Report HW3

Giorgio Giannone

July 28, 2018

Introduction

Our goal is to learn two Variational Autoencoders, one with Bernoulli likelihood and the other one with Gaussian likelihood. We optimize the Evidence Lower Bound (ELBO) using the Stochastic Gradient Variational Bayes (SGVB) estimator.

Problem Formulation

Given a joint distribution $p(x, z)$ where x is the observable variable (the distribution of the data) and z is the hidden variable (the topics), we want to learn $p(x)$. Marginalizing w.r.t. z

$$p(x) = \int_Z p(x, z)dz = \int_Z p(x|z)p(z)dz$$

we see that knowing the conditional probability of x given z, and the prior probability of z, we can solve this problem. In classical Bayesian theory, we update recoursively z and we compute the posterior of z given x to improve our prior estimate

$$p(z|x) = \frac{p(x, z)}{p(x)} = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int_Z p(x|z)p(z)dz} = q(z|x)$$

in general the denominator of this integral cannot be evaluated and we need to deal with approximated algorithms (MCMC).

With modern Variational Autoencoders, we learn the parameters of $p(x|z)$ and $p(z|x)$ using Neural Networks in an Unsupervised way. We build an encoder-decoder architecture, where the encoder generates $p(x|z)$ and the decoder generates $q(z|x)$. To optimize the weights of these networks, we apply Variational Methods, in particular we maximize a lower bound for the log-likelihood of $p(x)$.

$$\mathcal{LB} = \mathbb{E}_{q(z|x)}[\log p(x, z) - \log q(z|x)]$$

splitting this term, we obtain $\mathbb{E}_{q(z|x)}[\log p(x, z)]$, that is an expectation w.r.t. the conditional probability of the hidden variable given the data (so a $f(x)$); and a second term $\mathbb{E}_{q(z|x)}[\log q(z|x)]$ that is a constant.

Results

Bernoulli

For the first VAE, we obtain the result in Fig. 2

Gaussian

For the second VAE, we obtain the result in Fig. 3

Alternative Case

Removing the sampling method and the hidden layers for the encoding network, the result for the VAE with Gaussian likelihood improves, as we can see in Fig. 4. This can be explained considering that we are not using activation functions and that the two fully connected layers can build only a linear combination of the input.

```
Epoch 1 (5.8s): Lower bound = -153.1615448
Epoch 2 (4.2s): Lower bound = -118.733398438
Epoch 3 (4.2s): Lower bound = -112.259757996
Epoch 4 (4.2s): Lower bound = -109.203109741
Epoch 5 (4.4s): Lower bound = -107.385444641
Epoch 6 (4.3s): Lower bound = -106.162147522
Epoch 7 (4.3s): Lower bound = -105.091300964
Epoch 8 (4.2s): Lower bound = -104.374458313
Epoch 9 (4.2s): Lower bound = -103.789009094
Epoch 10 (4.3s): Lower bound = -103.084960938
For this assignment, training 10 epochs is sufficient.
```

Figure 1: lower bound with Bernoulli likelihood

```
Epoch 1 (6.6s): Lower bound = 296.658813477
Epoch 2 (4.9s): Lower bound = 487.8074646
Epoch 3 (4.9s): Lower bound = 584.559509277
Epoch 4 (4.9s): Lower bound = 660.177307129
Epoch 5 (5.0s): Lower bound = 665.776245117
Epoch 6 (4.9s): Lower bound = 680.364440918
Epoch 7 (5.0s): Lower bound = 698.304077148
Epoch 8 (5.0s): Lower bound = 656.803710938
Epoch 9 (4.9s): Lower bound = 709.943481445
Epoch 10 (4.9s): Lower bound = 705.224060059
For this assignment, training 10 epochs is sufficient.
```

Figure 2: lower bound with Gaussian likelihood

```
Epoch 1 (4.4s): Lower bound = 473.598754883
Epoch 2 (3.0s): Lower bound = 1059.59606934
Epoch 3 (3.0s): Lower bound = 1144.70080566
Epoch 4 (3.0s): Lower bound = 1238.96435547
Epoch 5 (3.1s): Lower bound = 1305.62097168
Epoch 6 (3.1s): Lower bound = 1319.58215332
Epoch 7 (3.1s): Lower bound = 1519.4420166
Epoch 8 (3.1s): Lower bound = 1496.24597168
Epoch 9 (2.9s): Lower bound = 1431.38146973
Epoch 10 (3.0s): Lower bound = 1547.20837402
For this assignment, training 10 epochs is sufficient.
```

Figure 3: lower bound optimization with Gaussian and removing the hidden layers in the encoder