AMS 691 HW 3

Zhiwei Lin

November 2023

Q1

We are given the matrix:

$$A = U\Sigma V^*$$

where

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{bmatrix},$$

$$\Sigma = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix},$$

$$V^* = \begin{bmatrix} u_{11} & u_{21} & u_{31} \\ u_{12} & u_{22} & u_{32} \\ u_{13} & u_{23} & u_{33} \end{bmatrix}.$$

we know $A = S \times \Sigma \times V^*$ where S and V^* are orthogonal matrix and Σ are non-negative diagonal matrix.A is equivalent to Q^T. where Q is orthogonal. Hence, we can express A as $Q\Lambda Q^T$, where Q is orthogonal. Applying the given conditions, we have:

$$A = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \times \begin{bmatrix} u_{11} & u_{21} & u_{31} \\ u_{12} & u_{22} & u_{32} \\ u_{13} & u_{23} & u_{33} \end{bmatrix}$$

$$\text{A}= \begin{bmatrix} u_{11} & u_{12} & -u_{13} \\ u_{21} & u_{22} & -u_{23} \\ u_{31} & u_{32} & -u_{33} \end{bmatrix} \times \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \times \begin{bmatrix} u_{11} & u_{21} & u_{31} \\ u_{12} & u_{22} & u_{32} \\ u_{13} & u_{23} & u_{33} \end{bmatrix}$$

$$\text{A}= \begin{bmatrix} u_{11} & -u_{13} & u_{12} \\ u_{21} & -u_{23} & u_{22} \\ u_{31} & -u_{33} & u_{32} \end{bmatrix} \times \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} u_{11} & u_{21} & u_{31} \\ u_{12} & u_{22} & u_{32} \\ u_{13} & u_{23} & u_{33} \end{bmatrix}$$

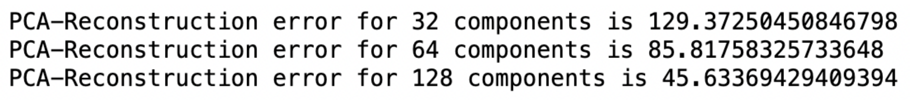
Q2

(a)

The `_do_pca()` function is completed. I centered the data by subtracting the mean of each feature and then applied Singular Value Decomposition (SVD) to find U, Σ, and, V^T. U contains the left singular vectors (principal components), and we selected the first n components columns. Then, project the centered data onto the principal components to obtain the reduced data matrix.

(b)

the `reconstruction()` is completed.



```
PCA-Reconstruction error for 32 components is 129.37250450846798
PCA-Reconstruction error for 64 components is 85.81758325733648
PCA-Reconstruction error for 128 components is 45.63369429409394
```

Figure 1: PCA-Reconstruction Error for 32, 64, 128 components

(c)

The `network()` and `forward()` methods are completed. We consider three cases: weight sharing, weight not sharing, and more linear layers/activation.

In the weight-sharing case, the weights are shared between the encoder and the decoder, with weight matrices transposed to each other.
Encoding: $W \cdot X$ **Decoding:** $W^T \cdot W \cdot X$

In the weight not sharing case, the weights for the encoder and the decoder are initialized independently.

In the more linear layers/activation case, we add more linear layers and activation functions to the network.
Encoding: $W3 \cdot W2 \cdot W1 \cdot X$ **Decoding:** $W1^T \cdot W2^T \cdot W3^T \cdot W3 \cdot W2 \cdot W1 \cdot X$

(d)

Weight Sharing:

- For $d = 32$: AE-Reconstruction error is 130.85101320279526.
- For $d = 64$: AE-Reconstruction error is 87.41427810195496.
- For $d = 128$: AE-Reconstruction error is 47.171194230767405.

(e)

The Reconstruction error of AE and PCA are reported in (d) and (b) respectively. Reconstruction errors of AE for weight sharing are a bit larger than the reconstruction error of PCA. This is expected because using the gradient descent usually find a point near optimal, but PCA will always give optimal solution.

(f)

Weight Sharing:

- For $p = d = 32$, relation between the projection matrix G in PCA and the optimized weight matrix W in AE is 7.841077966249748
- For $p = d = 64$, relation between the projection matrix G in PCA and the optimized weight matrix W in AE is 11.368028124657926
- For $p = d = 128$, relation between the projection matrix G in PCA and the optimized weight matrix W in AE is 16.03697079844863

As can be seen, projection matrix G in PCA and optimized weight matrix W in AE are not equal.

As discussed in Section 4.1, PCA solutions are not unique and can differ by an orthogonal matrix. Consequently, the optimal W obtained through gradient descent might differ from a PCA solution represented by G. However, these solutions are interchangeable through an orthogonal matrix transformation. This ensure many possible solution, but also implies the relationship $W^TW = G^TG$.

Weight Sharing:

- For $p = d = 32$, the relation between the projection matrix G^TG in PCA and the optimized weight matrix W^TW in AE is 0.27178496428837895.
- For $p = d = 64$, the relation between the projection matrix G^TG in PCA and the optimized weight matrix W^TW in AE is 0.040841201430559804.
- For $p = d = 128$, the relation between the projection matrix G^TG in PCA and the optimized weight matrix W^TW in AE is 0.02034998541328963.

The results experimentally justify the relationship $W^TW = G^TG$.

(g)

Weight Not Sharing:

- AE-Reconstruction error for 32-dimensional hidden representation is 135.8005863847178.
- AE-Reconstruction error for 64-dimensional hidden representation is 90.2442145831695.
- AE-Reconstruction error for 128-dimensional hidden representation is 49.0768200327175.

Reconstruction errors for weight not sharing perform slightly worse than weight sharing. They both using gradient decent technique to find optimal weights. It might indicate weight sharing is slightly better in this question.

(h)

More Linear Layers/activations:

- AE-Reconstruction error for 32-dimensional hidden representation is 74.06002060196884.
- AE-Reconstruction error for 64-dimensional hidden representation is 54.64028546048364.
- AE-Reconstruction error for 128-dimensional hidden representation is 35.17007941759635.

More linear layers/activations have a much smaller reconstruction error compared to PCA. This due to more linear layers and non-linear activation function able to capture the nature of non-linearity in dataset.

Through playing with hyperparameters, I set batch size = 64 and epoch = 1000. The logic behind it is letting model to learn small details by decreasing batch size, but set epoch larger so it has enough time to learn. AE-Reconstruction error for 64-dimensional hidden representation is 41.26656976587299