Machine Learning: Chenhao Tan

University of Colorado Boulder

LECTURE 6

Slides adapted from Jordan Boyd-Graber, Chris Ketelsen

- HW1 turned in
- HW2 released
- Office hour
- Group formation signup

Overview

Feature engineering

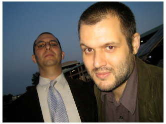
Revisiting Logistic Regression

Feed Forward Networks

Layers for Structured Data

Outline

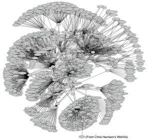
Feature Engineering



$\longrightarrow \langle 1.5, 3.2, -5.1, \ldots, 4.2 \rangle$

Republican nominee George Bush said he felt nervous as he voted today in his adopted home state of Texas, where he ended...

$\longrightarrow \langle 1, 0, 0, 0, 5, 0, 9, 3, 1, \ldots, 0 \rangle$



$$\longrightarrow \begin{bmatrix} 1 & 0 & 1 & \ldots & 0 \\ 0 & 1 & 1 & \ldots & 0 \\ 1 & 0 & 0 & \ldots & 1 \\ & \ldots & & & \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix}$$

Brainstorming

What are features useful for sentiment analysis?

Top critical review

See all 2,023 critical reviews ›

15,029 people found this helpful

★★☆☆☆ **Angle is wrong**

By Jim Anderson on August 1, 2012

I tried the banana slicer and found it unacceptable. As shown in the picture, the slicer is curved from left to right. All of my bananas are bent the other way.

What are features useful for sentiment analysis?

- Unigram
- Bigram
- Normalizing options
- Part-of-speech tagging
- Parse-tree related features
- Negation related features
- Additional resources

Sarcasm detection

"Trees died for this book?" (book)

Sarcasm detection

"Trees died for this book?" (book)

- find high-frequency words and content words
- replace content words with "CW"
- extract patterns, e.g., "does not CW much about CW"

[Tsur et al., 2010]

More examples: Which one will be retweeted more?

[Tan et al., 2014]

https://chenhaot.com/papers/wording-for-propagation.html

Outline

Feature engineering

Revisiting Logistic Regression

Feed Forward Networks

Layers for Structured Data

Revisiting Logistic Regression

$$P(\mathrm{Y}=0 \mid \boldsymbol{x}, \beta) = \frac{1}{1+\exp\left[\beta_0 + \sum_i \beta_i X_i\right]}$$

$$P(\mathrm{Y}=1 \mid \boldsymbol{x}, \beta) = \frac{\exp\left[\beta_0 + \sum_i \beta_i X_i\right]}{1+\exp\left[\beta_0 + \sum_i \beta_i X_i\right]}$$

$$\mathscr{L} = -\sum_j \log P(y^{(j)} \mid X^{(j)}, \beta)$$

Revisiting Logistic Regression

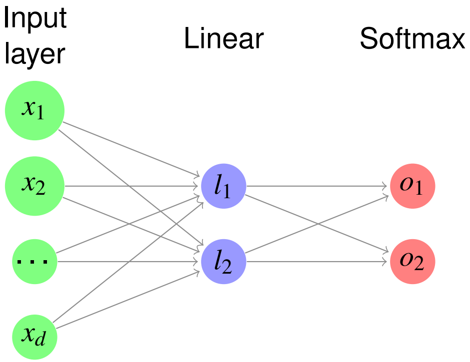
- Transformation on $\boldsymbol{x}$ (we map class labels from $\{0, 1\}$ to $\{1, 2\}$):

$$l_i = \beta_i^T \boldsymbol{x}, i = 1, 2$$
$$o_i = \frac{\exp l_i}{\sum_{c \in \{1,2\}} \exp l_c}, i = 1, 2$$

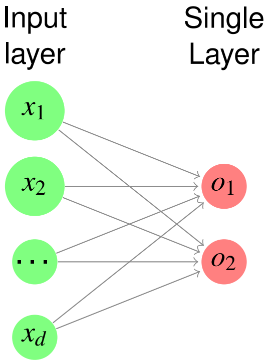
- Objective function (using cross entropy $-\sum_i p_i \log q_i$):

$$\mathscr{L}(Y, \hat{Y}) = -\sum_j P(y^{(j)} = 1) \log P(\hat{y}_i = 1 \mid x^{(j)}, \beta) + P(y^{(j)} = 0) \log \hat{P}(y_i = 0 \mid X_i)$$

Logistic Regression as a Single-layer Neural Network



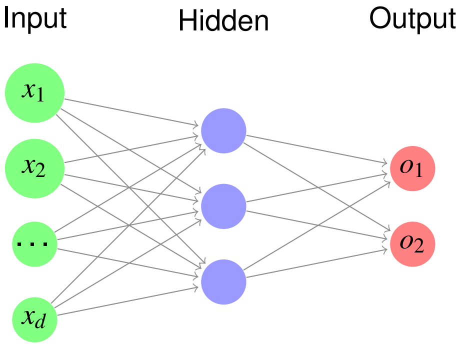
Logistic Regression as a Single-layer Neural Network



Outline

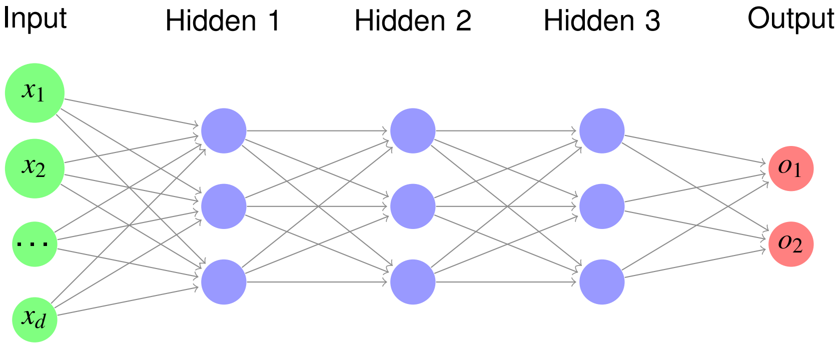
Deep Neural networks

A two-layer example (one hidden layer)



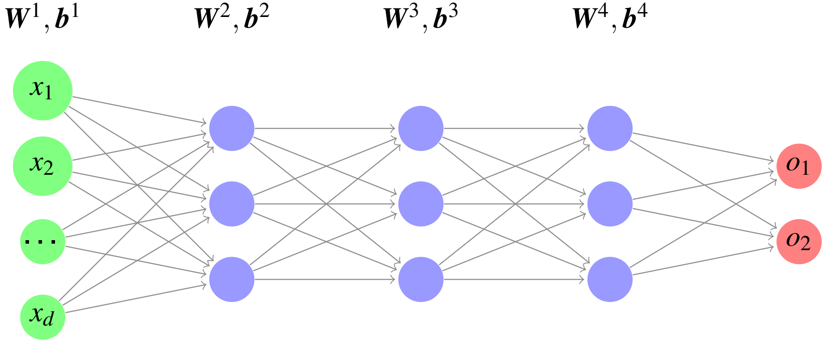
Deep Neural networks

More layers:



Forward propagation algorithm

How do we make predictions based on a multi-layer neural network?
Store the biases for layer l in $\boldsymbol{b}^l$, weight matrix in $\boldsymbol{W}^l$



Forward propagation algorithm

Suppose your network has L layers
Make a prediction based on text point $\boldsymbol{x}$

Initialize $\boldsymbol{a}^0 = \boldsymbol{x}$
for $l = 1$ to L **do**
$\quad \boldsymbol{z}^l = \boldsymbol{W}^l \boldsymbol{a}^{l-1} + \boldsymbol{b}^l$
$\quad \boldsymbol{a}^l = g(\boldsymbol{z}^l)$
end for
The prediction $\hat{y}$ is simply $\boldsymbol{a}^L$

Nonlinearity

What happens if there is no nonlinearity?

Nonlinearity

What happens if there is no nonlinearity?
Linear combinations of linear combinations are still linear combinations.

Neural networks in a nutshell

- Training data $S_{\text{train}} = \{(\boldsymbol{x}, y)\}$
- Network architecture (model)

$$\hat{y} = f_w(\boldsymbol{x})$$

- Loss function (objective function)

$$\mathscr{L}(y, \hat{y})$$

- Learning (next lecture)

Nonlinearity Options

- Sigmoid

$$f(x) = \frac{1}{1 + \exp(x)}$$

- tanh

$$f(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

- ReLU (rectified linear unit)

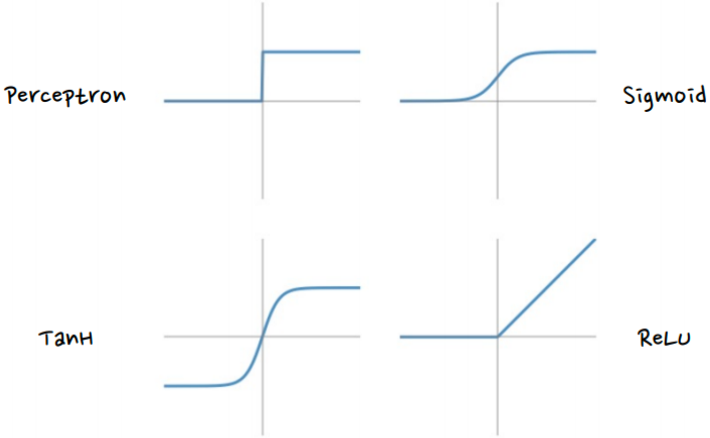
$$f(x) = \max(0, x)$$

- softmax

$$\boldsymbol{x} = \frac{\exp(\boldsymbol{x})}{\sum_{x_i} \exp(x_i)}$$

`https://keras.io/activations/`

Nonlinearity Options



Loss Function Options

- ℓ_2 loss

$$\sum_i (y_i - \hat{y}_i)^2$$

- ℓ_1 loss

$$\sum_i |y_i - \hat{y}_i|$$

- Cross entropy

$$-\sum_i y_i \log \hat{y}_i$$

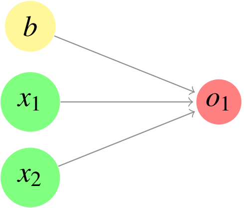
- Hinge loss (more on this during SVM)

$$\max(0, 1 - y\hat{y})$$

`https://keras.io/losses/`

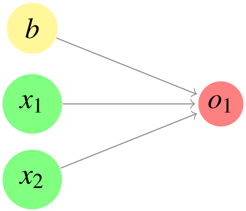
A Perceptron Example

$$\boldsymbol{x} = (x_1, x_2), y = f(x_1, x_2)$$



A Perceptron Example

$\boldsymbol{x} = (x_1, x_2), y = f(x_1, x_2)$



We consider a simple activation function

$$f(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

A Perceptron Example

Simple Example: Can we learn `OR`?

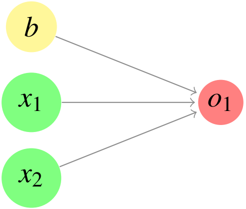
x_1	0	1	0	1
x_2	0	0	1	1
$y = x_1 \vee x_2$	0	1	1	1

A Perceptron Example

Simple Example: Can we learn OR?

x_1	0	1	0	1
x_2	0	0	1	1
$y = x_1 \vee x_2$	0	1	1	1

$$\boldsymbol{w} = (1, 1), b = -0.5$$



A Perceptron Example

Simple Example: Can we learn `AND`?

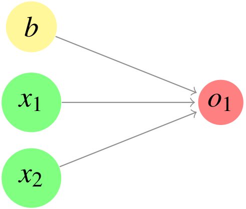
x_1	0	1	0	1
x_2	0	0	1	1
$y = x_1 \wedge x_2$	0	0	0	1

A Perceptron Example

Simple Example: Can we learn AND?

x_1	0	1	0	1
x_2	0	0	1	1
$y = x_1 \wedge x_2$	0	0	0	1

$$\boldsymbol{w} = (1, 1), b = -1.5$$



A Perceptron Example

Simple Example: Can we learn `NAND`?

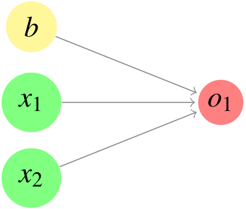
x_1	0	1	0	1
x_2	0	0	1	1
$y = \neg(x_1 \land x_2)$	1	0	0	0

A Perceptron Example

Simple Example: Can we learn `NAND`?

x_1	0	1	0	1
x_2	0	0	1	1
$y = \neg(x_1 \wedge x_2)$	1	0	0	0

$$\boldsymbol{w} = (-1, -1), b = 0.5$$



A Perceptron Example

Simple Example: Can we learn XOR?

x_1	0	1	0	1
x_2	0	0	1	1
x_1 XOR x_2	0	1	1	0

A Perceptron Example

Simple Example: Can we learn XOR?

x_1	0	1	0	1
x_2	0	0	1	1
x_1 XOR x_2	0	1	1	0

NOPE!

A Perceptron Example

Simple Example: Can we learn XOR?

x_1	0	1	0	1
x_2	0	0	1	1
x_1 XOR x_2	0	1	1	0

NOPE!
But why?

A Perceptron Example

Simple Example: Can we learn XOR?

x_1	0	1	0	1
x_2	0	0	1	1
x_1 XOR x_2	0	1	1	0

NOPE!
But why?
The single-layer perceptron is just a linear classifier, and can only learn things that are linearly separable.

A Perceptron Example

Simple Example: Can we learn XOR?

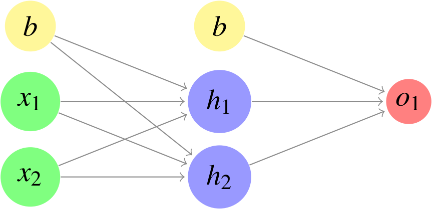
x_1	0	1	0	1
x_2	0	0	1	1
x_1 XOR x_2	0	1	1	0

NOPE!
But why?
The single-layer perceptron is just a linear classifier, and can only learn things that are linearly separable.
How can we fix this?

A Perceptron Example

Increase the number of layers.

x_1	0	1	0	1
x_2	0	0	1	1
x_1 XOR x_2	0	1	1	0



$$\boldsymbol{W}^1 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \boldsymbol{b}^1 = \begin{bmatrix} -0.5 \\ 1.5 \end{bmatrix}$$

$$\boldsymbol{W}^2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \boldsymbol{b}^2 = -1.5$$

General Expressiveness of Neural Networks

Neural networks with a single hidden layer can approximate any measurable functions [Hornik et al., 1989, Cybenko, 1989].

Outline

Feature engineering

Revisiting Logistic Regression

Feed Forward Networks

Layers for Structured Data

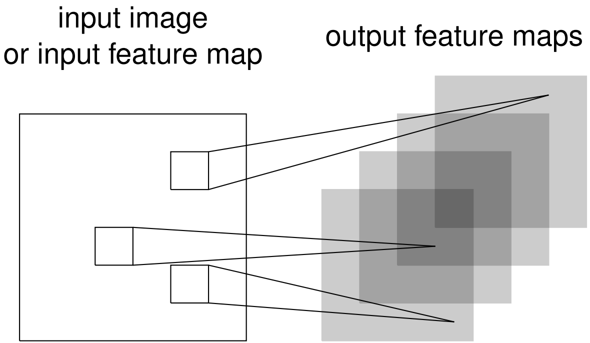
Structured data

Spatial information

https://www.reddit.com/r/aww/comments/6ip2la/before_and_after_she_was_told_she_was_a_good_girl/

Convolutional Layers

Sharing parameters across patches



https://github.com/davidstutz/latex-resources/blob/master/tikz-convolutional-layer/convolutional-layer.tex

Structured data

Sequential information
"My words fly up, my thoughts remain below: Words without thoughts never to heaven go."

—Hamlet

Structured data

Sequential information
"My words fly up, my thoughts remain below: Words without thoughts never to heaven go."

—Hamlet

- language
- activity history

Structured data

Sequential information
"My words fly up, my thoughts remain below: Words without thoughts never to heaven go."

—Hamlet

- language
- activity history

$$\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$$

Recurrent Layers

Sharing parameters along a sequence

$$h_t = f(x_t, h_{t-1})$$

Recurrent Layers

Sharing parameters along a sequence

$$h_t = f(x_t, h_{t-1})$$

Long short-term memory

What is missing?

- How to find good weights?
- How to make the model work (regularization, architecture, etc)?

References

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, 1989.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Chenhao Tan, Lillian Lee, and Bo Pang. The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. In *Proceedings of ACL*, 2014.

Oren Tsur, Dmitry Davidov, and Ari Rappoport. ICWSM-A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews. In *Proceedings of ICWSM*, 2010.