Deep Learning & Engineering Applications

13. Variational AutoEncoders – Part II

JIDONG J. YANG

COLLEGE OF ENGINEERING

UNIVERSITY OF GEORGIA

Outline

Another derivation (Jensen)

IWAE

- Importance sampling
- Another lower bound

Optimization

- Likelihood ratio gradient
- Reparameterization trick

Information Theory & Mutual Information

Expressivity of the Decoder

Recall from previous lecture

$$\log p_\theta\left(x^{(i)}\right) = \boldsymbol{E}_{z\sim q_{\boldsymbol{\phi}}(z|x^{(i)})}\left[\log p_\theta(x)\right]$$

$$= \boldsymbol{E}_z\left[\log \frac{p_\theta\left(x^{(i)}|z\right)p(z)}{p_\theta\left(z|x^{(i)}\right)}\right] \qquad \text{Bayes' Rule}$$

$$= \boldsymbol{E}_z\left[\log \frac{p_\theta\left(x^{(i)}|z\right)p(z)}{p_\theta\left(z|x^{(i)}\right)}\frac{q_{\boldsymbol{\phi}}\left(z|x^{(i)}\right)}{q_{\boldsymbol{\phi}}\left(z|x^{(i)}\right)}\right]$$

$$= \boldsymbol{E}_z\left[\log\left(p_\theta\left(x^{(i)}|z\right)\frac{p(z)}{q_{\boldsymbol{\phi}}\left(z|x^{(i)}\right)}\frac{q_{\boldsymbol{\phi}}\left(z|x^{(i)}\right)}{p_\theta\left(z|x^{(i)}\right)}\right)\right]$$

$$= \boldsymbol{E}_z\left[\log p_\theta\left(x^{(i)}|z\right)\right] - \boldsymbol{E}_z\left[\log\frac{q_{\boldsymbol{\phi}}\left(z|x^{(i)}\right)}{p(z)}\right] + \boldsymbol{E}_z\left[\log\frac{q_{\boldsymbol{\phi}}\left(z|x^{(i)}\right)}{p_\theta\left(z|x^{(i)}\right)}\right]$$

$$= \boxed{\boldsymbol{E}_z\left[\log p_\theta\left(x^{(i)}|z\right)\right] - \boldsymbol{D}_{\boldsymbol{KL}}\left(q_{\boldsymbol{\phi}}\left(z|x^{(i)}\right)||p(z)\right)} + \boldsymbol{D}_{\boldsymbol{KL}}\left(q_{\boldsymbol{\phi}}\left(z|x^{(i)}\right)||\, p_\theta\left(z|x^{(i)}\right)\right)$$

VLB or ELBO

Intractable; KL $\geq$ 0

Another Derivation of VLB (Jensen)

$$
\begin{aligned}
\max_{\theta} \sum_{i} \log p_{\theta}\left(x^{(i)}\right) &= \max_{\theta} \sum_{i} \log\left(\int p(z) p_{\theta}\left(x^{(i)}|z\right) dz \right) \\
&= \max_{\theta} \sum_{i} \log\left(\int \frac{q_{\boldsymbol{\phi}}(z|x^{(i)})}{q_{\boldsymbol{\phi}}(z|x^{(i)})} p(z) p_{\theta}\left(x^{(i)}|z\right) dz \right) \\
&= \max_{\theta} \sum_{i} \log\left(\int q_{\boldsymbol{\phi}}\left(z|x^{(i)}\right) \frac{p(z)}{q(z|x^{(i)})} p_{\theta}\left(x^{(i)}|z\right) dz \right) \\
&= \max_{\theta} \sum_{i} \log\left(E_{z\sim q(z)} \frac{p(z)}{q_{\boldsymbol{\phi}}(z|x^{(i)})} p_{\theta}\left(x^{(i)}|z\right) \right) \\
&\geq \max_{\theta} \sum_{i} E_{z\sim q(z)} \log\left(\frac{p(z)}{q_{\boldsymbol{\phi}}(z|x^{(i)})} p_{\theta}\left(x^{(i)}|z\right) \right) \qquad \text{Jensen's inequality} \\
&= \max_{\theta} \sum_{i} E_{z\sim q(z)} \left[\log p(z) - \log q_{\boldsymbol{\phi}}\left(z\middle|x^{(i)}\right) + \log p_{\theta}\left(x^{(i)}|z\right)\right]
\end{aligned}
$$

Importance Weighted AutoEncoder (IWAE) - another lower bound

IWAE has the same architecture as the VAE, but uses a strictly tighter log-likelihood lower bound derived from importance weighting.

Increased flexibility to model complex posteriors which do not fit the VAE modeling assumptions.

IWAE learns richer latent space representations than VAEs.

[Burda et al., 2015] IWAE https://arxiv.org/abs/1509.00519

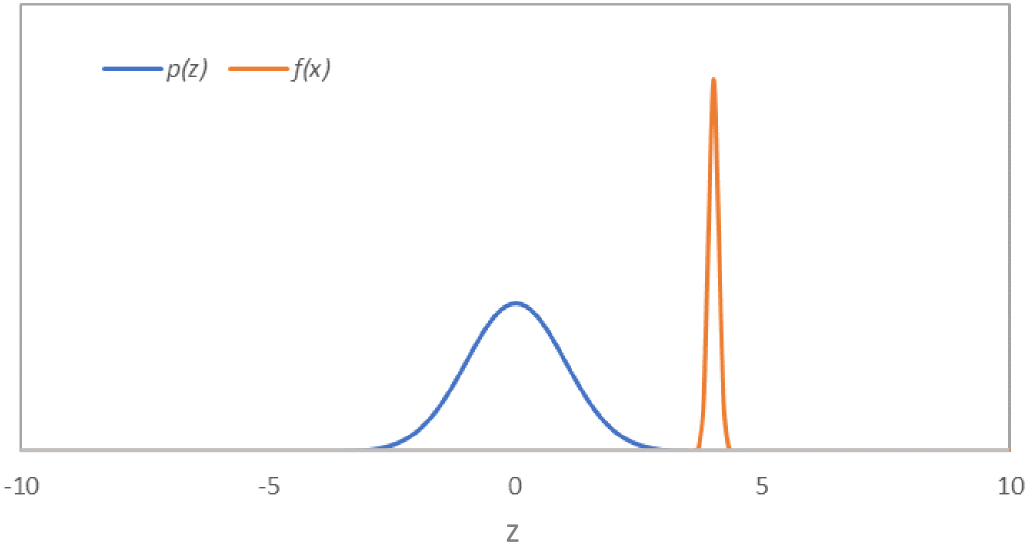
Importance Sampling

Let assume we want to compute the following expectation:

$$\boldsymbol{E}_{z\sim p(z)}f(z)$$

However, sampling from $p(z)$ is hard or not informative.

For example



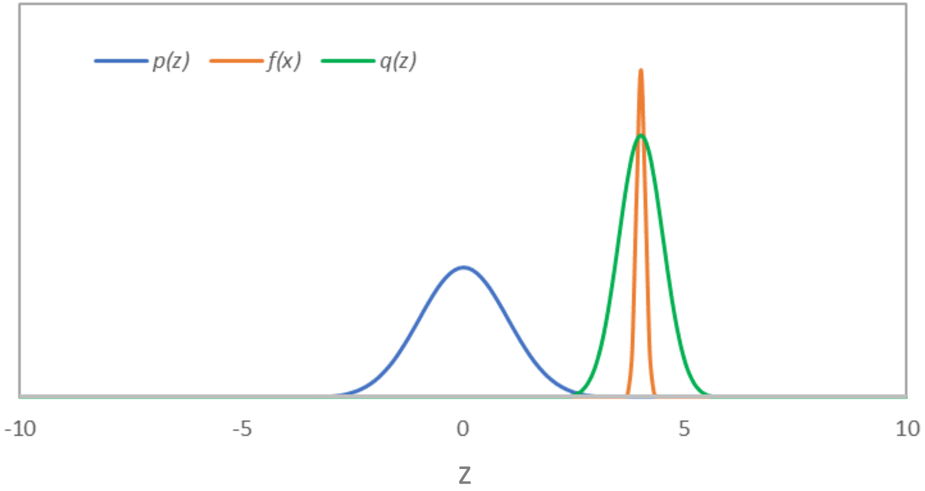
Importance Sampling

$$\boldsymbol{E}_{z\sim p(z)} f(z) = \int p(z) f(z)\, dz$$

$$= \int \frac{q(z)}{q(z)} p(z) f(z)\, dz$$

$$= \int q(z) \frac{p(z)}{q(z)} f(z)\, dz$$

$$= \boldsymbol{E}_{z\sim q(z)} \frac{p(z)}{q(z)} f(z) \qquad \approx \frac{1}{K} \sum_{k=1}^{K} \frac{p\left(z^{(k)}\right)}{q\left(z^{(k)}\right)} f\left(z^{(k)}\right) \qquad z^{(k)} \sim q(z)$$



Here, we sample from $q(z)$ to compute expectation w.r.t. $p(z)$

Importance Sampling

Recall Objective: maximize the sum of log probability as follows:

$$\sum_i \log p_\theta\left(x^{(i)}\right) = \sum_i \log \int p(z)\, p_\theta\left(x^{(i)}|z\right) dz \approx \sum_i \log \frac{1}{K} \sum_{k=1}^{K} \frac{p(z_k^{(i)})}{q(z_k^{(i)})} p_\theta\left(x^{(i)}|z_k^{(i)}\right)$$

What might be a good choice of $q(z)$?

We would like the samples from $q(z)$ to be compatible with $x^{(i)}$.

We could choose: $q(z) = p_\theta\left(z|x^{(i)}\right) = \frac{p_\theta(x^{(i)}|z)p(z)}{p_\theta\left(x^{(i)}\right)}$.

This becomes Catch-22.

Select $q(z)$

How about choosing $q(z)$ to be simple and easy to sample?

Such as, $q(z) \sim N(\mu, \sigma^2)$

But, we also want $q(z)$ to be as close as possible to $p_\theta\left(z|x^{(i)}\right) = \frac{p_\theta\left(x^{(i)}|z\right)p(z)}{p_\theta\left(x^{(i)}\right)}$

Next question is how can we make $q(z)$ to be close to $p_\theta\left(z|x^{(i)}\right)$?

We can minimize the "distance" between $q(z)$ and $p_\theta\left(z|x^{(i)}\right)$ using *KL* divergence.

$$\min_{q(z)} KL\left(q(z)||p_\theta\left(z|x^{(i)}\right)\right)$$

$$\min_{q(z)} E_{z\sim q(z)} log\left(\frac{q(z)}{p_\theta\left(z|x^{(i)}\right)}\right)$$

$$\min_{q(z)} E_{z\sim q(z)} log\left(\frac{q(z)}{p_\theta\left(x^{(i)}|z\right)p(z)/p_\theta\left(x^{(i)}\right)}\right)$$

Independent of z

$$\min_{q(z)} E_{z\sim q(z)} \log q(z) - \log p(z) - \log p_\theta\left(x^{(i)}|z\right) + \log p_\theta\left(x^{(i)}\right)$$

$$\min_{q(z)} E_{z\sim q(z)} \log q(z) - \log p(z) - \log p_\theta\left(x^{(i)}|z\right)$$

Parameterize $q(z)$

$$\min_{q(z)} KL\left(q(z)||p_\theta\left(z|x^{(i)}\right)\right)$$

Note the above minimization is for a particular $x^{(i)}$, i.e., the posterior is different for each $x^{(i)}$.

We could parametrize a neural network (inference network) to minimize *KL* for all $x^{(i)}$ (**Amortized Variational Inference**).

$$\min_{\Phi} \sum_i KL\left(q_\Phi\left(z|x^{(i)}\right)||p_\theta\left(z|x^{(i)}\right)\right)$$

$$q_\Phi(z|x) = N\left(\mu_\Phi(x), \sigma_\Phi^2(x)\right)$$

This is equivalent to $z = \mu_\Phi(x) + \varepsilon\sigma_\Phi^2(x)$ $\varepsilon \sim N(0, I)$

Reformulate the optimization problem

Recall the objective again. We aim to maximize the following:

$$\max_{\theta}\sum_{i}\log p_{\theta}\left(x^{(i)}\right)\approx\max_{\theta}\sum_{i}\log\frac{1}{K}\sum_{k=1}^{K}\frac{p_{\theta}(z_k^{(i)})}{q_{\Phi}(z_k^{(i)}|x^{(i)})}p_{\theta}\left(x^{(i)}|z_k^{(i)}\right)$$

But we also want to minimize the KL to ensure $q(z|x)$ stays close to $p(z|x)$.

$$\min_{\Phi}\sum_{i}KL\left(q_{\Phi}\left(z|x^{(i)}\right)||p_{\theta}\left(z|x^{(i)}\right)\right)$$

Combine both terms, we maximize the following:

$$\max_{\theta,\Phi}\left(\sum_{i}\log\frac{1}{K}\sum_{k=1}^{K}\frac{p_{\theta}(z_k^{(i)})}{q_{\Phi}(z_k^{(i)}|x^{(i)})}p_{\theta}\left(x^{(i)}|z_k^{(i)}\right)-\sum_{i}KL\left(q_{\Phi}\left(z|x^{(i)}\right)||p_{\theta}\left(z|x^{(i)}\right)\right)\right)$$

$$L_K \le L_{K+1} \le \log p_{\theta}(\boldsymbol{x})$$

If p(z, x)/q(z/x) is bounded, then : L_K *approach* $log\,p(\boldsymbol{x})$ *as K goes to infinity.*

[Burda et al., 2015] IWAE
https://arxiv.org/abs/1509.00519

Optimization

Likelihood Ratio Gradient

Reparameterization Trick

Likelihood Ratio Gradient

Given a generic function, $f(z)$, $z \sim q_{\boldsymbol{\phi}}(z)$.

Suppose we try to maximize the expectation of $f(z)$:

$$\max_{\boldsymbol{\phi}} \boldsymbol{E}_{z \sim q_{\boldsymbol{\phi}}(z)}[f(z)]$$

If we sample $z^{(i)} \sim q_{\boldsymbol{\phi}}(z)$ directly and then approximate the expectation by averaging, we have:

$$\max_{\boldsymbol{\phi}} \frac{1}{K} \sum_{i=1}^{K} f(z^{(i)})$$

What is the problem here?

Instead, we could follow the definition of expectation by maximizing the following integral.

$$\max_{\boldsymbol{\phi}} \int q_{\boldsymbol{\phi}}(z) f(z) dz$$

Likelihood Ratio Gradient

$$\max_{\phi} \int q_{\phi}(z) f(z) dz$$

Let denote: $L = \int q_{\phi}(z) f(z) dz$

$$\nabla_{\phi} L = \int \nabla_{\phi} q_{\phi}(z) f(z) dz$$

$$= \int \frac{q_{\phi}(z)}{q_{\phi}(z)} \nabla_{\phi} q_{\phi}(z) f(z) dz$$

$$= \int q_{\phi}(z) \frac{\nabla_{\phi} q_{\phi}(z)}{q_{\phi}(z)} f(z) dz$$

$$\nabla_{\phi} L = \int q_{\phi}(z) \nabla_{\phi} log[q_{\phi}(z)] f(z) dz$$

$$= \boldsymbol{E}_{z \sim q_{\phi}(z)}\left(\nabla_{\phi} log[q_{\phi}(z)] f(z)\right)$$

$$\approx \frac{1}{K} \sum_{i=1}^{K} \nabla_{\phi} log[q_{\phi}(z^{(i)})] f(z^{(i)})$$

reward

Note: this is similar to maximizing the reward in the Reinforcement Learning policy gradient setting.

Very noisy, high variance

Reparameterization Trick

$$\max_{\boldsymbol{\phi}} \boldsymbol{E}_{z\sim q_{\boldsymbol{\phi}}(z)}[f(z)]$$

$$q_{\boldsymbol{\phi}}(z) = N\left(\mu, \sigma^2\right)$$

$$\max_{\mu,\sigma} \boldsymbol{E}_{\varepsilon\sim N(0,1)}[f(\mu + \varepsilon\sigma)]$$

$$\boxed{z = \mu + \varepsilon\sigma \quad \varepsilon\sim N(0,1)}$$

$$\max_{\mu,\sigma} \frac{1}{K}\sum_{i=1}^{K} f\left(\mu + \varepsilon^{(i)}\sigma\right)$$

$$\nabla_{\mu,\sigma} f\left(\mu + \varepsilon^{(i)}\sigma\right)$$

VAE with Likelihood Ratio Gradient

$$\max_{\phi,\theta} \boldsymbol{E}_{z\sim q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z) - \log q_\phi(z|x^{(i)}) + \log p(z)]$$

$$VLB = \boldsymbol{E}_{z\sim q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z) + \log p(z) - \log q_\phi(z|x^{(i)})]$$

Decoder network

$$\nabla_\theta[\,VLB\,] = \nabla_\theta\left[\boldsymbol{E}_{z\sim q_\phi(z|x^{(i)})}\log p_\theta(x^{(i)}|z)\right] \approx \frac{1}{K}\sum_{k=1}^{K}\nabla_\theta log\, p_\theta(x^{(i)}|z^{(k)})$$

$$\nabla_\phi[\,VLB\,] = \nabla_\phi \int q_\phi(z|x^{(i)})[\log p_\theta(x^{(i)}|z) + \log p(z) - \log q_\phi(z|x^{(i)})]\,dz$$

$$= \int \nabla_\phi\{q_\phi(z|x^{(i)})[\log p_\theta(x^{(i)}|z) + \log p(z) - \log q_\phi(z|x^{(i)})]\}\,dz$$

$$= \int\{\nabla_\phi q_\phi(z|x^{(i)})[\log p_\theta(x^{(i)}|z) + \log p(z) - \log q_\phi(z|x^{(i)})] - q_\phi(z|x^{(i)})\,\nabla_\phi \log q_\phi(z|x^{(i)})\}\,dz$$

$$= \int \nabla_\phi q_\phi(z|x^{(i)})[\log p_\theta(x^{(i)}|z) + \log p(z) - \log q_\phi(z|x^{(i)})]dz - \int q_\phi(z|x^{(i)})\,\nabla_\phi \log q_\phi(z|x^{(i)})\,dz$$

$$= \int \nabla_\phi q_\phi(z|x^{(i)})[\log p_\theta(x^{(i)}|z) + \log p(z) - \log q_\phi(z|x^{(i)})]dz$$

$$= \int q_\phi(z|x^{(i)})\frac{\nabla_\phi q_\phi(z|x^{(i)})}{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z) + \log p(z) - \log q_\phi(z|x^{(i)})]\,dz$$

$$= \boldsymbol{E}_{z\sim q_\phi(z|x^{(i)})}\nabla_\phi \log q_\phi(z|x^{(i)})\,[log\, p_\theta(x^{(i)}|z) + log\, p(z) - log\, q_\phi(z|x^{(i)})] \;\approx \frac{1}{K}\sum_{k=1}^{K}\nabla_\phi \log q_\phi(z^{(k)}|x^{(i)})\,[log\, p_\theta(x^{(i)}|z^{(k)}) + log\, p(z^{(k)}) - log\, q_\phi(z^{(k)}|x^{(i)})]$$

$$\int q_\phi(z|x^{(i)})\,\nabla_\phi \log q_\phi(z|x^{(i)})\,dz$$

$$= \int q_\phi(z|x^{(i)})\frac{\nabla_\phi\, q_\phi(z|x^{(i)})}{q_\phi(z|x^{(i)})}dz$$

$$= \int \nabla_\phi\, q_\phi(z|x^{(i)})\,dz$$

$$= \nabla_\phi \int q_\phi(z|x^{(i)})dz$$

$$= \nabla_\phi\; 1 = 0$$

VAE with Reparameterization Trick

$$\max_{\boldsymbol{\phi},\theta} \boldsymbol{E}_{z\sim q_{\boldsymbol{\phi}}(z|x^{(i)})}\left[\log p_\theta\left(x^{(i)}|z\right) - \log q_{\boldsymbol{\phi}}\left(z|x^{(i)}\right) + \log p(z)\right]$$

$$z = \mu(x;\phi) + \varepsilon\,\Sigma^{1/2}(x;\phi) \qquad \varepsilon\sim N(0,I)$$

↓ substitute

$$\max_{\boldsymbol{\phi},\theta} \boldsymbol{E}_{\varepsilon\sim N(0,I)}\left[\log p_\theta\left(x^{(i)}|z\right) - \log q_{\boldsymbol{\phi}}\left(z|x^{(i)}\right) + \log p(z)\right]$$

Now, we can sample and compute: $\nabla_{\boldsymbol{\phi}}[VLB]$ $\nabla_\theta[VLB]$

VAE Variants

VQ-VAE 2	[Razavi et al., 2019] https://arxiv.org/abs/1906.00446
VQ-VAE	[van den Oord et al., 2017] https://arxiv.org/abs/1711.00937
beta-VAE	[Higgins et al., 2017] https://openreview.net/forum?id=Sy2fzU9gl
PixelVAE	[Gulrajani et al., 2016] https://arxiv.org/abs/1611.05013

etc.

Information Theory

Intuition: “learning that an unlikely event has occurred” is more informative than “learning that a likely event has occurred.” In other words, an unlikely event contains more information.

How to quantify the above intuition?

Define the self-information (information content) of an event X = x to be:

$$I(x) = -\log[P(x)]$$

Shannon entropy: (The amount of uncertainty in an entire probability distribution, also known as “differential entropy” when x is continuous.)

$$H(P) = E_{x \sim P}[I(x)] = E_{x \sim P}[-\log[P(x)]]$$

Entropy of a data source

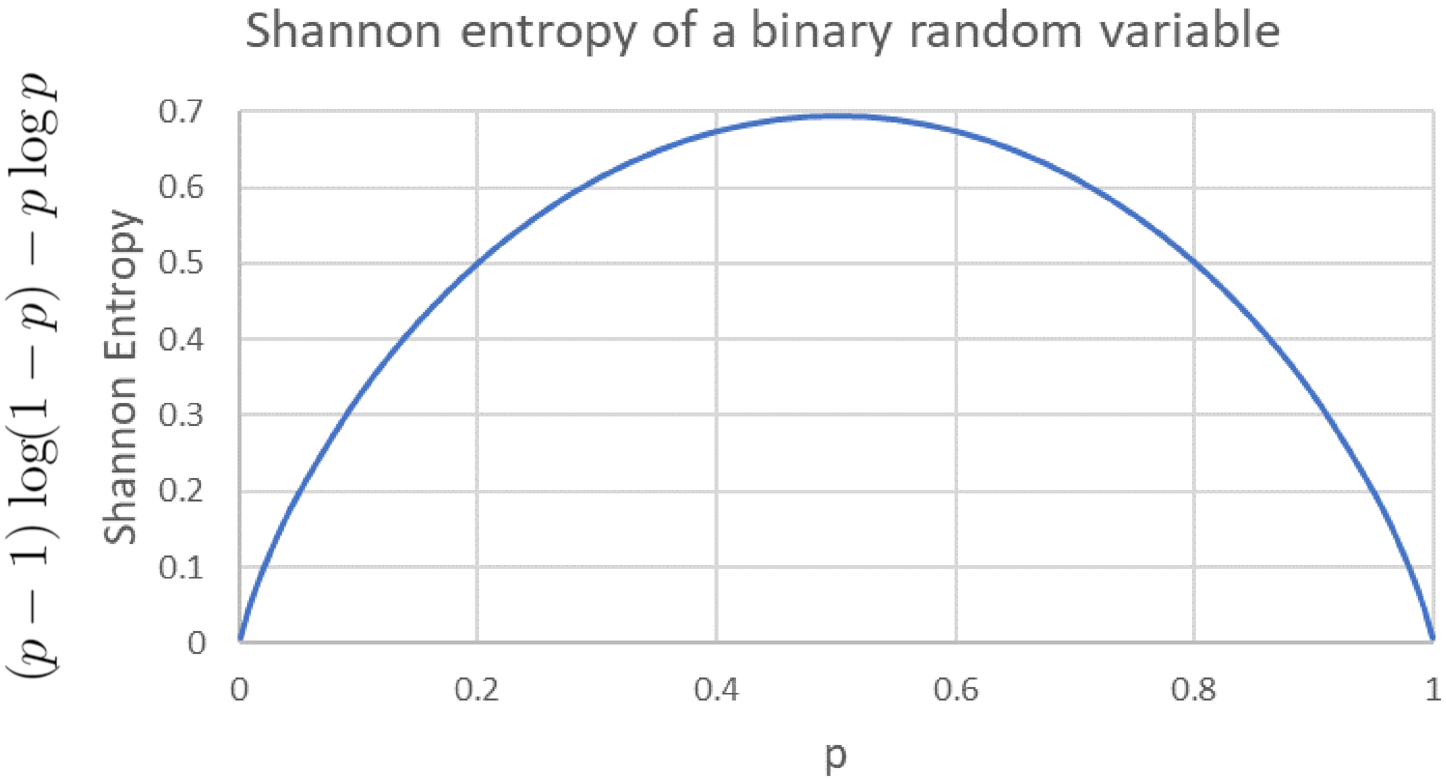
The entropy of a data source can be interpreted as the average number of bits needed to encode it, i.e., more information will need more bits to encode.

- 0 bit for a data source that emits the same signal (a certain event).
- 1 bit $\left(-log_2\left(\frac{1}{2}\right)\right)$ for flipping a coin.
- 2.58 bits $\left(-log_2\left(\frac{1}{6}\right)\right)$ for tossing a dice.

How about generating a word with a vocab = 10,000?

How about generating a RGB image of 1000x1000?

Example


Shannon entropy of a binary random variable
$(p-1)\log(1-p)-p\log p$
Shannon Entropy
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
p

Normal distribution is kind of special

Per central limit theorem, the sum of many independent random variables is approximately normally distributed.

- In practice, many complicated systems can be modeled successfully with normally distributed noise.

Out of all possible probability distributions with the same variance, the normal distribution encodes the maximum amount of uncertainty over the real numbers.

- The normal distribution inserts the least amount of prior knowledge into a model.

Kullback-Leibler (KL) Divergence

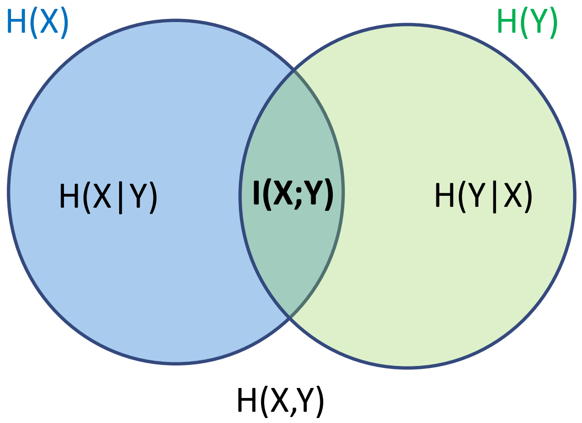
$$D_{KL}(P||Q) = E_{x\sim P}\left[log\left(\frac{P(x)}{Q(x)}\right)\right] = E_{x\sim P}[log(P(x)) - log(Q(x))]$$

- Measure how different two distributions, P(x) and Q(x), are.
- Non-negative
- Not symmetric, i.e., $D_{KL}(P||Q) \neq D_{KL}(Q||P)$ for some P and Q.
- The KL divergence is 0 if and only if P and Q are the same.

Mutual Information

Mutual information between two random variables X, Y, denoted as I(X;Y) is defined as:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$



Unlike the commonly used covariance, mutual information is a general way to measure dependency between two random variables.

Estimating mutual information between z and x in VAE

$$
\begin{aligned}
I(z;x) = H(z) - H(z|x) &= H(z) - E_{(z,x)\sim p(z,x)}[-\log p(z|x)] \\
&= H(z) + E_{(z,x)\sim p(z,x)}[\log p(z|x)] \\
&= H(z) + E_{(z,x)\sim p(z,x)}\left[\log p(z|x)\frac{\log q(z|x)}{\log q(z|x)}\right] \\
&= H(z) + E_{(z,x)\sim p(z,x)}[\log p(z|x) - \log q(z|x) + \log q(z|x)] \\
&= H(z) + E_{(z,x)\sim p(z,x)}[\log p(z|x) - \log q(z|x)] + E_{(z,x)\sim p(z,x)}[\log q(z|x)] \\
&\geq H(z) + E_{(z,x)\sim p(z,x)}[\log q(z|x)]
\end{aligned}
$$

Recall the posterior $p(z|x)$ is intractable, but can be estimated by a variational distribution $q(z|x)$.

Information encoded in z depends on expressivity of the decoder $p(x|z)$

- For simple decoder $p(x|z)$ (e.g., deterministic NN with simple distribution for individual pixels at the end), all entropy will be pushed to z.
- For a powerful decoder, such as $p(x|z) = p_{data}(x)$, z encodes no information.

What is the maximum VLB?

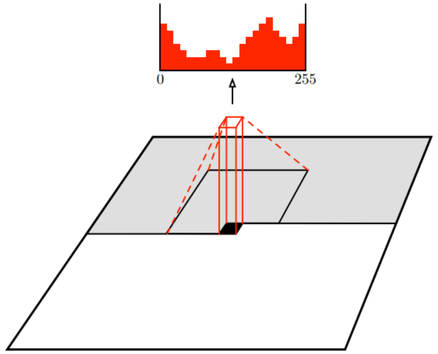
$$E_{x\sim p_{data}}[VLB] \leq E_{x\sim p_{data}}[\log p_\theta(x)] \leq E_{x\sim p_{data}}[\log p_{data}(x)]$$

If $p(x|z) = p_{data}(x)$

$$E_{x\sim p_{data}}[VLB] = E_{x\sim p_{data},\, z\sim p(z,x)}[\log p(x|z) + \log p(z) - \log q(z|x)]$$

$$= E_{x\sim p_{data}}\left[\log p_{data}(x) + E_{z\sim q(z|x)}[\log p(z) - \log q(z|x)]\right]$$

PixelCNN Autoencoder vs Autoencoder with MSE



PixelCNN

$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, ..., x_{i-1})$$

where x_i is a single pixel.

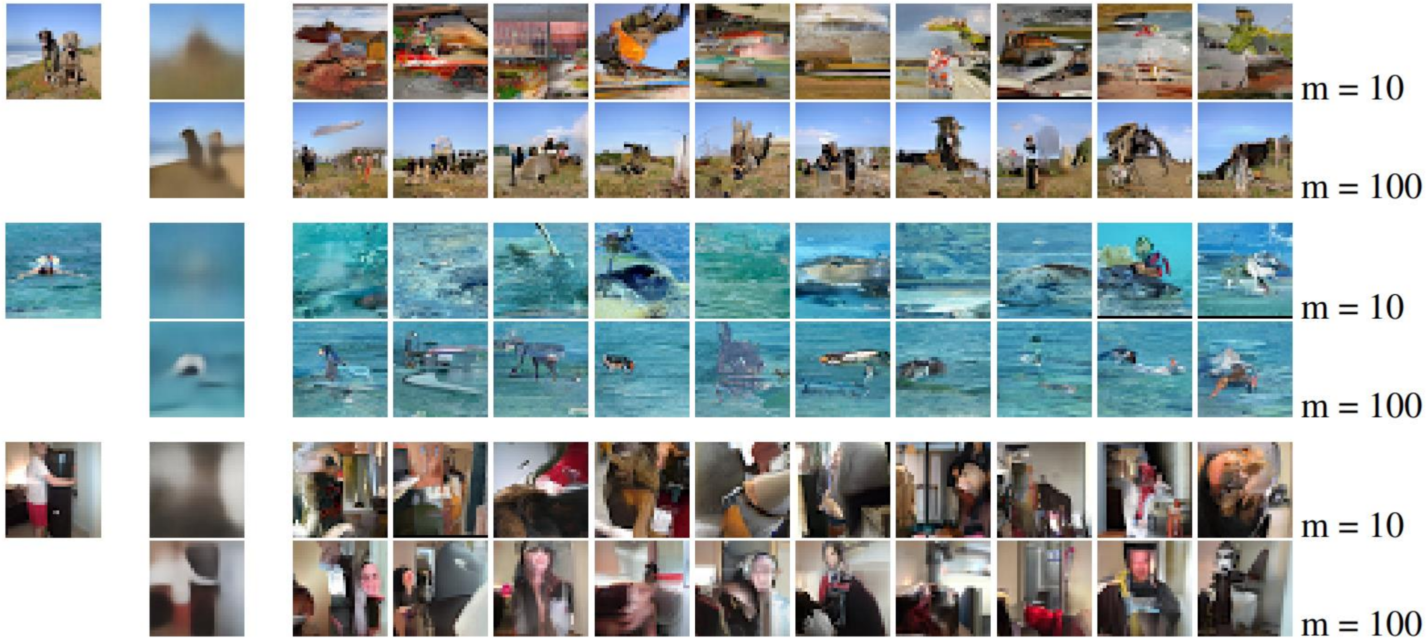


Figure 6: Left to right: original image, reconstruction by an auto-encoder trained with MSE, conditional samples from a PixelCNN auto-encoder. Both auto-encoders were trained end-to-end with a $m = 10$-dimensional bottleneck and a $m = 100$ dimensional bottleneck.

[van den Oord et al., 2016] PixelCNN https://arxiv.org/pdf/1606.05328.pdf

VQ-VAE-2

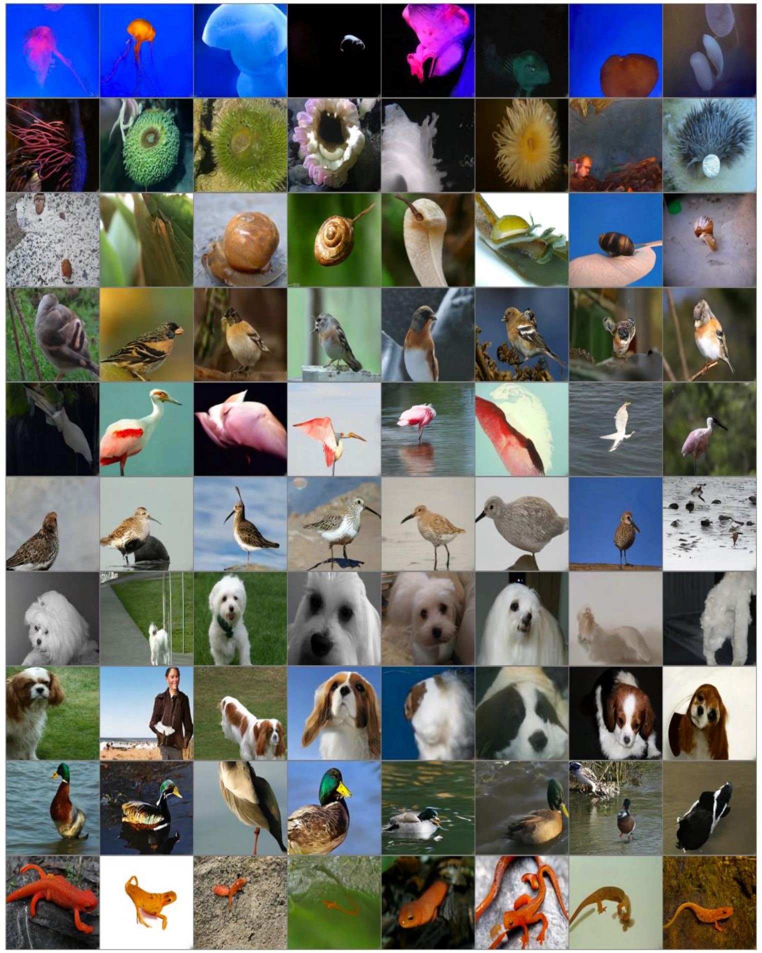


Figure 4: Class conditional random samples. Classes from the top row are: 108 sea anemone, 109 brain coral, 114 slug, 11 goldfinch, 130 flamingo, 141 redshank, 154 Pekinese, 157 papillon, 97 drake, and 28 spotted salamander.

[Razavi et al., 2019] https://arxiv.org/abs/1906.00446