EE412 Foundation of Big Data Analytics, Fall 2023
HW3

Name: 권혁태

Student ID: 20180036

Discussion Group (People with whom you discussed ideas used in your answers):

On-line or hardcopy documents used as part of your answers:

Answer to Problem 1

a) Solve following problems

- Exercise 5.1.2 (5 points)
  Compute the PageRank of each page in Fig 5.7, assuming $\beta = 0.8$.
  You can use programs for simple calculations. If you use any code to solve the problem, please attach your code in `hw3.pdf` with a brief explanation.

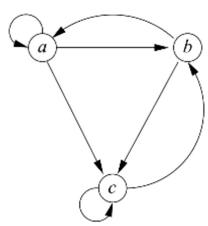


Figure 5.7: An example graph for exercises

a: 0.25925926
b: 0.30864198
c: 0.43209877

```
def matrix_multiply(matrix_a, matrix_b):
    # Check if the matrices can be multiplied
    if len(matrix_a[0]) != len(matrix_b):
        raise ValueError("Matrices cannot be multiplied. Inner dimensions must
match.")

    # Initialize the result matrix with zeros
    result = [[0 for _ in range(len(matrix_b[0]))] for _ in range(len(matrix_a))]

    # Perform matrix multiplication
    for i in range(len(matrix_a)):
        for j in range(len(matrix_b[0])):
```

```
            for k in range(len(matrix_b)):
                result[i][j] += matrix_a[i][k] * matrix_b[k][j]

    return result



def power_iteration(beta, matrix,random_walk,num_iterations):
    n = len(matrix)
    v = []
    for i in range(n):
        v.append([1/n])

    for i in range(num_iterations):
        computed_matrix = matrix_multiply(matrix, v)
        pageRank = []
        for i in range(n):
            pageRank.append([beta * computed_matrix[i][0] + (1-beta) *
random_walk[i][0]])
            v = pageRank
    return v


beta = 0.8

M =  [[1/3, 1/2, 0],
                [1/3, 0, 1/2],
                [1/3, 1/2, 1/2]]
random = [[1/3],[1/3],[1/3]]


pageRank = power_iteration(0.8, M, random, 100);
print(pageRank)
```

Exercise 5.3.1 (5 points)
Compute the topic-sensitive PageRank for the graph of Fig 5.15, assuming $\beta = 0.8$ and the teleport set is:

(a) A only

(b) A and C

You can use programs for simple calculations. If you use any code to solve the problem, please attach your code in `hw3.pdf` with a brief explanation.

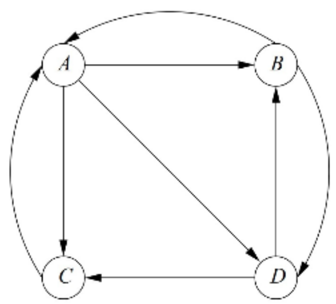


Figure 5.15: Repeat of example Web graph

a) A only

A: 0.42857143

B: 0.19047619

C: 0.19047619

D: 0.19047619

```
def matrix_multiply(matrix_a, matrix_b):
    # Check if the matrices can be multiplied
    if len(matrix_a[0]) != len(matrix_b):
        raise ValueError("Matrices cannot be multiplied. Inner dimensions must 
match.")

    # Initialize the result matrix with zeros
    result = [[0 for _ in range(len(matrix_b[0]))] for _ in range(len(matrix_a))]

    # Perform matrix multiplication
    for i in range(len(matrix_a)):
        for j in range(len(matrix_b[0])):
            for k in range(len(matrix_b)):
                result[i][j] += matrix_a[i][k] * matrix_b[k][j]

    return result
```

```
def power_iteration(beta, matrix,random_walk,num_iterations):
    n = len(matrix)
    v = []
    for i in range(n):
        v.append([1/n])

    for i in range(num_iterations):
        computed_matrix = matrix_multiply(matrix, v)
        pageRank = []
        for i in range(n):
            pageRank.append([beta * computed_matrix[i][0] + (1-beta) *
random_walk[i][0]])
            v = pageRank
    return v

beta = 0.8

M = ([[0, 1/2, 1,0 ],
                [1/3, 0, 0, 1/2],
                [1/3, 0, 0, 1/2],
                [1/3, 1/2,0 ,0]]

random = [[1],[0],[0],[0]]

pageRank = power_iteration(0.8, M, random, 100);
print(pageRank)
```

b) A and C
   A: 0.38571429
   B: 0.17142857
   C: 0.27142857
   D: 0.17142857

```
def matrix_multiply(matrix_a, matrix_b):
    # Check if the matrices can be multiplied
    if len(matrix_a[0]) != len(matrix_b):
        raise ValueError("Matrices cannot be multiplied. Inner dimensions must
match.")

    # Initialize the result matrix with zeros
    result = [[0 for _ in range(len(matrix_b[0]))] for _ in range(len(matrix_a))]

    # Perform matrix multiplication
    for i in range(len(matrix_a)):
```

```
        for j in range(len(matrix_b[0])):
            for k in range(len(matrix_b)):
                result[i][j] += matrix_a[i][k] * matrix_b[k][j]

    return result


def power_iteration(beta, matrix,random_walk,num_iterations):
    n = len(matrix)
    v = []
    for i in range(n):
        v.append([1/n])

    for i in range(num_iterations):
        computed_matrix = matrix_multiply(matrix, v)
        pageRank = []
        for i in range(n):
            pageRank.append([beta * computed_matrix[i][0] + (1-beta) *
random_walk[i][0]])
            v = pageRank
    return v

beta = 0.8

M =  [[0, 1/2, 1,0 ],
                [1/3, 0, 0, 1/2],
                [1/3, 0, 0, 1/2],
                [1/3, 1/2,0 ,0]]

random = [[1/2],
                   [0],
                   [1/2],
                   [0]]

pageRank = power_iteration(0.8, M, random, 50);
print(pageRank)
```

Answer to Problem 2

Figure 2 shows a simple neural network which consists of two input nodes, two hidden nodes and two output nodes. Given input data x_1, x_2, we want the network to predict the target label y_1, y_2 correctly. In other words, the network outputs o_1, o_2 are equal to the target labels y_1, y_2. In order to train the network using gradient descent, we have to compute the gradient of the loss function with respect to the parameters of the network using the backpropagation algorithm.

Suppose we use the sigmoid activation function in the hidden and output layers. Also say we use the mean squared error for the loss function. Express the answer using the parameters in Figure 2.

- Compute the gradient of the loss with respect to w_{ij}^2 $(i, j = 1, 2)$
- Compute the gradient of the loss with respect to w_{ij}^1 $(i, j = 1, 2)$

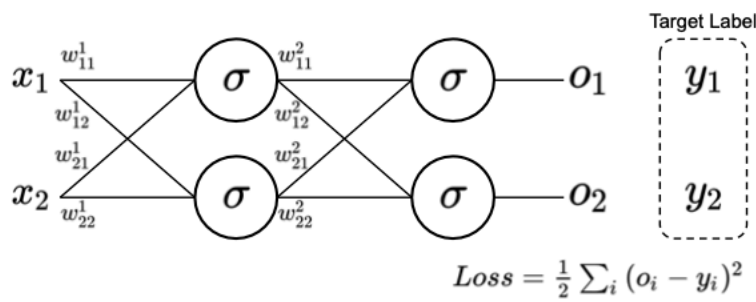
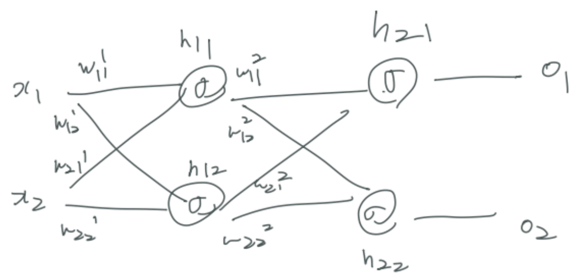


Figure 2: A simple neural network

$$\text{Loss} = \frac{1}{2}\sum_i (o_i - y_i)^2$$

$$\nabla_{o_1} l = o_1 - y_1$$

$$\nabla_{o_2} l = o_2 - y_2$$



$o_1 = h_{21}$	$h_{21} = \sigma(w_{11}^2 h_{11} + w_{21}^2 h_{12})$	$h_{11} = \sigma(w_{11}^1 x_1 + w_{21}^1 x_2)$
$o_2 = h_{22}$	$h_{22} = \sigma(w_{12}^2 h_{11} + w_{22}^2 h_{12})$	$h_{12} = \sigma(w_{12}^1 x_1 + w_{22}^1 x_2)$

$$\boxed{\begin{aligned}\nabla_{o_1} l &= |o_1 - y_1| \\ \nabla_{o_2} l &= |o_2 - y_2|\end{aligned}}$$

$$\nabla_{h_{21}} l = \frac{\partial l}{\partial h_{21}} = \frac{\partial l}{\partial o_1} \cdot \frac{\partial o_1}{\partial h_{21}} = \frac{\partial l}{\partial o_1}$$

$$\nabla_{w_{11}^2} l = \frac{\partial l}{\partial w_{11}^2} = \frac{\partial l}{\partial o_1} \cdot \frac{\partial o_1}{\partial h_{21}} \cdot \frac{\partial h_{21}}{\partial w_{11}} = \frac{\partial l}{\partial o_1} \cdot \underline{(h_{21} \cdot (1 - h_{21}))} \cdot h_{11}$$

$$\nabla_{w_{21}^2} l = \frac{\partial l}{\partial o_1} \cdot (h_{21}(1 - h_{21})) \cdot h_{12}$$

$$\nabla_{w_{12}^2} l = \frac{\partial l}{\partial o_2} \cdot h_{22} \cdot (1 - h_{22}) \cdot h_{11}$$

$$\nabla_{w_{22}^2} l = \frac{\partial l}{\partial o_2} \cdot h_{22}(1 - h_{22}) \cdot h_{12}.$$

$$\nabla_{w_{11}^1} l = \frac{\partial l}{\partial w_{11}^1} = \frac{\partial l}{\partial o_2} \cdot \frac{\partial o_2}{\partial h_{22}} \cdot \left(\frac{\partial h_{22}}{\partial h_{11}} \cdot \frac{\partial h_{11}}{\partial w_{11}^1} + \frac{\partial h_{22}}{\partial h_{12}} \cdot \frac{\partial h_{12}}{\partial w_{11}^1} \right) +$$
$$\frac{\partial l}{\partial o_1} \cdot \frac{\partial o_1}{\partial h_{21}} \cdot \left(\frac{\partial h_{21}}{\partial h_{11}} \cdot \frac{\partial h_{11}}{\partial w_{11}^1} + \frac{\partial h_{21}}{\partial h_{12}} \cdot \frac{\partial h_{12}}{\partial w_{11}^1} \right)$$
$$= \frac{\partial l}{\partial o_2} \cdot \left(h_{22} \cdot (1 - h_{22}) \cdot w_{12}^2 \cdot h_{11}(1 - h_{11}) \cdot x_1 \right) + \frac{\partial l}{\partial o_1} \cdot \left(h_{21}(1 - h_{21}) \cdot w_{11}^2 \cdot h_{11}(1 - h_{11}) \cdot x_1 \right.$$

$$\nabla_{w_{12}^1} l = \frac{\partial l}{\partial w_{12}^1} = \frac{\partial l}{\partial o_2} \cdot \frac{\partial o_2}{\partial h_{22}} \cdot \left(\frac{\partial h_{22}}{\partial h_{11}} \cdot \frac{\partial h_{11}}{\partial w_{12}^1} + \frac{\partial h_{22}}{\partial h_{12}} \cdot \frac{\partial h_{12}}{\partial w_{12}^1} \right) +$$
$$\frac{\partial l}{\partial o_1} \cdot \frac{\partial o_1}{\partial h_{21}} \left(\frac{\partial h_{21}}{\partial h_{11}} \cdot \frac{\partial h_{11}}{\partial w_{12}^1} + \frac{\partial h_{21}}{\partial h_{12}} \cdot \frac{\partial h_{12}}{\partial w_{12}^1} \right)$$
$$= \frac{\partial l}{\partial o_2} \cdot \left(h_{22} \cdot (1 - h_{22}) \cdot w_{22}^2 \cdot h_{12}(1 - h_{12}) \cdot x_1 \right) + \frac{\partial l}{\partial o_1} \left(h_{21}(1 - h_{21}) \cdot w_{21}^2 h_{12}(1 - h_{12}) \cdot x_1 \right)$$

$$\nabla_{w_{21}^1} l = \frac{\partial l}{\partial w_{21}^1} = \frac{\partial l}{\partial o_1} \cdot \frac{\partial o_1}{\partial h_{21}} \cdot \left(\frac{\partial h_{21}}{\partial h_{11}} \cdot \frac{\partial h_{11}}{\partial w_{21}^1} + \frac{\partial h_{21}}{\partial h_{12}} \cdot \frac{\partial h_{12}}{\partial w_{21}^1} \right) + \frac{\partial l}{\partial o_2} \cdot \frac{\partial o_2}{\partial h_{22}} \left(\frac{\partial h_{22}}{\partial h_{11}} \cdot \frac{\partial h_{11}}{\partial w_{21}^1} + \frac{\partial h_{22}}{\partial h_{12}} \cdot \frac{\partial h_{12}}{\partial w_{21}^1} \right)$$

$$= \frac{\partial l}{\partial o_1} \left(h_{21}(1-h_{21}) \cdot w_{11}^2 \cdot h_{11} \cdot (1-h_{11}) \cdot x_2 \right) + \frac{\partial l}{\partial o_2} \left(h_{22} \cdot (1-h_{22}) \cdot w_{12}^2\, h_{11}(1-h_{11}) \cdot x_2 \right)$$

$$\nabla_{w_{22}^1} l = \frac{\partial l}{\partial w_{22}^1} = \frac{\partial l}{\partial o_1} \cdot \frac{\partial o_1}{\partial h_{21}} \left(\frac{\partial h_{21}}{\partial h_{11}} \cdot \frac{\partial h_{11}}{\partial w_{22}^1} + \frac{\partial h_{21}}{\partial h_{12}} \cdot \frac{\partial h_{12}}{\partial w_{22}^1} \right) + \frac{\partial l}{\partial o_2} \cdot \frac{\partial o_2}{\partial h_{22}} \cdot \left(\frac{\partial h_{22}}{\partial h_{11}} \cdot \frac{\partial h_{11}}{\partial w_{22}^1} + \frac{\partial h_{22}}{\partial h_{12}} \cdot \frac{\partial h_{12}}{\partial w_{22}^1} \right)$$

$$= \frac{\partial l}{\partial o_1} \cdot \left(h_{22} \cdot (1-h_{22}) \cdot w_{22}^2\, h_{12} \cdot (1-h_{12})\, x_2 \right) + \frac{\partial l}{\partial o_2} \left(h_{22}(1-h_{22}) \cdot w_{21}^2\, h_{12}(1-h_{12}) \cdot x_2 \right)$$