Bayesian Neural Net.

1) W : no longer deterministic. $\{\mu_w, \Sigma_w\}$ ss
2) Given $D = \{x^i, t^i\}_{i=1}^N$

3) Likelihood

$$p(D|w,\beta) \propto \exp(-\beta E_D(w)) \quad , \text{ where } E_D(w) = \frac{1}{2}\sum (f(x^i,w) - t^i)^2$$

4) Prior

$$p(w|A) \propto \exp(-E_W(w)) \quad , \text{ where } E_W(w) = \frac{1}{2} w^T A w$$

5) Posterior

$$p(w|D,A,\beta) \propto \exp(-\beta E_D(w) - E_W(w))$$

6) Evaluation at x_*

$$\langle f(x_*) \rangle = \int_w f(x_*, w)\, p(w|D,A,\beta)\, dw$$

7)
- Laplace's Approx.
- MCMC
- Ensemble Learning ⇐ Variation Inference.
  Introduce $Q(w)$ that approximates $p(w|\cdot)$

$$\ln p(D|A,\beta) = \ln \int p(D|w,A,\beta)\, p(w|A)\, dw$$

$$= \ln \int \left(\frac{p(D|w,A,\beta)\, p(w|A)}{Q(w)} \right) Q(w)\, dw$$

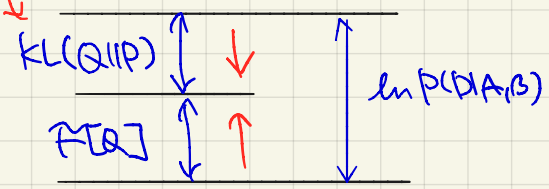
$$\varphi(\mathbb{E}[x]) \le \mathbb{E}[\varphi(x)]$$

$$\geq \int \ln \left(\frac{p(D|w,A,\beta)\, p(w|A)}{Q(w)} \right) Q(w)\, dw \quad (*)$$

$$= \mathcal{F}[Q]$$

$\mathcal{F}[Q]$ ← maximize $Q(w)$

set $P(w|\cdot)$ ← $x \in W$



8) $\ln p(D) = \int \ln p(D)\, q(w|\theta)\, dw$

$= \int q(w|\theta) \ln \frac{p(D)\, p(w|D)}{p(w|D)}\, dw$

$= \int q(w|\theta) \ln \frac{p(w)\, p(D|w)}{p(w|D)}\, dw$

$= \int q(w|\theta) \ln \frac{q(w|\theta)\, p(w)\, p(D|w)}{q(w|\theta)\, p(w|D)}\, dw$

$= \int q(w|\theta) \ln \frac{q(w|\theta)}{p(w|D)}\, dw + \int q(w|\theta) \ln \frac{p(D|w)\, p(w)}{q(w|\theta)}\, dw$

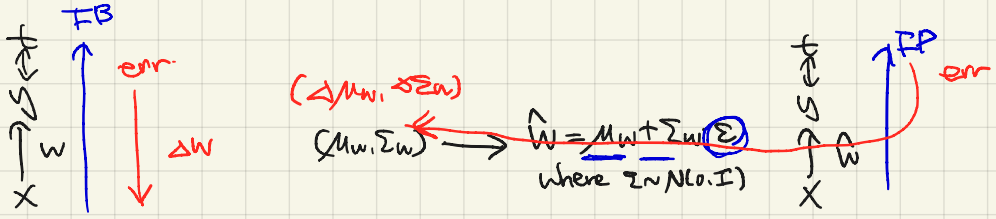
$= \underbrace{D_{KL}(q(w|\theta) \,||\, p(w|D))}_{\text{KL btw/ } q(\cdot) \text{ \& posterior.} \downarrow} + \underbrace{\mathcal{F}[q]}_{\text{ELBO}} \uparrow$

9) $\mathcal{F}[q] = \int q(w|\theta) \ln \frac{p(D|w)\, p(w)}{q(w|\theta)}\, dw$

$= \int \left(\ln p(D|w) + \ln \frac{p(w)}{q(w|\theta)} \right) q(w|\theta)\, dw$

$= \underbrace{\mathbb{E}_{w\sim q(\cdot)}[\ln p(D|w)]}_{\text{Expectation of Log lik. under } q_i \uparrow} - \underbrace{D_{KL}(q(w|\theta) \,||\, p(w))}_{\text{prior fitting term}, \downarrow}$

10) $\ln p(D) = D_{KL}(q(w|\theta) \,||\, p(w|D)) \downarrow + \mathcal{F}[q] \uparrow$



11) Weight Uncertainty in Neural Networks. (Being Bayesian by Backpropagate) (BBB)

$\mathcal{F}[q] = \int q(w|\theta) \ln \frac{p(D|w)\, p(w)}{q(w|\theta)}\, dw$

$= \mathbb{E}_q [\ln p(D|w) + \ln p(w) - \ln q(w|\theta)]$

$\simeq \left(\ln p(D|\tilde{w}) + \ln p(\tilde{w}) - \ln q(\tilde{w}|\theta) \right), \quad \tilde{w} \sim q(w|\theta)$

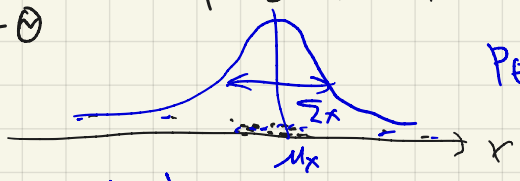
log lik, log prior, $\subset$ tractable.

Summary: $x \xrightarrow{w} y$, $p(w|D) \rightsquigarrow q_\phi(w) = N(w|\underbrace{\mu_w, \Sigma_w}_{\text{optimize}})$

(2) Deep Latent Variable Model (DLVM), e.g., VAE

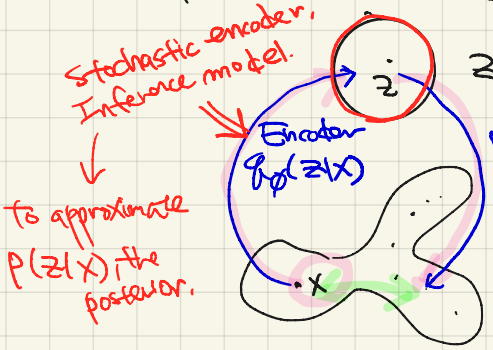
- $D = \{x_i\}_{i=1}^N \longrightarrow p_\theta(x)$ to sample $\tilde{x} \sim p_\theta(x)$
- How? $\theta \leftarrow \underset{\theta \in \Theta}{\text{argmax}} \ln p(D|\theta)$.

example)



$p_\theta(x) = N(x; \mu_x, \Sigma_x)$
$\theta = \{\mu_x, \Sigma_x\}$.

- $p(z)$: z. $z \rightsquigarrow x$: $\boxed{p_\theta(x|z)}$ (decoder), $p_\theta(x) = \int p_\theta(x|z)p(z)\,dz$.



z-space (latent space)
$p(z)$ prior.

x-Space.

Goal is to find $q_\phi(z|x)$ that approximates $p(z|x)$.

Similarly, $\ln p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}[\ln p_\theta(x)]$

$= \mathbb{E}_{q_\phi(z|x)}\left[\ln \frac{p_\theta(x,z)}{p_\theta(z|x)}\right]$

$= \mathbb{E}_{q_\phi(z|x)}\left[\ln \frac{p_\theta(x,z)}{q_\phi(z|x)} \frac{q_\phi(z|x)}{p_\theta(z|x)}\right]$

$= \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\ln \frac{p_\theta(x,z)}{q_\phi(z|x)}\right]}_{\text{ELBO}} + \underbrace{\mathbb{E}_{q}\left[\ln \frac{q_\phi(z|x)}{p_\theta(z|x)}\right]}_{D_{KL}(q_\phi(z|x) \,\|\, p_\theta(z|x))}$

$D_{KL}(q_\phi(z|x) \,\|\, p_\theta(z|x))$: encoder, posterior. ↑ Intractable.

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

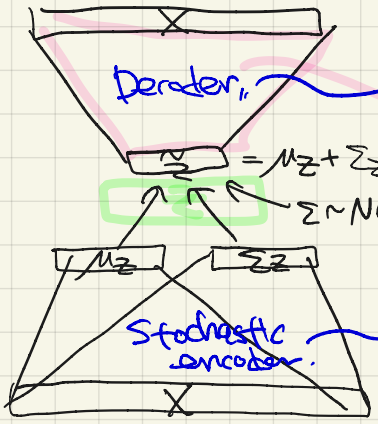
$= \int \ln \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)} q_\phi(z|x)\,dz$

VAE.

$= \underbrace{\mathbb{E}_{q_\phi(z|x)}[p_\theta(x|z)]}_{\text{reconstruction term}} - D_{KL}(q_\phi(z|x) \,\|\, p(z))$ ← prior fitting.

$x \xrightarrow[\text{encoder}]{q_\phi(z|x)} z \xrightarrow[\text{decoder}]{p_\theta(x|z)} \tilde{x}$

(3)



$p_\theta(x|z) = N(x \mid \text{Dec}_\theta(z), I)$

$z \in \mathbb{R}^D$

$q_\phi(z|x) = N(z; \mu_z(x), \Sigma_z(x))$

ϕ $\quad$ θ

$x \in \mathbb{R}^{784}$

$$KL(q_\phi(z|x) \,||\, N(0,I)) = \frac{1}{2} \sum_{i=1}^{D} (\sigma_{z_i}^2 + \mu_{z_i}^2 - \ln(\sigma_{z_i}^2) - 1)$$

network output of stochastic encoder.

Big Limitation!! $p(z)$.

	Given	Prior	variational distribution (goal)
1) BNN:	$D = \{x_i, y_i\}_{i=1}^N$	$p(w)$	$Q(w)$
2) VAE:	$D = \{x_i\}_{i=1}^N$	$p(z)$	$q_\phi(z\|x)$ & $p_\theta(x\|z)$

Objectives of VAE: 1) Reconstruction
2) KLD (prior fitting)

$$KL(q_\phi(z|x) \,\|\, N(0,I)) = \frac{1}{2}\sum_{i=1}^{D}\left(\sigma_{z_i}^2 + \mu_{z_i}^2 - \ln(\sigma_{z_i}^2) - 1\right) \quad -(*)$$

* outputs of neural networks.

WAE$_{II}$ → (WAE-GAN)

* GAN: Generative Adversarial Net.

Optimal transport and its dual formulation,

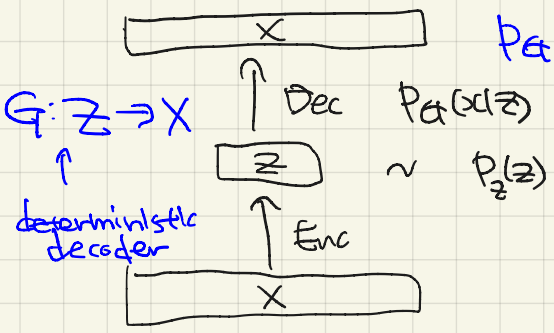
$$W_c(P_X, P_G) = \inf_{\Gamma \in \mathcal{P}(X\sim P_X, Y\sim P_G)} \mathbb{E}_{(X,Y)\sim\Gamma}[c(X,Y)]$$

W_c: cost, P_X: data dist., P_G: Generator dist., $c(X,Y)$: cost

If $c(x,y) = d(x,y)$, then the following Kantorovich-Rubinstein duality hold.

$$W_1(P_X, P_G) = \sup_{f\in F_L} \left(\mathbb{E}_{X\sim P_X}[f(x)] - \mathbb{E}_{Y\sim P_G}[f(y)]\right) \quad -(**)$$

F_L: 1-Lipschitz.



$$P_G(x) = \int_Z P_G(x|z)\, P_Z(z)\, dz$$

Thm1. For P_G as defined above with deterministic $P_G(X|Z)$ and any function $G: Z \to X$

$$\inf_{\Gamma\in\mathcal{P}(X\sim P_X, Y\sim P_G)} \mathbb{E}_{(X,Y)\sim\Gamma}[c(X,Y)] \Leftarrow \text{W.D.}$$

$$= \inf_{Q: Q_Z = P_Z} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}[c(X, G(Z))] \quad (***)$$

P_X: data, $Q(Z|X)$: encoder, $c(X, G(Z))$: decoder, Reconstruction, $Q_Z = P_Z$: 잠재공간의 prior fitting

where Q_Z is the marginal distribution of Z when $X\sim P_X$ and $Z\sim Q(Z|X)$.

From (***),

Recon.

$$D_{WAE}(P_X, P_G) = \inf_{Q(Z|X)\in\mathcal{Q}} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}\left[c(X, G(Z))\right] + \lambda D_Z(Q_Z, P_Z)$$

prior fitting term, 다양한 divergence.

⇓ GAN-based.

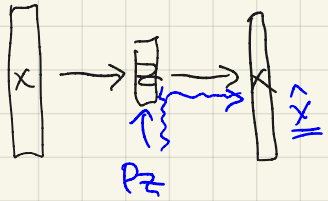
$D_Z(Q_Z, P_Z) = D_{JS}(Q_Z, P_Z)$ ← adversarial training.

*GAN

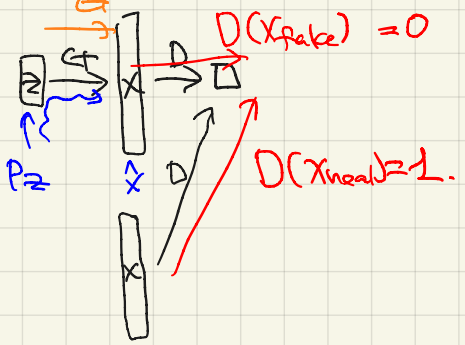
$$\min_G \max_D \left\{ \log(D(x)) + \log(1 - D(G(z))) \right\} \quad \text{— (1)}$$

where $D: x \to \mathbb{R}$, $G: Z \to X$.

VAE:



GAN:



i) Fix G, optimize D,

$$\min_G \max_D \left\{ \left[\mathbb{E}_{x\sim P_X}[\log(D(x))] + \mathbb{E}_{z\sim P_Z}[\log(1 - D(G(z)))] \right\}$$

$$= \min_G \max_D V(G, D)$$

Since G is fixed, let $y = G(z)$

$$\Rightarrow \mathbb{E}_{z\sim P_Z}[\log(1 - D(G(z)))] = \mathbb{E}_{y\sim P_y}[\log(1 - D(y))]$$

$$\downarrow$$

$$V(G,D) = \mathbb{E}_{x\sim P_X}[\log(D(x))] + \mathbb{E}_{y\sim P_y}[\log(1 - D(y))]$$

$$= \int_x P_{data}(x) \log D(x)\, dx + \int_y P_g(y) \log(1 - D(y))\, dy$$

$$= \int_x P_{data}(x) \log D(x) + P_g(x) \log(1 - D(x))\, dx$$

$$D_G^*(x) = \arg\max_D V(G,D) = \frac{P_{data}(x)}{P_{data}(x) + P_G(x)}.$$

$\left(\because\ y \to a\log(y) + b\log(1-y)\ \text{achieves its maximum in } [0,1] \text{ at } \frac{a}{a+b}\right)$

ii) Then, fix $D = D^*_G$.

$$V(G, D^*) = \mathbb{E}_{x\sim p_{data}}[\log(D^*(x))] + \mathbb{E}_{x\sim p_g(x)}[\log(1-D^*(x))]$$

$$C(G) = \mathbb{E}_{x\sim p_{data}}\left[\log\left(\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right)\right] + \mathbb{E}_{x\sim p_g(x)}\left[\log\left(\frac{p_g(x)}{p_{data}(x)+p_g(x)}\right)\right]$$

$$= -\log(4) + KL\left(p_{data} \,\|\, \frac{p_{data}+p_g}{2}\right) + KL\left(p_g \,\|\, \frac{p_{data}+p_g}{2}\right)$$

$$= -\log(4) + 2\cdot JSD(p_{data} \,\|\, p_g)$$

$\therefore$ GAN objective minimizes JSD between p_{data} and p_g.