Guidelines to implement the code

Md Mahmudul Hasan

November 30, 2021

1 Background

Variational auto-encoders proposed in [1] are meant to compress the input information into a constrained multivariate latent distribution (encoding) to reconstruct it as accurately as possible (decoding). The model has two prominent neural structures i.e. the encoder network and the decoder network connected by a lower dimensional latent space. The training of the model is carried out by maximizing the objective function known as the evidence lower-bound (ELBO) as given by (1).

For any given data point x,

$$\mathcal{L}(\theta, \phi, x) = -D_{KL}[q_\phi(z|x)||p_\theta(z)] + E_{q_\phi(z|x)}[\log p_\theta(x|z)] = \text{ELBO} \tag{1}$$

where

- $\mathcal{L}(.)$ is the objective function i.e. the ELBO. For any given data batch, for each data point (1) is calculated, then the expectation is found over the batch to find the ELBO for the data batch.
- ϕ, θ are the encoder and decoder parameter sets respectively.
- $q_\phi(z|x)$ and $p_\theta(x|z)$ are encoder and decoder neural networks respectively.
- x is a data point from the ground truth data distribution and z is the low dimensional latent random variable.

2 About the code

Though the paper [1] maximizes the objective function given by (1), we for the convenience of coding minimize the $-\mathcal{L}(.)$. Typical VAE work with images as inputs, we have a special case here where the input set is 10 dimensional jointly Gaussian samples. Our low dimensional random variable z is of dimension 2.

Since GitHub does not allow me to upload larger files, I have uploaded a small dataset of only 100 data points for your reference. You can generate and use your own data to implement the code.

3 How the code works

the ’vae-main’ file has all the hyper-parameters with default values that you can change as required. ’vae-layers’ file has the structure of the neural networks i.e. the encoder and the decoder laid down. ’vae-models’ file defines the model input output relationships, calculates the cost function (in this case the negative of the ELBO) and calls in the encoder and decoder structures form ’vae-layers’ to do so.

The ’vae-main’ file has the ’training step’ defined in it that requires the the calculation of cost function defined in ’vae-models’. ’vae-main’ has the training process ordered in terms of the number of epochs and iterations. ’vae-main’ calls in the ’vae-models’ to calculate and update the gradients of the cost function with respect to the model parameters in every iteration. The program stops after it runs for the default number of ’epochs’ set at the parameter list in the ’vae-main’ file. The file ’data-processing’ is called in by the main file ’vae-main’ to get the dataset shuffled and batched according to the batch size before every epoch.

The ’vae-decoder-sample-generation’ file is used as a generative network after the actual VAE training is done. Using the trained parameters of the decoder network, with latent samples as input to the decoder we can produce the VAE generated data.

References

[1] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.