Exercise Sheet 10

Exercise 1: Auto-Encoders and PCA (25 P)

In this exercise, we would like to show an equivalence between linear autoencoders with tied weights (same parameters for the encoder and decoder) and PCA. We consider the special case of an autoencoder with a single hidden unit. In that case, the autoencoder consists of the two layers:

$$s_i = \boldsymbol{w}^\top \boldsymbol{x}_i \quad \text{(encoder)}$$
$$\hat{\boldsymbol{x}}_i = \boldsymbol{w} \cdot s_i \quad \text{(decoder)}$$

where $\boldsymbol{w} \in \mathbb{R}^d$. We consider a dataset $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$ assumed to be centered (i.e. $\sum_i \boldsymbol{x}_i = 0$), and we define the objective that we would like to minimize to be the mean square error between the data and the reconstruction:

$$J(\boldsymbol{w}) = \frac{1}{N} \sum_{i=1}^{N} ||\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i||^2 \tag{1}$$

Furthermore, to make the objective closer to PCA, we can always rewrite the weight vector as $\boldsymbol{w} = \alpha \mathbf{u}$ where $\mathbf{u}$ is a unit vector (of norm 1) and α is some positive scalar, and search instead for the optimal parameters $\mathbf{u}$ and α.

(a) Show that the optimization problem can be equally rewritten as

$$\arg\min_{\alpha,\mathbf{u}} J(\boldsymbol{w}) = \arg\max_{\alpha,\mathbf{u}} \mathbf{u}^\top S \mathbf{u}$$

where $S = \frac{1}{N} \sum_{i=1}^{N} x_i x_i^\top$, which is a common formulation of PCA.

$$\begin{aligned}
\arg\min_{\alpha,\mathbf{u}} J(\boldsymbol{w}) &= \arg\min_{\alpha,\mathbf{u}} \frac{1}{N} \sum_i ||\boldsymbol{x}_i - \boldsymbol{w}\boldsymbol{w}^\top \boldsymbol{x}_i||^2 \\
&= \arg\min_{\alpha,\mathbf{u}} \frac{1}{N} \sum_i ||\boldsymbol{x}_i - \alpha^2 \mathbf{u}\mathbf{u}^\top \boldsymbol{x}_i||^2 \\
&= \arg\min_{\alpha,\mathbf{u}} \frac{1}{N} \sum_i (||\boldsymbol{x}_i||^2 - 2\alpha^2 \boldsymbol{x}_i^\top \mathbf{u}\mathbf{u}^\top \boldsymbol{x}_i + \alpha^4 \boldsymbol{x}_i^\top \mathbf{u}\mathbf{u}^\top \mathbf{u}\mathbf{u}^\top \boldsymbol{x}_i) \\
&= \arg\min_{\alpha,\mathbf{u}} \frac{1}{N} \sum_i (||\boldsymbol{x}_i||^2 - 2\alpha^2 \mathbf{u}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top \mathbf{u} + \alpha^4 \mathbf{u}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top \mathbf{u}) \\
&= \arg\min_{\alpha,\mathbf{u}} \frac{1}{N} \sum_i (-2\alpha^2 \mathbf{u}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top \mathbf{u} + \alpha^4 \mathbf{u}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top \mathbf{u}) \\
&= \arg\min_{\alpha,\mathbf{u}} \frac{1}{N} \sum_i (-2\alpha^2 + \alpha^4)(\mathbf{u}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top \mathbf{u}) \\
&= \arg\min_{\alpha,\mathbf{u}} (-2\alpha^2 + \alpha^4) \mathbf{u}^\top (\frac{1}{N} \sum_i \boldsymbol{x}_i \boldsymbol{x}_i^\top) \mathbf{u} \\
&= \arg\min_{\alpha,\mathbf{u}} (-2\alpha^2 + \alpha^4) \mathbf{u}^\top S \mathbf{u} \\
&= \arg\max_{\mathbf{u}} \mathbf{u}^\top S \mathbf{u}
\end{aligned}$$

Exercise 2: Lecture Questions (15 P)

(a) Imagine that we want to learn a representation that is invariant to rotations with an autoencoder. How would you train the autoencoder? What would be the input and what would be the objective function?

We would train an autoencoder by presenting it with rotated images and letting it reconstruct the image without a rotation. This forces the autencoder to have the same representation for the same image which is rotated in two different angles. Given an image $\boldsymbol{x}_i$, the input would be randomly rotated images $r(\boldsymbol{x}_i)$ and the objective function $J(\theta) = \frac{1}{N} \sum_{i=1}^{N} ||\boldsymbol{x}_i - f_\theta(r(\boldsymbol{x}_i))||^2$

(b) What is the purpose of skip connections in a U-Net?

The skip connections allow the model to pass low-level details to the decoder part directly. Only information that requires more layers (often abstract concepts) are therefore retained in the representation.

(c) Name three different applications of autoencoders.

- **Compression**
- **Anomaly Detection**
- **Denoising**
- **Segmentation**

Exercise 3: Programming (60 P)

Download the programming files on ISIS and follow the instructions.