Exercises
Deep Learning
Fall 2018

Machine Learning Institute
Dept. of Computer Science, ETH Zürich
Fernando Pérez-Cruz
Web http://www.da.inf.ethz.ch/teaching/2018/DeepLearning/

Series Monday, Dec 10, 2018
(Deep Learning, Exercise series 10 - solutions)

Solution 1 (Evidence Lower Bound):

$$\begin{aligned}
\mathcal{L} :&= \log p_\theta(\mathbf{x}) && (1)\\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}) d\mathbf{z} && (2)\\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left(\frac{p_\theta(\mathbf{z},\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})} \right) d\mathbf{z} && (3)\\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left(\frac{p_\theta(\mathbf{z},\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})} \right) d\mathbf{z} && (4)\\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left(\frac{p_\theta(\mathbf{z},\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} \right) d\mathbf{z} + \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left(\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})} \right) d\mathbf{z} && (5)\\
&= \mathcal{F} + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x})) && (6)\\
&\geq \mathcal{F} && (7)
\end{aligned}$$

The ELBO $\mathcal{F}$ can be further split into a reconstruction error term and a KL-divergence (regularization) term, as you have seen in the lecture. Compare also with Eqs. (1)-(3) in Kingma D. P., and Welling M. “Auto-encoding variational bayes.” (2013).

Solution 2 (Variational Autoencoders):

VAE vs. classical autoencoder. Autoencoders try to learn a “compressed representation” of the input (e.g. images, text sequences etc.) by first compressing the input (encoder) and then decompressing it back (decoder) with the aim to match the original input. The learning objective is specified in terms of a distance function that quantifies the information loss that occurs from the compression/decompression.

Instead of learning a deterministic function representing the data (i.e. a compressed representation) as in classical autoencoders, variational autoencoders (VAEs) learn the parameters of a probability distribution representing the data. Since VAEs learn to model the data, we can sample from the distribution and generate new input data samples. Hence, VAEs are generative models.

Reparameterization Trick. The reparameterization trick is needed in order to be able to backpropagate through a random node, as backprop cannot flow through a randomly drawn number. Mathematically, the reparameterization trick leverages the fact that the following two expressions are equivalent

$$\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbb{1}\boldsymbol{\sigma}^2) \quad (8)$$
$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbb{1}) \quad (9)$$

where $\odot$ denotes elementwise multiplication. Thus, introducing the variable $\boldsymbol{\epsilon}$ allows us to reparameterize $\mathbf{z}$ in a way that allows backprop to flow through deterministic nodes, see Figure 1. See also Section 2.4 in Kingma D. P., and Welling M. “Auto-encoding variational bayes.” (2013).

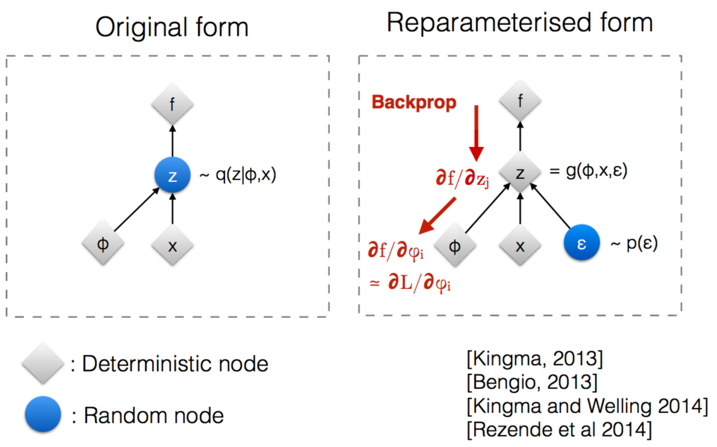

Figure 1: Source: Kingma's NIPS15 Black Box Inference and Learning workshop talk

Latent space dimension. If the dimension of the latent space is larger than the dimension of the input space, the autoencoder will learn to simply be the identity mapping which copies the input to the latent space and back, achieving zero reconstruction error. By introducing a bottleneck, the autoencoder is forced to learn compact and expressive representations.

Reconstruction loss function. The commonly used L_2 loss function (reflected also in the Gaussian output distribution) is generally considered to be a "bad" proxy for visual similarity between original and reconstructed images. Small L_2 distance is sufficient but not necessary for visual similarity: a pair of images with small L_2 distance looks visually similar, but the opposite is not necessarily true. Two visually similar images (take for instance a rotated or translated image) can be quite far in L_2 distance.

Solution 3 (Practical: Autoencoders in Tensorflow):

See the jupyter notebook: *mnist-autoencoder.ipynb* or the python script: *mnist-autoencoder.py*