# Series 9: Neural Networks

Computational Statistics, FS 2022

May 2022

## 1 Measuring Distances between Distributions

Recall from Exercise Class 10 that the Kullback-Leiber Divergence gives us a way to quantify the difference between two probability distributions. For continuous probability densities $P(x)$ and $Q(x)$, it is defined as

$$\mathrm{KL}(P,Q) = \int P(x) \log\left(\frac{P(x)}{Q(x)}\right) dx. \tag{1}$$

**1**. Show that the Kullback-Leibler Divergence satisfies $\mathrm{KL}(P,Q) \geq 0$, where equality happens if and only if $P(x) = Q(x)$ for all $x$. You may assume that $P$ and $Q$ are continuous.

**2**. Is it true that $\mathrm{KL}(P,Q) = \mathrm{KL}(Q,P)$?

**3**. Show that the Kullback-Leibler Divergence between multivariate Gaussian distributions $P = \mathcal{N}(\boldsymbol{\mu}_P, \boldsymbol{\Sigma}_P)$ and $Q = \mathcal{N}(\boldsymbol{\mu}_Q, \boldsymbol{\Sigma}_Q)$ is

$$\mathrm{KL}(P,Q) = \frac{1}{2}\Big\{ \log\frac{\det \boldsymbol{\Sigma}_Q}{\det \boldsymbol{\Sigma}_P} - d + \mathrm{Tr}(\boldsymbol{\Sigma}_Q^{-1}\boldsymbol{\Sigma}_P) + (\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q)^\top \boldsymbol{\Sigma}_Q^{-1}(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q)\Big\}, \tag{2}$$

where figuring out the meaning of $d$ is part of the exercise.

The equality (2) plays an important role in implementing a variational auto-encoder (VAE).

## 2 Deriving the Auto-Encoding Variational Bayes

In this exercise, we derive the auto-encoding variational Bayes algorithm by following the original paper by Kingma and Welling (see references below). The reason we derive this algorithm is that the variational auto-encoder is a special case (see Exercise 3 below).

The framework is a probabilistic model with latent variables. Specifically, we model the input vector $\mathbf{x} \in \mathbb{R}^p$ and latent variables $\mathbf{z} \in \mathbb{R}^d$ with a family of joint distributions

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) \tag{3}$$

indexed by $\theta$.

We think of the latent variables $\mathbf{z}$ as a lower-dimensional "code" representing $\mathbf{x}$. The conditional probability $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ represents *decoding* $\mathbf{x}$ from the code $\mathbf{z}$. Conversely, the reverse conditional $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ specifies the *encoding* distribution. To fit our probability model, we approximate the true encoding distribution $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ by a simpler approximate model $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. In a variational auto-encoder, this simpler encoding model is a Gaussian distribution, but we stick with the general framework in this derivation.

**1**. Assume that $n$ training examples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(n)} \in \mathbb{R}^p$ are sampled i.i.d. Show that the marginal log likelihood $\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(n)})$ equals

$$\sum_{i}^{n} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = \sum_{i=1}^{n} \log \left( \frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})} \frac{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)}) p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})} \right). \tag{4}$$

**2**. Focus on the marginal log-likelihood $\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$ for a single training example $\mathbf{x}^{(i)}$. By rewriting the expression inside the log as in Equation (4), show that

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)}), p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) \tag{5}$$

where

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \sum_{\mathbf{z}} q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \log \left( \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \right). \tag{6}$$

Note that since $\mathrm{KL}(\cdot, \cdot) \geq 0$, the term $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$ is a lower bound on the marginal log-likelihood. This a specific instance of the *evidence lower bound* (see Wikipedia).

Hint: As long as the log in Equation (4) does not depend on $\mathbf{z}$, you can take its expectation with respect to any distribution over $\mathbf{z}$ and its value will not change. You may take a discrete expectation, *i.e.* $\sum_{\mathbf{z}} p(\mathbf{z}) f(\mathbf{z})$ rather than $\int p(\mathbf{z}) f(\mathbf{z}) \mathrm{d}\mathbf{z}$.

**3**. To fit this model, we maximize the log likelihood by maximizing its lower bound $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$. In other words, $-\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$ is our loss function for training. Rewrite the right hand side of Equation (6) to conclude that

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)}), p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z})]. \tag{7}$$

Interpret the two terms on the right hand side as follows. The KL term regularizes our encoding distribution towards a prior distribution over the latent variables. The expectation term measures the reconstruction error: it assesses how closely the decoder's output resembles the encoder's input.

# 3 Deriving the Variational Auto-Encoder

The variational auto-encoder (VAE) is a special case of the auto-encoding variational Bayes model derived in Exercise 2 above.

In a variational auto-encoder, optimizing for $\boldsymbol{\phi}$ in $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ means fitting an encoder neural network that computes a Gaussian mean vector and covariance matrix as a function of $\mathbf{x}$, to specify the encoding distribution. The parameter $\phi$ represents the weights of this neural network.

Similarly, optimizing for $\boldsymbol{\theta}$ in $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ means fitting the decoder neural network. We take the decoding distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ to be Gaussian with mean vector $\mathbf{x}'(\mathbf{z})$ depending on the latent representation $\mathbf{z}$ and unit covariance matrix. The decoder neural network computes $\mathbf{x}'(\mathbf{z})$ from $\mathbf{z}$. The parameter $\boldsymbol{\theta}$ represents the weights of this neural network.

More formally, a VAE models the encoding and decoding distributions as

$$\begin{aligned} q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) &\sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \operatorname{diag}(\boldsymbol{\sigma}^{\mathbf{2}}(\mathbf{x}))) \\ p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) &\sim \mathcal{N}(\mathbf{x}'(\mathbf{z}), \mathbf{I}), \end{aligned}$$

where $\boldsymbol{\sigma}^{\mathbf{2}}(\mathbf{x})$ is a vector of variances. In addition, we choose a Gaussian prior $p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ over the latent representation. The output of the decoder neural network is the vector $\mathbf{x}'$. After training a VAE, its main purpose lies in computing lower-dimensional latent representations for input data points. To compute a latent representation $\mathbf{z}$ for an input point $\mathbf{x}$, we sample from the encoding distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \operatorname{diag}(\boldsymbol{\sigma}^{\mathbf{2}}(\mathbf{x})))$.

In this exercise, we build on the results of Exercise 2 to derive the loss function for training a VAE.

**1**. Use the results of Exercise 1 to express the KL term in Equation (7) as

$$\mathrm{KL}(\boldsymbol{\phi}) = \frac{1}{2}\left[\left(\sum_{j=1}^{d} \sigma_j^2(\mathbf{x}^{(i)}) - \log \sigma_j^2(\mathbf{x}^{(i)})\right) - d + ||\boldsymbol{\mu}(\mathbf{x}^{(i)})||^2\right], \tag{8}$$

where $\boldsymbol{\phi}$ specifies the encoder neural network that maps $\mathbf{x}$ to $\boldsymbol{\mu}(\mathbf{x})$ and $\boldsymbol{\sigma}^{\mathbf{2}}(\mathbf{x})$.

Note: There is no dependence on $\boldsymbol{\theta}$ (representing the decoder neural network) because our prior distribution over the latent representation, $p_{\boldsymbol{\theta}}(\mathbf{z})$, is simply $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

**2**. Express the reconstruction error (expectation term in Equation (7)) for training example $\mathbf{x}^{(i)}$ as

$$-\frac{1}{2}\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}[\ ||\mathbf{x}^{(i)} - \mathbf{x}'(\mathbf{z})||^2\ ] + \text{constant term}. \tag{9}$$

**3**. Conclude that maximizing the lower bound $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$ of the log likelihood is equivalent to minimizing the loss function

$$\ell(\boldsymbol{\theta}, \boldsymbol{\phi}) = \Big(\sum_{j=1}^{d} \sigma_j^2(\mathbf{x}^{(i)}) - \log \sigma_j^2(\mathbf{x}^{(i)})\Big) + ||\boldsymbol{\mu}(\mathbf{x}^{(i)})||^2 + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}[\ ||\mathbf{x}^{(i)} - \mathbf{x}'(\mathbf{z})||^2\ ],$$

where the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ specify the mappings $\mathbf{x} \mapsto \boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\sigma}^{\mathbf{2}}(\mathbf{x})$ and $\mathbf{z} \mapsto \mathbf{x}'(\mathbf{z})$ in the form of neural networks (encoder and decoder). Optimizing for $\boldsymbol{\theta}, \boldsymbol{\phi}$ corresponds to optimizing the weights of these two neural networks.

**4**. (Bonus) It is not obvious how to compute the gradient with respect to $\boldsymbol{\phi}$ of the expectation term in our loss function. Read the paper by Kingma and Welling to familiarize yourself with their *reparametrization trick*. You are then ready to train a variational auto-encoder with gradient descent.

# 4 References

[1] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations*, ICLR 2014, 2014.
→ access here: `https://arxiv.org/abs/1312.6114`