LAB 1

Question 1

```
def sgd_factorise(A: torch.Tensor, rank: int, num_epochs = 1000, lr = 0.01):

    U = torch.rand(A.shape[0], rank)
    V = torch.rand(A.shape[1], rank)

    for i in range(num_epochs):
        for r in range(A.shape[0]):
            for c in range(A.shape[1]):
                e = A[r][c] - U[r]@V[c].t()
                U[r] = U[r] + lr*e*V[c]
                V[c] = V[c] + lr*e*U[r]
    return U, V
```

```
matrix = torch.tensor([[0.3374, 0.6005, 0.1735], [3.3359, 0.0492, 1.8374], [2.9407, 0.5301, 2.2620]])
U, V = sgd_factorise(matrix, 2)
torch.nn.functional.mse_loss(matrix, torch.mm(U, V.t()), reduction = 'sum')
```

```
tensor(0.1220)
```

Question 2

```
In [11]: U, S, V = torch.svd(matrix)
         S[-1] = 0
         temp = torch.mm(U, torch.diag(S))
         matrix_const = torch.mm(temp, V.t())
         torch.nn.functional.mse_loss(matrix, matrix_const, reduction = 'sum')

Out[11]: tensor(0.1219)
```

The reconstruction loss for the truncated SVD is similar to the loss produced in question 1. The SVD technique can produce the representation of A as a sum of rank one matrices up to its rank r. A matrix of rank r can be approximated by the matrix with rank k where k ≤ r according to Eckart Young Mirsky theorem. Since k = 1 and r = 2 and k ≤ r, the reconstructed matrix in question 2 is a good approximation for the original matrix.

Question 3

```
def sgd_factorise_masked(A: torch.Tensor , M: torch.Tensor , rank : int ,num_epochs =1000, lr =0.01):

    U = torch.rand(A.shape[0], rank)
    V = torch.rand(A.shape[1], rank)

    for i in range(num_epochs):
        for r in range(A.shape[0]):
            for c in range(A.shape[1]):
                if M[r, c] == 1:
                    e = A[r][c] - U[r]@V[c].t()
                    U[r] = U[r] + lr*e*V[c]
                    V[c] = V[c] + lr*e*U[r]
    return U, V
```

```
matrix_masked = torch.tensor([[0.3374, 0.6005, 0.1735], [float('nan'), 0.0492, 1.8374], [2.9407, float('nan'),  2.2620]])
mask = torch.tensor([[1,1,1], [0,1,1], [1,0,1]])
U, V = sgd_factorise_masked(matrix_masked, mask, 2)
completion_matrix = torch.mm(U, V.t())
print(completion_matrix)
torch.nn.functional.mse_loss(matrix, completion_matrix, reduction = 'sum')
```

```
tensor([[0.3334, 0.5913, 0.1803],
        [2.3540, 0.0492, 1.8385],
        [2.9415, 0.3304, 2.2605]])

tensor(1.0042)
```

The error between the original matrix and the estimated matrix is 1.0042. The 2 matrices are not identical; however, rank 2 factorization is able to complete the masked matrix and produce reasonable result similar to the original matrix. Based on the results in question 3, gradient-based approach to matrix factorization is a good algorithm to do matrix completion and produce good approximation of the original matrix.

Question 4

The rating is 3.4869. The sum of squared error of the resultant matrix computed over valid values is 3926469