Natural Language Processing

Tutorial 6 (Week 9): ML & DL

Summary and Recap for Week 8

- Classification vs Regression in supervised learning
- Linear Regression
- Logistic Regression
- Neural Networks
- Chain rule for computing gradients

Recap – Linear Regression

Problem Setup

Training Data is represented by:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$$ Supervised

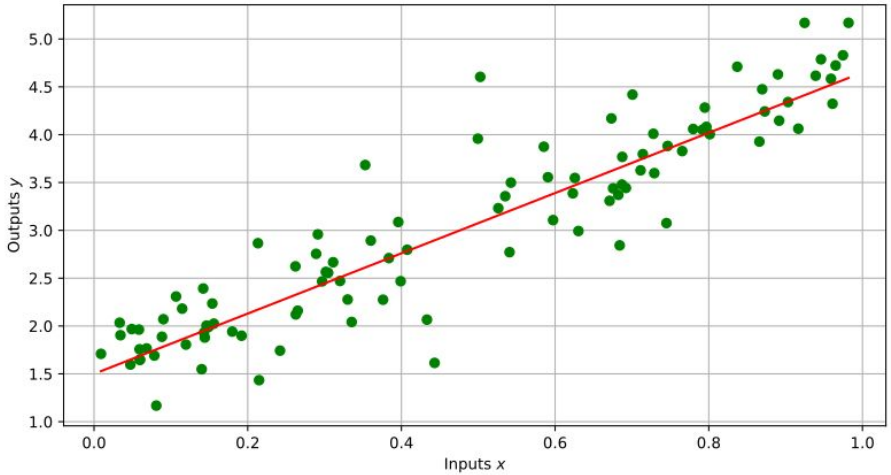
Here,
$\mathcal{D}$ is called the training set.
N is the number of training example.
$\mathbf{x}_i$ is a d-dimensional vector of features (also called attributes or covariates) i.e. #features is d.
y_i is called output or response variable.

For regression: y_i is real-valued i.e. $y_i \in \mathbb{R}$

Recap – Linear Regression

A linear regression relates y to a linear predictor function of **x**.
For a given data point i, the linear function is of the form:

$$\hat{y}_i = w_0 + w_1 x_{i1} + \ldots + w_d x_{id}$$



Recap – Linear Regression

Least Square Formulation

So our **objective function** is:

What is theta?

$$J(\theta) = \arg\min_{\theta} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

Where is the bias term?

$$= \sum_{i=0}^{N} (y_i - \mathbf{x}_i^\intercal \mathbf{w})^2 = (y - \mathbf{X}\mathbf{w})^\intercal (y - \mathbf{X}\mathbf{w})$$

Vector notation

Recap – Logistic Regression

Problem Setup

Training Data is represented by

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$$ Supervised

Here

$\mathcal{D}$ is called the training set.

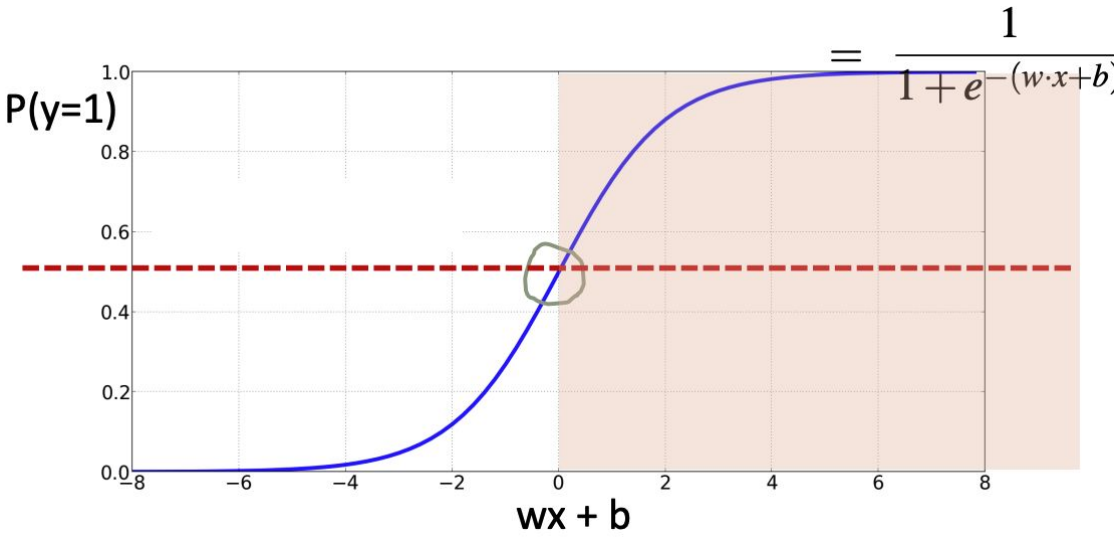
N is the number of training examples.

x_i is a d-dimensional vector of features (also called attributes or covariates) i.e. #features is d.

y_i is called output or response variable.

For classification: y_i is categorial i.e. $y_i \in \{1, .. C\}$

Recap – Logistic Regression



$$= \frac{1}{1+e^{-(w \cdot x+b)}}$$

$$P(y=1|x) = \sigma(\mathbf{w}x+b)$$

$$= \frac{1}{1+e^{-(\mathbf{w}x+b)}}$$

$$P(y=0|x) = 1-\sigma(\mathbf{w}x+b)$$

$$p(y=y_i|x_i) = \mu_i^{y_i}(1-\mu_i)^{1-y_i}$$

$$\mu_i = \sigma(\mathbf{w}x_i+b)$$

$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

if $w \cdot x + b > 0$

if $w \cdot x + b \leq 0$

Recap – Logistic Regression

Maximum Likelihood Estimation

$$p(y = y_i|x_i) = \mu_i^{y_i}(1-\mu_i)^{1-y_i}$$

$$\mu_i = \sigma(\mathbf{w}x_i + b)$$

$$\begin{aligned}
\mathbf{w}^* &= \text{argmax}_{\mathbf{w}} \log p(\mathcal{D}|\mathbf{w}) \\
&= \text{argmax}_{\mathbf{w}} \log \textstyle\prod_{i=1}^{N} p(y = y_i|x_i) \\
&= \text{argmax}_{\mathbf{w}} \textstyle\sum_{i}^{N} \log p(y = y_i|x_i) \\
&= \text{argmax}_{\mathbf{w}} \textstyle\sum_{i=1}^{N} \log(\mu_i^{y_i} \cdot (1-\mu_i)^{1-y_i}) \\
&= \text{argmin}_{\mathbf{w}} - \textstyle\sum_{i=1}^{N} (y_i \log \mu_i + (1-y_i)\log(1-\mu_i))
\end{aligned}$$

Negative Log Likelihood (NLL)

This is also called the Cross Entropy Error Function.

Recap – Multiclass Logistic Regression

Logistic regression can be extended to handle more than two classes. This is called **Multi-class logistic regression**.

Also called **Multinomial logistic regression**.

In this case, the response variable $y_i \in \{1, 2, \ldots, C\}$

Different from binary logistic regression, we model the probability using **Softmax** as

$$\mathcal{P}(y_i = c|\mathbf{x}_i, \mathbf{W}) = \mu_{ic} = \frac{\exp(\mathbf{w}_c^\top \mathbf{x}_i)}{\sum_{c'=1}^{C} \exp(\mathbf{w}_{c'}^\top \mathbf{x}_i)}$$

$$\mathcal{P}(y_i|\mathbf{x}_i, \mathbf{W}) = \prod_{c=1}^{C} \mu_{ic}^{y_{ic}}$$

Recap – Neural Networks

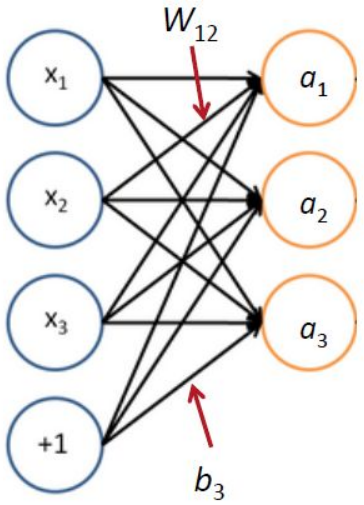
$$a_1 = f(W_{11}x_1 + W_{12}x_2 + W_{13}x_3 + b_1)$$
$$a_2 = f(W_{21}x_1 + W_{22}x_2 + W_{23}x_3 + b_2)$$
.....

In Matrix Notation $z = Wx + b$

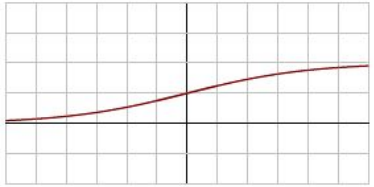
$$a = f(z)$$

Where f() is applied element-wise

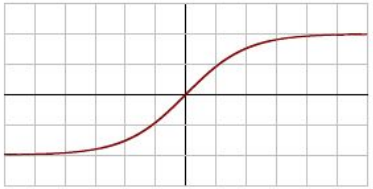
$$f([z_1, z_2, z_3]) = [f(z_1), f(z_2), f(z_3)]$$



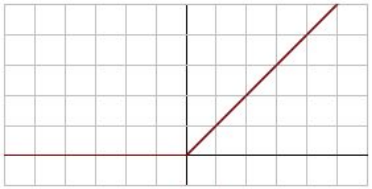
Recap – Neural Networks



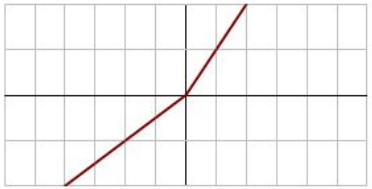
(a) Sigmoid



(b) tanh



(c) ReLU



(d) Leaky ReLU

- Sigmoid

$$\sigma(x) = \frac{1}{1 + \exp^{-x}}$$

- Hyperbolic Tangent:

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

- Rectified Linear Unit (ReLU):

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

- Leaky ReLU:

$$f(x) = \begin{cases} \alpha x & x < 0 \\ x & x \geq 0 \end{cases}$$

, where α is small constant

Recap – Train Neural Networks

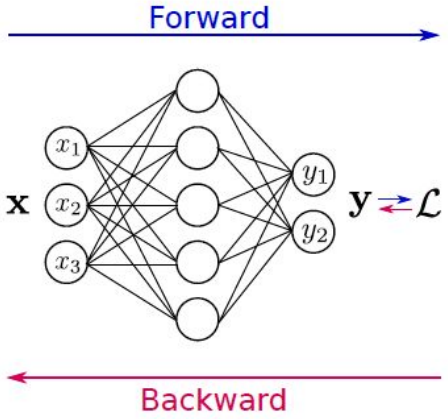
Two information flow directions:

Forward propagation

- NN accepts an input **x** and produces an output **y**
- During training, it continues onward until it produces a scalar cost **L**

Back-propagation

- Information (**gradients** with respect to the parameters) from the cost flows backward through the network

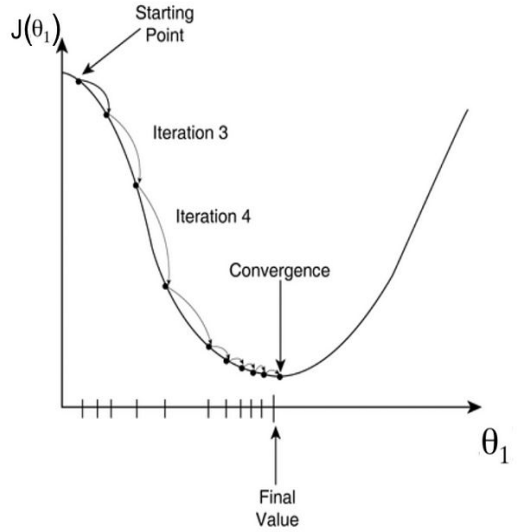


Recap – Gradient Descent

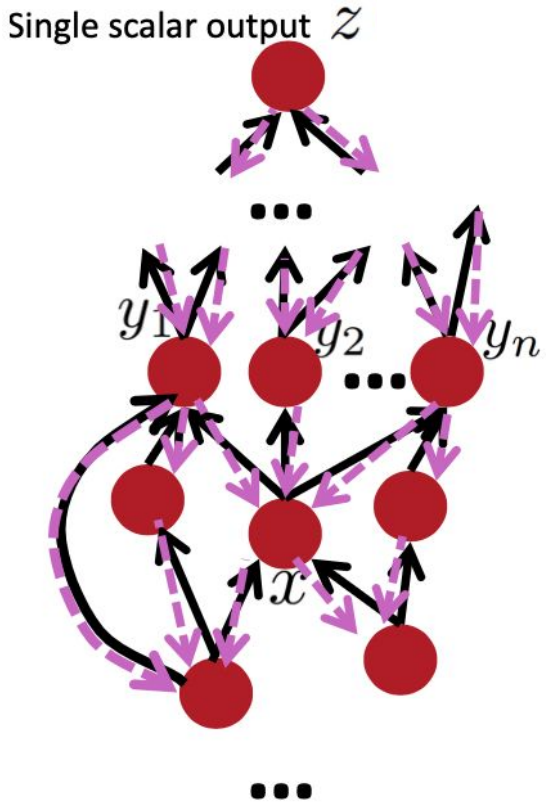
The most commonly used method for unconstrained optimization

Goal: minimize $\sum_{i=1}^{N} J_i(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \nabla_{\boldsymbol{\theta}} J_{\pi(i)}$$



Recap – Gradient Descent (Chain Rule)



1. Fprop: visit nodes in topo-sort order
   - Compute value of node given predecessors
2. Bprop:
   - initialize output gradient = 1
   - visit nodes in reverse order:
     Compute gradient wrt each node using gradient wrt successors

$\{y_1, y_2, \ldots y_n\}$ = successors of x

$$\frac{\partial z}{\partial x} = \sum_{i=1}^{n} \frac{\partial z}{\partial y_i} \frac{\partial y_i}{\partial x}$$

Source: Socher & Manning

Question 1

Imagine you're working on a basic sentiment analysis task. You have collected a small dataset where each data point consists of the number of positive words in a movie review and the corresponding rating given by the reviewer on a scale of 1 to 10. Now use a simple **linear regression model** to **predict the movie rating based on the number of positive words**. Your model should be defined as: $y = wx + b$, where x = number of positive words.

Question 1

Use the following data to derive

(1) The optimal weight w and bias b using the least squares method.

Number of Positive Words	Movie Rating
1	3
3	6
5	8
6	9

Solution 1 (1)

$$J(w,b) = \sum_i (y_i - \hat{y}_i)^2$$
$$= \sum_i (y_i - (wx_i + b))^2$$

This is convex, compute partial derivatives and equate to 0

$$\frac{\partial J}{\partial w} = -2 \sum_i x_i (y_i - (wx_i + b)) = 0$$
$$\frac{\partial J}{\partial b} = -2 \sum_i (y_i - (wx_i + b)) = 0$$

Solution 1 (1)

This is convex, compute partial derivatives and equate to 0

$$\frac{\partial J}{\partial w} = -2\sum_i x_i(y_i - (wx_i + b)) = 0$$

$$\frac{\partial J}{\partial b} = -2\sum_i (y_i - (wx_i + b)) = 0$$

Rearrange the above equations to get

$$\sum_i x_i y_i = w\sum_i x_i^2 + b\sum_i x_i$$

$$\sum_i y_i = w\sum_i x_i + nb$$

Solution 1 (1)

Rearrange the above equations to get

$$\sum_i x_i y_i = w \sum_i x_i^2 + b \sum_i x_i$$

$$\sum_i y_i = w \sum_i x_i + nb$$

Solving this system of 2 equations over 2 variables, we get

$$w = \frac{n \sum_i x_i y_i - \sum_i x_i \sum_i y_i}{n \sum_i x_i^2 - \left(\sum_i x_i\right)^2} \qquad b = \frac{\sum_i y_i - w \sum_i x_i}{n}$$

Solution 1 (1)

Solving this system of 2 equations over 2 variables, we get

$$w = \frac{n\sum_i x_i y_i - \sum_i x_i \sum_i y_i}{n\sum_i x_i^2 - \left(\sum_i x_i\right)^2} \qquad b = \frac{\sum_i y_i - w\sum_i x_i}{n}$$

Now plug in the values from the table

$n = 4$

x_i Number of Positive Words	y_i Movie Rating
1	3
3	6
5	8
6	9

$$w = \frac{n\sum_i x_i y_i - \sum_i x_i \sum_i y_i}{n\sum_i x_i^2 - (\sum_i x_i)^2}$$

$$b = \frac{\sum_i y_i - w\sum_i x_i}{n}$$

$$w = \frac{4(13+36+58+69) - (1+3+5+6)(3+6+8+9)}{4(1+9+25+36) - (1+3+5+6)^2}$$

$$\approx 1.2$$

$$b = \frac{(3+6+8+9) - 1.2(1+3+5+6)}{4} = 2$$

Now plug in the values from the table

$$n = 4$$

x_i Number of Positive Words	y_i Movie Rating
1	3
3	6
5	8
6	9

Question 1

Use the following data to derive

(2) Once you have the parameters, what would be the predicted rating for a review with 4 positive words?

Number of Positive Words	Movie Rating
1	3
3	6
5	8
6	9

Solution 1 (2)

$$x = 4, w = 1.2, b = 2, y = ?$$

Number of Positive Words	Movie Rating
1	3
3	6
5	8
6	9

Solution 1 (2)

$$x = 4, w = 1.2, b = 2, y = ?$$

$$y = wx + b = 1.2 \times 4 + 2 \approx 7$$

Number of Positive Words	Movie Rating
1	3
3	6
5	8
6	9

Question 2

You're trying to predict the sentiment label for the following document:

“It's hokey . There are virtually no surprises , and the writing is second-rate . So why was it so enjoyable ? For one thing , the cast is great . Another nice touch is the music . I was overcome with the urge to get off the couch and start dancing . It sucked me in , and it'll do the same to you .”

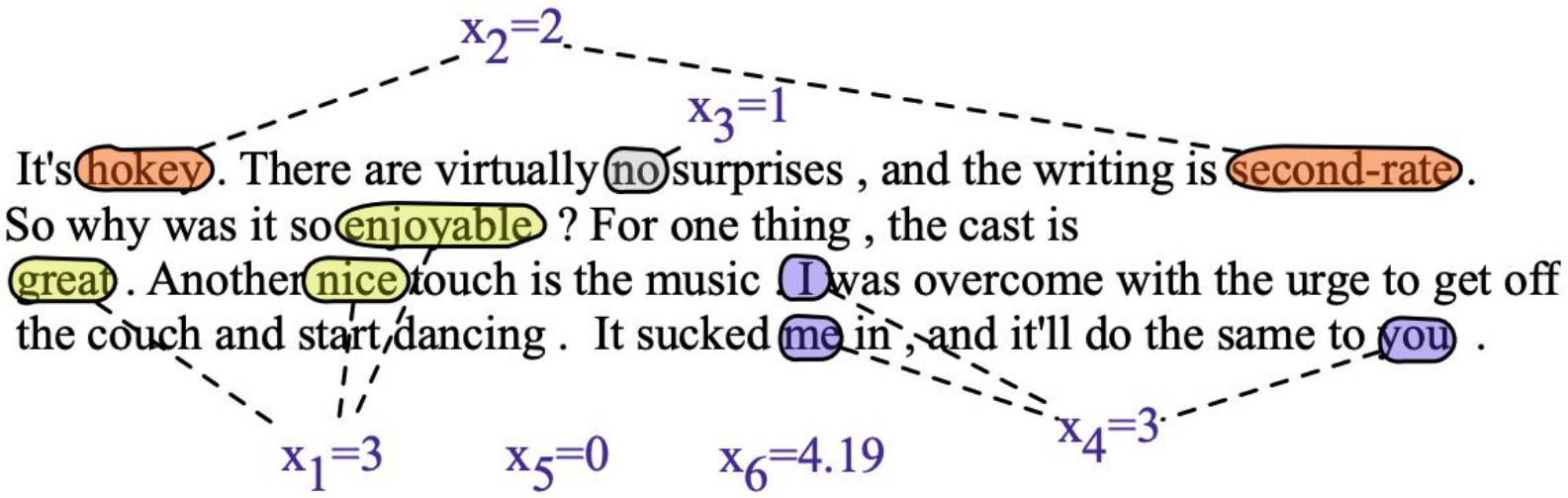
There are 6 features x1, ..., x6 capturing key factors for determining sentiment as detailed in the table below.

Question 2

Feature	Definition	Value
x_1	# positive lexicons	
x_2	# negative lexicons	
x_3	Is “no” in the document? 1 if yes, 0 otherwise	
x_4	# first and second pronouns	
x_5	Is “!” in the document? 1 if yes, 0 otherwise	
x_6	log(word count in total)	

(1) Fill in the value for each feature according to the given document.

Solution 2 (1)


$x_2=2$
$x_3=1$
It's hokey . There are virtually no surprises , and the writing is second-rate .
So why was it so enjoyable ? For one thing , the cast is
great . Another nice touch is the music . I was overcome with the urge to get off
the couch and start dancing . It sucked me in , and it'll do the same to you .
$x_1=3$
$x_5=0$
$x_6=4.19$
$x_4=3$

Solution 2 (1)

Feature	Definition	Value
x_1	# positive lexicons	3
x_2	# negative lexicons	2
x_3	Is “no” in the document? 1 if yes, 0 otherwise	1
x_4	# first and second pronouns	3
x_5	Is “!” in the document? 1 if yes, 0 otherwise	0
x_6	log(word count in total)	ln(66)=4.19

(1) Fill in the value for each feature according to the given document.

Question 2 (2, 3)

Feature	Definition	Value
x_1	# positive lexicons	3
x_2	# negative lexicons	2
x_3	Is “no” in the document? 1 if yes, 0 otherwise	1
x_4	# first and second pronouns	3
x_5	Is “!” in the document? 1 if yes, 0 otherwise	0
x_6	log(word count in total)	ln(66)=4.19

(2) Suppose w=[2.5, -5.0, -1.2, 0.5, 2.0, 0.7], b=0.1, write the logistic regression function for the above input document.

(3) What is the predicted probability for positive sentiment? What is the predicted label?

Solution 2 (2, 3) $\sigma(z) = \frac{1}{1+e^{-z}} \quad z = wx + b$

$$
\begin{aligned}
p(+|x) = P(Y=1|x) &= \sigma(w \cdot x + b) \\
&= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\
&= \sigma(.833) \\
&= 0.70
\end{aligned}
$$

$$
\begin{aligned}
p(-|x) = P(Y=0|x) &= 1 - \sigma(w \cdot x + b) \\
&= 0.30
\end{aligned}
$$

Question 2 (4)

(4) If we want to use this document to train the model and the ground-truth sentiment is positive, what is the loss computed for this single example?

Solution 2 (4)

$$y = 1 \quad L(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

$$\begin{aligned} L_{\text{CE}}(\hat{y}, y) &= -[y \log \sigma(w \cdot x + b) + (1 - y) \log (1 - \sigma(w \cdot x + b))] \\ &= -[\log \sigma(w \cdot x + b)] \\ &= -\log(.70) \\ &= .36 \end{aligned}$$

$$\begin{aligned} p(+|x) = P(Y = 1|x) &= \sigma(w \cdot x + b) \\ &= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\ &= \sigma(.833) \\ &= 0.70 \end{aligned}$$

Question 2 (5)

(5) If we change the binary prediction task to 3-class classification, involving positive, negative and neutral, what is the predicted probability for each class? What is the loss value under this setting? Suppose the weight matrix for 3-class classification is W=[[1.3, -2.2, -1.0, 0.1, 0.7, 0.5], [-2.4, 1.9, 0.5, -0.4, -1.0, 0.2], [1,0, 0.8, -1.5, 1.3, -2.0, 0.6]].

Solution 2 (5)

W=[[1.3, -2.2, -1.0, 0.1, 0.7, 0.5],
[-2.4, 1.9, 0.5, -0.4, -1.0, 0.2],
[1.0, 0.8, -1.5, 1.3, -2.0, 0.6]]

b=[0.1, 0.2, 0.3]

$$\mathcal{P}(y_i = c|\mathbf{x}_i, \mathbf{W}) = \mu_{ic} = \frac{\exp(\mathbf{w}_c^\top \mathbf{x}_i)}{\sum_{c'=1}^{C} \exp(\mathbf{w}_{c'}^\top \mathbf{x}_i)}$$

$$s_1 = [1.3, -2.2, -1.0, 0.1, 0.7, 0.5] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1 = 0.995$$
$$s_2 = [-2.4, 1.9, 0.5, -0.4, -0.1, 0.2] \cdot [3, 2, 1, 3, 0, 4.19] + 0.2 = -3.062$$
$$s_3 = [1.0, 0.8, -1.5, 1.3, -2.0, 0.6] \cdot [3, 2, 1, 3, 0, 4.19] + 0.3 = 9.814$$

$$P(y = c|x) = \frac{\exp(s_c)}{\exp(s_1)+\exp(s_2)+\exp(s_3)}$$

Solution 2 (5)

W=[[1.3, -2.2, -1.0, 0.1, 0.7, 0.5],
[-2.4, 1.9, 0.5, -0.4, -1.0, 0.2],
[1.0, 0.8, -1.5, 1.3, -2.0, 0.6]]

b=[0.1, 0.2, 0.3]

$$\mu_c = P(y = c|x) = \frac{\exp(s_c)}{\exp(s_1)+\exp(s_2)+\exp(s_3)}$$

$$\text{argmax}_{\mathbf{W}} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} \log \mu_{ic}$$

$$loss = -\log \mu_1$$

Coding Practice

- Linear Regression: https://colab.research.google.com/drive/1XjVJO8CRAcHjua2iX_Ul8BtRRxjLbrVb?usp=sharing
- Logistic Regression: https://colab.research.google.com/drive/1d_1yt5z1U8cD4ybC1fq2vUBpJpxGJ-k3?usp=sharing