CS109B Advanced Section : A Tour of Variational Inference

Professor : Pavlos Protopapas, TF : Srivatsan Srinivasan

CS109B, IACS

April 10, 2019

Information Theory

Information Theory

How much information can be communicated between any two components of any system ?

QUESTION : Assume you have N forks (left or right) on road. An oracle tells you which paths you take to reach a final destination. How many prompts do you need ?

SHANNON INFORMATION (SI) : Consider a coin which lands heads 90% times. What is the surprise when you see its outcome?

SI Quantifies surprise of information - $SI = -\log_2 p(x_h)$

Entropy

Assume I transmit 1000 bits (0s and 1s) of information from A to B. What is the quantum of information that has been transmitted ?

- When all the bits are known ? (0 shannons)
- When each bit is i.i.d. and equally distributed (P(0) = P(1) =0.5) i.e. all messages are equi-probable? (1000 shannons)
- Entropy defines a general uncertainty measure over this information. When is it maximized ?

$$H(X) = -\mathbb{E}_X \log p(x) = -\sum_x p(x) \log p(x) \quad \text{or} \quad -\int_x p(x) \log p(x) dx \tag{1}$$

EXERCISE : Calculate entropy of a dice roll.

REMEMBER THIS ? $-p(x) \log p(x) - (1 - p(x)) \log p(x)$

Joint and Conditional Entropy

- Joint Entropy - Entropy of joint distribution

$$H^{joint}(X,Y) = -\mathbb{E}_{X,Y} \log p(X,Y) = -\sum_{x,y} p(x,y) \log p(x,y) \tag{2}$$

- Conditional Entropy - Conditional Uncertainty of X given Y

$$\begin{aligned} H(X|Y) &= -\mathbb{E}_Y\, H(X|Y=y) \\ &= -\sum_y p(y) \sum_x p(x|y) \log p(x|y) \\ &= -\sum_{x,y} p(x,y) \log p(x|y) \end{aligned} \tag{3}$$

$$H(X|Y) = H(X,Y) - H(Y)$$

Mutual Information

Pointwise Mutual Information - Between two events, the discrepancy between joint likelihood and independent joint likelihood

$$pmi(x, y) = \log \frac{p(x, y)}{p(x)p(y)} \tag{4}$$

Mutual Information - Expected amount of information that can be obtained about one random variable by observing another.

$$\begin{aligned} I(X;Y) &= \mathbb{E}_{x,y}\, pmi(x, y) = \mathbb{E}_{x,y} \log \frac{p(x, y)}{p(x)p(y)} \\ I(X;Y) &= I(Y;X) \quad \text{(symmetric)} \\ &= H(X) - H(X|Y) = H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X, Y) \end{aligned} \tag{5}$$

Cross Entropy

Average number of bits needed to identify an event drawn from p when a coding scheme used is for optimizing a different distribution q.

$$H(p,q) = \mathbb{E}_p - \log(q) = \sum_x -p(x) \log q(x) \tag{6}$$

Example : Take any code over which you communicate a equiprobable number between 1 and 8 (true). But your receiver uses a different code scheme and hence needs a longer message length to get the message.

REMEMBER ? $y \log \hat{y} + (1-y) \log(1-\hat{y})$

Understanding cross entropy

- Game 1 : 4 coins of different color each(blue, yellow, red, green) - probability each 0.25. Ask me yes/no questions to figure out the answer.
  - Q1 : Is it green or blue ?
  - Q2 : Yes : Is it green? No : Is it red ?
  - Expected number of questions 2 H(P)
- Game 2 : 4 coins of different color each - probability each [0.5 -blue, 0.125-red, 0.125-green, 0.25-yellow]. Ask me yes/no questions to figure out the answer.
  - Q1 : Is it blue ?
  - Q2 : Yes : over, No : Is it red ?
  - Q3 : Yes : over, No : Is it yellow ?
  - Expected number of questions 1.75. H(Q)
- Game 3 : Use strategy used in game 1 on game 2 and the expected number of questions is $2 > 1.75$. H(Q,P)

KL Divergence

Measure of Discrepancy between two probability distributions.

$$D_{KL}(p(X)||q(X)) = -\mathbb{E}_P \log \frac{q(X)}{p(X)}$$
$$= -\sum_x p(x) \log \frac{q(x)}{p(x)} \quad \text{or} \quad -\int_x p(x) \log \frac{q(x)}{p(x)} dx \tag{7}$$

$$D_{KL}(P||Q) = H(P,Q) - H(P) \geq 0 \tag{8}$$

Remember entropy of P quantifies the least possible message length for encoding information from P.

KL - Extra message-length per datum that must be communicated if a code that is optimal for a given (wrong) distribution Q is used, compared to using a code based on the true distribution P.

Variational Inference

Latent Variable Inference

- Latent Variables - Random variables which are not observed.
- Example - Data of Children's score on an exam - Latent Variable : Intelligence of a child
- Example

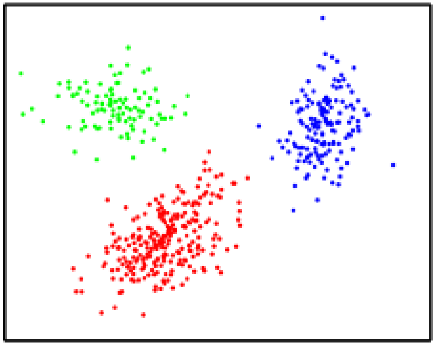


Figure 1: Mixture of cluster centers

- Break down :

$$p(x,z) = \underbrace{p(z)}_{\text{latent}} p(x|z) = p(z|x)p(x); \quad p(x) = \int_z p(x,z)dz$$

Latent Variable Inference

- Assuming a prior on z since it is under our control.
- **INFERENCE :** Learn posterior of the latent distribution - $p(z|x)$. How does our belief about the latent variable change after observing data ?

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \underbrace{\frac{p(x|z)p(z)}{\sum_z p(x|z)p(z)}}_{\text{Could be intractable}} \tag{9}$$

Variational Inference - Central Idea

Minimize $KL(q(z)||p(z|x))$

$$q^*(\mathbf{z}) = \arg\min_{q \sim \mathcal{Q}} \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) \tag{10}$$

$$\begin{aligned}\mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_{\mathbf{z}\sim q} \log q(\mathbf{z}) - \mathbb{E}_{\mathbf{z}\sim q} \log p(\mathbf{z}|\mathbf{x}) \\ &= \underbrace{\mathbb{E}_{\mathbf{z}\sim q} \log q(\mathbf{z}) - \mathbb{E}_{\mathbf{z}\sim q} \log p(\mathbf{z},\mathbf{x})}_{\text{(a) --- -1*ELBO}} + \underbrace{\log p(\mathbf{x})}_{\text{(b)}} \\ &= -\mathrm{ELBO}(q) + \underbrace{\log p(\mathbf{x})}_{\text{Does not depend on z}}\end{aligned} \tag{11}$$

Idea

Minimizing $KL(q(z)||p(z|x))$ = Maximizing ELBO !

ELBO

$$
\begin{aligned}
\mathrm{ELBO}(p,q) &= \mathbb{E}_q \log p(\mathbf{z},\mathbf{x}) - \mathbb{E}_q \log q(\mathbf{z}) \\
&= \mathbb{E}_q \log p(\mathbf{z}) + \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}) - \mathbb{E}_q \log q(\mathbf{z}) \qquad (12) \\
&= \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}) - \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}))
\end{aligned}
$$

Idea

$\mathbb{E}_q \log p(\boldsymbol{z},\boldsymbol{x}) - \mathbb{E}_q \log q(\boldsymbol{z})$- *Energy encourages q to focus probability mass where the joint mass is, $p(\mathbf{x},\mathbf{z})$. The entropy encourages q to spread probability mass and avoid concentration to one location.*

Idea

ELBO Term $\mathbb{E}_q \log p(\boldsymbol{x}|\boldsymbol{z}) - KL(q(\boldsymbol{z})||p(\boldsymbol{z})$- Conditional Likelihood Term and KL Term. Trade-off between maximizing the conditional likelihood and not deviating from the true latent distribution (prior).

Variational Parameters

- Parametrize q(z) using variational parameters λ - $q(z;\lambda)$
- Learn variational parameters during training (using some gradient based optimization for example)
- Example - $q(z;\lambda=[\mu,\sigma]) \sim \mathcal{N}(\mu,\sigma)$. Here μ,σ are variational parameters $\lambda=[\mu,\sigma]$.
- $ELBO(\lambda) = \mathbb{E}_{q(z;\lambda)} \log p(\mathbf{x}|\mathbf{z}) - \mathrm{KL}(q(\mathbf{z};\lambda)||p(\mathbf{z}))$
- Gradients :
  $\nabla_\lambda ELBO(\lambda) = \nabla_\lambda \left[\mathbb{E}_{q(z;\lambda)} \log p(\mathbf{x}|\mathbf{z}) - \mathrm{KL}(q(\mathbf{z};\lambda)||p(\mathbf{z}))\right]$
- Not directly differentiable via backpropagation : WHY ?

VI Gradients and Reparametrization

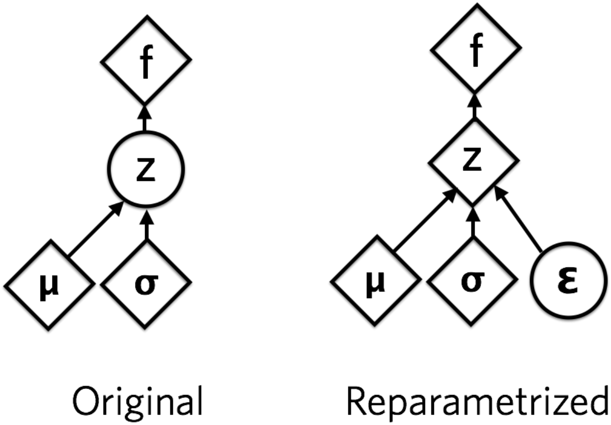


Figure 2: Reparametrization Trick : $z = \mu + \sigma * \epsilon; \quad \epsilon \sim \mathcal{N}(0, 1)$

- Gradients : $\nabla_\lambda ELBO(\lambda) = \mathbb{E}_\epsilon \Big[\nabla_\lambda \big[\log p(\mathbf{x}|\mathbf{z}) - \mathrm{KL}(q(\mathbf{z}; \lambda)||p(\mathbf{z})) \big] \Big]$
- Disadvantage : Not flexible for any black box distribution.

VI Gradients and Score Function a.k.a REINFORCE

$$\begin{aligned}
\nabla_\lambda ELBO(\lambda) &= \nabla_\lambda \mathbb{E}_{q(z;\lambda)}\big[-\log q_\lambda(z) + \log p(z) + \log p(x|z)\big] \\
&= \int_z \nabla_\lambda q_\lambda(z)\big[-\log q_\lambda(z) + \log p(z) + \log p(x|z)\big]dz \\
&\text{Use} \nabla_\lambda(q_\lambda(z)) = q_\lambda(z)\log q_\lambda(z) \\
&= \mathbb{E}_{q(z;\lambda)}\Big[\big(\nabla_\lambda q_\lambda(z)\big)\cdot\big(-\log q_\lambda(z) + \log p(z) + \log p(x|z)\big)\Big]
\end{aligned} \tag{13}$$

- Only need ability to take derivative of q with respect to λ.
- Works for any black box variational family.
- Use MC sampling to update parameters in each step and take empirical mean.

Mean Field Variational Inference

- Mean Field Approximation - A simplifying approximation for the variational distribution.
- Assumes all the variational components are independent of each other.
- Then, mean field assumption assumes

$$p(z|X) \approx q(z) = \prod_{i=1}^{N} q_i(z_i) \tag{14}$$

Mean Field VI - GMM

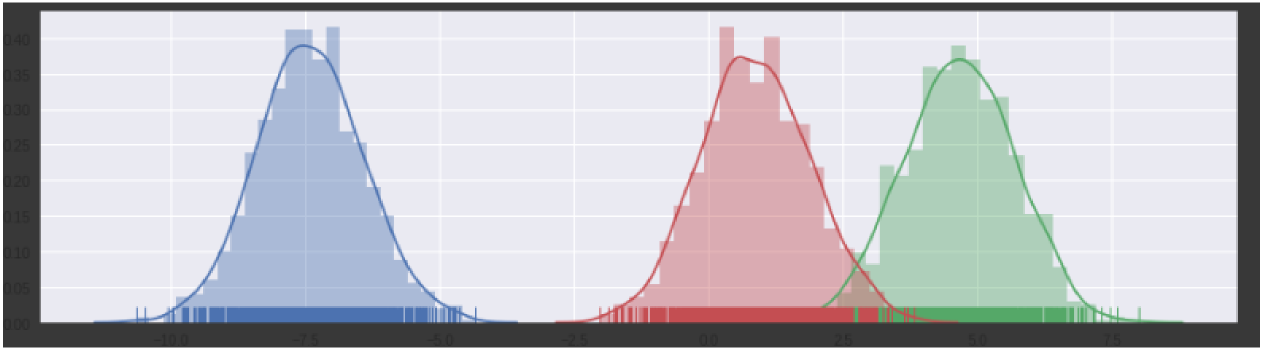


Figure 3: 1-D GMM with three cluster centers

Generative Model : For each datapoint $x^{(i)}$ where i = 1,2......N

- Sample a cluster assignment i.e. the membership of a given point to a mixture component $c^{(i)}$ uniformly. $c^{(i)} \sim Uniform(K)$
- Sample its value from the correpsonding component: $x^{(i)} \sim \mathcal{N}(\mu_{c^{(i)}}, 1)$

Mean Field VI - GMM

To reiterate, the full parametrization of the model could be written as

- $\mu_j \sim \mathcal{N}(0, \sigma^2)\ \forall j = 1, 2....K$ - totally K (3) cluster centers. Known variance σ - Not learning them.
- $c_i \sim \mathcal{U}(K)\ \forall i = 1, 2....N$ - one cluster assignment for each point.
- $x_i \sim \mathcal{N}(c_i^T \mu, 1) \forall i = 1, 2....N$ - each datapoint comes from a Gaussian whose mean is a mixture of the cluster centers with a known variance.
- **PROBLEM :** You are provided X$(x_1, ...x_n)$. You need to eventually learn P(X) using latent variables $\mu, \mathbf{c}$ which you don't observe. You don't know any of the information that you see above in real life.

Mean Field Approximations

- Mean Field Definition : $q(z) = \prod_j q_j(z_j)$.
- Latent variables in this case :
  $q(\mu, c) = q(\mu; m, s^2) = \prod_j q(\mu_j; m_j, s_j^2) \times \prod_i q(c_i, \phi_i)$
- $\mu_j; m_j, s_j^2 \sim \mathcal{N}(m_j, s_j^2)$
- $c_i; \phi_i \sim MultiNomial(\phi_i)$
- Thus, ϕ_i is a vector of probabilities such that $p(c_i = j) = \phi_{ij}$ such that $\sum_j \phi_{ij} = 1$. Learns the likelihood of each point belonging to one cluster center.

Mean Field VI for GMM - A sketch

- Use $ELBO(\lambda) = \mathbb{E}_{q(z;\lambda})\log p(x, z) + H(q; \lambda)$
- Calculate $\log p(x, c, \mu) = \log p(\mu) \log p(c) \log p(x|c, \mu)$ based on our mean field approximations.
- Calculate the entropy term.

$$\log q(c, \mu) = \log q(c) + \log q(\mu) = \sum_{i=1}^{N} \log q(c_i; \phi_i) + \sum_{j=1}^{K} \log q(\mu_j; m_j, s_j^2)$$

.

- Final ELBO is an expectation over sum of both these terms i.e.

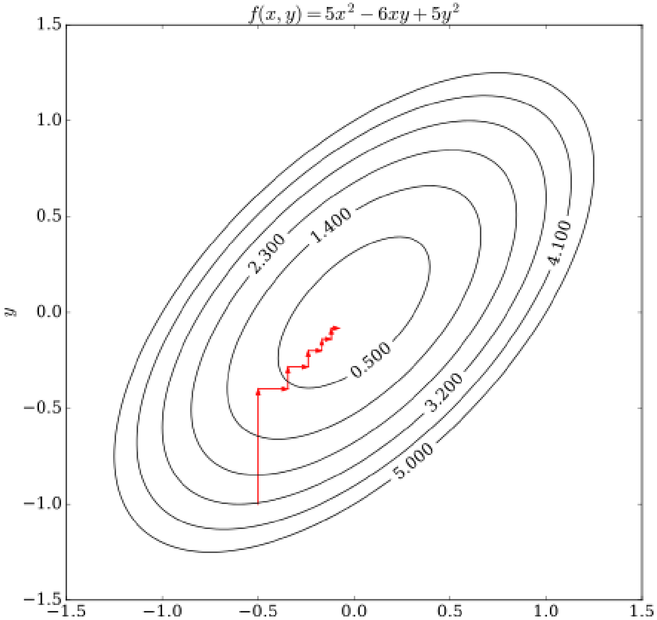
$$\begin{aligned} ELBO \propto \sum_j -\mathbb{E}_q \frac{\mu_j}{2\sigma^2} + \sum_i \sum_j \mathbb{E}_q[C_{ij}] \mathbb{E}_q\left[\frac{(x_i - \mu_j)^2}{2}\right] - \\ \sum_i \sum_j \mathbb{E}_q[\log \phi_{ij}] + \sum_j \frac{1}{2} \log(s_j^2) \end{aligned} \tag{15}$$

Parameter Updates and CAVI

- Gradient Update ϕ_{ij} using $\frac{\partial ELBO}{\partial \phi_{ij}}$
- Gradient update m_j using $\frac{\partial ELBO}{\partial m_j}$
- Gradient Update s_j^2 using $\frac{\partial ELBO}{\partial s_j^2}$
- Remember we are doing Coordinate Ascent here (Maximization Problem).

Coordinate Ascent

1. Choose initial parameter vector x. Repeat steps 2 to 4.
2. Choose an index i from 1 to n.
3. Choose a step size α.
4. Update x_i to $x_i + \alpha \frac{\partial F(\mathbf{x})}{\partial x_i}$



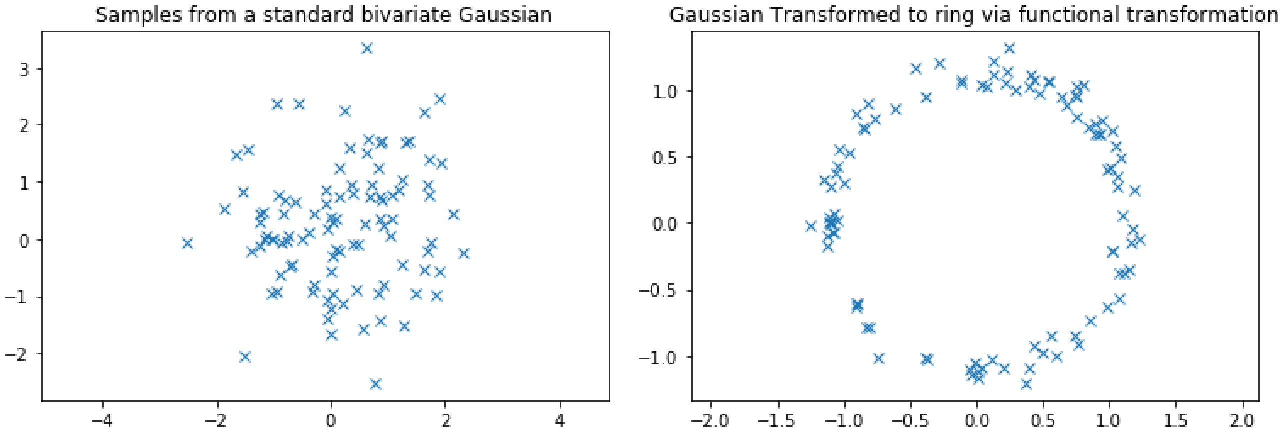
Variational Autoencoders

Generative Models

- Learns the generative form of the data distribution - P(X)
- Remember AutoEncoders learned in class.
- Why latent variable models are needed ?
- What are the latent variables expected to learn ? Eg: MNIST
- Remember $p(x) = \int_z p(x, z; \theta) p(z; \theta) dz$. θ can be any parametric form - could be a neural network.

VAEs

- Define $p(z) = \mathcal{N}(0, I)$
- Transform a simple $p(z)$ into a complicated $p(x)$


Samples from a standard bivariate Gaussian

Gaussian Transformed to ring via functional transformation

Figure 5: Given a random variable Z with one distribution (on the left - standard bivariate Gaussian), we can always create another random variable X = g(Z) with an entirely different distribution through appropriate functional transformation(on the right. $g(z) \to z/10 + z/||z||$.

VAEs

Where is the Autoencoder?

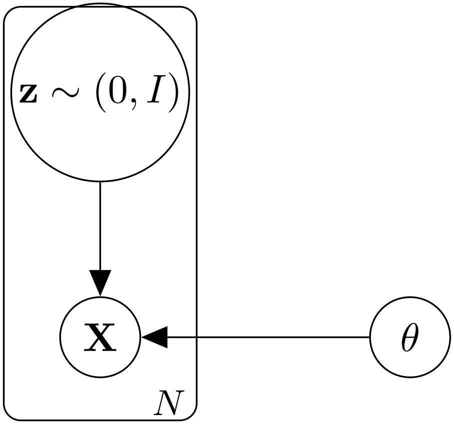


Figure 6: Graphical Model of VAE

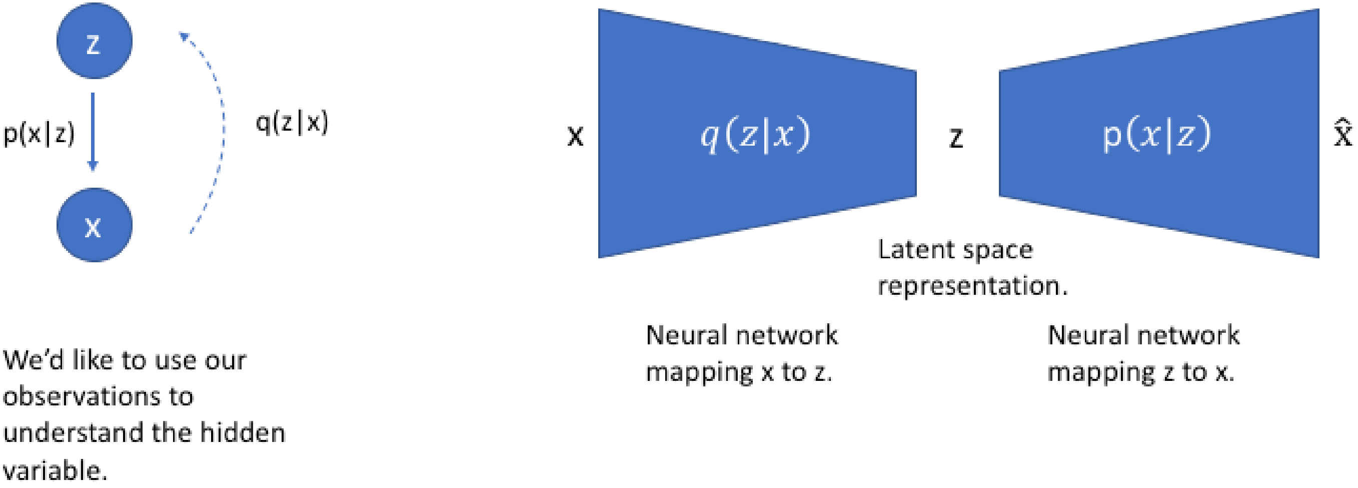
Need to infer the posterior after observing data.

$$p(z|x) = \frac{p(x|z)p(z)}{\underbrace{\int_z p(x|z;\theta)p(z)dz}_{\text{Intractable}}} \tag{16}$$

VAEs

Assume variational approximation for p(z—x). We have got our encoder decoder setup back. q is the encoder and p is the decoder.

$$\mathcal{L}(\mathbf{x};\theta,\lambda) = D_{KL}\big(\underbrace{q(\mathbf{z}|\mathbf{x};\lambda)}_{\text{decoder}}\,||p(\mathbf{z})\big) - \mathbb{E}_{\mathbf{z}\sim q}\log \underbrace{p(\mathbf{x}|\mathbf{z};\theta)}_{\text{encoder}} \tag{17}$$



We'd like to use our observations to understand the hidden variable.

Latent space representation.

Neural network mapping x to z.

Neural network mapping z to x.

Figure 7: VAE in a nutshell

VAEs

$$\mathcal{L}(\mathbf{x};\theta,\lambda) = D_{KL}\big(\underbrace{q(\mathbf{z}|\mathbf{x};\lambda)}_{\text{decoder}}||p(\mathbf{z})\big) - \mathbb{E}_{\mathbf{z}\sim q}\log \underbrace{p(\mathbf{x}|\mathbf{z};\theta)}_{\text{encoder}}$$

$$D_{KL}((\mathcal{N}(\mu(X),\Sigma(X))||\mathcal{N}(0,I)) = \frac{1}{2}\Big(\mathrm{Tr}(\Sigma(X)) + (\mu(X))^T(\mu(X)) - k - \log\det(\Sigma(X))\Big) \tag{18}$$

What about the reconstruction term ?

VAE Reconstruction - Training

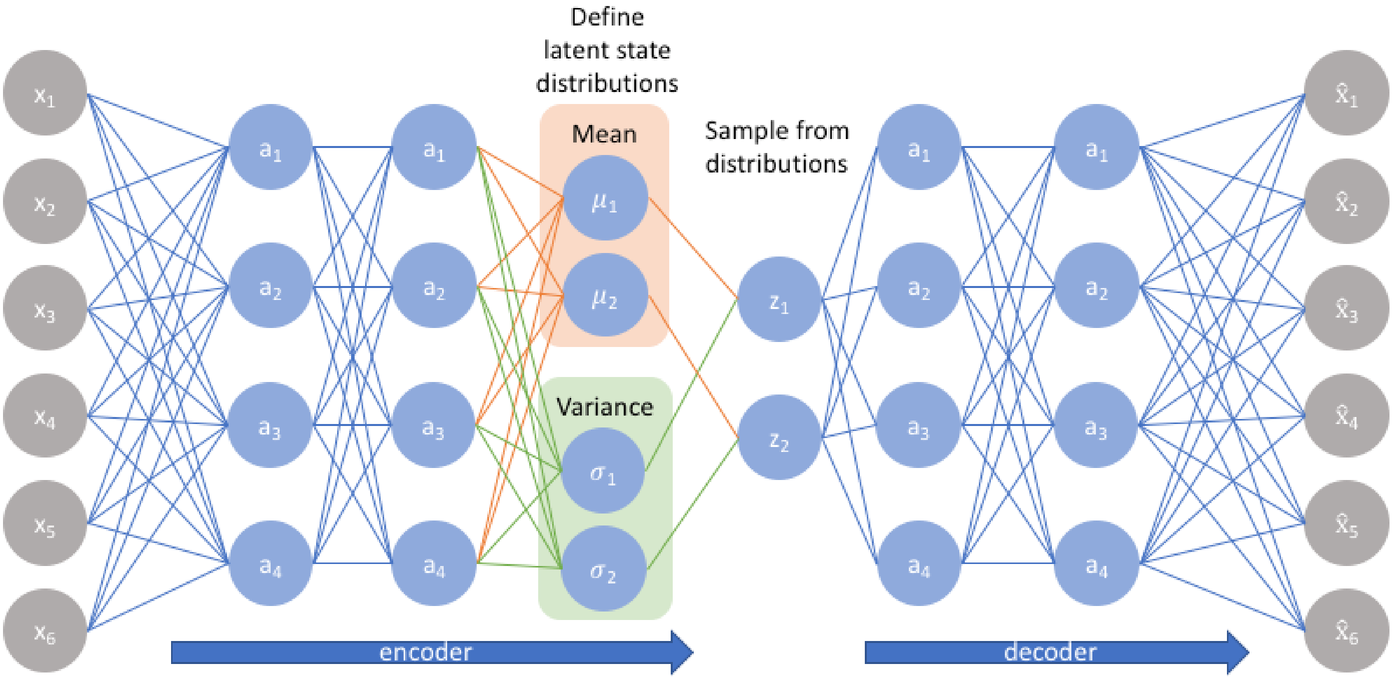
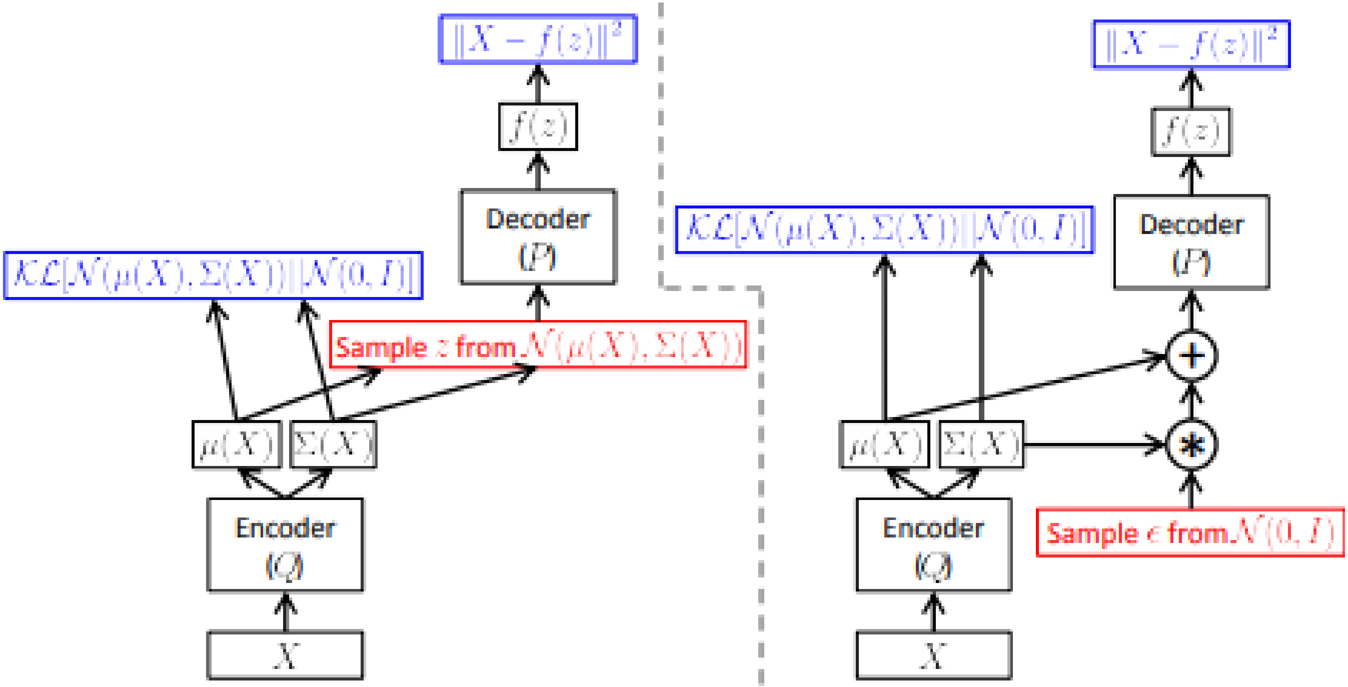


Figure 8: Training of VAE with Gaussian Variational Family

Reparametrization



$\|X - f(z)\|^2$
$f(z)$
Decoder
(P)
$\mathcal{KL}[\mathcal{N}(\mu(X), \Sigma(X))\|\mathcal{N}(0, I)]$
Sample z from $\mathcal{N}(\mu(X), \Sigma(X))$
$\mu(X)$
$\Sigma(X)$
Encoder
(Q)
X
$\|X - f(z)\|^2$
$f(z)$
$\mathcal{KL}[\mathcal{N}(\mu(X), \Sigma(X))\|\mathcal{N}(0, I)]$
Decoder
(P)
+
*
$\mu(X)$
$\Sigma(X)$
Encoder
(Q)
Sample ϵ from $\mathcal{N}(0, I)$
X

Figure 9: Reparametrization(Right)

VAE - Visualization

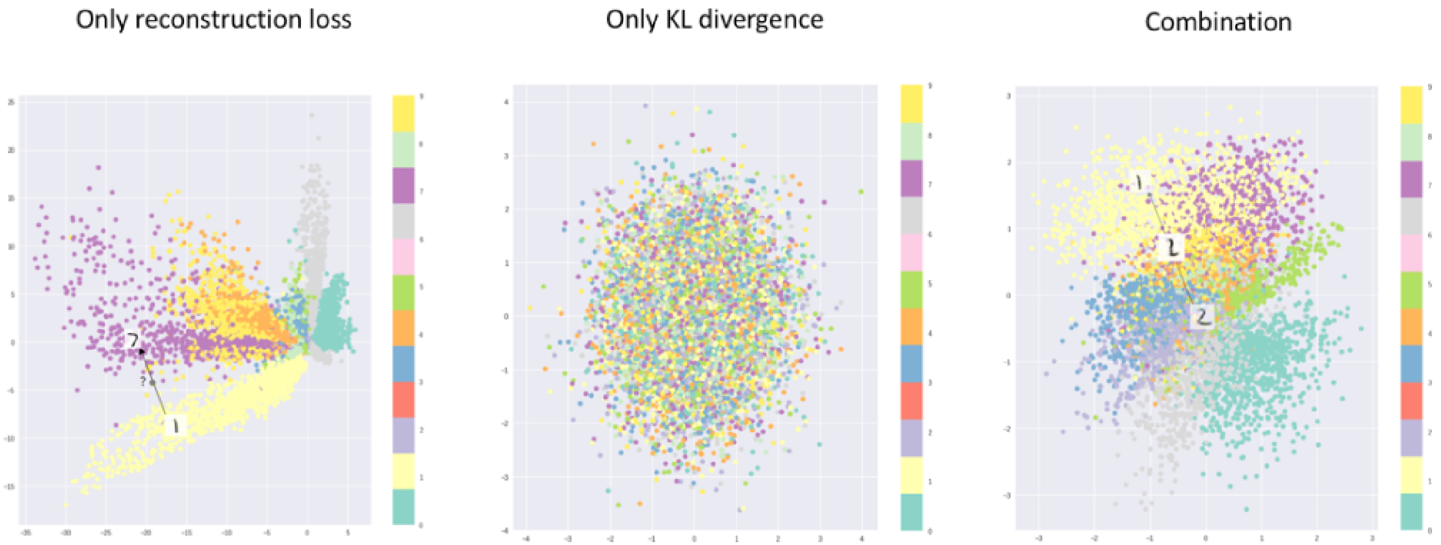
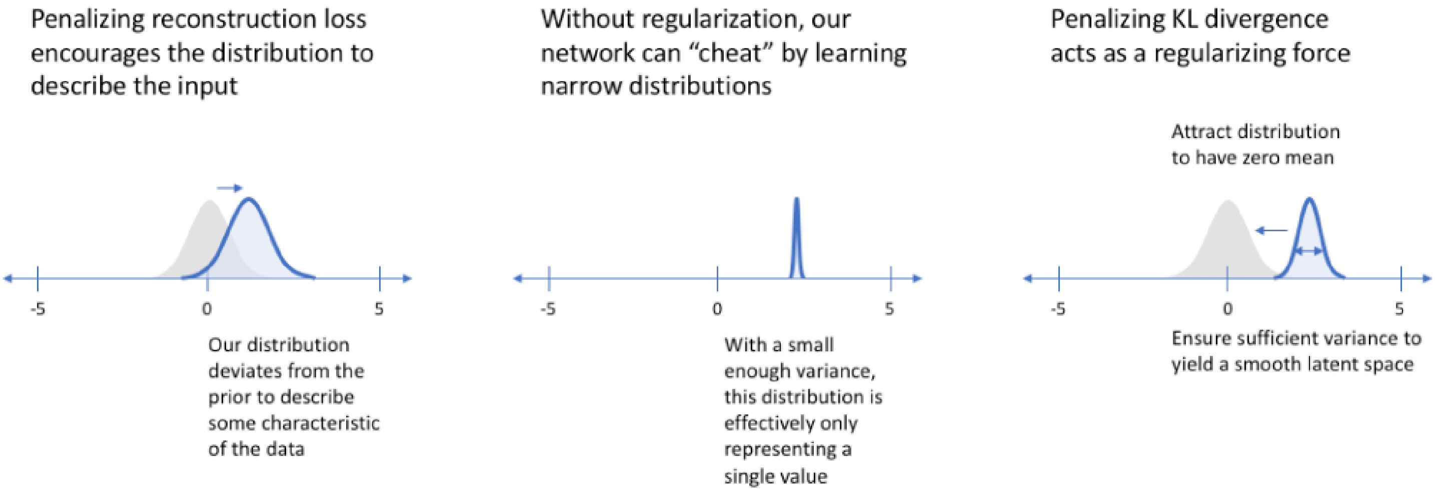


Figure 10: Contributions of reconstruction and KL

VAE - Visualization


Penalizing reconstruction loss encourages the distribution to describe the input

Our distribution deviates from the prior to describe some characteristic of the data

Without regularization, our network can “cheat” by learning narrow distributions

With a small enough variance, this distribution is effectively only representing a single value

Penalizing KL divergence acts as a regularizing force

Attract distribution to have zero mean

Ensure sufficient variance to yield a smooth latent space

Figure 11: Contributions of reconstruction and KL

VAE-Results

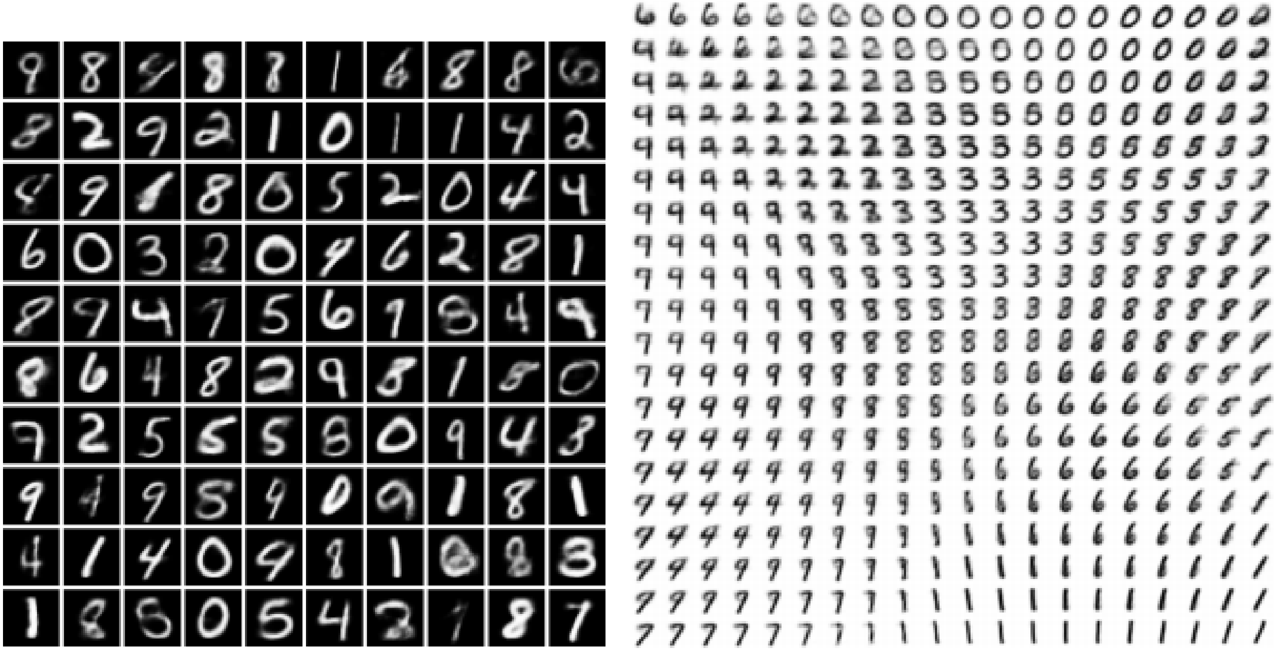
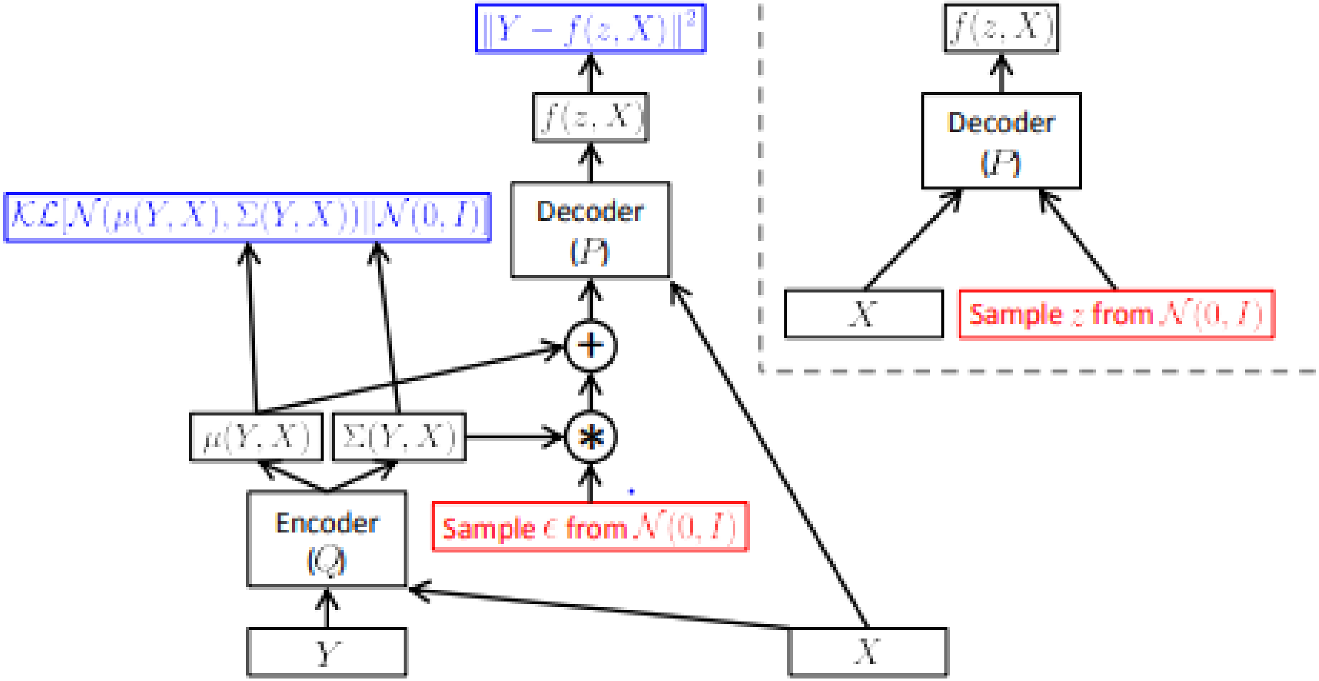


Figure 12: Left: MNIST generative results from VAE. Right : Latent code interpolation - Results generated from sampling latent codes and interpolating between those two codes.

Music-VAE (Google, 2018)

https://youtu.be/G5JT16flZwM

Conditional VAE



$\|Y - f(z, X)\|^2$
$f(z, X)$
$\mathcal{KL}[\mathcal{N}(\mu(Y, X), \Sigma(Y, X)) \| \mathcal{N}(0, I)]$
Decoder (P)
+
$\mu(Y, X)$
$\Sigma(Y, X)$
*
Encoder (Q)
Sample ϵ from $\mathcal{N}(0, I)$
Y
X
$f(z, X)$
Decoder (P)
X
Sample z from $\mathcal{N}(0, I)$

Conditional VAE

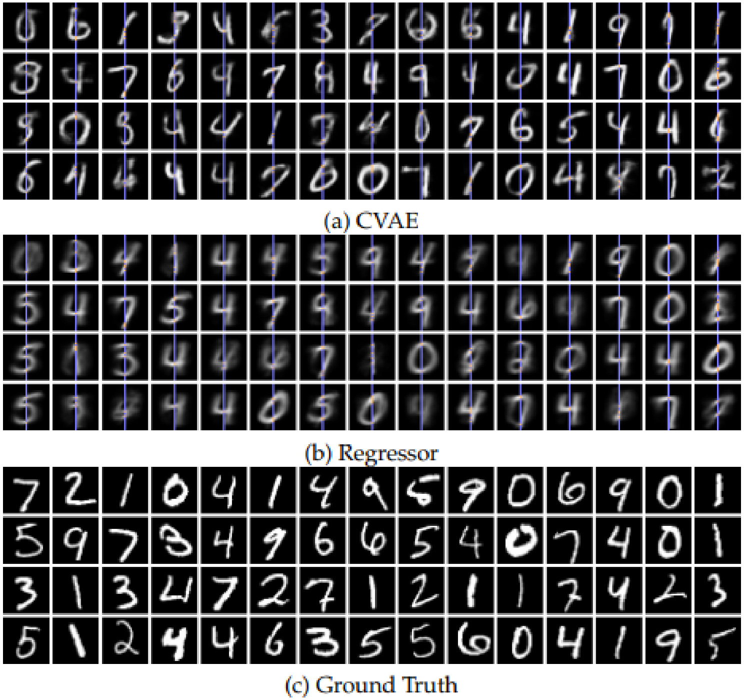


Figure 14: A Conditional VAE. Image Completion - The inputs(incomplete image) to CVAE are the pixels in the middle column shown in the images in blue.

Bayesian Neural Networks

QUESTION : How do you learn uncertainty of what your deep network learns ?

IDEA : Have a prior over weights and do MAP inference.

- Confidence of your predictions.
- Richer and regularized representation of weights since you control the prior
- Model Averaging (since the lilely prediction of y is the expected value of distribution over functions)

How does it look like ?

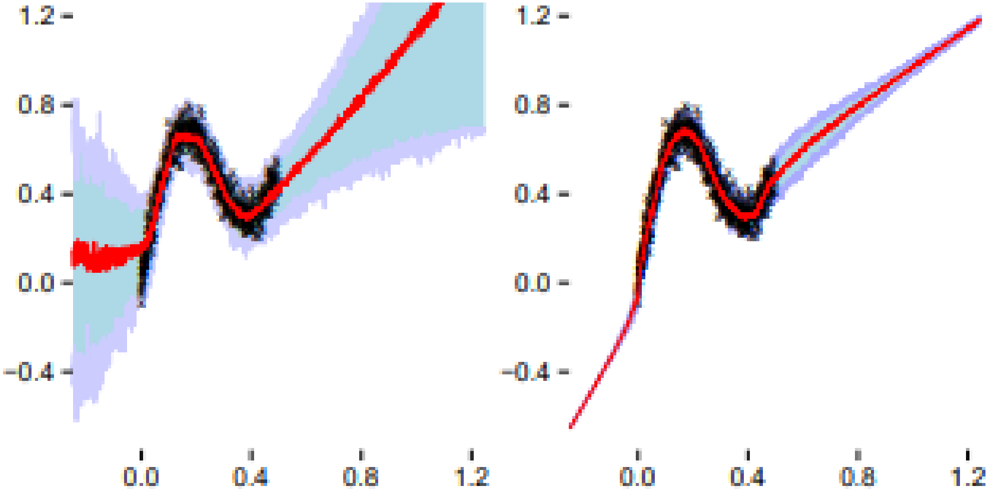


Figure 15: Left : Fit via BBB. Right:Fit via Neural Nets. Red indicates the median prediction. Blue boundaries indicate quartile ranges. Look how BBB is less confident in out of distribution regions and more confident around evidence.Credits

How do you do it ?

$$\begin{aligned} p(\mathbf{w}|\mathbf{x},\mathbf{y}) &\propto \mathbb{P}\Big(\mathbf{y}_{1:n}|\mathbf{x}_{1:n},;\mathbf{w}\Big) * p(\mathbf{w}) \\ \mathbf{w}^* &= \arg\max_{\mathbf{w}} \quad \underbrace{P(\mathbf{w}|\mathbf{x},\mathbf{y})}_{\text{As usual, intractable}} \end{aligned} \tag{19}$$

$$\begin{aligned} \theta^* &= \arg\min_{\theta} D_{KL}\big(q(\mathbf{w};\theta)||p(\mathbf{w}|\mathcal{D})\big) \\ &= \arg\min_{\theta} \underbrace{D_{KL}\big[q(\mathbf{w};\theta)||p(\mathbf{w})\big] - \mathbb{E}_{q(\mathbf{w};\theta)} \log p(\mathcal{D}|\mathbf{w})}_{\mathcal{L}(\mathcal{D},\theta)} \end{aligned} \tag{20}$$

(derived similar to VI)

Perform SGD via re-parametrization to train the network. Bayes by backpropagation - $https://arxiv.org/pdf/1505.05424.pdf$.(pseudo-code)

Credits

1. https://www.jeremyjordan.me/variational-autoencoders/ (Images and Text)
2. https://arxiv.org/abs/1606.05908 (Images and Text)
3. Other references in the notes (Largely text)