VAE Tutorial

Sumit Chaturvedi

Abstract. A VAE tutorial suited for undergrads. I do experiments on a synthetic mixture-of-gaussian dataset.

Contents

1 Introduction

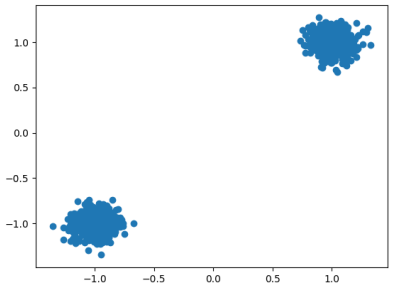


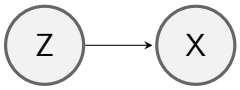
Figure 1: 1000 points generated by this process.

Our problem statement is to capture a data generating process simply by looking at the dataset.

Suppose, for example, we are interested in sampling a random variable X which is generated by the following process.

1. Z is drawn from a **bernoulli distribution** with $p = 0.5$.
2. If $z = 1$, $X \sim \mathcal{N}([1, 1], 0.1)$, else $X \sim \mathcal{N}([-1, -1], 0.1)$.

The **bayesian network** for this data generating process is as follows.



The **VAE** method ([KW13]) provides a way to generate data with very few assumptions on the data generating process.

2 Simple Methods

1. The simplest thing would be do is **MLE**. Assume that the data is drawn from some standard distribution (ex: gaussian) and distribution parameters (ex: μ and σ for gaussian) which maximize log likelihood. **MLE** is not ideal because we cannot capture complex, real-world distributions. Moreover, we aren't making use of the knowledge that data is produced by the process mentioned above.
2. Since, $p(x) = \int_z p(x|z) * p(z)$ and in our case $z \in \{0, 1\}$, we can assume a distribution for $p(x|z = 0)$ and $p(x|z = 1)$ and do **MLE**. This will work extremely well for us because it exactly captures our data generation process. But what if $\int_z p(x|z) * p(z)$ is intractable?

3 VAE Method

3.1 Preliminary Stuff

The method suggested by **VAE** is more complex. It is suited to handle arbitrarily complicated multi-dimensional distributions.

It relies on the following equation for likelihood for one datapoint $x^{(i)}$:

$$log(p(x^{(i)})) = D_{KL}(q(z|x^{(i)})||p(z|x^{(i)})) + \mathcal{L}(x^{(i)}) \tag{1}$$

Here,

1. $p(x^{(i)})$ is the true probability of the data point $x^{(i)}$.
2. $p(z|x^{(i)})$ is the true posterior probability.
3. $q(z|x^{(i)})$ is some arbitrary probability distribution.
4. $\mathcal{L}(x^{(i)})$ is known as the **variational lower bound**.

Try to derive what $\mathcal{L}(x^{(i)})$ should be for this equation to make sense. Use the definition of D_{KL} in terms of **expectation** and apply Bayes Rule. The derivation is in the Appendix. The answer is:

$$\mathcal{L}(x^{(i)}) = -D_{KL}(q(z|x^{(i)})||p(z)) + \mathbb{E}_{q(z|x^{(i)})}[p(x^{(i)}|z)] \tag{2}$$

For complicated distributions with multi-dimensional random variables, calculating $D_{KL}(q(z|x^{(i)})||p(z|x^{(i)}))$ is computationally expensive. But since this quantity is non-negative, we have:

$$log(p(x^{(i)})) \geq \mathcal{L}(x^{(i)}) \tag{3}$$

By maximizing $\mathcal{L}(x^{(i)})$, we can indirectly maximize $log(p(x^{(i)}))$.
Note that if $\mathcal{L}(x^{(i)}) \rightarrow log(p(x^{(i)}))$ then $q(z|x^{(i)}) \rightarrow p(z|x^{(i)})$. This is because the KL-Divergence between the two distribution will go to 0. Therefore, it is helpful to think of $q(z|x^{(i)})$ as the approximate posterior distribution.

3.2 Modelling Assumptions

In order to calculate $\mathcal{L}(x^{(i)})$, there are 3 distributions we have to find. We are free to choose these distributions and how to parameterize them.

1. $q(z|x^{(i)})$: We choose some standard distribution for this with ϕ as the distribution parameters. For example, if we chose $q(z|x^{(i)})$ to be a gaussian, then ϕ would be μ and σ. Similarly, if we chose $q(z|x^{(i)})$ as the exponential distribution, then ϕ would be λ, the rate parameter.
2. $p(z)$: Mostly, this distribution is chosen to be standard normal. This is a blatantly nonsensical choice. For example, in the mixture-of-gaussians example in the Introduction, $p(z)$ is clearly a bernoulli distribution! We'll come to we can make this assumption and get away later.

3. $p(x^{(i)}|z)$: As for $q(z|x^{(i)})$, we choose some standard distribution parameterized by θ to represent this.

From now on, we'll refer to $q(z|x^{(i)})$ as $q_\phi(z|x^{(i)})$ and $p(x^{(i)}|z)$ as $p_\theta(x^{(i)}|z)$ to emphasize that these are parametrized distributions.

3.3 Maximizing $\mathcal{L}(x^{(i)})$

Figure 2 shows the **VAE** architecture. f, g, h, k are functions approximated by neural networks. The weights of f and g are collectively known as ϕ. Similarly the weights of h and k are collectively known as θ. We use gradient ascent to find ϕ^* and θ^* such that the combination of the two network outputs (KL loss and the conditional likelihood i.e. the stuff in the red boxes) is maximized.

The interesting part is that in training the $p_\theta(x^{(i)}|z)$ model, instead of sampling from $p(z)$, we are sampling from $q_\phi(z|x^{(i)})$. The rationale behind this is the following. Recall that $q_\phi(z|x^{(i)})$ is meant to approximate $p(z|x^{(i)})$. By sampling from $q_\phi(z|x^{(i)})$, we are sampling those values of z which the dataset deems more likely. As it is, the $p_\theta(x^{(i)}|z)$ model has finite capacity. It seems wasteful for it to learn distributions for z which will occur rarely (as would happen if we sampled z from $p(z)$.

Note that the sampling step is non-differentiable. One would think that due to this, we can't back-propagate the loss from the conditional log likelihood back to ϕ. Luckily we have a solution. By reordering the sampling step, we can ensure that gradients can flow all the way back to ϕ. Exact details are presented in the Appendix.

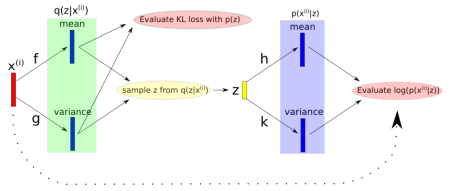


Figure 2: VAE Architecture

4 Experiment

$q_\phi(z|x)$ and $p_\theta(x|z)$ were assumed to be gaussian and the the means and log of variances were approximated by 2 linear layers. The model was trained using Adam Optimizer with learning rate 4.10^{-3}. After training for 1000 epochs, we used the equation

$$p(x) = p(x|z).p(z) \tag{4}$$

Where, $p(z)$ was the standard gaussian (as mentioned earlier) and $p(x|z)$ was the trained model $p_\theta(x|z)$.

We sampled 1000 points from the trained model. They can be seen in Figure 4.

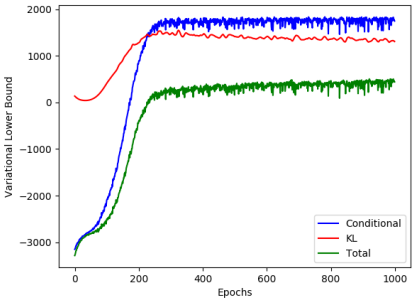


Figure 3: Training Curves

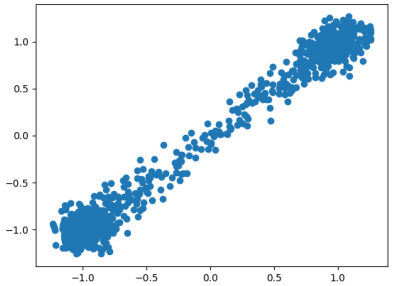


Figure 4: Not a perfectly faithful representation of the original dataset but close!

5 Discussion

5.1 Role of posterior $q_{phi}(z|x)$

Note that to effectively sample X according to this method, we want to train an accurate $p_\theta(x|z)$. For that $q_\phi(z|x)$ would need to be a good approximation of the true posterior $p(z|x)$. In our case, with our mixture-of-gaussian data (Figure 1), if x_1 and x_2 are both negative, then it is highly likely that $z = 0$. Likewise, when x_1 and x_2 are both positive, it is highly likely that $z = 1$.

This can be captured by our assumption that $z|x \sim \mathcal{N}(\mu_\phi, \sigma_\phi)$. It is plausible that when x_1 and x_2 are negative, the network learns to output $\mu = 0$. Similarly when x_1 and x_2 are positive, the network might output $\mu = 1$. In both cases, if the network additionally outputs a really small σ, then this would be a good approximation of $z|x$.

5.2 Role of prior $p(z)$

Actually note that all the approximate posterior has to do is to separate out the two individual gaussians from which x is drawn. In the sense that the z values for xs drawn from the lower gaussian should be different than the z values for the xs drawn from the upper gaussian. should be different. Even if in one case $z = -10$ and in the other $z = 100$, it doesn't matter because the $p_\theta(x|z)$ can learn a suitable transformation if need be.

In such a case, multiple possible $q_\theta(z|x)$ exist, which will all perform equally well. This abundance of options will make the learning procedure harder. In light of this, we can view the KL term involving $q_\phi(z|x)$ and $p(z)$ as a regularizer which puts the constraint that the support of the two distributions should be similar.

In Figure 5, for each $x^{(i)}$ in the dataset, I have sampled a $z \sim q_\phi(z|x^{(i)})$. I have added noise in the plot along y-direction so that the z-clusters are visible more clearly.

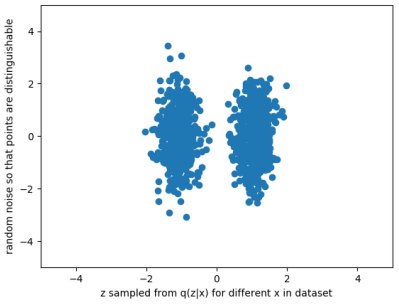


Figure 5: We can see that two clusters formed for z values. The clusters represent the two types of x that exist in the dataset.

6 Appendix

6.1 Derivation of $\mathcal{L}$

Our original equation was

$$log(p(x^{(i)})) = D_{KL}(q(z|x^{(i)})||p(z|x^{(i)})) + \mathcal{L}(x^{(i)}) \tag{5}$$

By the definition of D_{KL}:

$$= \mathbb{E}_{q(z|x^{(i)})}[log(\frac{q(z|x^{(i)})}{p(z|x^{(i)})})] + \mathcal{L}(x^{(i)}) \tag{6}$$

By using Baye's Rule on $p(z|x^{(i)})$ and noticing that:

$$\mathbb{E}_{q(z|x^{(i)})}[log(p(x^{(i)}))] = log(p(x^{(i)})) \tag{7}$$

We get:

$$= \mathbb{E}_{q(z|x^{(i)})}[log(\frac{q(z|x^{(i)})}{p(z).p(x^{(i)}|z)})] + log(p(x^{(i)})) + \mathcal{L}(x^{(i)}) \tag{8}$$

Cancelling the $log(p(x^{(i)}))$ on both sides:

$$\mathcal{L}(x^{(i)}) = -D_{KL}(q(z|x^{(i)})||p(z)) + \mathbb{E}_{q(z|x^{(i)})}[p(x^{(i)}|z)] \tag{9}$$

6.2 KL-Divergence between two Univariate Gaussians

Let $X_1 \sim \mathcal{N}(\mu_1, \sigma_1)$ and $X_2 \sim \mathcal{N}(\mu_2, \sigma_2)$ be two Random Variables.

The KL-Divergence between the two **pdf**s is given by:

$$D_{KL}(p_1||p_2) = \int p_1 log(\frac{p_1}{p_2}) \tag{10}$$

In our case $p_1(x) = \frac{e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}}{\sqrt{2\pi}\sigma_1}$ and $p_2(x) = \frac{e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}}}{\sqrt{2\pi}\sigma_2}$.

Plug this into the above formula and simplify to get

$$D_{KL}(p_1||p_2) = \int_{-\infty}^{+\infty} p_1(x) log(e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2} + \frac{(x-\mu_2)^2}{2\sigma_2^2}} . \frac{\sigma_2}{\sigma_1}) dx \tag{11}$$

Further, we can write.

$$= log(\frac{\sigma_2}{\sigma_1}) + \int_{-\infty}^{+\infty} p_1(x)(-\frac{(x-\mu_1)^2}{2\sigma_1^2} + \frac{(x-\mu_2)^2}{2\sigma_2^2}) dx \tag{12}$$

Observe that the first term in the integral is very close to the variance.

$$= log(\frac{\sigma_2}{\sigma_1}) - \frac{\mathbb{E}_{p_1}[(x-\mu_1)^2]}{2\sigma_1^2} + \int_{-\infty}^{+\infty} p_1(x) \frac{(x-\mu_2)^2}{2\sigma_2^2} dx \tag{13}$$

We'll add and subtract μ_1 in the second term so that we can make it closer to the variance.

$$= log(\frac{\sigma_2}{\sigma_1}) - \frac{\sigma_1^2}{2\sigma_1^2} + \int_{-\infty}^{+\infty} p_1(x) \frac{(x-\mu_1+\mu_1-\mu_2)^2}{2\sigma_2^2} dx \tag{14}$$

On expanding the square

$$= log(\frac{\sigma_2}{\sigma_1}) - \frac{1}{2} + \frac{1}{2\sigma_2^2}[\sigma_1^2 + (\mu_1 - \mu_2)^2 + I] \tag{15}$$

Where $I = 2.(\mu_1 - \mu_2) \int_{-\infty}^{+\infty} p_1(x).(x - \mu_1)dx$. Convince yourself that this integral evaluates to 0.

Hence,

$$D_{KL}(p_1||p_2) = log(\frac{\sigma_2}{\sigma_1}) - \frac{1}{2} + \frac{1}{2\sigma_2^2}[\sigma_1^2 + (\mu_1 - \mu_2)^2] \tag{16}$$

6.3 Making sampling from q_ϕ differentiable

This step is known as the reparametrization trick in the literature.

6.3.1 q_ϕ represents a gaussian distribution

In this case, we sample $\epsilon \sim \mathcal{N}(0, 1)$ and then generate samples of z by:

$$z = \mu_\phi + \epsilon.\sigma_\phi \tag{17}$$

It is easy to show that $z \sim \mathcal{N}(\mu_\phi, \sigma_\phi)$

6.3.2 q_ϕ represents a bernoulli distribution

In this case we can sample $\epsilon \sim \mathcal{U}(0, 1)$. The network will output p_ϕ which is the bernoulli distribution parameter. If $\epsilon > p_\phi$, we will set $z = 0$, else $z = 1$. This IF condition is a differentiable operation so there is no problem there.

7 Acknowledgement

I benefited greatly from the tutorial by Carl Doersch ([Doe16]).

References

[KW13] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2013. arXiv: 1312.6114 [stat.ML].

[Doe16] Carl Doersch. "Tutorial on Variational Autoencoders". In: *ArXiv* abs/1606.05908 (2016).