Lab 1 Exercise - Playing with gradients and matrices in PyTorch

Amishapriya Singh
Student id: 33333904
as6u21@soton.ac.uk

1. Implement a matrix factorisation using gradient descent

1.1. Implement gradient-based factorisation

```
def sgd_factorise(A: torch.Tensor, rank: int, epochs = 1000, lr = 0.01):
  [m, n] = A.shape
  U = torch.rand(m, rank)
  V = torch.rand(rank, n)
  for epoch in range(epochs):
    for r in range(m):
      for c in range(n):
        e = A[r, c] - torch.dot(U[r, :], V[:, c])
        U[r, :] = U[r, :] + (lr * e * V[:, c])
        V[:, c] = V[:, c] + (lr * e * U[r, :])
  return U, V
```

1.2. Factorise and compute reconstruction error

Rank 2 factorisation of the following matrix using *sgd_factorise*:

$$\begin{bmatrix}0.3374 & 0.6005 & 0.1735\\ 3.3359 & 0.0492 & 1.8374\\ 2.9407 & 0.5301 & 2.2620\end{bmatrix} \approx \begin{bmatrix}0.6966 & -0.2414\\ 0.5549 & 1.4885\\ 1.1043 & 1.1078\end{bmatrix} \begin{bmatrix}0.9785 & 0.7543 & 0.8107\\ 1.7955 & -0.2584 & 1.0558\end{bmatrix}$$

Reconstruction loss (squared L2 norm) = 0.1310

2. Compare your result to truncated SVD

Reconstruction loss (squared L2 norm) = 0.1219

Reconstruction loss for truncated SVD is lower than that for gradient based factorisation. This is in accordance with the Eckart-Young-Mirsky Theorem which says that for either the 2-norm or the Frobenius norm, $||A - A_k|| \leq ||A - B||$ for all rank k matrices B.

3. Matrix Completion

3.1. Implement masked factorisation

```
def masked_factorisation(A: torch.Tensor, mask:torch.Tensor, rank:int, epochs = 1000, lr = 0.01):
  [m, n] = A.shape
  U = torch.rand(m, rank)
  V = torch.rand(rank, n)
  for epoch in range(epochs):
    for r in range(m):
      for c in range(n):
        if (mask[r, c] == 1):
          e = A[r, c] - torch.dot(U[r, :], V[:, c])
          U[r, :] = U[r, :] + (lr * e * V[:, c])
          V[:, c] = V[:, c] + (lr * e * U[r, :])
  return U, V
```

3.2. 2 Reconstruct a matrix

$$\text{Estimate of completed matrix: } \begin{bmatrix}0.3364 & 0.6006 & 0.1747\\ 2.3216 & 0.0492 & 1.8377\\ 2.9410 & 0.4924 & 2.2616\end{bmatrix}$$

The squared L2 norm between the estimate and the original matrix A is 1.0302. This tells us that, assuming a low rank (=2) factorisation based latent variable model, we can quite accurately approximate an incomplete matrix by imputing missing values. We could also use this technique to filter out noise from observations.