COGS 181 Fall 2017: Neural Networks and Deep Learning

Practice questions for Midterm I
Instructor: Zhuowen Tu
TA: Zeyu Chen, Long Jin
IA: Adilijiang Ainihaer

We provide below some practice questions for you to prepare midterm 1. Please note that questions to be asked in the exam are not limited to the scope of these sample questions. Please study the class notes by visiting the course webpage at: https://sites.google.com/site/ucsdcogs181fall2017/calendar-notes.

1 Basic Matrix Operations Using Numpy

Suppose $\mathbf{A} = \begin{bmatrix} 0 & -2 & 5 \\ 1 & 0 & -6 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 1 & -1 & 2 \\ 2 & 1 & 5 \end{bmatrix}$, we can convert $\mathbf{A}$ and $\mathbf{B}$ into numpy arrays by

```
import numpy as np
A = np.array([[0, -2, 5], [1, 0, -6]])
B = np.array([[1,-1, 2], [2, 1, 5]])
```

Please write down the python code to compute the following equations: (If the matrix multiplication is not possible, write 'impossible' in your answer)

1. $A - B$
2. $A \circ B$
3. $A^T B$
4. AB

2 One-hot Encoding

Considering the Car-Spec dataset you have seen in homework 1: $S = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}$.

	Length (inch)	Height (inch)	Color
$\mathbf{x}_1$	182.3	62	Silver
$\mathbf{x}_2$	181	66	Blue
$\mathbf{x}_3$	186	56	Red
$\mathbf{x}_4$	179	59	Blue
$\mathbf{x}_5$	182	50	Black

If we use one-hot encoding to represent the categorical features for the color with the order of Red, Blue, Silver, Black being at the first, second, third, and forth place.

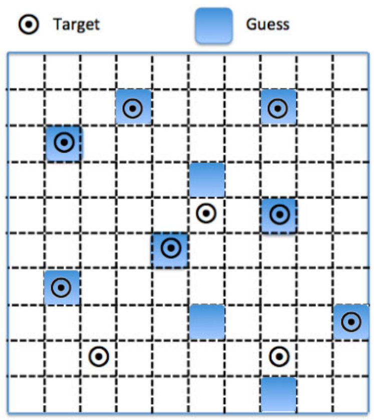
1. Please write down the feature representation in a matrix form for this given set.

2. Now imagine you are given a new feature vector $\mathbf{x}_6 = \{180, 63, 0.5, 0, 0.5, 0\}$, can you try to explain the semantic meaning of this feature vector represents?

3. Do you understand why we need to use one-hot encoding, instead of ordinal index to represent categorical features?

3 Error Metrics

The grid below shows 10x10 = 100 possible, discrete locations for targets to appear. i.e. Any of the 100 locations without targets are thus non-targets. We can make guesses at any of the locations that a target will be there, and all locations without guesses, those where we think we will not find a target, are non-guesses. Evaluate the following metrics for our guesses:



(a) Compute the Sensitivity (=Recall) using the following formula

$$\text{Sensitivity} = \frac{\text{number of true positives (correct guesses)}}{\text{number of targets}}$$

(b) Compute the Specificity using the following formula

$$\text{Specificity} = \frac{\text{number of true negatives (correct non-guesses)}}{\text{number non-targets}}$$

(c) Compute the Precision using the following formula

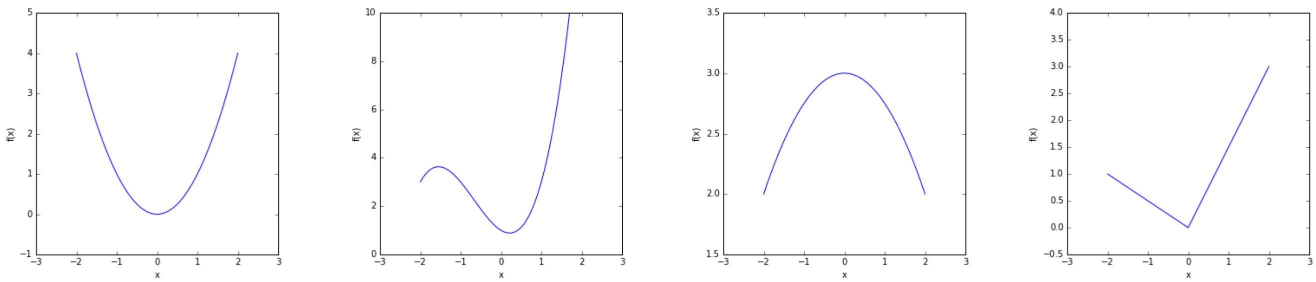
$$\text{Precision} = \frac{\text{number of true positives (correct guesses)}}{\text{number of guesses}}$$

(d) Compute the F-value using the following formula

$$\text{F-value} = \frac{2\text{*Precision*Recall}}{\text{Precision+Recall}}$$

4 Convex functions

Please identify the convexity for the following five functions (a-d). Simply write down whether the function is convex or non-convex in each figure.



5 Logistic Regression

For a logistic regression function with input $x \in R$ and output $y \in \{0, 1\}$, the probability of $P(y = 1|x) = \dfrac{e^{\alpha+\beta x}}{1 + e^{\alpha+\beta x}}$.

1. Please write down the formulation of $P(y = 0|x)$.

2. Show that $P(y = 1|x)]^y \times [P(y = 0|x)]^{1-y} = \dfrac{1}{1 + e^{-(2y-1)(\alpha+\beta x)}}$

3. What is the decision boundary for classifier:

$$y = \begin{cases} 1, & \alpha + \beta x \geq 0, \\ 0, & else \end{cases}$$

   and how is it related to $P(y = 1|x)$.

6 Perceptron

We want to learn a linear classifier on the small synthesized dataset shown below by using a perceptron algorithm.

index	x_1	x_2	target
1	-1	1	1
2	-1	-1	-1
3	1	2	1
4	2	-1	-1

Table 1: The Synthesized Dataset.

The activation rule is defined as:

$$f(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{w}^T\mathbf{x} + b \geq 0, \\ -1, & \text{otherwise,} \end{cases}$$

1. Write down the update rule for the weights and bias in the perceptron learning algorithm.
2. We initialize the weights and bias to be both 0. Suppose we pick up each data point once, and update the weights and bias for 4 steps. Write down the updated weights and bias after each step. (The order is: data point 1 $\rightarrow$ data point 2 $\rightarrow$ data point 3 $\rightarrow$ data point 4.)