Ulrich Oberhofer
ulrich.oberhofer@kit.edu
Sebastian Pütz
sebastian.puetz@kit.edu

Übungsblatt 4

PCA und Variational Autoencoder

Abgabe online auf ILIAS bis 9. Januar 2025, 12:00 Uhr

Für die Abgabe bitte sowohl die handgeschriebene Lösung zu Aufgabe 1 und Aufgabe 2, als auch das bearbeitete Jupyter Notebook zu Aufgabe 3 im Ilias hochladen.

Aufgabe 1 *Principal Component Analysis* (4 Punkte)

1. (2 Punkte) Wir wollen für einen zweidimensionalen Datensatz mit der folgenden Kovarianzmatrix Σ eine PCA durchführen:

$$\begin{pmatrix} 4 & 3 \\ 3 & 12 \end{pmatrix}$$

   Was ist die erste Hauptkomponente und welcher Anteil der Varianz wird durch sie erklärt?

2. (2 Punkte) Angenommen, wir führen eine PCA an einem zweidimensionalen Datensatz durch und erhalten zwei gleiche Eigenwerte. Was bedeutet das in Bezug auf die Relevanz der zwei Dimensionen? Wäre es in diesem Fall sinnvoll eine Dimensionreduktion durchzuführen? Zeichne einen Beispieldatensatz, dessen Eigenwerte gleich sind.

Aufgabe 2 *Herleitung der Objective-Funktion des Variational Autoencoder* (6 Punkte)

Wir möchten in der praktischen Übung einen Variational Autoencoder (VAE) implementieren und trainieren. Die Verlustfunktion aus der Vorlesung (siehe VL 8, Folien 25 und 37) ist gegeben durch

$$\mathcal{L}(\boldsymbol{q}, \boldsymbol{\theta}) = \sum_{i=1}^{N} \left(\mathbb{E}_{\boldsymbol{z}_i \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[\log(p_\theta(\boldsymbol{x}_i|\boldsymbol{z}_i) p(\boldsymbol{z}_i)) \right] - \mathbb{E}_{\boldsymbol{z}_i \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[\log q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i) \right] \right) \tag{1}$$

Hier sind $\boldsymbol{\phi}$ die Parameter des Encoder, $\boldsymbol{\theta}$ die Parameter des Decoder. $\boldsymbol{x}_i$ sind unsere Daten, auf denen wir den VAE trainieren, und $\boldsymbol{z}_i \in \mathbb{R}^d$ ist ein Sample der encodierten Verteilung $q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)$ gegeben dem Input $\boldsymbol{x}_i$. Außerdem haben wir eine Prior-Verteilung $p(z)$ über den Latent Space. Im VAE ist der Prior $p(z) = \mathcal{N}(z|\boldsymbol{0}, \boldsymbol{I})$ eine d-dimensionale Normalverteilung mit Einheitsmatrix als Kovarianzmatrix. Außerdem wählen wir als Encoder-Verteilung

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}_i), \operatorname{diag}(\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\boldsymbol{x}_i))) \tag{2}$$

eine Normalverteilung mit diagonaler Kovarianzmatrix. Die Parameter der Verteilung (Mittelwert $\mu_{\boldsymbol{\phi}}(\boldsymbol{x}_i)$ und Varianz $\sigma^2_{\boldsymbol{\phi}}(\boldsymbol{x}_i)$) werden vom Encoder berechnet.

1. (3 Punkte) Um das Objective in (1) zu implementieren, müssen wir es etwas umformen. Zeige, dass folgendes gilt:

$$\mathcal{L}(\boldsymbol{q}, \boldsymbol{\theta}) = \sum_{i=1}^{N} \left(\mathbb{E}_{\boldsymbol{z}_i \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i) \right] - \mathbb{E}_{\boldsymbol{z}_i \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[\log q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i) - \log p(\boldsymbol{z}_i) \right] \right) \quad (3)$$

2. (3 Punkte) Wir wollen nun den zweiten Term aus (3), der im Übrigen als Kullback-Leibler-Divergenz bezeichnet wird, vereinfachen. Zeige dafür, dass für eindimensionale Verteilungen

$$q(z) = \mathcal{N}(z|\mu, \sigma^2), \qquad p(z) = \mathcal{N}(z|0, 1)$$

gilt:

$$\mathbb{E}_{z \sim q(z)}[\log q(z) - \log p(z)] = -\frac{1}{2}(1 + \log(\sigma^2) - \sigma^2 - \mu^2)$$

Hinweis: Verwende die Formeln der Varianz $\sigma^2 = \mathbb{E}_z[(z-\mu)^2] = \mathbb{E}_z[z^2] - \mu^2$.

Da wir diagonale Normalverteilungen verwenden, folgt für d-dimensionale Normalverteilungen:

$$\begin{aligned} \mathbb{E}_{\boldsymbol{z}_i \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[\log q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i) - \log p(\boldsymbol{z}_i) \right] &= \sum_{j=1}^{d} \mathbb{E}_{z_j \sim q_j(z_j)} \left[\log q_j(z_j) - \log p_j(z_j) \right] \\ &= -\frac{1}{2} \sum_{j=1}^{d} \left(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2 \right) \end{aligned}$$

Hierfür verwenden wir die Notation $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}_i) = [\mu_1, \ldots, \mu_d]$ und $\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\boldsymbol{x}_i) = [\sigma_1^2, \ldots, \sigma_d^2]$. So können wir diesen Erwartungswert in geschlossener Form berechnen. Den ersten Term im Objective

$$\mathbb{E}_{\boldsymbol{z}_i \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{z}_i) \right]$$

müssen wir mittels Sampling approximieren und dabei den sogenannten "Reparameterization Trick" verwenden, damit wir durch den Sampling Prozess Gradienteninformationen erhalten können. Mehr Infos dazu gibt es im Notebook.

Aufgabe 3 *Variational Autoencoder* (10 Punkte)

Bitte löst für diese Aufgabe das im Ilias zur Verfügung gestellte Jupyter Notebook.