Lab 1 Exercise - Playing with gradients and matrices in PyTorch

Luke McClure
29573904

$$\mathbf{A} = \begin{bmatrix} 0.3374 & 0.6005 & 0.1735 \\ 3.3359 & 0.0492 & 1.8374 \\ 2.9407 & 0.5301 & 2.2620 \end{bmatrix}$$

Matrix A: Base matrix used for experimentation

1 EXERCISE 1

1.1 Implement gradient-based factorisation

```
def sgd_factorise(A: torch.Tensor, rank: int, num_epochs = 1000, lr = 0.01):
    m = A.shape[0]
    n = A.shape[1]
    U = torch.rand(m, rank)
    V = torch.rand(n, rank)
    for epoch in range(num_epochs):
        for r in range(m):
            for c in range(n):
                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]
```

1.2 Factorise and compute reconstruction error

Applying sgd_factorise to matrix $\mathbf{A}$ to produce a rank 2 factorisation produces the matrices $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$.

$$\hat{\mathbf{U}} = \begin{bmatrix} 0.6168 & -0.1530 \\ 0.4108 & 1.5961 \\ 1.0798 & 1.1800 \end{bmatrix}$$

$$\hat{\mathbf{V}} = \begin{bmatrix} 0.8126 & 1.8290 \\ 0.7836 & -0.2088 \\ 0.8384 & 1.0195 \end{bmatrix}$$

The reconstructed matrix $\mathbf{R}$ can be computed using these matrices by calculating $\hat{\mathbf{U}}\hat{\mathbf{V}}^T$.

$$\mathbf{R} = \begin{bmatrix} 0.2214 & 0.5153 & 0.3612 \\ 3.2531 & -0.0114 & 1.9717 \\ 3.0356 & 0.5997 & 2.1083 \end{bmatrix}$$

Comparing this reconstructed matrix to the original matrix $\mathbf{A}$ using $||\mathbf{A} - \mathbf{R}||_F^2$ produces the mse loss of this method.

$$loss = 0.1220$$

2 EXERCISE 2

2.1 Compare to the truncated-SVD

SVD decomposes a matrix into the matrices $\hat{\mathbf{U}}$, Σ & $\hat{\mathbf{V}}$. Using the SVD method on matrix $\mathbf{A}$, the reconstructed matrix with the third element of Σ set to 0 is as follows.

$$\mathbf{R} = \begin{bmatrix} 0.2245 & 0.5212 & 0.3592 \\ 3.2530 & -0.0090 & 1.9737 \\ 3.0378 & 0.5983 & 2.1023 \end{bmatrix}$$

When compared to matrix $\mathbf{A}$, the mse loss of this reconstructed matrix is as follows.

$$loss = 0.1219$$

The loss and reconstruction using truncated SVD are remarkably similar to the result produced in 1.2, with only 0.0001 separating the loss between the two results.

This act of altering Σ has the same effect as producing a lower rank approximation, this is formalised by the Eckart-Young theorem. By removing the third value within Σ produced by SVD and then reconstructing based on that effectively removes that third rank from the reconstruction and produces a rank 2 reconstruction of matrix $\mathbf{A}$.

3 EXERCISE 3

3.1 Implement masked factorisation

```
def sgd_factorise_masked(A: torch.Tensor, M: torch.Tensor, rank: int, num_epochs = 1000, lr = 0.01):
    m = A.shape[0]
    n = A.shape[1]
    U = torch.rand(m, rank)
    V = torch.rand(n, rank)
    for epoch in range(num_epochs):
        for r in range(m):
            for c in range(n):
                if(M[r][c]):
                    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]
```

3.2 Reconstruct a matrix

Applying sgd_factorise_masked to matrix $\mathbf{A}$ using $rank = 2$ and a mask $\mathbf{M}$ produces the matrices $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$. Reconstructing using these matrices produces $\mathbf{R}$.

$$\mathbf{R} = \begin{bmatrix} 0.3561 & 0.5951 & 0.1518 \\ 2.1802 & 0.0496 & 1.8334 \\ 2.9360 & 0.8643 & 2.2685 \end{bmatrix}$$

The loss of this matrix $\mathbf{R}$ compared to matrix $\mathbf{A}$ produces a much higher loss than in 1.2.

$$loss = 1.4484$$

This is to be expected considering two of the values of this original matrix were hidden for this evaluation, this algorithm cannot completely compensate for the masking although the decomposition produces a close estimate within 50% of each masked value. Disregarding the masked values, this algorithm produces remarkably close values. When the two masked values in $\mathbf{R}$ are set to their value in $\mathbf{A}$, the loss outside these values:

$$loss = 0.0009$$