Variational Auto-Encoders

Homework 3 for Machine Learning Summer School

Instructor: Prof. Jun Zhu

July 13, 2017

1 Backgrounds

Let x denote the observable data and z denote the corresponding latent variable that can describe the data. The generative process, i.e. $p(x, z)$, of the Variational Auto-Encoders (VAEs) with Gaussian prior of z and Bernoulli likelihood of x is defined as follows:

$$\begin{aligned} z &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \mu_x &= f(z) \\ x &\sim \mathcal{B}(\mu_x), \end{aligned}$$

where $\mathcal{N}(\mathbf{0}, \mathbf{I})$ is the standard Gaussian distribution, f is parameterized by a deep neural network and $\mathcal{B}(\mu_x)$ is the multivariate Bernoulli distribution[1].

To infer the latent variable z given observable x, VAEs build a recognition model $q(z|x)$ as follows:

$$\begin{aligned} \mu_z &= g_\mu(x) \\ \sigma_z &= g_\sigma(x) \\ z &\sim \mathcal{N}(\mu_z, \sigma_z), \end{aligned}$$

where g_μ and g_σ are parameterized as deep neural networks[2] and $\mathcal{N}(\mu_z, \sigma_z)$ is a multivariate distribution with diagonal covariance given μ_z and σ_z.

To jointly learn the parameters in all networks, VAEs defines a variational lower-bound of the marginal data likelihood for per data, i.e. $p(x)$ and sums over them. Formally, the objective $\mathcal{L}$ is defined as follows:

$$\log p(x) \geq E_{q(z|x)}[\log p(x, z) - \log q(z|x)] = \mathcal{L}.$$

[1] As in our case where x is multi-dimensional, each dimension of x is sampled independently given the corresponding dimension of μ_x following one-dimensional Bernoulli distribution.

[2] These networks can share most of the parameters.

2 Install ZhuSuan and Get the Code Files

ZhuSuan is a python library for Generative Models, built upon Tensorflow. Unlike existing deep learning libraries, which are mainly designed for supervised tasks, ZhuSuan is featured for its deep root into Bayesian Inference, thus supporting various kinds of generative models: both the traditional hierarchical Bayesian models and recent deep generative models.

Run the following comment to download ZhuSuan from Github:

```
git clone https://github.com/thu-ml/zhusuan.git
```

Run the following comments to install ZhuSuan:

```
cd zhusuan/
```

and

```
sudo pip install .
```
[3]

Run the following comment to download the code files from Github:

```
git clone https://github.com/zhenxuan00/vae-zhusuan-mlss
```

3 Preparation

- Case if you are not familiar with python, please scan the python tutorial to get started with the language: `https://docs.python.org/2.7/tutorial/`
- Case if you are not familiar with numpy (scientific computation in python), please scan the numpy tutorial: `https://docs.scipy.org/doc/numpy-dev/user/quickstart.html`
- Case if you are not familiar with Tensorflow, learn it by these tutorials:
  - `https://www.tensorflow.org/get_started/mnist/beginners`
  - `https://www.tensorflow.org/get_started/mnist/pros`
  - `https://www.tensorflow.org/get_started/mnist/mechanics`
  - `https://www.tensorflow.org/programmers_guide/dims_types`
  - `https://www.tensorflow.org/programmers_guide/variables`
  - `https://www.tensorflow.org/programmers_guide/variable_scope`
- To get started with ZhuSuan, follow this tutorial: `http://zhusuan.readthedocs.io/`

[3] Note that "." is part of the comment instead of the full stop of the sentence.

4 Assignment

4.1 Training of VAEs

Please implement the VAEs on ZhuSuan. Please see detailed instructions in the code file named by "`vae_basic_mlss.py`". You need fill some code in the space with the "TODO" comments and do NOT modify other parts of the code. Note that VAEs use thousands of epochs to converge completely while training tens of epochs is sufficient in this assignment due to the time limit.

4.2 Changing the Likelihood to Gaussian (Optional)

Please change the likelihood function to Gaussian with diagonal covariance matrix. That is, the generative process should be:

$$\begin{aligned} z &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \mu_x &= f_\mu(z) \\ \sigma_x &= f_\sigma(z) \\ x &\sim \mathcal{N}(\mu_x, \sigma_x). \end{aligned}$$

Please see detailed instructions in the code file named by "`vae_gaussian_mlss.py`".