Practice problems for Midterm 1 ECE 490/590 Spring 2024

Date: Feb 29, 2024

Instructor: Vikas Dhiman (vikas.dhiman@maine.edu)

1. Total marks are 75.
2. Total time allowed is 75 min.
3. One page 8"x11" cheatsheet is allowed.
4. Calculators are allowed.
5. Computers are not allowed. You must know approximately know what the Python code will output. Minor formatting errors will not be penalized.

1. Write your name here:

2. Write your email here:

Python Basics

The midterm will be on paper, no computers will be allowed. Make sure you know what the python code output should be.

Python questions will be restriced to content covered in Python_1.ipynb, Python_2.ipynb, NumpyTutorial.ipynb

Q1. What will the following code print?

In [32]:
```
hello = "'Hello'"
name = '"ECE"'
pi = 3.1419
print(f'{hello:s} {name}. pi is {pi:.03f}')  # string formatting
```

```
'Hello' "ECE". pi is 3.142
```

Q2. What will the following code print?

In [33]:
```
xs = [1, 2, 3, 'hello', [4, 5, 6]]    # Create a list
print(xs[-1])
```

```
[4, 5, 6]
```

Q3. What will the following code print?

In [34]:
```
nums = list(range(5))    # range is a built-in function that creates a list
print(nums[-2:])
```

```
[3, 4]
```

Q4. Which code is faster for very large lists and dictionaries ?
Option 1 or Option 2? Why?

In [35]:
```
# Code Option 1:
d = {'cat': 'cute', 'dog': 'furry'}  # Create a new dictionary with some dat
print(d['dog'])
# Code option 2:
keys = ['cat', 'dog'] # Create the dictionary with keys as lists
values = ['cute', 'furry'] # # Create the dictionary with values as lists
print(values[keys.index('dog')])
```

```
furry
furry
```

Q5. Which code is faster for very large lists and dictionaries ?
Option 1 or Option 2? Why?

In [36]:
```
# Code Option 1:
d = {0: 'cute', 1: 'furry'}  # Create a new dictionary with some data
print(d[1])
# Code option 2:
values = ['cute', 'furry'] # # Create the dictionary with values as lists
print(values[1])
```

```
furry
furry
```

Q6. What is the output of the following code?

In [37]:
```
class Value:
    def __init__(self, v):
        self.v = v

    def __add__(self, other):
        return self.v * other

print(Value(3) + 2)
```

```
6
```

Numpy basics

Python questions will be restriced to content covered in NumpyTutorial.ipynb

Q7: What is the output of the following code?

In [38]:
```
import numpy as np
x = np.array([[1, 2], [3, 4]])
y = np.array([[5, 6]])
np.concatenate((x.T, y.T), axis=-1)
```

Out[38]:
```
array([[1, 3, 5],
       [2, 4, 6]])
```

Q8. What is the output of the following code?

In [39]:
```
x = np.array([[1, 2], [3, 4]])
y = np.array([[5, 6]])
x @ y.T
```

Out[39]:
```
array([[17],
       [39]])
```

Q9. What is the output of the following code?

In [40]:
```
x = np.array([[1, 2], [3, 4]])
y = np.array([[5, 6]])
(x * y).sum(axis=-1)
```

Out[40]:
```
array([17, 39])
```

Q9: What is the output of the following code

In [41]:
```
import numpy as np
_as = np.array([[2, 3], # a_1
                [3, 5] # a_2
               ])
bs = np.array([[7, 11], # b_1
               [11, 13] # b_2
              ])
print((_as * bs).sum(axis=-1))
```

```
[47 98]
```

$[47, 98]$

Because $[47, 98] = [2 \times 7 + 3 \times 11, 3 \times 11 + 5 \times 13]$

Q9: What is the output of the following code

In [42]:
```
import numpy as np
mat = np.array([[1, 2],
                 [3, 4],
                 [5, 6]])
column_vector = np.array([[1],
                          [-1],
```

```
                        [0]])

mat + column_vector
```

Out[42]:
```
array([[2, 3],
       [2, 3],
       [5, 6]])
```

Perceptron variations

Q10. Show that for any vector dot product with itself is same as its magnitude

$\mathbf{a} = [a_1, a_2, \ldots, a_n]$, it's magnitude squared is same as dot product with itself i.e. $\|\mathbf{a}\|^2 = \mathbf{a}^\top\mathbf{a}$

A10. The mangitude of n-D vector is given by $\|\mathbf{a}\| = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}$ and dot product the vector with itself is given by $\mathbf{a}^\top\mathbf{a} = a_1a_1 + a_2a_2 + \cdots + a_na_n = a_1^2 + a_2^2 + \cdots + a_n^2$. Squaring the magnitude gives us $\|\mathbf{a}\|^2 = a_1^2 + a_2^2 + \cdots + a_n^2$, which is same as $\mathbf{a}^\top\mathbf{a}$.

Q12. Convert the following scalar equation into vector form.

Your end result should contain the vectors $\mathbf{m} = [m; c]$, $\mathbf{y} = [y_1; y_2; \ldots; y_n]$ and $\mathbf{x} = [x_1; x_2, \ldots, x_n]$. You can define other vectors and matrices as needed, included a vector of ones like 1_n.

$$e(m, c, (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)) = (y_1 - (x_1m + c))^2 + (y_2 - (x_2m + c))^2$$

A12. Recall that the magnitude of a vector $\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^n}$ has a similar form to the error function. This suggests that we can define an error vector with the signed error for each data point as it's elements

$$\mathbf{e} = \begin{bmatrix} y_1 - (mx_1 + c) \\ y_2 - (mx_2 + c) \\ \vdots \\ y_n - (mx_n + c) \end{bmatrix}$$

The total error is same as minimizing the square of error vector magnitude which is further same as vector product with itself.

$$e(m, c, (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)) = \|\mathbf{e}\|^2 = \mathbf{e}^\top \mathbf{e}$$

Let us define $\mathbf{x} = [x_1; \dots; x_n]$ to denote the vector of all x coordinates of the dataset and $\mathbf{y} = [y_1; \dots; y_n]$ to denote y coordinates. Then the error vector is:

$$\mathbf{e} = \mathbf{y} - (\mathbf{x}m + \mathbf{1}_n c)$$

where $\mathbf{1}_n$ is a n-D vector of all ones. Finally, we vectorize parameters of the line $\mathbf{m} = [m; c]$. We will also need to horizontally concatenate $\mathbf{x}$ and $\mathbf{1}_n$. Let's call the result $\mathbf{X} = [\mathbf{x}, \mathbf{1}_n] \in \mathbb{R}^{n \times 2}$. Now, the error vector looks like this:

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{m}$$

Expanding the error magnitude:

$$\begin{aligned} \|\mathbf{e}\|^2 &= (\mathbf{y} - \mathbf{X}\mathbf{m})^\top (\mathbf{y} - \mathbf{X}\mathbf{m}) \\ &= \mathbf{y}^\top \mathbf{y} + \mathbf{m}^\top \mathbf{X}^\top \mathbf{X}\mathbf{m} - 2\mathbf{y}^\top \mathbf{X}\mathbf{m} \end{aligned}$$

Q13: Convert the following scalar equation into vector form.

Convert the following scalar equation into vector form. Your end result should contain $\mathbf{m} = [a; b; c]$, $\mathbf{z} = [z_1; z_2; \dots; z_n]$, $\mathbf{y} = [y_1; y_2; \dots; y_n]$ and $\mathbf{x} = [x_1; x_2, \dots, x_n]$. You can define other vectors and matrices as needed, included a vector of all ones like 1_n.

$$e(a, b, c, (x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n)) = (z_1 - (x_1 a + y_1 b + c))^2 + (z$$

A13: A variation of A12

Q14 Convert the following vector equation into even more vectorized form.

$$e(m_0, \mathbf{m}, (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)) = (y_1 - (\mathbf{x}_1^\top \mathbf{m} + m_0))^2 + (y_2 - (\mathbf{x}_2^\top \mathbf{m}$$

where $\mathbf{m} = [m_1; m_2; \dots; m_p] \in \mathbb{R}^p$ is a p-dimensional vector and $\mathbf{x}_i = [x_{i1}; x_{i2}; \dots; x_{ip}] \in \mathbb{R}^p$ are p-dimensional vectors for all $i = \{1, 2, \dots n\}$

Your end result should contain $\mathbf{q} = [m_0, m_1, m_2, \dots, m_p] \in \mathbb{R}^{p+1}$, $\mathbf{y} = [y_1; y_2; \dots; y_n] \in \mathbb{R}^n$ and

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix} \in \mathbb{R}^{n \times p}$$

.

You can define other vectors and matrices as needed, included a vector of all ones like 1_n.

A15. Recall that the magnitude of a vector $\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \dots + v_n^n}$ has a similar form to the error function. This suggests that we can define an error vector with the signed error for each data point as it's elements

$$\mathbf{e} = \begin{bmatrix} y_1 - (\mathbf{x}_1^\top \mathbf{m} + m_0) \\ y_2 - (\mathbf{x}_2^\top \mathbf{m} + m_0) \\ \vdots \\ y_n - (\mathbf{x}_2^\top \mathbf{m}_2 + m_0) \end{bmatrix}$$

The total error is same as minimizing the square of error vector magnitude which is further same as vector product with itself.

$$e(m_0, \mathbf{m}, (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)) = \|\mathbf{e}\|^2 = \mathbf{e}^\top \mathbf{e}$$

Let us define $\mathbf{X} = [\mathbf{x}_1^\top; \dots; \mathbf{x}_n^\top]$ to denote the vector of all x coordinates of the dataset and $\mathbf{y} = [y_1; \dots; y_n]$ to denote y coordinates. Then the error vector is:

$$\mathbf{e} = \mathbf{y} - (\mathbf{1}_n m_0 + \mathbf{X}\mathbf{m})$$

where $\mathbf{1}_n$ is a n-D vector of all ones. Finally, we call parameters of the line $\mathbf{q} = [m_0; \mathbf{m}]$. We will also need to horizontally concatenate $\mathbf{X}$ and $\mathbf{1}_n$. Let's call the result $\bar{\mathbf{X}} = [\mathbf{1}_n, \mathbf{X}] \in \mathbb{R}^{n \times (p+1)}$. Now, the error vector looks like this:

$$\mathbf{e} = \mathbf{y} - \bar{\mathbf{X}}\mathbf{q}$$

Expanding the error magnitude:

$$\|\mathbf{e}\|^2 = (\mathbf{y} - \bar{\mathbf{X}}\mathbf{q})^\top (\mathbf{y} - \bar{\mathbf{X}}\mathbf{q})$$
$$= \mathbf{y}^\top \mathbf{y} + \mathbf{q}^\top \bar{\mathbf{X}}^\top \bar{\mathbf{X}}\mathbf{q} - 2\mathbf{y}^\top \bar{\mathbf{X}}\mathbf{q}$$

Q16: Convert the following scalar equation into vector form.

Your end result should contain $\mathbf{m} = [m; c]$, the matrix $\mathbf{W} = \mathrm{Diag}([w_1; w_2; \ldots; w_n])$, $\mathbf{y} = [y_1; y_2; \ldots; y_n]$ and $\mathbf{x} = [x_1; x_2, \ldots, x_n]$. You can define other vectors and matrices as needed, included a vector of all ones like 1_n.

$$e(m, c, (x_1, y_1, w_1), (x_2, y_2, w_2), \ldots, (x_n, y_n, w_n)) = w_1^2(y_1 - (x_1 m + c))^2 + w_2^2($$

The matrix $\mathbf{W}$ is defined as $\mathrm{Diag}([w_1; w_2; \ldots; w_n])$ which indicates that $\mathbf{W}$ is diagonal matrix of $[w_1; w_2; \ldots; w_n]$.

$$\mathbf{W} = \mathrm{Diag}([w_1; w_2; \ldots; w_n]) = \begin{bmatrix} w_1 & 0 & \ldots & 0 \\ 0 & w_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & w_n \end{bmatrix}$$

A16:

Recall that the magnitude of a vector $\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^n}$ has a similar form to the error function. This suggests that we can define an error vector with the signed error for each data point as it's elements

$$\mathbf{e} = \begin{bmatrix} y_1 - (mx_1 + c) \\ y_2 - (mx_2 + c) \\ \vdots \\ y_n - (mx_n + c) \end{bmatrix}$$

and let $\mathbf{W} = \mathrm{Diag}([w_1; w_2; \ldots; w_n])$.

Note that

$$\mathbf{We} = \begin{bmatrix} w_1(y_1 - (mx_1 + c)) \\ w_2(y_2 - (mx_2 + c)) \\ \vdots \\ w_3(y_n - (mx_n + c)) \end{bmatrix}$$

The total error is same as the square of error vector magnitude

$$e(m, c, (x_1, y_1, w_1), (x_2, y_2, w_2), \ldots, (x_n, y_n, w_n)) = w_1^2(y_1 - (x_1 m + c))^2 + w_2^2($$

The square of error vector magnitude is same as dot product with itself,

$$\|\mathbf{We}\|^2 = (\mathbf{We})^\top (\mathbf{We}) = \mathbf{e}^\top \mathbf{W}^\top \mathbf{We}$$

Let us define $\mathbf{x} = [x_1; \ldots; x_n]$ to denote the vector of all x coordinates of the dataset and $\mathbf{y} = [y_1; \ldots; y_n]$ to denote y coordinates. Then the error vector is:

$$\mathbf{e} = \mathbf{y} - (\mathbf{x}m + \mathbf{1}_n c)$$

where $\mathbf{1}_n$ is a n-D vector of all ones. Finally, we vectorize parameters of the line $\mathbf{m} = [m; c]$. We will also need to horizontally concatenate $\mathbf{x}$ and $\mathbf{1}_n$. Let's call the result $\mathbf{X} = [\mathbf{x}, \mathbf{1}_n] \in \mathbb{R}^{n \times 2}$. Now, the error vector looks like this:

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{m}$$

Expanding the error magnitude:

$$\begin{aligned} \|\mathbf{W}\mathbf{e}\|^2 &= (\mathbf{y} - \mathbf{X}\mathbf{m})^\top \mathbf{W}^\top \mathbf{W}(\mathbf{y} - \mathbf{X}\mathbf{m}) \\ &= \mathbf{y}^\top \mathbf{W}^\top \mathbf{W}\mathbf{y} + \mathbf{m}^\top \mathbf{X}^\top \mathbf{W}^\top \mathbf{W}\mathbf{X}\mathbf{m} - 2\mathbf{y}^\top \mathbf{W}^\top \mathbf{W}\mathbf{X}\mathbf{m} \end{aligned}$$

Q17: Using vector derivatives find the minimum of the following vector quadratic function

$$\arg\ \min_{\mathbf{m}} e(\mathbf{m}) = \mathbf{y}^\top \mathbf{W}^\top \mathbf{W}\mathbf{y} + \mathbf{m}^\top \mathbf{X}^\top \mathbf{W}^\top \mathbf{W}\mathbf{X}\mathbf{m} - 2\mathbf{y}^\top \mathbf{W}^\top \mathbf{W}\mathbf{X}\mathbf{m}$$

The dimensions of the each of the variables are given $\mathbf{m} \in \mathbb{R}^p$, $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{W} \in \mathbb{R}^{n \times n}$, $\mathbf{X} \in \mathbb{R}^{n \times p}$.

A17:

$$\begin{aligned} \mathbf{0}^\top &= \frac{\partial}{\partial \mathbf{m}}(\mathbf{y}^\top \mathbf{W}^\top \mathbf{W}\mathbf{y} + \mathbf{m}^\top \mathbf{X}^\top \mathbf{W}^\top \mathbf{W}\mathbf{X}\mathbf{m} - 2\mathbf{y}^\top \mathbf{W}^\top \mathbf{W}\mathbf{X}\mathbf{m}) && (1) \\ &= 2\mathbf{m}^{*\top} \mathbf{X}^\top \mathbf{W}^\top \mathbf{W}\mathbf{X} - 2\mathbf{y}^\top \mathbf{W}^\top \mathbf{W}\mathbf{X} && (2) \end{aligned}$$

This gives us the solution

$$\mathbf{m}^* = (\mathbf{X}^\top \mathbf{W}^\top \mathbf{W}\mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}^\top \mathbf{W}\mathbf{y}$$

Vector derivatives

Q17: Define a gradient, Jacobian and Hessian in terms of partial derivatives

Gradient is defined for a scalar-valued vector function $f(\mathbf{x})$ ($\mathbf{x} \in \mathbb{R}^n$ and $f(\mathbf{x}) \in \mathbb{R}$) as the arrangement of partial derivatives as a vector

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

Jacobian is defined for a vector-valued vector function $\mathbf{f}(\mathbf{x})$ ($\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m$) as the arrangement of the partial derviatives as the following matrix,

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \mathcal{J}_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m \times n}$$

Note that Jacobian can be written in terms of gradients of each element of the vector function.

$$\mathcal{J}_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = \begin{bmatrix} (\nabla_{\mathbf{x}} f_1(\mathbf{x}))^\top \\ (\nabla_{\mathbf{x}} f_2(\mathbf{x}))^\top \\ \vdots \\ (\nabla_{\mathbf{x}} f_m(\mathbf{x}))^\top \end{bmatrix} \in \mathbb{R}^{m \times n}$$

Hessian matrix of a scalar-valued vector function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as the following arrangement of second derivatives,

$$\mathcal{H} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

It is sometimes also written as $\nabla^2 f(\mathbf{x})$, and hessian can be computed by taking the Jacobian of the gradient,

$$\mathcal{H} f(\mathbf{x}) = \mathcal{J}^\top (\nabla f(\mathbf{x}))$$

If the second partial derivatives are continuous then the Hessian matrix is symmetric.

Q18:

Find the derivative of $f(\mathbf{x}) = (\mathbf{x} - \mathbf{a}_1)^\top A(\mathbf{x} - \mathbf{a}_2)$ with respecto to $\mathbf{x}$.

You can assume $A \in \mathbb{R}^{n \times n}$ to be symmetric. The size of vectors are $\mathbf{x}, \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{b} \in \mathbb{R}^n$

A18

$$\begin{aligned} f(\mathbf{x}) &= (\mathbf{x} - \mathbf{a}_1)^\top A(\mathbf{x} - \mathbf{a}_2) \\ &= \mathbf{x}^\top A\mathbf{x} - \mathbf{a}_1^\top A\mathbf{x} - \mathbf{x}^\top A\mathbf{a}_2 + \mathbf{a}_1^\top A\mathbf{a}_2 \end{aligned}$$

Note that $\mathbf{x}^\top A\mathbf{a}_2$ is a scalar. That's why we can replace it with its transpose $\mathbf{x}^\top A\mathbf{a}_2 = \mathbf{a}_2^\top A\mathbf{x}$

$$= \mathbf{x}^\top A\mathbf{x} - (\mathbf{a}_1 + \mathbf{a}_2)^\top A\mathbf{x} + \mathbf{a}_1^\top A\mathbf{a}_2$$

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{x}} &= 2\mathbf{x}^\top A - (\mathbf{a}_1 + \mathbf{a}_2)^\top A \\ &= (2\mathbf{x} - (\mathbf{a}_1 + \mathbf{a}_2))^\top A \end{aligned}$$

Q20

Show that for $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{c}^\top \mathbf{x} = \mathbf{c}^\top \tag{3}$$

A20: Let $\mathbf{c} = [c_1, c_2, \ldots, c_n]$ and $\mathbf{x} = [x_1, x_2, \ldots x_n]$

Let $f(\mathbf{x}) = \mathbf{c}^\top \mathbf{x} = c_1 x_1 + c_2 x_2 + \ldots c_n x_n$

$$\begin{aligned} \frac{\partial f}{\partial x_1} &= c_1 \\ \frac{\partial f}{\partial x_2} &= c_2 \\ &\vdots \\ \frac{\partial f}{\partial x_n} &= c_n \end{aligned}$$

By Jacobian convention, we arrange the partial derivatives in a row vector:

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{x}} \mathbf{c}^{\top} \mathbf{x} &= \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \dots & \frac{\partial f}{\partial x_n} \end{bmatrix} \\
&= \begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix} = \mathbf{c}^{\top}
\end{aligned}
$$

Q21:

Show that for $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{x} \in \mathbb{R}^n$

$$
\frac{\partial}{\partial \mathbf{x}} \mathbf{A}\mathbf{x} = \mathbf{A} \tag{4}
$$

A21: Let $\mathbf{x} = [x_1; x_2; \dots x_n]$

Let $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^{\top} \\ \mathbf{a}_2^{\top} \\ \vdots \\ \mathbf{a}_n^{\top} \end{bmatrix}$, where $\mathbf{a}_i^{\top} \in \mathbb{R}^{1 \times n}$ are the row vectors of matrix $\mathbf{A}$.

Then

$$
\mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{a}_1^{\top} \\ \mathbf{a}_2^{\top} \\ \vdots \\ \mathbf{a}_n^{\top} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{a}_1^{\top}\mathbf{x} \\ \mathbf{a}_2^{\top}\mathbf{x} \\ \vdots \\ \mathbf{a}_n^{\top}\mathbf{x} \end{bmatrix}
$$

Let

$$
\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix} = \mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{a}_1^{\top}\mathbf{x} \\ \mathbf{a}_2^{\top}\mathbf{x} \\ \vdots \\ \mathbf{a}_n^{\top}\mathbf{x} \end{bmatrix}
$$

By Jacobian convention we arrange the partial derivatives of each function component column-wise

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial \mathbf{x}} \\ \frac{\partial f_2(\mathbf{x})}{\partial \mathbf{x}} \\ \vdots \\ \frac{\partial f_n(\mathbf{x})}{\partial \mathbf{x}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{a}_1^\top \mathbf{x}}{\partial \mathbf{x}} \\ \frac{\partial \mathbf{a}_2^\top \mathbf{x}}{\partial \mathbf{x}} \\ \vdots \\ \frac{\partial \mathbf{a}_2^\top \mathbf{x}}{\partial \mathbf{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_n^\top \end{bmatrix} = \mathbf{A}$$

Q22:

Show that for $\mathbf{x} \in \mathbb{R}^n$ amd $\mathbf{A} \in \mathbb{R}^{n\times n}$

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^\top \mathbf{A}\mathbf{x} = \mathbf{x}^\top(\mathbf{A}^\top + \mathbf{A}) \tag{5}$$

A22:

For product of any two vectors

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^\top \mathbf{y} = \mathbf{y}^\top \tag{6}$$

If $\mathbf{y}$ is a function of $\mathbf{x}$, then

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^\top \mathbf{y} = \mathbf{y}^\top + \left(\frac{\partial}{\partial \mathbf{y}} \mathbf{x}^\top \mathbf{y}\right)\left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right) \tag{7}$$

$$= \mathbf{y}^\top + \mathbf{x}^\top \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right) \tag{8}$$

If $\mathbf{y} = \mathbf{A}\mathbf{x}$, then

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} \mathbf{A}\mathbf{x} = \mathbf{A}$$

and

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^\top \mathbf{A}\mathbf{x} = \mathbf{y}^\top + \mathbf{x}^\top \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right) = \mathbf{x}^\top \mathbf{A}^\top + \mathbf{x}^\top \mathbf{A} = \mathbf{x}^\top(\mathbf{A}^\top + \mathbf{A})$$

Perceptron

Q23:

You are given 2D points and corresponding labels as a training dataset $\{(x_1, y_1, l_1), (x_2, y_2, l_2), \ldots, (x_n, y_n, l_n)\}$, where $x_i \in \mathbb{R}$, $y_i \in \mathbb{R}$ and the labels $l_i \in \{-1, 1\}$. Use the model $\hat{l}_i = \operatorname{sign}(y_i - (mx_i + c))$ to construct a Hinge loss (or error) function. Find the gradient of the Hinge loss function with respect to the vector $\mathbf{m} = [m; c]$.

A23

$$e(y_i, x_i; m, c) = \begin{cases} 0 & \text{if } \operatorname{sign}(y_i - (mx_i + c)) = l_i \\ |y_i - (mx_i + c)| & \text{if } \operatorname{sign}(y_i - (mx_i + c)) \neq l_i \end{cases}$$

$$\mathbf{m} = \begin{bmatrix} m \\ c \end{bmatrix}$$

$$e(y_i, x_i; \mathbf{m}) = \begin{cases} 0 & \text{if } \operatorname{sign}(y_i - [\, x_i \quad 1 \,]\, \mathbf{m}) = l_i \\ |y_i - [\, x_i \quad 1 \,]\, \mathbf{m}| & \text{if } \operatorname{sign}(y_i - [\, x_i \quad 1 \,]\, \mathbf{m}) \neq l_i \end{cases}$$

If $l_i \in \{-1, 1\}$, then $\operatorname{sign}(y_i - [\, x_i \quad 1 \,]\, \mathbf{m}) = l_i$ is same as saying $l_i(y_i - [\, x_i \quad 1 \,]\, \mathbf{m}) > 0$.

$$e(y_i, x_i; \mathbf{m}) = \begin{cases} 0 & \text{if } l_i(y_i - [\, x_i \quad 1 \,]\, \mathbf{m}) > 0 \\ |l_i(y_i - [\, x_i \quad 1 \,]\, \mathbf{m})| & \text{if } l_i(y_i - [\, x_i \quad 1 \,]\, \mathbf{m}) < 0 \end{cases}$$

Also when $z < 0$, then $|z| = -z$.

$$e(y_i, x_i; \mathbf{m}) = \begin{cases} 0 & \text{if } l_i(y_i - [\, x_i \quad 1 \,]\, \mathbf{m}) > 0 \\ -l_i(y_i - [\, x_i \quad 1 \,]\, \mathbf{m}) & \text{if } l_i(y_i - [\, x_i \quad 1 \,]\, \mathbf{m}) < 0 \end{cases}$$

$$\nabla_{\mathbf{m}} e(y_i, x_i; \mathbf{m}) = \begin{cases} 0 & \text{if } l_i(y_i - [\, x_i \quad 1 \,]\, \mathbf{m}) > 0 \\ l_i([\, x_i \quad 1 \,]) & \text{if } l_i(y_i - [\, x_i \quad 1 \,]\, \mathbf{m}) < 0 \end{cases}$$

It is acceptable to leave the answer in above form.

$$e(y_i, x_i; \mathbf{m}) = \max\{0, -l_i(y_i - [\, x_i \quad 1 \,]\, \mathbf{m})\}$$

$$\nabla_{\mathbf{m}} e(y_i, x_i; \mathbf{m}) = \max\{0, l_i([\, x_i \quad 1 \,])\}$$

It is acceptable to leave the answer in above form.

For the entire dataset, we have $\mathbf{y} = [y_1; \ldots; y_n]$ and $\mathbf{x} = [x_1; \ldots; x_n]$, $\mathbf{l} = [l_1; \ldots; l_n]$ the average error is:

$$e(\mathbf{x}, \mathbf{y}; \mathbf{m}) = \frac{1}{n} 1_n^\top \max\{0, -\mathbf{l} \odot (\mathbf{y} - [\, \mathbf{x} \quad 1_n \,]\, \mathbf{m})\},$$

where $\odot$ is the element-wise product. and 1_n is a vector of ones.

and the average gradient is:

$$\nabla_{\mathbf{m}}^{\top} e(\mathbf{x}, \mathbf{y}; \mathbf{m}) = \frac{1}{n} 1_n^{\top} \max\{0, \mathbf{l} \odot ([\mathbf{x} \quad 1_n])\}$$

Q24

You are given p-D points $\mathbf{x}_i \in \mathbb{R}^p$ and corresponding labels as a training dataset $\{(\mathbf{x}_1, l_1), (\mathbf{x}_2, l_2), \ldots, (\mathbf{x}_n, l_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^p$, and the labels $l_i \in \{-1, 1\}$. Use the model $\hat{l}_i = \mathrm{sign}(\mathbf{x}_i^{\top} \mathbf{m} + m_0))$ to construct a Hinge loss (or error) function. Find the gradient of the Hinge loss function with respect to the vector $\mathbf{q} = [m_0; \mathbf{m}]$.

A24:

$$e(m_0, \mathbf{m}; \mathbf{x}_i) = \begin{cases} 0 & \text{if } \mathrm{sign}(\mathbf{x}_i^{\top} \mathbf{m} + m_0) = l_i \\ |\mathbf{x}_i^{\top} \mathbf{m} + m_0| & \text{if } \mathrm{sign}(\mathbf{x}_i^{\top} \mathbf{m} + m_0) \neq l_i \end{cases}$$

$$e(y_i, x_i; m, c) = \begin{cases} 0 & \text{if } \mathrm{sign}(\mathbf{x}_i^{\top} \mathbf{m} + m_0) = l_i \\ |\mathbf{x}_i^{\top} \mathbf{m} + m_0| & \text{if } \mathrm{sign}(\mathbf{x}_i^{\top} \mathbf{m} + m_0) \neq l_i \end{cases}$$

$$\mathbf{q} = \begin{bmatrix} m_0 \\ \mathbf{m} \end{bmatrix}$$

$$e(m_0, \mathbf{m}; \mathbf{x}_i) = \begin{cases} 0 & \text{if } \begin{bmatrix} 1 & \mathbf{x}_i^{\top} \end{bmatrix} \mathbf{q} = l_i \\ | \begin{bmatrix} 1 & \mathbf{x}_i^{\top} \end{bmatrix} \mathbf{q} | & \text{if } \begin{bmatrix} 1 & \mathbf{x}_i^{\top} \end{bmatrix} \mathbf{q} \neq l_i \end{cases}$$

$$\nabla_{\mathbf{q}} e(m_0, \mathbf{m}; \mathbf{x}_i) = \begin{cases} 0 & \text{if } \begin{bmatrix} 1 & \mathbf{x}_i^{\top} \end{bmatrix} \mathbf{q} = l_i \\ | \begin{bmatrix} 1 & \mathbf{x}_i^{\top} \end{bmatrix} | & \text{if } \begin{bmatrix} 1 & \mathbf{x}_i^{\top} \end{bmatrix} \mathbf{q} \neq l_i \end{cases}$$

It is acceptable to leave the answer in above form.

If $l_i \in \{-1, 1\}$, then we can write

$$e(m_0, \mathbf{m}; \mathbf{x}_i) = \max\{0, -l_i(\begin{bmatrix} 1 & \mathbf{x}_i^{\top} \end{bmatrix} \mathbf{q})\}$$

$$\nabla_{\mathbf{m}} e(m_0, \mathbf{m}; \mathbf{x}_i) = \max\{0, -l_i(\begin{bmatrix} 1 & \mathbf{x}_i^{\top} \end{bmatrix})\}$$

It is acceptable to leave the answer in above form.

For the entire dataset, we have $\mathbf{X} = [\mathbf{x}_1^\top; \ldots; \mathbf{x}_n^\top]$, $\mathbf{l} = [l_1; \ldots; l_n]$ the average error is:

$$e(\mathbf{m}; \mathbf{X}, \mathbf{l}) = \frac{1}{n} 1_n^\top \max\{0, -\mathbf{l} \odot ([\, 1_n \quad \mathbf{X}\,]\, \mathbf{q})\}$$

and the average gradient is:

$$\nabla_{\mathbf{m}}^\top e(\mathbf{m}; \mathbf{X}, \mathbf{l}) = \frac{1}{n} 1_n^\top \max\{0, \mathbf{l} \odot ([\, 1_n \quad \mathbf{X}\,])\}$$

Q25: Define Positive definite, Negative definite and Indefinite matrices

Positiive definite

A square matrix $A \in \mathbb{R}^{n \times n}$ is called positive definite if for all $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^\top A \mathbf{x} \succ 0$.

Negative definite

A square matrix $A \in \mathbb{R}^{n \times n}$ is called negative definite if for all $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^\top A \mathbf{x} \prec 0$.

Indefinite

A square matrix $A \in \mathbb{R}^{n \times n}$ is called indefinite if it is neither positive definite nor negative definite.

Q26: How would you find out if a matrix is positive definite/negative definite using eigen values

A26: If all the eigen values are positive, then the matrix is positive definite. If all the eigen values are negative, then the matrix is negative definite.

Q27: Relationship between Hessian matrix and minimum; maximum and saddle points.

Suppose you found an extreme point $\mathbf{x}^*$ of a function $f(\mathbf{x})$, where the gradient is zero

$$\nabla_{\mathbf{x}} f(\mathbf{x})|_{\mathbf{x}^*} = \mathbf{0}$$

You are given the Hessian matrix $\mathcal{H} f(\mathbf{x})|_{\mathbf{x}^*}$ at the extreme point. How would you find out if the extrement point $\mathbf{x}^*$ is a minimum, maximum or a saddle point?

A27:

1. If all the eigen values of the Hessian matrix are positive, then $\mathbf{x}^*$ is a miniumum.
2. If all the eigen values of the Hessian matrix are negative, then $\mathbf{x}^*$ is a maximum.
3. If some of the the eigen values of the Hessian matrix are positive and others are negative, then $\mathbf{x}^*$ is a saddle point.

```
In [ ]:
```