Stocastic gradient-based optimization of ELBO

- Given a dataset with i.i.d. data, the ELBO objective is the sum (or average) of individual-datapoint ELBO's:

$$\mathcal{L}_{\theta,\phi}(D) = \sum_{i=1}^{n} \mathcal{L}_{\theta,\phi}(x_i) \tag{9}$$

- An important property of the ELBO, is that it allows joint optimization w.r.t. all parameters (ϕ and θ) using stochastic gradient descent (SGD).
- We can start out with random initial values of ϕ and θ and stochastically optimize their values until convergence.

ELBO gradient w.r.t to model parameters: θ

Using the expression of $\mathcal{L}_{\theta,\phi}(x)$ from (5):

$$\mathcal{L}_{\theta,\phi}(x) = \mathbf{E}_{q_\phi(z|x)}[log(p_\theta(x,z)) - log(q_\phi(z|x))]$$

Unbiased gradients of the ELBO w.r.t. the generative model parameters θ are simple to obtain:

$$\begin{aligned}
\nabla_\theta \mathcal{L}_{\theta,\phi}(x) &= \nabla_\theta \mathbf{E}_{q_\phi(z|x)}[log(p_\theta(x,z)) - log(q_\phi(z|x))] \\
&= \mathbf{E}_{q_\phi(z|x)}[\nabla_\theta(log(p_\theta(x,z)) - log(q_\phi(z|x)))] \\
&\approx \nabla_\theta(log(p_\theta(x,z)) - log(q_\phi(z|x))) \\
&= \nabla_\theta log(p_\theta(x,z))
\end{aligned}$$

Here z in the last two lines is a random sample from $q_\phi(z|x)$. Hence, $\nabla_\theta log(p_\theta(x,z))$ **is a simple Monte Carlo Estimate of** $\nabla_\theta \mathcal{L}_{\theta,\phi}(x)$. We will use this idea to train our model.

ELBO gradient w.r.t to variational parameters: ϕ

Unbiased gradients w.r.t. the variational parameters ϕ are more difficult to obtain, since the ELBO's expectation is taken w.r.t. the distribution $q_\phi(z|x)$, which is a function of ϕ. I.e., in general:

$$\begin{aligned}\nabla_\phi \mathcal{L}_{\theta,\phi}(x) &= \nabla_\phi \mathbf{E}_{q_\phi(z|x)}[log(p_\theta(x,z)) - log(q_\phi(z|x))] \\ &\neq \mathbf{E}_{q_\phi(z|x)}[\nabla_\phi(log(p_\theta(x,z)) - log(q_\phi(z|x)))]\end{aligned}$$

In the case of continuous latent variables, we can use a **reparameterization trick** for computing unbiased estimates of $\nabla_\phi \mathcal{L}_{\theta,\phi}(x)$, which we discuss now.

Reparameterization trick

For continuous latent variables and a differentiable encoder/inference and decoder/generative model, the ELBO can be straightforwardly differentiated w.r.t. both ϕ and θ through a change of variables, also called the reparameterization trick.

Change of variables:

- First we express the random variable $z \sim q_\phi(z|x)$ as some differentiable and invertible transformation of another random variable ϵ

$$z = g(\epsilon, \phi, x) \tag{10}$$

  where the distribution of random variable ϵ is independent of x or ϕ.
- Hence now we have: $\mathbf{E}_{q_\phi(z|x)}[f(z)] = \mathbf{E}_{p(\epsilon)}[f(z)]$

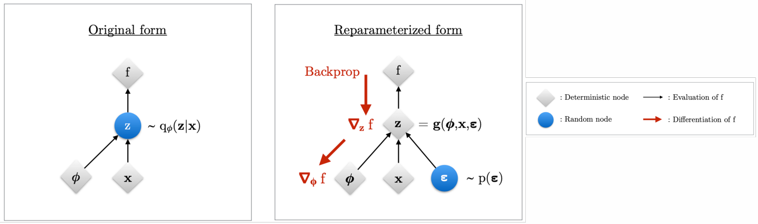
ELBO gradient w.r.t to ϕ with change of variables

$$
\begin{aligned}
\nabla_\phi \mathcal{L}_{\theta,\phi}(x) &= \nabla_\phi \mathbf{E}_{q_\phi(z|x)}[log(p_\theta(x,z)) - log(q_\phi(z|x))] \\
&= \nabla_\phi \mathbf{E}_{p(\epsilon)}[log(p_\theta(x,z)) - log(q_\phi(z|x))] \\
&= \mathbf{E}_{p(\epsilon)}[\nabla_\phi(log(p_\theta(x,z)) - log(q_\phi(z|x)))] \\
&\approx \nabla_\phi(log(p_\theta(x,z)) - log(q_\phi(z|x))) \\
&= -\nabla_\phi log(q_\phi(z|x))
\end{aligned}
$$

Here

- The second equation comes from change of variables $z = g(\phi, x, \epsilon)$ with random sampling noise $\epsilon \sim p(\epsilon)$.
- Because ϕ and ϵ are independent, so $\mathbf{E}_{p(\epsilon)}[\cdot]$ and $\nabla_\phi[\cdot]$ operators behave in a commutative way in third equation.
- In fourth equation we we take a monte carlo estimate of expectation by taking a single sample of ϵ and thus obtaining a single sample of $z = g(\phi, x, \epsilon)$.
- Last equation shows the final unbiased estimate of: $\nabla_\phi \mathcal{L}_{\theta,\phi}(x)$.

Why we needed reparameterization trick



- The variational parameters ϕ affect the objective f through the random variable $z \sim q_\phi(z|x)$.
- We wish to compute gradients $\nabla_\phi f$ to optimize the objective with SGD. In the original form (left), we cannot differentiate f w.r.t. ϕ, because we cannot directly backpropagate gradients through the random variable z.
- We can 'externalize' the randomness in z by re-parameterizing the variable as a deterministic and differentiable function of ϕ, x, and a newly introduced random variable ϵ. **This allows us to backprop through z, and compute gradients $\nabla_\phi f$.**

What to do next

Where we are:

- We have now established the idea of an encoder/inference model: $q_\phi(z|x)$ and a decoder/generative model: $p_\theta(x|z)$ along with the latent variable distribution $p_\theta(z)$.
- We know we need reparameterization to be able to update variational parameters ϕ in $q_\phi(z|x)$.
- We know if we find optimal values of θ and ϕ that maximizes ELBO, then:
  - We are approximately maximizing marginal likelihood of data: $p_\theta(x)$.
  - We are minimizing the distribution distance (KL) between our proposed inference model $q_\phi(z|x)$ and the true inference model $p_\theta(z|x)$

What we need now:

- Formally define what $q_\phi(z|x)$, $p_\theta(x|z)$ and $p_\theta(z)$ distributions looks like. These distributions will completely define the flexibility of the overall model.
- How to compute ELBO for the chosen distributions.
- **These choices will lead to multiple types of variational autoencoders.**

Variational Autoencoder (VAE) 03/21/2023

- Likelihood

$$\max_{\mu,\sigma^2} \prod_{i=1}^{100} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{(x_i-\mu)^2}{2\sigma^2}\right)}$$ optimization problem

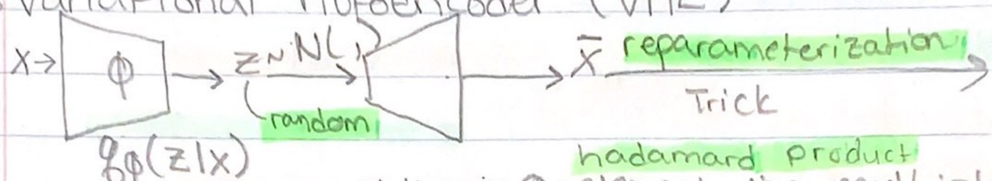
- ELBO gradient

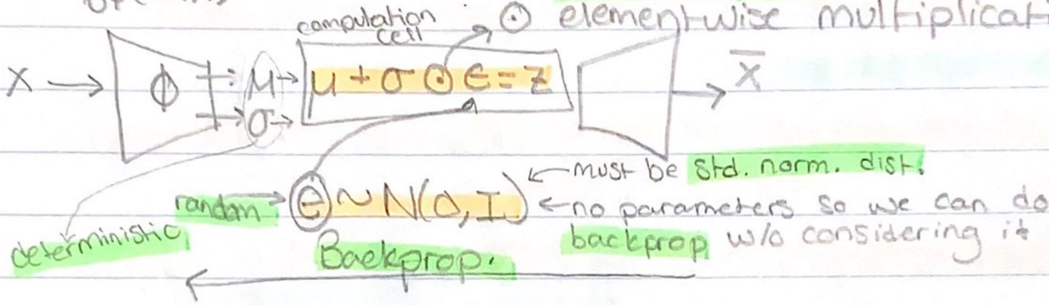
$$\mathcal{L}_{\theta,\phi}(D) = \sum_{i=1}^{n} \mathcal{L}_{\theta,\phi}(x_i)$$

$$\mathcal{L}_{\theta,\phi} = E_{q_\phi(z|x)}\left[\log(p_\theta(x,z)) - \log(q_\phi(z|x))\right]$$

$$\begin{aligned}\nabla_\theta \mathcal{L}_{\theta,\phi} &= \nabla_\theta E_{q_\phi(z|x)}\left[\log(p_\theta(x,z)) - \log(q_\phi(z|x))\right] \\ &= E_{q_\phi(z|x)}\left[\nabla_\theta(\log(p_\theta(x,z)) - \log(q_\phi(z|x)))\right] \\ &\approx \nabla_\theta(\log(p_\theta(x,z)) - \log(q_\phi(z|x))) \\ &= \nabla_\theta(\log(p_\theta(x,z)))\end{aligned}$$

03/23/23 Variational Autoencoder (VAE)





Side note: $\epsilon \sim N(0, I)$

$z = \mu + \sigma \odot \epsilon$ hadamard product

$z \sim N(\mu, \sigma^2 I)$

extra side note:

$x \sim N(\mu, \Sigma)$

$Ax + b \sim N(A\mu, A\Sigma A^T)$

$Ax + b \sim N(A\mu + b, A\Sigma A^T)$

$x = 3\times1 \Rightarrow \mu = 3\times1$

$A = 10\times3 \Rightarrow \Sigma = 3\times3$

$A\mu = 10\times1$

$A\Sigma A^T = (10\times3)\times(3\times3)(3\times10) = 10\times10$

- now we need $q_\phi(z|x)$, $p_\theta(z)$, and $p_\theta(x|z)$

VAE w/ Factorized Gaussian Posteriors 03/23/2023

- need to fix
  - $p_\theta(z)$, $q_\phi(z|x)$, $p_\theta(x|z)$
- we assume
  - $\text{EncoderNeuralNet}_\phi(x) \to (\mu, \log\sigma)$

    because we want positive values

    we calculate σ from here

$p_\theta(z) \to N(0, I)$

$q_\phi(z|x) \to$ neural network $\to N(\mu, \sigma^2 I)$

$p_\theta(x|z) \to$ neural network

- Kullback-Leibler (KL) distance/divergence
  - for 2 Gaussian Distributions

$$E_{p(x)}\left[\log\frac{P(x)}{Q(x)}\right] = \frac{1}{2}\left[\log\frac{|\Sigma_2|}{|\Sigma_1|} - d + \mathrm{Tr}(\Sigma_2^{-1}\Sigma_1) + (\mu_2-\mu_1)^T \Sigma^{-1}(\mu_2-\mu_1)\right]$$

$$P(x) = q_\phi(z|x) = \mu_1 = \mu, \; \Sigma_1 = \Sigma = \sigma^2 I$$