Exercises
Introduction to Machine Learning
Spring 2022

Institute for Machine learning
Dept. of Computer Science, ETH Zürich
Prof. Dr. Andreas Krause, Prof. Dr. Fanny Yang

Homework 4

(Convolutional Networks and Dimensionality Reduction)

For questions, please refer to Moodle.
Released on **26 April 2022**

General Instructions

- Submission of solutions is not mandatory but solving the exercises are highly recommended. The master solution will be released next week.
- Part of the exercises are available on Moodle as a quiz. These problems are marked with [✓].

Exercise 1: Clustering with k-Means

Assume we are given a dataset with three two-dimensional points as shown in Figure 1, and we want to cluster the dataset using the k-means algorithm. Note that if at any iteration no point is assigned to some center μ_j, this center is not updated during that iteration.

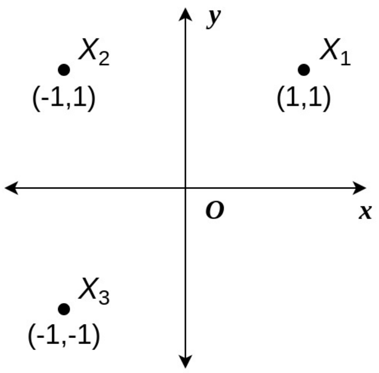

Figure 1: Dataset consisting of three points: X_1, X_2 and X_3.

(a) For $k = 1$, compute the global minimizer μ^* of the k-means objective.

(b) For $k = 2$, if we initialize one cluster center as $\mu_1 = (0, 0)$ and the other as $\mu_2 = (1, -1)$, how many unique cluster assignments will the algorithm have before converging?

(c) For $k = 3$, find the optimal cluster centers. Also compute the value of the k-means objective for the optimal clusters.

Exercise 2: Dimensionality Reduction with PCA

[Exam 2021] In principal component analysis (PCA), we map the data points $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, ..., n$ to $\mathbf{z}_i \in \mathbb{R}^k$, where typically $k \ll d$, by solving the following optimization problem:

$$C_* = \frac{1}{n} \min_{\substack{W \in \mathbb{R}^{d \times k}, W^\top W = I_k \\ \mathbf{z}_1, ..., \mathbf{z}_n \in \mathbb{R}^k}} \sum_{i=1}^{n} \|W\mathbf{z}_i - \mathbf{x}_i\|_2^2 \tag{1}$$

We denote by $W_*, \mathbf{z}_1^*, ..., \mathbf{z}_n^*$ the optimal solution of equation (1). We further assume that the data points are centered, i.e., $\sum_{i=1}^n \mathbf{x}_i = \mathbf{0}$.

(a) [✓] What is the value of $\mathrm{tr}(W_* W_*^\top)$?

1. n 2. k 3. d 4. $\max(n, d)$

(b) [✓] Which holds for $\mathbf{z}_i^*$?

1. $\mathbf{z}_i^* = W_*^\top (W_* W_*^\top)^{-1} \mathbf{x}_i$
2. $\mathbf{z}_i^* = (W_*^\top W_*)^{-1} W_*^\top \mathbf{x}_i$
3. $\mathbf{z}_i^* = (W_* W_*^\top)^{-1} W_*^\top \mathbf{x}_i$
4. $\mathbf{z}_i^* = W_*^\top (W_* W_*^\top)^{-1} W_* \mathbf{x}_i$

(c) [✓] Let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d \geq 0$ be the eigenvalues of the empirical covariance matrix $\Sigma = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \in \mathbb{R}^{d \times d}$. What is the value of C_*?

1. $\sum_{i=k+1}^d \lambda_i$ 2. $\sum_{i=1}^k \lambda_i$ 3. $\sum_{i=k+1}^d \lambda_i^2$ 4. $\sum_{i=1}^k \lambda_i^2$

(d) [✓] In standard PCA, we compute the spectral decomposition (eigenvalues and eigenvectors) of the empirical covariance matrix with size $d \times d$. In kernelized PCA, we instead compute the spectral decomposition of a matrix of size:

1. $d \times d$ 2. $d^2 \times d^2$ 3. $n \times n$ 4. $n^2 \times n^2$

(e) [✓] Imagine two features are identical in the whole dataset, i.e., they are identical among all data samples $\mathbf{x}_1, ..., \mathbf{x}_n$. Then, for the execution of PCA it is true that:

1. PCA never leads to non-zero reconstruction error.
2. We can strictly reduce the dimension of the dataset by at least one with zero reconstruction error.
3. We can strictly reduce the dimension of the dataset by at least two with zero reconstruction error.
4. PCA always leads to non-zero reconstruction error.

(f) [✓] What is true about PCA?

1. PCA helps us find a *linear* mapping to a lower dimensional space.
2. PCA is a supervised learning algorithm.
3. If the underlying data distribution is a Gaussian distribution with diagonal covariance matrix, then PCA is equivalent to k-means clustering.

Exercise 3: Convolutional layers and Fully Connected Layers

In this exercise we theoretically examine the relation between convolutional layers and fully connected linear layers to understand better their similarities and differences. We consider a convolutional layer that takes d channels as input and outputs n channels. Let the input images have size $I_{\text{in}} \times I_{\text{in}}$ and the output $I_{\text{out}} \times I_{\text{out}}$. Also, assume that each filter has dimension $m \times m$.

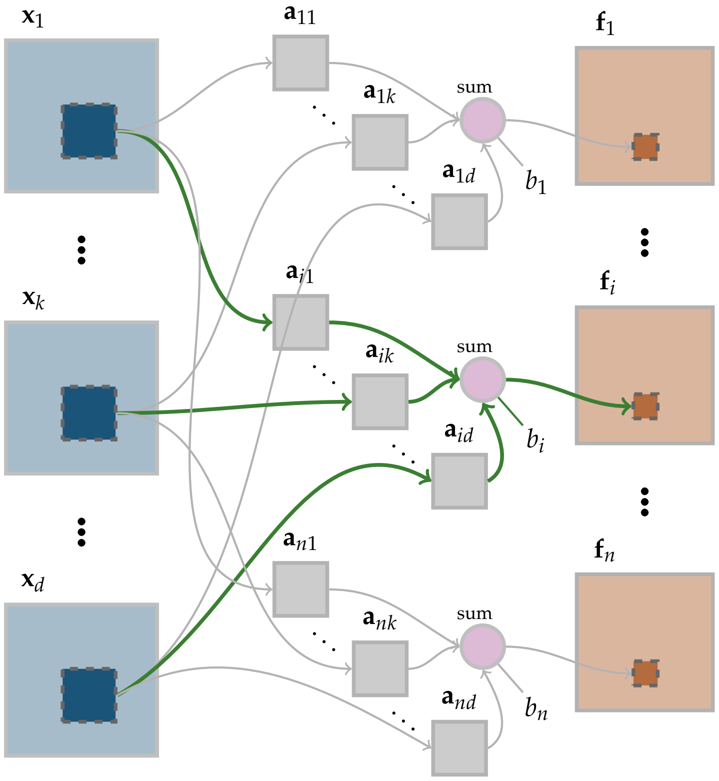

Figure 2: Illustration of convolutional layer with d input channels $\mathbf{x}_1, ..., \mathbf{x}_d$ and n output channels $\mathbf{f}_1, ..., \mathbf{f}_n$. The convolutional filters $\mathbf{a}_{i1}, ..., \mathbf{a}_{in}$ and bias b_i correspond to the $i-$th output channel.

(a) Prove that there exists a fully connected linear layer of proper input size and output size that is functionally equivalent to the described convolutional network.

(b) Deduce that the family of functions written as convolutional layers is a subset of those written as fully connected linear layers.

(c) Compute the number of parameters of the original convolutional layer and the equivalent linear layer in terms of the parameters in the exercise. Which layer has more parameters?

(d) From the previous questions it seems that fully connected layers are more expressive than convolutional layers. Why do we then choose to use convolutional layers and don't just stay with the linear ones?

Exercise 4: Linear Autoencoders and PCA

Recall neural network autoencoders discussed in class. Neural network autoencoders are known to generate very efficient encodings from unlabeled data. These autoencoders can learn complex non-linear functions, thanks to their non-linear activation units. However, without these non-linear activations, the representative power of neural network autoencoders is highly restricted. In this exercise, we draw parallels between an autoencoder with linear activations and principal component analysis.

Prove that an optimal autoencoder with linear activations and bottleneck dimension equal to m is the same as performing prinicipal component analysis with m components.

Hint: Use the Eckart-Young theorem which states the following:

$$\|X - X_r\|_F = \min_{\text{rank}(Y) \leq r} \|X - Y\|_F,$$

where X_r is obtained by taking the SVD of $X = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$, and setting $\sigma_q = 0$ for $q > r$, that is, $X_r = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$, and $\|X\|_F$ is the Frobenius norm of the matrix X, defined as $\|X\|_F^2 = \text{tr}(X^\top X)$.