / P(\odot)
\]
\[
= 0.4 \times 0.36 / 0.36
\]
\[
= 0.4
\]

\[P(\text{Yes}|\odot) > P(\text{No}|\odot)\] go outside and play!
Bayesian classification

\[ \hat{y} = \arg \max_y p(y | x) \]  

(Prediction)

\[ = \arg \max_y \frac{p(x | y) \cdot p(y)}{p(x)} \]  

(by Bayes’ rule)

\[ = \arg \max_y p(x | y)p(y) \]  

(Posterior)
Bayesian classification

What if $\mathbf{x}$ has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

\[
\hat{y} = \arg \max_y p(y | X_1, \ldots, X_k) \quad \text{(Posterior)}
\]

(Prediction)

Likelihood is hard to calculate for many attributes.
Bayesian classification

What if \( \mathbf{x} \) has multiple attributes \( \mathbf{x} = \{X_1, \ldots, X_k\} \)

\[
\hat{y} = \arg \max_y p(y \mid X_1, \ldots, X_k) \quad \text{(Posterior)}
\]

(Prediction)

\[
= \arg \max_y \frac{p(X_1, \ldots, X_k \mid y) \cdot p(y)}{p(X_1, \ldots, X_k)} \quad \text{(by Bayes’ rule)}
\]

Independent of \( y \)
Bayesian classification

What if $\mathbf{x}$ has multiple attributes $\mathbf{x} = \{X_1, \ldots, X_k\}$

$$\hat{y} = \arg \max_y p(y | X_1, \ldots, X_k) \quad \text{(Posterior)}$$

(Prediction)

$$= \arg \max_y \frac{p(X_1, \ldots, X_k | y) \cdot p(y)}{p(X_1, \ldots, X_k)} \quad \text{(by Bayes’ rule)}$$

$$= \arg \max_y p(X_1, \ldots, X_k | y) \cdot p(y)$$

Class conditional likelihood

Class prior
Naïve Bayes Assumption

Conditional independence of feature attributes

\[ p(X_1, \ldots, X_k \mid y)p(y) = \prod_{i=1}^{k} p(X_i \mid y)p(y) \]

Easier to estimate (using MLE!)
 Quiz break

Q1-1: Which of the following about Naive Bayes is incorrect?

• A Attributes can be nominal or numeric
• B Attributes are equally important
• C Attributes are statistically dependent of one another given the class value
• D Attributes are statistically independent of one another given the class value
• E All of above
Q1-1: Which of the following about Naive Bayes is incorrect?

- A Attributes can be nominal or numeric
- B Attributes are equally important
- C Attributes are statistically dependent of one another given the class value
- D Attributes are statistically independent of one another given the class value
- E All of above
Q1-2: Consider a classification problem with two binary features, \( x_1, x_2 \in \{0,1\} \). Suppose \( P(Y = y) = 1/32, \ P(x_1 = 1| Y = y) = y/46, \ P(x_2 = 1 | Y = y) = y/62 \). Which class will naive Bayes classifier produce on a test item with \( x_1 = 1 \) and \( x_2 = 0 \)?

- A 16
- B 26
- C 31
- D 32
Q1-2: Consider a classification problem with two binary features, \( x_1, x_2 \in \{0,1\} \). Suppose \( P(Y = y) = 1/32 \), \( P(x_1 = 1 \mid Y = y) = y/46 \), \( P(x_2 = 1 \mid Y = y) = y/62 \). Which class will naive Bayes classifier produce on a test item with \( x_1 = 1 \) and \( x_2 = 0 \)?

- A 16
- B 26
- C 31
- D 32
Q1-3: Consider the following dataset showing the result whether a person has passed or failed the exam based on various factors. Suppose the factors are independent to each other. We want to classify a new instance with Confident=Yes, Studied=Yes, and Sick=No.

<table>
<thead>
<tr>
<th>Confident</th>
<th>Studied</th>
<th>Sick</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Fail</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Pass</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Fail</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Pass</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Pass</td>
</tr>
</tbody>
</table>

- A Pass
- B Fail
Q1-3: Consider the following dataset showing the result whether a person has passed or failed the exam based on various factors. Suppose the factors are independent to each other. We want to classify a new instance with Confident=Yes, Studied=Yes, and Sick=No.

<table>
<thead>
<tr>
<th>Confident</th>
<th>Studied</th>
<th>Sick</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Fail</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Pass</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Fail</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Pass</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Pass</td>
</tr>
</tbody>
</table>

- A Pass
- B Fail
Part I: Single-layer Neural Network
How to classify
Cats vs. dogs?
Inspiration from neuroscience

- Inspirations from human brains
- Networks of **simple and homogenous** units

(wikipedia)
Cats vs. dogs?

Perceptron

\[ \text{Input} \]

\[ x_1 \]

\[ x_2 \]

\[ x_d \]

\[ w_1 \]

\[ w_2 \]

\[ w_d \]

\[ \text{Output} \]
Linear Perceptron

- Given input $x$, weight $w$ and bias $b$, perceptron outputs:

$$f = \langle w, x \rangle + b$$

Cats vs. dogs?
Perceptron

• Given input $x$, weight $w$ and bias $b$, perceptron outputs:

$$o = \sigma (\langle w, x \rangle + b)$$

$$\sigma(x) = \begin{cases} 
1 & \text{if } x > 0 \\
0 & \text{otherwise}
\end{cases}$$

Cats vs. dogs?
Perceptron

- Goal: learn parameters $\mathbf{w} = \{w_1, w_2, \ldots, w_d\}$ and $b$ to minimize the classification error.
Training the Perceptron

**Perceptron Algorithm**

Initialize $\vec{w} = \vec{0}$

while TRUE do

\[ m = 0 \]

for $(x_i, y_i) \in D$ do

if $y_i(\vec{w}^T \cdot \vec{x}_i) \leq 0$ then

$\vec{w} \leftarrow \vec{w} + y\vec{x}_i$

$m \leftarrow m + 1$

end if

end for

if $m = 0$ then

break

end if

end while

// Initialize $\vec{w}$. $\vec{w} = \vec{0}$ misclassifies everything.
// Keep looping
// Count the number of misclassifications, $m$
// Loop over each (data, label) pair in the dataset, $D$
// If the pair $(\vec{x}_i, y_i)$ is misclassified
// Update the weight vector $\vec{w}$
// Counter the number of misclassification

// If the most recent $\vec{w}$ gave 0 misclassifications
// Break out of the while-loop

// Otherwise, keep looping!
Perceptron

From wikipedia
Perceptron

From wikipedia
Perceptron
Example 2: Predict whether a user likes a song or not.
Example 2: Predict whether a user likes a song or not using Perceptron.
Learning AND function using perceptron

The perceptron can learn an AND function

\[ x_1 = 1, x_2 = 1, y = 1 \]
\[ x_1 = 1, x_2 = 0, y = 0 \]
\[ x_1 = 0, x_2 = 1, y = 0 \]
\[ x_1 = 0, x_2 = 0, y = 0 \]
Learning AND function using perceptron

The perceptron can learn an AND function

![Diagram of an AND function learned by a perceptron](image)
The perceptron can learn an AND function.

Output \( \sigma(x_1w_1 + x_2w_2 + b) \) where

\[
\sigma(x) = \begin{cases} 
1 & \text{if } x > 0 \\
0 & \text{otherwise}
\end{cases}
\]

What's \( w \) and \( b \)?
Learning AND function using perceptron

The perceptron can learn an AND function

\[
\sigma(x_1w_1 + x_2w_2 + b) = \begin{cases} 
1 & \text{if } x > 0 \\
0 & \text{otherwise}
\end{cases}
\]

\[
w_1 = 1, w_2 = 1, b = -1.5
\]
The perceptron can learn an OR function

\[ x_1 = 1, x_2 = 1, y = 1 \]
\[ x_1 = 1, x_2 = 0, y = 1 \]
\[ x_1 = 0, x_2 = 1, y = 1 \]
\[ x_1 = 0, x_2 = 0, y = 0 \]
Learning OR function using perceptron

The perceptron can learn an OR function
Learning OR function using perceptron

The perceptron can learn an OR function

\[
\sigma(x) = \begin{cases} 
1 & \text{if } x > 0 \\
0 & \text{otherwise}
\end{cases}
\]

What’s w and b?
Learning OR function using perceptron

The perceptron can learn an OR function

$$\text{Output } \sigma(x_1 w_1 + x_2 w_2 + b)$$

$$\sigma(x) = \begin{cases} 
1 & \text{if } x > 0 \\
0 & \text{otherwise}
\end{cases}$$

$$w_1 = 1, w_2 = 1, b = -0.5$$
Learning NOT function using perceptron

The perceptron can learn NOT function (single input)

\[ \sigma(x) = \begin{cases} 
1 & \text{if } x > 0 \\
0 & \text{otherwise}
\end{cases} \]

\( w_1 = -1, b = 0.5 \)
XOR Problem (Minsky & Papert, 1969)

The perceptron cannot learn an XOR function (neurons can only generate linear separators)

This contributed to the first AI winter
Quiz Break

Consider the linear perceptron with $x$ as the input. Which function can the linear perceptron compute?
(1) $y = ax + b$
(2) $y = ax^2 + bx + c$

A. (1)  
B. (2)  
C. (1)(2)  
D. None of the above
Consider the linear perceptron with $x$ as the input. Which function can the linear perceptron compute?

1. $y = ax + b$
2. $y = ax^2 + bx + c$

A. (1)
B. (2)
C. (1)(2)
D. None of the above

**Answer:** A. All units in a linear perceptron are linear. Thus, the model can not present non-linear functions.
Quiz Break

Perceptron can be used for representing:

A. AND function
B. OR function
C. XOR function
D. Both AND and OR function
Quiz Break

Perceptron can be used for representing:

A. AND function
B. OR function
C. XOR function
D. Both AND and OR function
Step Function activation

Step function is discontinuous, which cannot be used for gradient descent

\[
\sigma(x) = \begin{cases} 
1 & \text{if } x > 0 \\
0 & \text{otherwise}
\end{cases}
\]
Sigmoid/Logistic Activation

Map input into $[0, 1]$, a **soft** version of $\sigma(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$

$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)}$
Logistic regression

\( \mathbf{x} \in \mathbb{R}^d, y = \{-1, +1\} \)

\[ p(y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})} \]

\[ p(y = -1 \mid \mathbf{x}) = 1 - \sigma(\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + \exp(\mathbf{w}^T \mathbf{x})} \]
Logistic regression

Given: \{ (x_i, y_i) \}_{i=1}^n

Training: maximize likelihood estimate (on the conditional probability)

$$\max_w \sum_i \log \frac{1}{1 + \exp(-y_i w^T x_i)}$$
Logistic regression

Given: \( \{(x_i, y_i)\}_{i=1}^{n} \)

Training: maximize likelihood estimate (on the conditional probability)

When training data is linearly separable, many solutions
Logistic regression

Given: \( \{(x_i, y_i)\}_{i=1}^{n} \)

Training: maximum a posteriori (MAP)

\[
\min_{w} \sum_{i} - \log \frac{1}{1 + \exp(-y_i w^T x_i)} + \frac{\lambda}{2} \|w\|_2^2
\]

- Convex optimization
- Solve via (stochastic) gradient descent
**Tanh Activation**

Map inputs into (-1, 1)

\[
tanh(x) = \frac{1 - \exp(-2x)}{1 + \exp(-2x)}
\]
ReLU Activation

ReLU: rectified linear unit (commonly used in modern neural networks)

\[
ReLU(x) = \max(x, 0)
\]
Quiz Break

Which one of the following is a valid activation function?

a) Step function
b) Sigmoid function
C) ReLU function
D) all of above
Quiz Break

Which one of the following is a valid activation function?

a) Step function
b) Sigmoid function
C) ReLU function
D) all of above
Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. Which of the following functions is NOT an element-wise operation that can be used as an activation function?

A $f(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

B $f(x) = \begin{bmatrix} \max(0, x_1) \\ \max(0, x_2) \end{bmatrix}$

C $f(x) = \begin{bmatrix} \exp(x_1) \\ \exp(x_2) \end{bmatrix}$

D $f(x) = \begin{bmatrix} \exp(x_1 + x_2) \\ \exp(x_2) \end{bmatrix}$
Let $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. Which of the following functions is NOT an element-wise operation that can be used as an activation function?

A $f(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

B $f(x) = \begin{bmatrix} \max(0, x_1) \\ \max(0, x_2) \end{bmatrix}$

C $f(x) = \begin{bmatrix} \exp(x_1) \\ \exp(x_2) \end{bmatrix}$

D $f(x) = \begin{bmatrix} \exp(x_1 + x_2) \\ \exp(x_2) \end{bmatrix}$
Multilayer Perceptron
Single Hidden Layer

How to classify
Cats vs. dogs?
How to classify
Cats vs. dogs?

Single Hidden Layer
Single Hidden Layer

- Input $\mathbf{x} \in \mathbb{R}^d$
- Hidden $\mathbf{W} \in \mathbb{R}^{m \times d}, \mathbf{b} \in \mathbb{R}^m$
- Intermediate output

$$h = \sigma(\mathbf{Wx} + \mathbf{b})$$

$\sigma$ is an element-wise activation function
Single Hidden Layer

- Output $f = w_2^T h + b_2$
Single Hidden Layer

Why do we need an a nonlinear activation?
Why do we need an a nonlinear activation?

\[ h = Wx + b \]
\[ f = w_2^T h + b_2 \]

hence \[ f = w_2^T Wx + b' \]
Why multiple layers?
Brief history of neural networks

- 1943: S. McCulloch – W. Pitts
- 1957: F. Rosenblatt
- 1960: B. Widrow – M. Hoff
- 1969: M. Minsky – S. Papert
- 1970: V. Vapnik – C. Cortes

- **Golden Age**
- **Dark Age ("AI Winter")**

- **ADALINE**
- **XOR Problem**
- **Multi-layered Perceptron (Backpropagation)**
- **SVM**
- **Deep Neural Network (Pretraining)**

- **Electronic Brain**
- **Perceptron**
What we’ve learned today…

• Single-layer Perceptron
  • Motivation
  • Activation function
  • Representing AND, OR, NOT
• Brief history of neural networks
Thanks!

Based on slides from Xiaojin (Jerry) Zhu and Yingyu Liang (http://pages.cs.wisc.edu/~jerryzhu/cs540.html), and Alex Smola: https://courses.d2l.ai/berkeley-stat-157/units/mlp.html