Variational Autoencoder

Latent variable model

We consider the model

$$p(x, z) = p(x|z)p(z),$$

- Observed $x \in \mathcal{X}$.
- Latent $z = (z_1, z_2, \ldots, z_k) \in \mathbb{R}^k$.

For example, x is an image of a human face and z is a hidden feature, such as happy vs sad or male vs female, etc.

Learning generative models

Given a dataset $D = \{x^1, x^2, \ldots, x^n\}$. We are interested in the following inference and learning tasks:

- Learning the parameters θ of p.
- Approximate posterior inference over z: given an image x, what is $p(z|x)$?

We are also going to assume high-dimensional data i.e. computing the posterior probability $p(z \mid x)$ is intractable.

What can we try?

So far we have learned about

- EM algorithm to learn z from a given x. However...
  - E step requires computing $p(z \mid x)$ which is intractable.
  - M step requires optimization over entire dataset, which we might not have enough memory for.

What can we try?

So far we have learned about

- EM algorithm to learn z from a given x. However...
  - E step requires computing $p(z \mid x)$ which is intractable.
  - M step requires optimization over entire dataset, which we might not have enough memory for.
- Variational Inference but $z_i's$ depends on each other, so we have to compute

$$\mathbb{E}_{z_2,\ldots,z_k} \log \tilde{p}(z_1, z_2, \ldots, z_k).$$

What can we try?

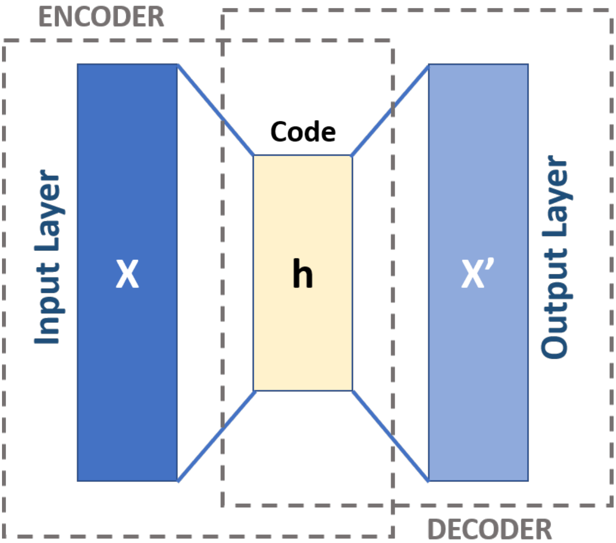
So far we have learned about

- EM algorithm to learn z from a given x. However...
  - E step requires computing $p(z \mid x)$ which is intractable.
  - M step requires optimization over entire dataset, which we might not have enough memory for.
- Variational Inference but $z_i's$ depends on each other, so we have to compute

$$\mathbb{E}_{z_2,\dots,z_k} \log \tilde{p}(z_1, z_2, \dots, z_k).$$

- MCMC does not scale well to large dataset, and MH algorithm requires a proposal distribution q which might be hard to choose.

Autoencoder



Auto-encoding variational Bayes

Recall the Evidence Lower Bound (ELBO):

$$\mathrm{ELBO}(p_\theta, q_\phi) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x,z) - \log q_\phi(z|x)\right]$$

In mean field variational inference, we assumed that

$$q_\phi(z \mid x) = q_1(z_1)q_2(z_2)\dots q_k(z_k)$$

but this might be too simple.

Auto-encoding variational Bayes

Instead, in AEVB, we assume that

$$q_\phi(z|x) = q(z|\phi(x)),$$

where q is a **base** distribution and the parater ϕ is now a function of x.

For example, if q is a Gaussian then

$$q_{\mu,\sigma^2}(z|x) = q(z|\mu(x), \sigma^2(x)).$$

Auto-encoding variational Bayes

Instead, in AEVB, we assume that

$$q_\phi(z|x) = q(z|\phi(x)),$$

where q is a **base** distribution and the parater ϕ is now a function of x.

For example, if q is a Gaussian then

$$q_{\mu,\sigma^2}(z|x) = q(z|\mu(x), \sigma^2(x)).$$

We will optimize the ELBO over ϕ. This method is called

black-box variational inference

Optimizing ELBO

We optimize ELBO with respect to ϕ and θ via **gradient descent**. Thus we need to compute the gradient

$$\nabla_{\theta,\phi}\text{ELBO} = \nabla_{\theta,\phi}\mathbb{E}_{q_\phi(z)}\left[\log p_\theta(x,z) - \log q_\phi(z)\right].$$

We can push the gradient in side the expectation and apply the chain rule

$$\nabla_{\theta,\phi}\text{ELBO} = \mathbb{E}_{q_\phi(z)} f(x,z,\theta,\phi),$$

which is again difficult to compute because of expectation. Thus we rely on Monte Carlo estimate

$$\mathbb{E}_{q_\phi(z)} f(x,z,\theta,\phi) \approx \frac{1}{N}\sum_{i=1}^{N} F(x,z_i,\theta,\phi).$$

Optimizing ELBO

However, it was shown by Mnih & Gregor (2014) that the variance of the Monte Carlo is high.

What this means is that, suppose that $\mathbb{E}f = 1$, you can sample f 100 times and get something like

$$0, 0, 0, \ldots, 100$$

The expectation is correct, but you have to sample for a long time to figure out that the true expectation is actually one.

Stochastic gradient variational Bayes

ELBO can be reformulated as

$$\begin{aligned}\mathrm{ELBO} &= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x,z) - \log q_\phi(z|x)\right] \\ &= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p(z)).\end{aligned}$$

This can be interpreted as **autoencoder**:

- Encoder $q_\phi(z|x)$ which turns x into a *code* z.
- Decoder $p_\theta(x|z)$ which tries to reconstruct x from the code z.

Stochastic gradient variational Bayes

ELBO can be reformulated as

$$\begin{aligned}\text{ELBO} &= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x,z) - \log q_\phi(z|x)\right] \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - KL(q_\phi(z|x)||p(z)).\end{aligned}$$

This can be interpreted as **autoencoder**:

- Encoder $q_\phi(z|x)$ which turns x into a *code* z.
- Decoder $p_\theta(x|z)$ which tries to reconstruct x from the code z.

Our goal is to find $q_\phi(z|x)$ that maximizes the expected reconstruction and minimizes the KL-divergence at the same time.

Expectation value = We still need Monte Carlo.
How can we reduce the variance in Monte Carlo?
Reparametrization trick Write z as

$$z = g_\phi(\epsilon, x)$$

where

$$\epsilon \sim N(0, 1).$$

*We have to make sure that $g_\phi(\epsilon, x) \sim q_\phi(z \mid x)$.

Reparametrization trick

Example: Gaussian variable $z \sim q_{\mu,\sigma^2}(z) = N(\mu, \sigma^2)$. We can write

$$z = g_{\mu,\sigma}(\epsilon) = \mu + \epsilon \cdot \sigma,$$

where $\epsilon \sim \mathcal{N}(0, 1)$.

Reparametrization trick

We may now write the gradient of the expectation as

$$\begin{aligned}\nabla_\phi \mathbb{E}_{z\sim q_\phi(z|x)}\left[f(x,z)\right] &= \nabla_\phi \mathbb{E}_{\epsilon\sim p(\epsilon)}\left[f(x,g_\phi(\epsilon,x))\right] \\ &= \mathbb{E}_{\epsilon\sim p(\epsilon)}\left[\nabla_\phi f(x,g_\phi(\epsilon,x))\right]\end{aligned}$$

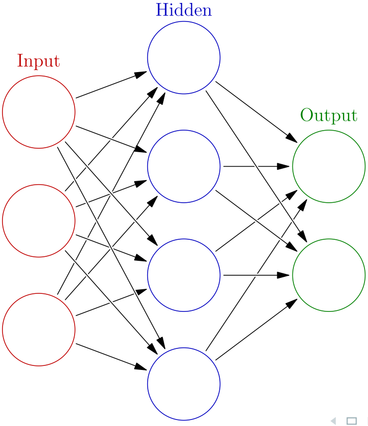
- Expectation value = We can use Monte Carlo **to sample** ϵ.
- The variance is lower than the original formulation (Rezende et al., 2014)

Choosing p and q

As mentioned before, q takes the form of

$$q_\phi(z|x) = q(z|\phi(x)),$$

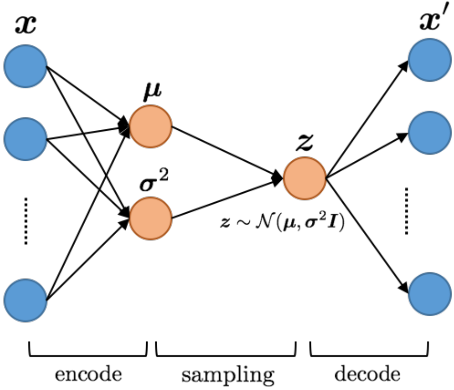
and we will take ϕ to be a neural network, which is a deterministic function of x.



Choosing p and q

For example, if the base distribution q is normal, then

$$q(z \mid x) = \mathcal{N}(z; \vec{\mu}(x), (\vec{\sigma}(x))^2).$$



What we are missing in the ELBO is $p_\theta(x|z)$.

Choosing p and q

We also model p using a neural network

$$
\begin{aligned}
p(x \mid z) &= N(x; \vec{\mu}(z), (\vec{\sigma}(z))^2) \\
p(z) &= N(z; 0, I),
\end{aligned}
$$

where $\vec{\mu}(z)$ and $\vec{\sigma}(z)$ are neural networks of z.

Summary

In summary, we want to maximize

$$\text{ELBO} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p(z))$$

using gradient descent on θ and ϕ

- Initialize all parameters and neural networks.
- Sample $\epsilon \sim N(0,1)$ for Monte Carlo estimate in order to compute the reconstruction term.
- Update $\mu, \sigma, \vec{\mu}(z)$ and $\vec{\sigma}(z)$, which contains all parameters of all neural network.
- Repeat.