Homework 1

Damion Huppert

In [1]:

```
import torch
```

Problem 1

Exercise 4

Answer: 2

In [ ]:

```
x = torch.arange(24).reshape(2, 3, 4)
len(x)
```

Out[ ]:

```
2
```

In [6]:

```
x
```

Out[6]:

```
tensor([[[ 0,  1,  2,  3],
         [ 4,  5,  6,  7],
         [ 8,  9, 10, 11]],

        [[12, 13, 14, 15],
         [16, 17, 18, 19],
         [20, 21, 22, 23]]])
```

Exercise 5

Answer: Yes it is always the first axis (0)

Exercise 6

This is dividing each item in the A by the sum of the row

In [13]:

```
A = torch.arange(6, dtype=torch.float32).reshape(2, 3)
print(A)
A / A.sum(axis=1).unsqueeze(1)
```

```
tensor([[0., 1., 2.],
        [3., 4., 5.]])
```

Out[13]:

```
tensor([[0.0000, 0.3333, 0.6667],
        [0.2500, 0.3333, 0.4167]])
```

Problem 2

Question A

In [18]:

```
A = torch.tensor([[1],[-2],[3],[2]])
B = torch.tensor([[2],[-3],[1],[-1]])
```

```
c = A @ B.T
d = A.T @ B

print("c:\n", c)
print("Shape of c:", c.shape)
print("\nd:\n", d)
print("Shape of d:", d.shape)
```

```
c:
 tensor([[ 2, -3,  1, -1],
        [-4,  6, -2,  2],
        [ 6, -9,  3, -3],
        [ 4, -6,  2, -2]])
Shape of c: torch.Size([4, 4])

d:
 tensor([[9]])
Shape of d: torch.Size([1, 1])
```

Question B

Matrix A (3 × 2):

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

Matrix D (2 × 2, diagonal):

$$D = \begin{bmatrix} d_{11} & 0 \\ 0 & d_{22} \end{bmatrix}$$

Matrix B (2 × 4):

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \end{bmatrix}$$

Matrix E = ADB

ADB =

$$\begin{bmatrix} a_{11}d_{11}b_{11} + a_{12}d_{22}b_{21} & a_{11}d_{11}b_{12} + a_{12}d_{22}b_{22} & a_{11}d_{11}b_{13} + a_{12}d_{22}b_{23} & a_{11}d_{11}b_{14} + a \\ a_{21}d_{11}b_{11} + a_{22}d_{22}b_{21} & a_{21}d_{11}b_{12} + a_{22}d_{22}b_{22} & a_{21}d_{11}b_{13} + a_{22}d_{22}b_{23} & a_{21}d_{11}b_{14} + a \\ a_{31}d_{11}b_{11} + a_{32}d_{22}b_{21} & a_{31}d_{11}b_{12} + a_{32}d_{22}b_{22} & a_{31}d_{11}b_{13} + a_{32}d_{22}b_{23} & a_{31}d_{11}b_{14} + a \end{bmatrix}$$

If we expand the summation we get:

$$E = d_{11}a_1b_1^T + d_{22}a_2b_2^T$$

We can see that the two matrices are equal

Question C

```
In [24]: a = torch.arange(20)
         print(a)
         a = a.reshape(5, 4)
         a
```

```
tensor([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
17,
        18, 19])
```

```
Out[24]: tensor([[ 0,  1,  2,  3],
                 [ 4,  5,  6,  7],
                 [ 8,  9, 10, 11],
                 [12, 13, 14, 15],
                 [16, 17, 18, 19]])
```

Question D

```
In [26]: c = a * a
         c
```

```
Out[26]: tensor([[  0,   1,   4,   9],
                 [ 16,  25,  36,  49],
                 [ 64,  81, 100, 121],
                 [144, 169, 196, 225],
                 [256, 289, 324, 361]])
```

Problem 3

```
In [36]: A = torch.tensor([
             [1, 2],
             [-2, 1],
             [3, -1]
         ],dtype=torch.float32)
```

Question A

```
In [38]: rank = torch.linalg.matrix_rank(A)
         rank
```

```
Out[38]: tensor(2)
```

Question B

```
In [60]: U, S, Vt = torch.linalg.svd(A, full_matrices=False)

         print("SVD matrices:")
         print("U matrix:\n ", U)
         print("Singular values:\n ", S)
         print("V^T matrix:\n ", Vt)

         print("SVD Shapes:")
         print("U matrix: ", U.shape)
         print("Singular values: ", S.shape)
         print("V^T matrix: ", Vt.shape)
print("K = 2")
```

```
SVD matrices:
U matrix:
  tensor([[-8.1650e-02, -9.8995e-01],
        [ 5.7155e-01, -1.4142e-01],
        [-8.1650e-01, -1.9372e-07]])
Singular values:
  tensor([3.8730, 2.2361])
V^T matrix:
  tensor([[-0.9487,  0.3162],
        [-0.3162, -0.9487]])
SVD Shapes:
U matrix:  torch.Size([3, 2])
Singular values:  torch.Size([2])
V^T matrix:  torch.Size([2, 2])
K = 2
```

Question C

In [63]:

```
B = A @ A.T
print(B)
eigvals, eigvecs = torch.linalg.eigh(B)
print("Eigenvalues:\n", eigvals)
print("Eigenvectors:\n", eigvecs)
print("M = 3")
```

```
tensor([[ 5.,  0.,  1.],
        [ 0.,  5., -7.],
        [ 1., -7., 10.]])
Eigenvalues:
 tensor([-4.9202e-07,  5.0000e+00,  1.5000e+01])
Eigenvectors:
 tensor([[-1.1547e-01,  9.8995e-01,  8.1650e-02],
        [ 8.0829e-01,  1.4142e-01, -5.7155e-01],
        [ 5.7735e-01, -1.4901e-08,  8.1650e-01]])
M = 3
```