Time Series Analysis & Recurrent Neural Networks

lecturer: Daniel Durstewitz
tutors: Georgia Koppe, Leonard Bereska
WS2019/2020

Exercise 12

To be uploaded before the exercise group on January 29th, 2020.

1. **Laplace Approximation**

$$\int e^{Mf(x)} = \sqrt{\frac{2\pi}{M|f''(x_0)|}} e^{Mf(x_0)} \text{ as } M \to \infty, \text{with } x_0 \text{ as the global maximum of } f. \tag{I}$$

Apply Laplace's method (Eq. I) to approximate the integral $N! = \int_0^\infty e^{-t} t^N dt$.

2. **Variational Autoencoder**
In the file *vae.py* you will find a code template for a variational autoencoder (VAE)* applied to hand-written digits of the MNIST dataset†. However, the loss function is still missing and has to be implemented (see task 2d).
The VAE is optimizing the ELBO (Eq. II). The approximate posterior is assumed as a Gaussian $q(Z|X) = \mathcal{N}(\mu, \Sigma)$, with diagonal covariance Σ. The parameters μ, Σ are the output of a neural network (also called *encoder*). The prior $p(Z) = \mathcal{N}(0, I)$ is assumed to be unit Gaussian. The distribution $p(X|Z)$ is also a neural network (*decoder*). In our example, the images of the digits are binary: 1 for a white pixel, 0 for a black pixel. Hence, we take $p(X|Z)$ for each pixel as a Bernoulli distribution, the decoder output is a vector, each dimension characterizing the probability of a pixel being white.

   (a) Show mathematically that minimizing the Kullback-Leibler divergence $KL[q(Z|X)||p(Z|X)]$ between the approximate posterior $q(Z|X)$ and the true posterior $p(Z|X)$ is equivalent to minimizing the evidence lower bound (ELBO, Eq. II),

$$ELBO = \mathbb{E}_{z \sim q(Z|X)} \log(p(X|Z)) + KL(q(Z|X)||p(Z)). \tag{II}$$

   (b) Derive the $KL(q(Z|X)||p(Z))$ in the case of $q(Z|X) = \mathcal{N}(\mu, \Sigma)$, where Σ is diagonal and assuming $p(Z) = \mathcal{N}(0, I)$. This is a more general result of what we showed in exercise 5.

   (c) Confirm that maximizing the likelihood of a Bernoulli distribution $p(X|Z) = \phi^x + (1-\phi)^{(1-x)}$ (where ϕ is the predicted probability of a pixel being white) is giving rise to the *cross-entropy loss* $\mathcal{L}(x, \phi) = x\log(\phi) + (1-x)\log(1-\phi)$, for target x and prediction ϕ .

   (d) Implement the ELBO as a loss function for the VAE in the given code template. Use results from task 2b for the KL-divergence term and from task 2c for the reconstruction term (Hint: Use `torch.nn.functional.binary_cross_entropy`).

   (e) Train the VAE. After training, plot samples from the learned distribution $p(X) = p(X|Z)p(Z)$.

---

*Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2013. arXiv: `1312.6114 [stat.ML]`.
†`https://en.wikipedia.org/wiki/MNIST_database`