

Assignment 1 - Introduction to Deep Learning

Tsinghua University

Allan Li, 2025403473

```
In [1]: # imports
import numpy as np
import matplotlib.pyplot as plt
import math
import cv2
```

Problem 0 Objective Questions

0.1: B

0.2: B

0.3: A

Problem 1

```
In [2]: list1 = [10, 20, 30, 40, 50]
reversed_list = []

for i in range(len(list1)-1, -1, -1):
    reversed_list.append(list1[i])

list1 = reversed_list

print(list1)
```

[50, 40, 30, 20, 10]

Problem 2

```
In [3]: d1 = {"a":100,"b":200,"c":300}
d2 = {"a":300,"b":200,"d":400}
d3 = {}

def insertDict(dictToInsert, globalDict):
    for key, value in dictToInsert.items():
        if key in globalDict:
            globalDict[key] += value
        else:
            globalDict[key] = value
    return globalDict
```

```
d3 = insertDict(d1, d3)
d3 = insertDict(d2, d3)

print(d3)
```

```
{'a': 400, 'b': 400, 'c': 300, 'd': 400}
```

Problem 3

```
In [4]: class Point:
        def __init__(self, x, y):
            self.x = x
            self.y = y

        # prints the point out
        def show(self):
            print("(" + str(self.x) + ", " + str(self.y) + ")")

        def dist(self, otherPoint):
            xDist = abs(self.x - otherPoint.x)
            yDist = abs(self.y - otherPoint.y)
            return math.sqrt(xDist ** 2 + yDist ** 2)

        # demonstration of functions
        p1 = Point(0, 0)
        p2 = Point(10, 10)

        p1.show()
        p2.show()
        print(p1.dist(p2))
        print(p2.dist(p1))
```

```
(0, 0)
(10, 10)
14.142135623730951
14.142135623730951
```

Problem 4

```
In [5]: A = np.random.normal(loc = 0, scale = 1, size = (4, 4))

print("1: the two by two matrix")
print(A[1:3, 1:3])

columnMax = A.max(axis = 0)
rowMax = A.min(axis = 1)

print("2: the maximum of each column and the minimum of each row")
print(columnMax)
print(rowMax)

averageCol2 = A[:, 2].mean()
```

```

averageRow3 = A[3, :].mean()
averageAll = A.mean()

print("3: the average of the third column, the average of the fourth row, and the a
print(averageCol2)
print(averageRow3)
print(averageAll)

```

```

1: the two by two matrix
[[1.75094482 0.54672575]
 [1.6523932  0.98915873]]
2: the maximum of each column and the minimum of each row
[0.42083307 1.75094482 0.98915873 1.3219561 ]
[-0.36821198 -1.46656395 -0.86394299 -1.28210716]
3: the average of the third column, the average of the fourth row, and the average o
f all entries
0.3504652658366355
-0.03006694656684772
0.11187503092116834

```

Problem 5

```

In [6]: img = cv2.imread("polar bear.jpg")
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

height = img.shape[0]
width = img.shape[1]

resized = cv2.resize(img, (width // 2, height // 2))
resized_rgb = cv2.cvtColor(resized, cv2.COLOR_BGR2RGB)

plt.imshow(img_rgb)
plt.title("Original Image")
plt.axis("off")
plt.show()

plt.imshow(resized_rgb)
plt.title("Resized Image (Half)")
plt.axis("off")
plt.show()

print("Original shape:", img_rgb.shape)
print("Resized shape:", resized_rgb.shape)

```

Original Image



Resized Image (Half)



Original shape: (2000, 3000, 3)

Resized shape: (1000, 1500, 3)

Problem 6

```
In [7]: y = np.array([1, 2, 3, 0, 2, 1])  
height = np.max(y) + 1  
  
one_arr = np.zeros((y.size, height))
```

```
ohe_arr[np.arange(y.size), y] = 1
print(ohe_arr)
```

```
[[0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]
 [1. 0. 0. 0.]
 [0. 0. 1. 0.]
 [0. 1. 0. 0.]]
```

Problem 7

```
In [8]: x = np.linspace(-5, 5, 1000)

y1 = 1 / (1 + np.exp(-x))
y2 = np.maximum(0, x)

plt.figure(figsize=(5,5))
plt.plot(x, y1, label="y1: 1/(1+e^-x)", color="blue")
plt.plot(x, y2, label="y2: max(0,x)", color="red")

plt.title("Problem 7")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.legend()
plt.grid(True)
plt.show()
```

