CS 190I
Deep Learning
Variational Auto-Encoder

Lei Li (leili@cs)
UCSB

Course Evaluation

- https://esci.id.ucsb.edu
- https://bit.ly/3FSqFs0
- Feedback is important and helpful for improving the course
- Encourage narrative comments
- Bonus 5% to final exam, if response rate > 90% (48% today)

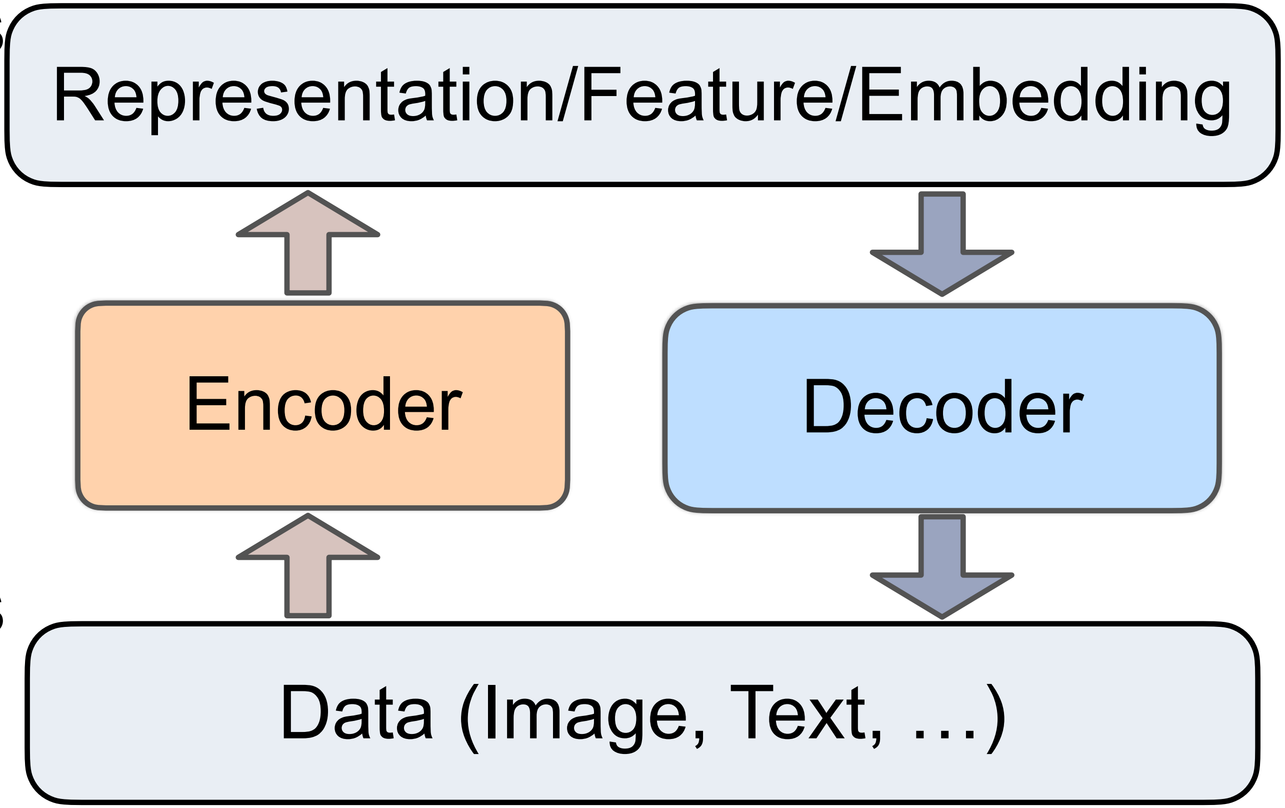
Recap

- Graph neural network
  - message passed along graph edges
  - aggregate message/embedding by FFN
  - many variants: GCN, GAT, GraphSAGE

Variational Auto-Encoder (VAE)

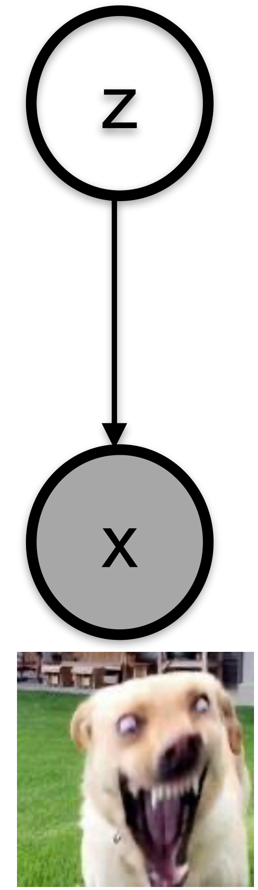
VAE

- Hidden representations follow a prior distribution
- Encoder will produce a distribution of representations (posterior distribution)



Deep Latent Model

- z follows a prior distribution, e.g. Gaussian(0, I)
- p(x|z) is defined by a deep neural network $f(z;\theta)$
- To learn θ, use $E_{(Z|X)}[\log p(X,Z;\theta)]$

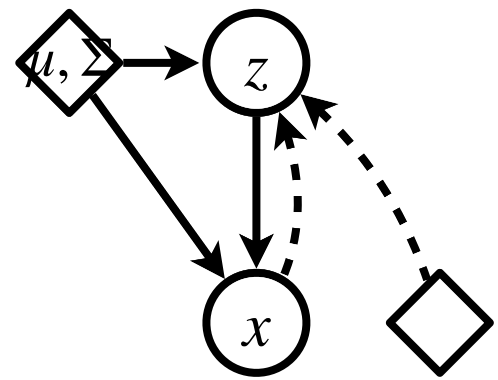


Graphical Model for VAE

- Assuming data X is generated from a latent variable Z
- Generation process
  - draw $Z \sim N(\mu, \Sigma)$
  - draw $X \mid Z \sim p(f(Z))$, defined by a neural network f
- The goal is to maximize the data log-likelihood

$$\log p(X;\theta) = \log \int p(X \mid Z) p(Z) dZ$$

- Hard to optimize over θ, if f(Z) is very complex such as a CNN, RNN, or Transformer.



VAE

Objective: maximize the data loglikelihood

$$\max \ell(\theta) = \sum_n \log p(x_n; \theta)$$

$$= \sum_n \log \int p(x_n \mid z_n; \theta) p(z_n; \theta) dz_n$$

VAE

$$\max \ell(\theta) = \sum_n \log p(x_n; \theta)$$

$$= \sum_n \log \int p(x_n \mid z_n; \theta) p(z_n; \theta) dz_n$$

- But $\log p(x; \theta)$ is intractable.

$q(z \mid x; \phi)$ is the posterior distribution from encoder!

- For any distribution $q(z \mid x, \phi)$:

$$\log p(x; \theta) \geq \mathrm{E}_{q(z|x;\phi)} \left[\log \frac{p(x, z; \theta)}{q(z \mid x; \phi)} \right] = \mathrm{ELBO}$$

- Derivation via Jensen's inequality.
- Maximizing the ELBO instead of maximizing $\log p(x; \theta)$

Understanding ELBO

$$\log p(X;\theta) \geq \mathrm{E}_q[\log \frac{p(X,Z;\theta)}{q(Z \mid X;\phi)}]$$

$$\max_\theta \max_\phi \mathrm{ELBO} = \sum_n \mathrm{E}_q \left[\log \frac{p(x_n \mid z_n;\theta) p_0(z_n)}{q(z_n \mid x_n;\phi)}\right]$$

$$=$$

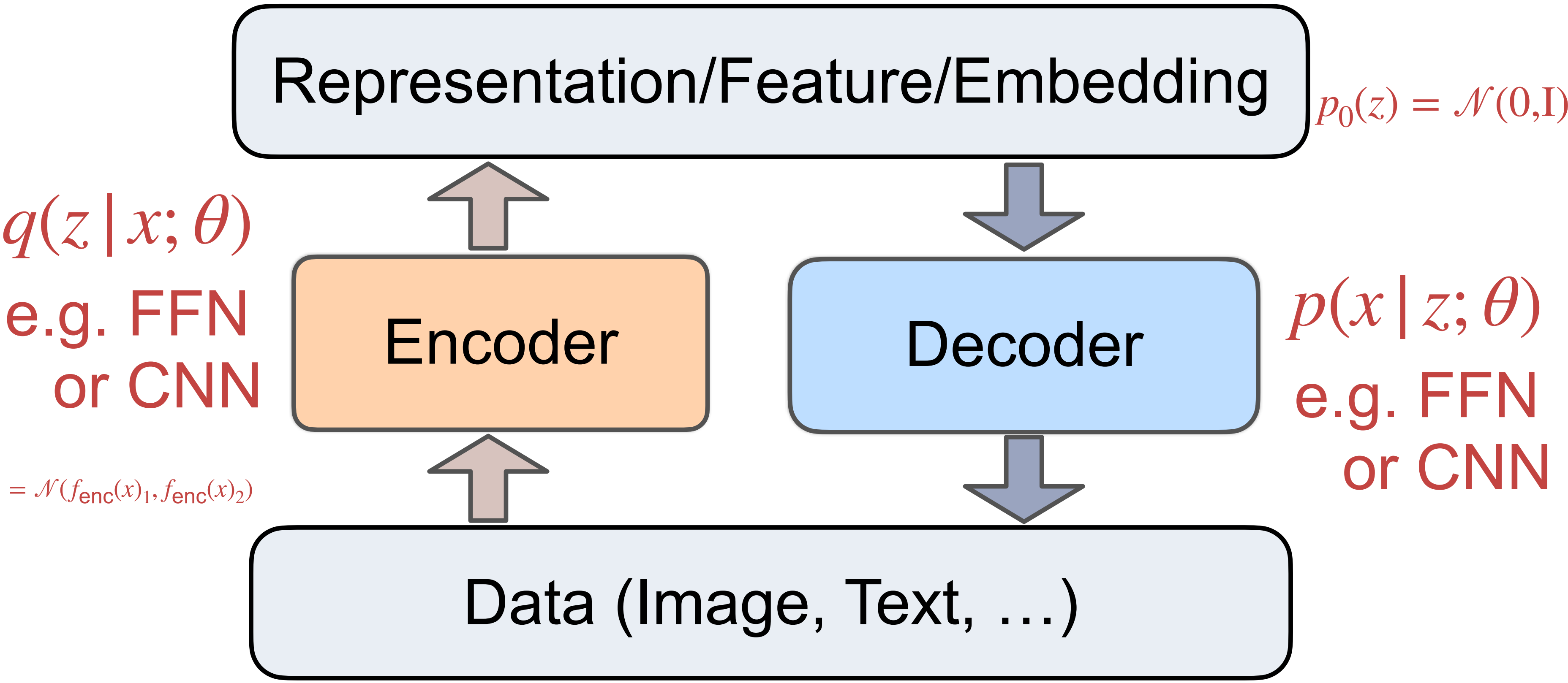
$$=$$

$$= \mathrm{E}_q \left[\log p(x_n \mid z_n;\theta)\right] - \mathrm{KL}\left(q(z_n \mid x_n;\phi) \| p_0(z_n)\right)$$

Reconstruction loss Regularization

VAE

Let $q(z \mid x; \phi)$ and $p(x \mid z; \theta)$ share the same parameter θ



Training VAE

gradient descent(ascent for max)

$$\max_{\theta} \max_{\phi} \text{ELBO} = \sum_n \mathrm{E}_{q(z_n|x_n;\theta)} \left[\log \frac{p(x_n \mid z_n; \theta) p_0(z_n)}{q(z_n \mid x_n; \theta)} \right]$$

$$= \sum_n \mathrm{E}_{q(z_n|x_n;\theta)} \left[r(\theta, z_n, x_n) \right]$$

$$r(\theta, z_n, x_n) = \log \frac{p(x_n \mid z_n; \theta) p_0(z_n)}{q(z_n \mid x_n; \theta)}$$

Computing gradient:

$$\nabla_\theta \mathrm{E}_{q(z_n|x_n;\theta)} \left[r(\theta, z_n, x_n) \right]$$

Gradient of ELBO

$$r(\theta, z_n, x_n) = \log \frac{p(x_n \mid z_n; \theta) p_0(z_n)}{q(z_n \mid x_n; \theta)}$$

Computing gradient:

$$\nabla_\theta \mathrm{E}_{q(z_n \mid x_n; \theta)} \left[r(\theta, z_n, x_n) \right]$$

Gradient of ELBO

$$r(\theta, z_n, x_n) = \log \frac{p(x_n \mid z_n; \theta) p_0(z_n)}{q(z_n \mid x_n; \theta)}$$

Computing gradient:

$$\nabla_\theta \mathrm{E}_{q(z_n|x_n;\theta)} \left[r(\theta, z_n, x_n) \right] = \mathrm{E}_{q(z_n|x_n;\theta)} \left[\nabla_\theta r(\theta, z_n, x_n) \right] + \int r(\theta, z_n, x_n) \nabla_\theta q(z_n \mid x_n; \theta) d_{z_n}$$

1. sample $z_n \sim q(z_n \mid x_n; \theta) = \mathcal{N}(f(x_n)_1, f(x_n)_2)$, then compute average of $\nabla_\theta r(\theta, z_n, x_n)$

Gradient of ELBO

$$r(\theta, z_n, x_n) = \log \frac{p(x_n \mid z_n; \theta) p_0(z_n)}{q(z_n \mid x_n; \theta)}$$

Computing gradient:

$$\nabla_\theta \mathrm{E}_{q(z_n|x_n;\theta)} \left[r(\theta, z_n, x_n) \right] = \mathrm{E}_{q(z_n|x_n;\theta)} \left[\nabla_\theta r(\theta, z_n, x_n) \right] + \int r(\theta, z_n, x_n) \nabla_\theta q(z_n \mid x_n; \theta) d_{z_n}$$

2. rewrite as

$$\int r(\theta, z_n, x_n) \nabla_\theta q(z_n \mid x_n; \theta) d_{z_n} = \mathrm{E}_{q(z_n|x_n;\theta)} \left[r(\theta, z_n, x_n) \nabla_\theta \log q(z_n \mid x_n; \theta) \right]$$

then sample $z_n \sim q(z_n \mid x_n; \theta) = \mathcal{N}(f(x_n)_1, f(x_n)_2)$

compute average of $r(\theta, z_n, x_n) \nabla_\theta q(z_n \mid x_n; \theta)$

Problem — high variance

Reparameterization Trick

$$q(z_n | x_n; \theta) = \mathcal{N}(f(x_n)_1, f(x_n)_2) = \mathcal{N}(\mu_\theta(x_n), \Sigma_\theta(x_n))$$

Treating $\epsilon \sim N(0,1)$, standard Gaussian distribution, then

$$\mathrm{E}_{q(z_n|x_n;\theta)}\left[r(\theta, z_n, x_n)\right] = \mathrm{E}_{\epsilon\sim N(0,1)}\left[r(\theta, z_n, x_n)\right]$$

where $z_n = \Sigma_\theta^{\frac{1}{2}}(x_n)\epsilon + \mu_\theta(x_n)$

Taking gradient does not depend on the distribution

Reparameterization Trick

$$\nabla_\theta \mathrm{E}_{q(z_n|x_n;\theta)}\left[\log \frac{p(x_n\,|\,z_n;\theta)p_0(z_n)}{q(z_n\,|\,x_n;\theta)}\right]$$

$$= \nabla_\theta \mathrm{E}_{q(z_n|x_n;\theta)}\left[\log p(x_n\,|\,z_n;\theta)\right] - \mathrm{KL}\left(q(z_n\,|\,x_n;\phi)\|p_0(z_n)\right)$$

$$= \nabla_\theta \mathrm{E}_{\epsilon\sim N(0,1)}\left[\log p(x_n\,|\,z_n;\theta)\right] - \mathrm{KL}\left(\mathcal{N}(\mu_\theta(x_n),\Sigma_\theta(x_n))\|\mathcal{N}(0,1)\right)$$

$$= \nabla_\theta \mathrm{E}_{\epsilon\sim N(0,1)}\left[\log p(x_n\,|\,z_n;\theta)\right] - \frac{1}{2}\left(\mu_\theta(x_n)^T\mu_\theta(x_n) + \mathrm{tr}(\Sigma_\theta)(x_n) - M - \mathrm{logDet}(\Sigma_\theta(x_n))\right)$$

$$= \mathrm{E}_{\epsilon\sim N(0,1)}\left[\nabla_\theta \log p(x_n\,|\,z_n;\theta)\right] - \nabla_\theta\frac{1}{2}\left(\mu_\theta(x_n)^T\mu_\theta(x_n) + \mathrm{tr}(\Sigma_\theta(x_n)) - M - \mathrm{logDet}(\Sigma_\theta(x_n))\right)$$

where $z_n = \Sigma_\theta^{\frac{1}{2}}(x_n)\epsilon + \mu_\theta(x_n)$

Compute Gradient using Reparameterization Trick

For each data point x_n, current parameter θ

Step 1: sample $\epsilon \sim N(0,1)$

Step 2: using encoder forward to compute $\mu, \Sigma = f_{\text{enc}}(x_n; \theta)$

Step 3: $z(\theta) = \Sigma^{\frac{1}{2}}\epsilon + \mu$

Step 4: using decoder forward to compute $p(x_n \mid z(\theta); \theta)$

Step 5: define

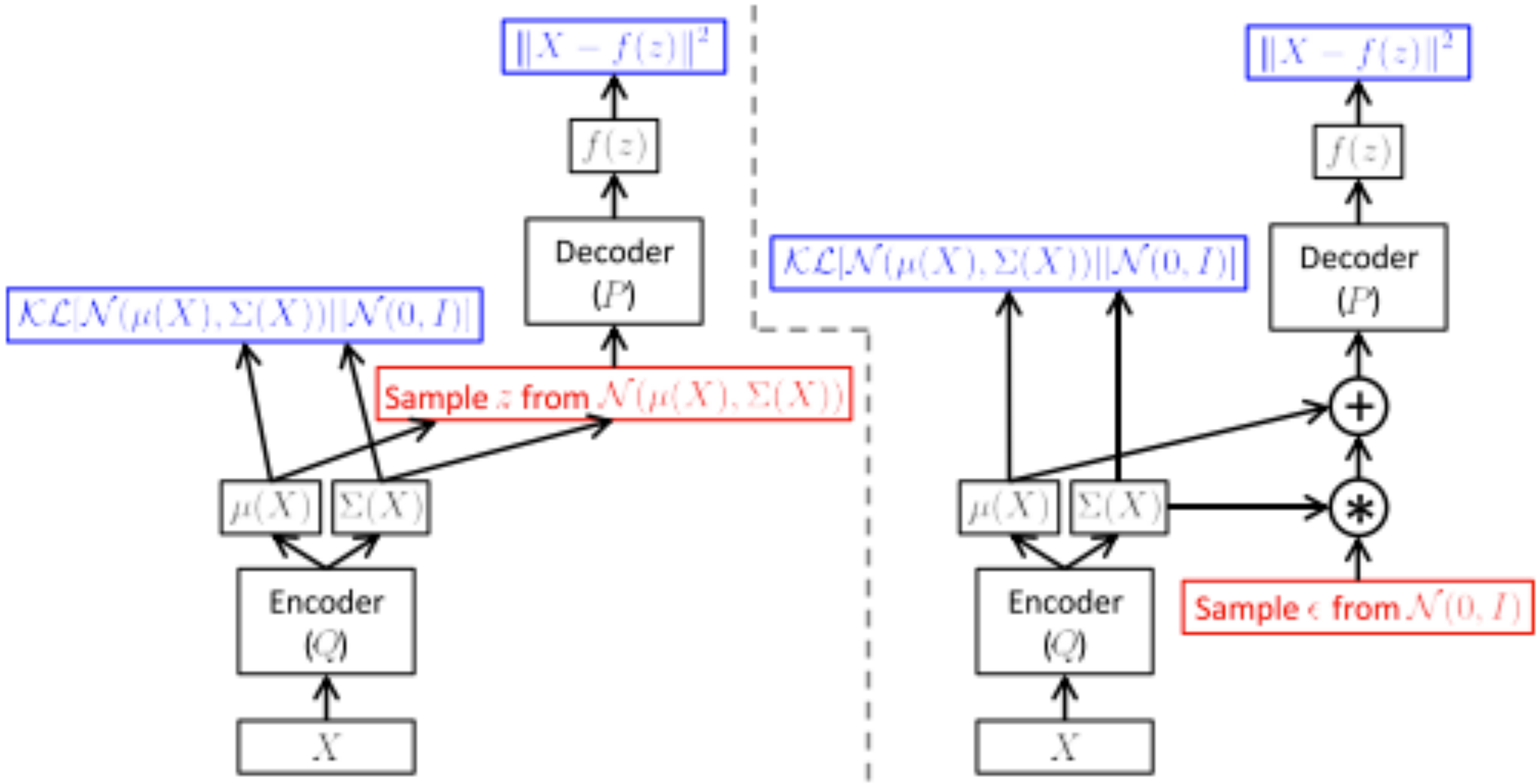
$\text{err} = \log p(x_n \mid z_n; \theta) - \beta \cdot \text{KL}\left(q(z \mid x_n; \theta) \| p_0(z)\right)$, then

using back-propagation to compute gradient for θ

$$\frac{1}{2}\left(\mu_\theta(x_n)^T \mu_\theta(x_n) + \text{tr}(\Sigma_\theta)(x_n) - M - \text{logDet}(\Sigma_\theta(x_n))\right)$$

Training VAE

- Reparameterization trick

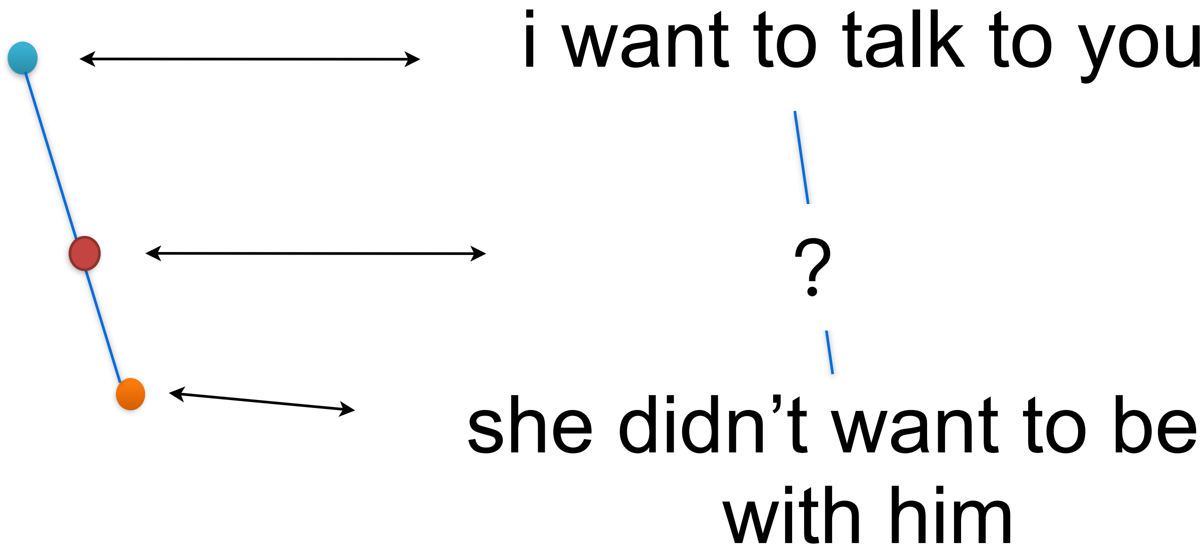


Tutorial on Variational Autoencoders
(Doersch Carl, 2016)

Sentence VAE

Generating Sentence from Continuous vectors

- Key challenge: Interpolation in continuous space should yield reasonable sentences



Conditional Sequence Generation

Given a latent variable z, a sequence of text tokens $x = (x_1, x_2, \ldots, x_t)$ can be generated with RNN (or LSTM, transformer), CRNN model:

$$p\left(x \mid z; \theta\right) = \prod_t p(x\_t \mid x_{<t}, z; \theta)$$

$$p\left(x_t \middle| x_{<t}, z; \theta\right) = \mathrm{softmax}(W \cdot h_t)$$

$$h_t = RNN(h_{t-1}, \left[x_{t-1},\ z\right], \theta)$$

VAE for Sentence Generation

Decoding:

$z \sim N(0, I)$

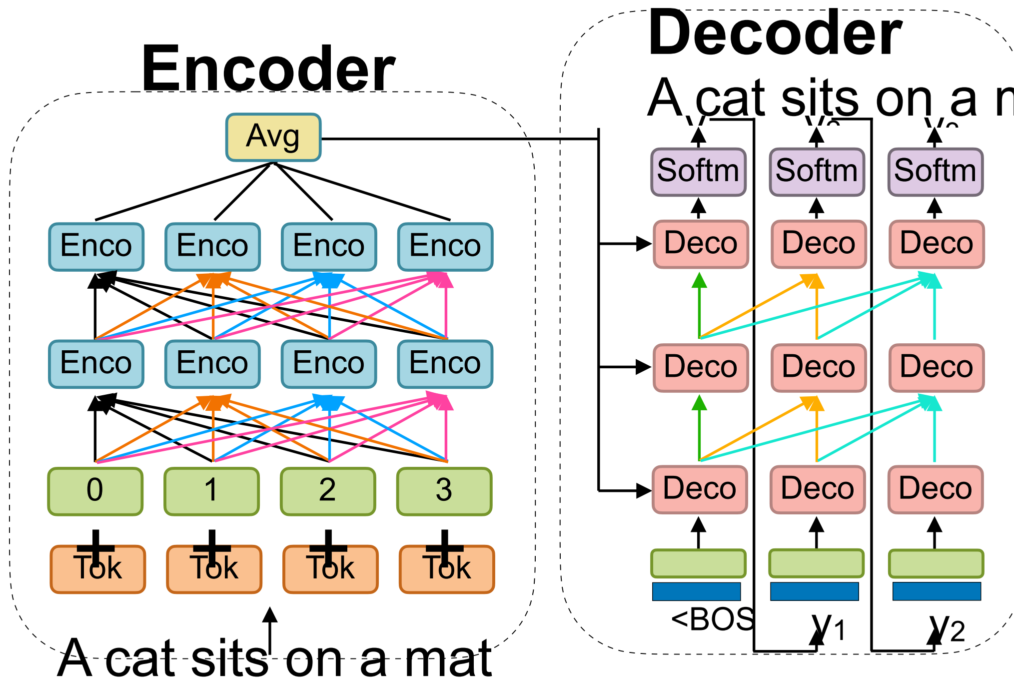
generate x from Transformer(z) or LSTM(z)

Encoding:

$$q(z \mid x) = N(\mu, \sigma^2)$$

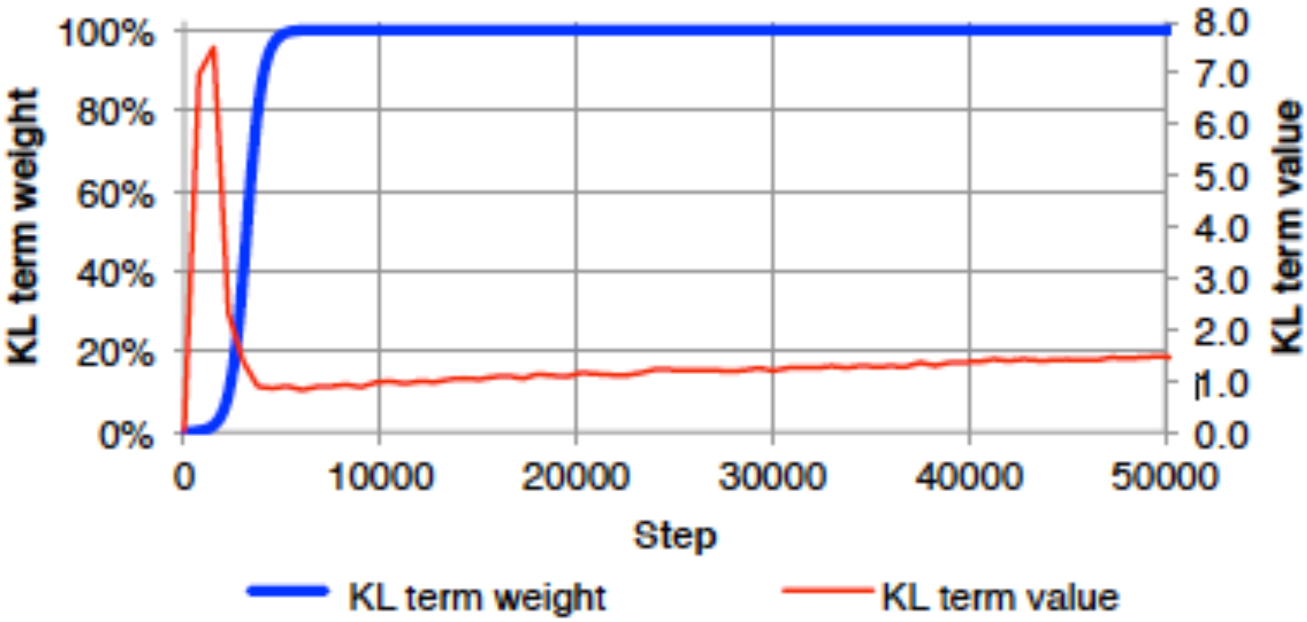
$$\mu = W_1 \cdot h_t, \ \sigma^2 = \exp W_2 \cdot h_t$$

$$h_t = \text{Transformer}(x; \theta)$$



Training VAE: Posterior Collapse

- KL term in ELBO collapses to zero and latent variable encodes little information.
- Solution: KL annealing & word dropout



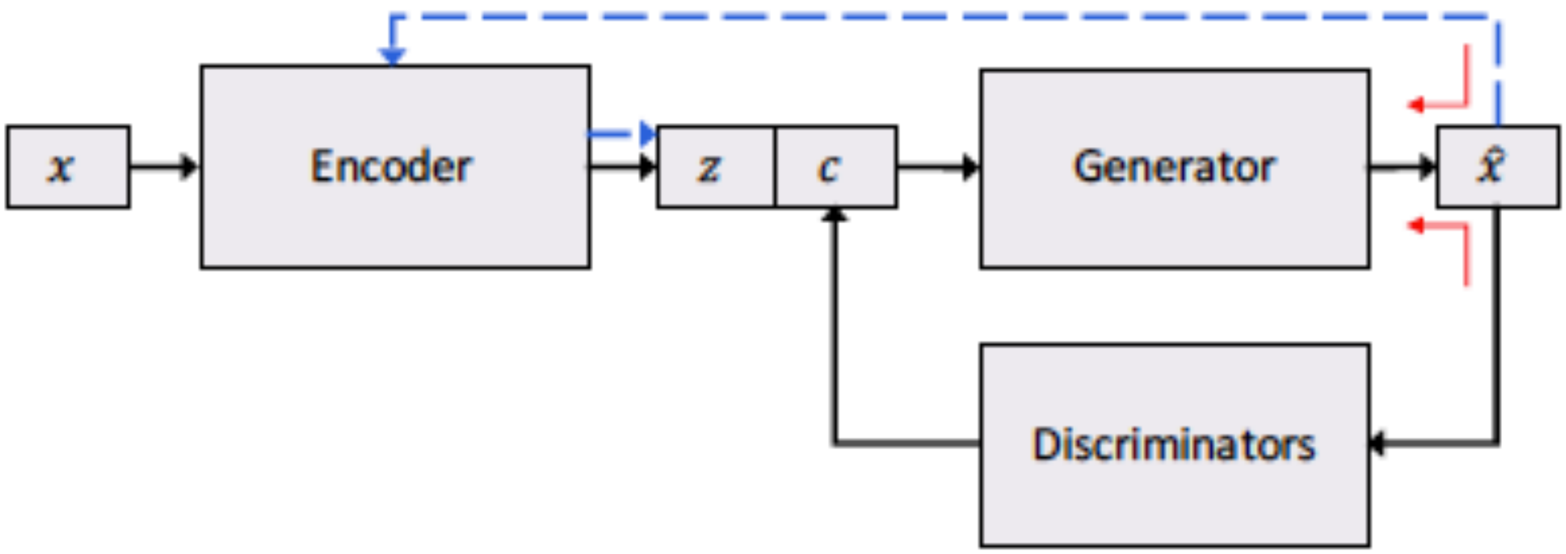
Examples on Sentence Interpolation

" i want to talk to you . "
"i want to be with you . "
"i do n't want to be with you . "
i do n't want to be with you .
she did n't want to be with him .

he was silent for a long moment .
he was silent for a moment .
it was quiet for a moment .
it was dark and cold .
there was a pause .
it was my turn .

Variants

- Controllable sentence generation with both continuous and discrete labels



Toward Controlled Generation of Text, (Hu et. al. ICML 2017)

Generating with Varying Semantic Label

the film is strictly routine !
the film is full of imagination .

after watching this movie , i felt that disappointed .
after seeing this film , i 'm a fan .

the acting is uniformly bad either .
the performances are uniformly good .

this is just awful .
this is pure genius .

the acting is bad .
the movie is so much fun .

none of this is very original .
highly recommended viewing for its courage , and ideas .

too bland
highly watchable

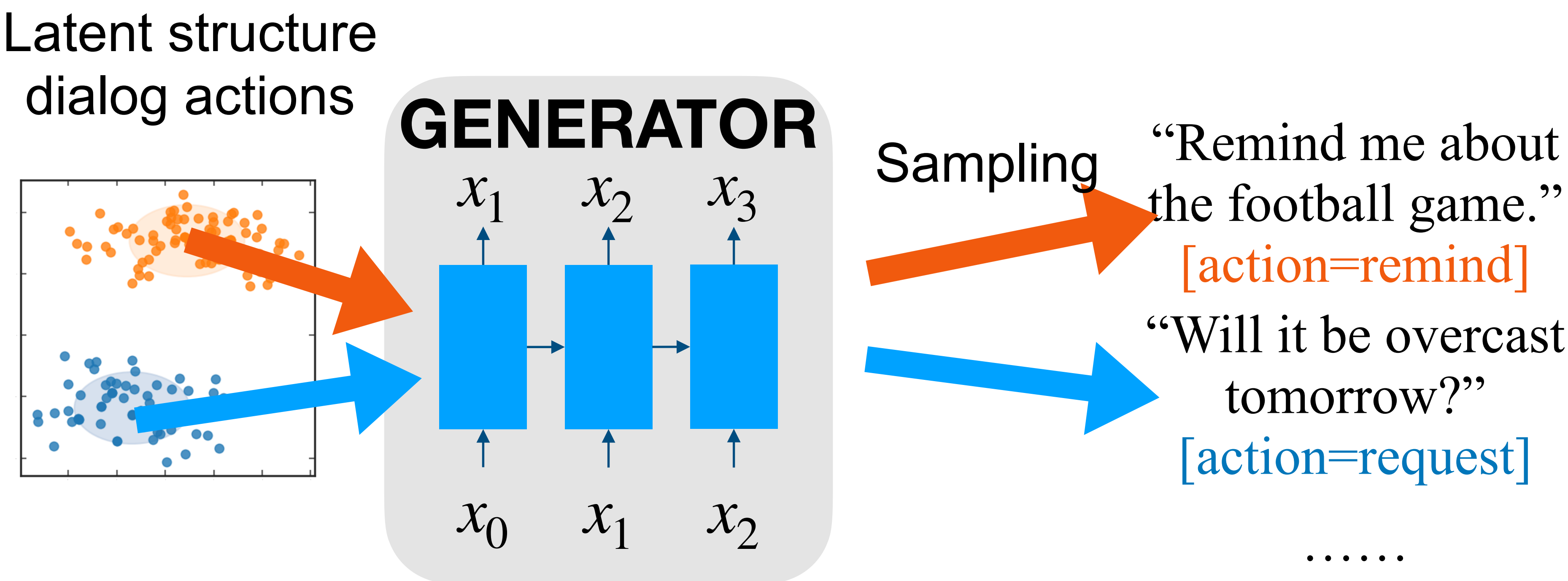
i can analyze this movie without more than three words .
i highly recommend this film to anyone who appreciates music .

Toward Controlled Generation of Text, (Hu et. al. ICML 2017)

Deep Latent Variable Models for Text

- Interpretable Deep Latent Representation from Raw Text
  - Learning Exponential Family Mixture VAE [ICML 20]
- Disentangled Representation Learning for Text Generation
  - Data to Generation: VTM [ICLR 20b]
  - Learning syntax-semantic representation [ACL 19c]
- One model to acquire 4 language skills
  - Mirror Generative NMT [ICLR 20a]

Learning Interpretable Latent Representation

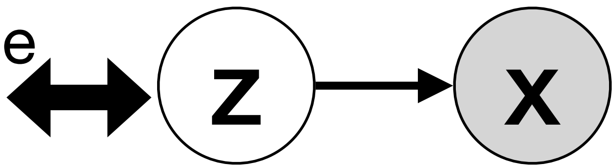


Generate Sentences with interpretable factors

How to Interpret Latent Variables in VAEs?

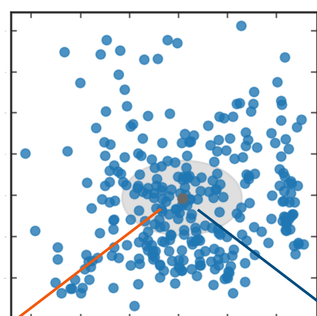
Variational Auto-encoder (VAE)

interpretable structure



(Kingma & Welling, 2013)

z: **continuous** latent variables



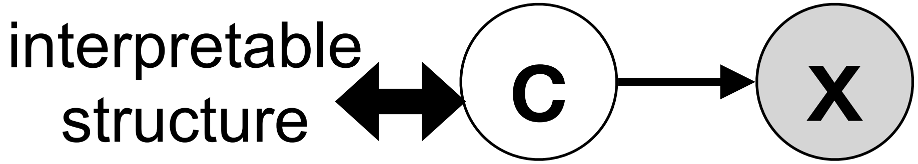
Will it be humid in New York today?

Remind me about my meeting.

difficult to interpret discrete factors

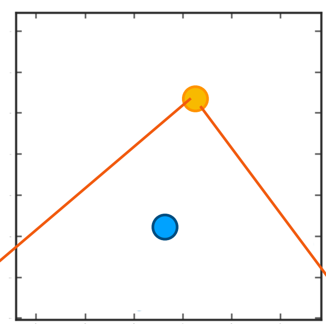
VAEs Introduce Latent Variables

Variational Auto-encoder (VAE)



(Zhao et al, 2018b)

c: **discrete** latent variables



Remind me about my meeting.

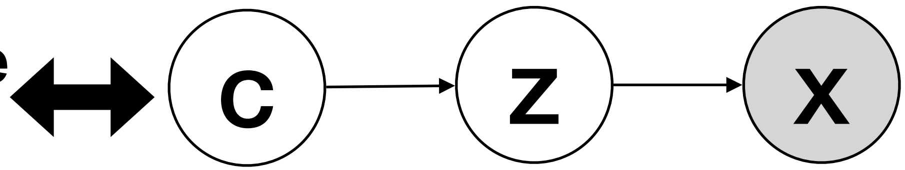
Remind me about the football game.

expressiveness is limited.

Discrete Variables Could Enhance Interpretability - but one has to do it right!

Gaussian Mixture Variational Auto-encoder (GM-VAE)

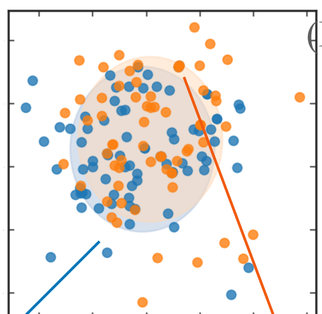
interpretable structure



(Dilokthanakul et al., 2016; Jiang et al., 2017)

c: **discrete** component

z: **continuous** latent variable



Will it be overcast tomorrow?

Remind me about the football game.

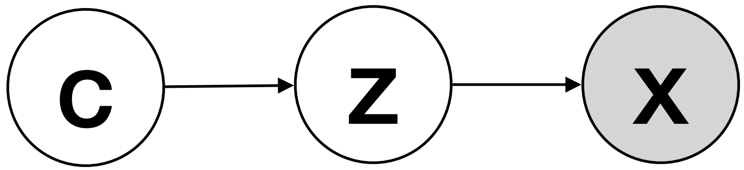
Why?

How to fix it?

mode-collapse

Do it right for VAE w/ hierarchical priors - Dispersed Exponential-family Mixture VAE

Exponential-family Mixture VAE



adding dispersion term in training

Dispersed EM-VAE

$$L(\theta; x) = \text{ELBO} + \beta \cdot \boxed{L_d},$$

dispersion term

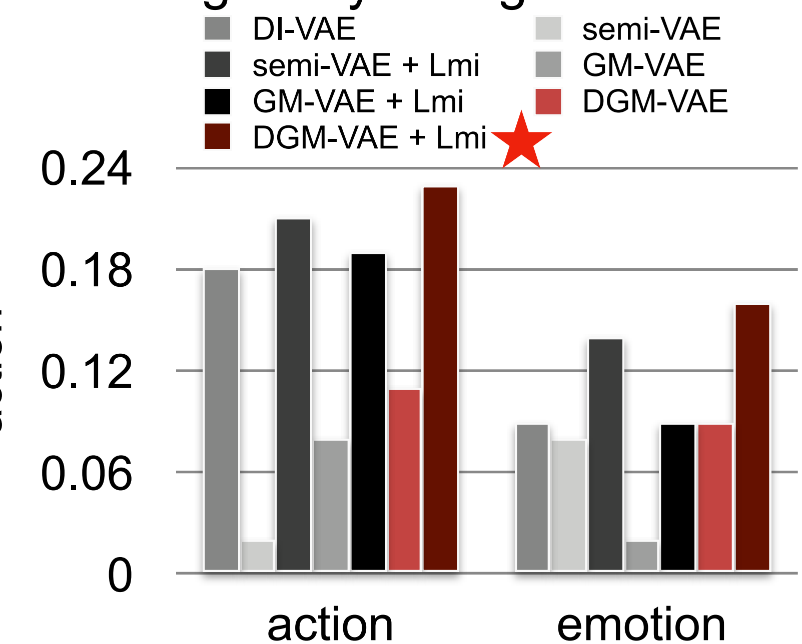
$$L_d = \mathbb{E}_{q_\phi(c|x)} A(\boldsymbol{\eta}_c) - A(\mathbb{E}_{q_\phi(c|x)} \boldsymbol{\eta}_c) \, .$$

DEM-VAE [W. Shi, H. Zhou, N. Miao, **Lei Li**, ICML 2020]

Generation Quality and Interpretability

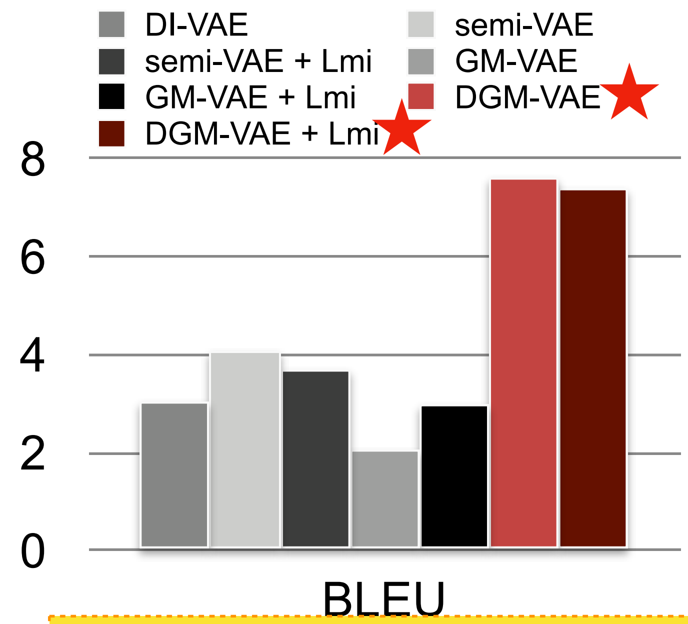
DGM-VAE obtains the best performance in interpretability and reconstruction

Homogeneity with golden label in DD



Best interpretability

BLEU of reconstruction in DD



Best reconstruction

DEM-VAE [W. Shi, H. Zhou, N. Miao, **Lei Li**, ICML 2020]

Latent Variables Learned by DEM-VAE are Semantically Meaningful

Example actions and corresponding utterances (classified by $q_\phi(c \mid x)$)

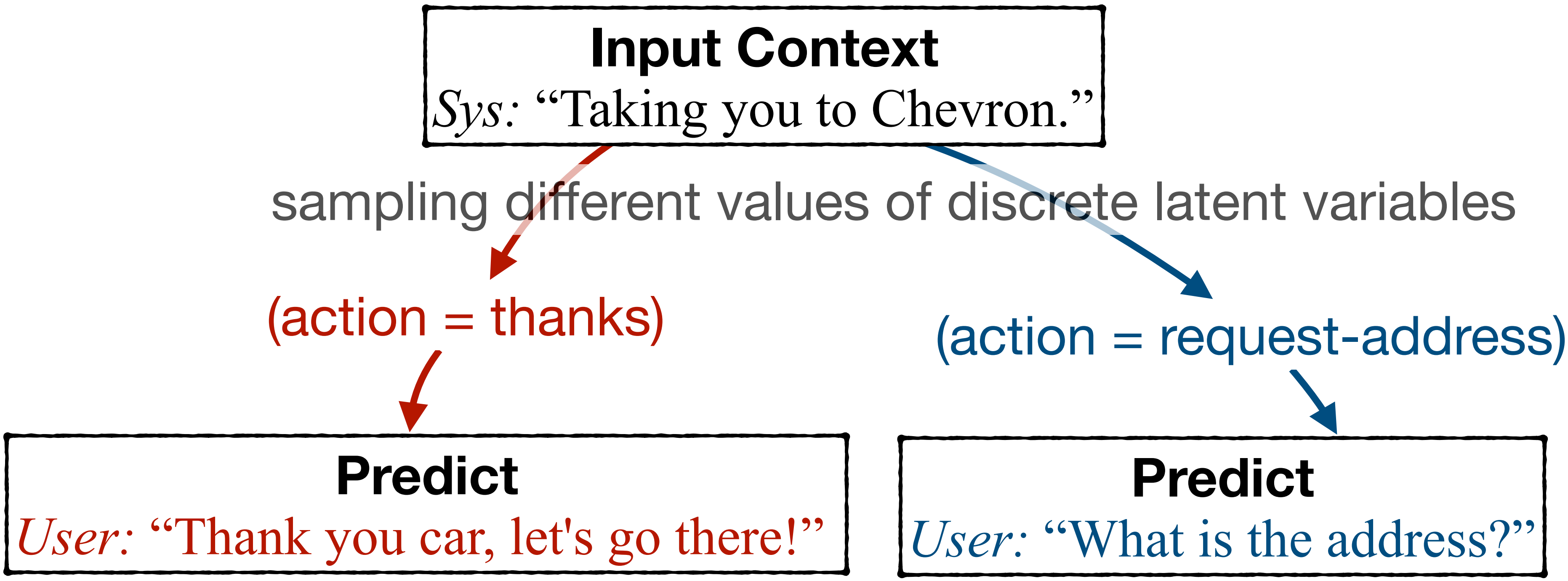
Inferred action=Inform-route/address
"There is a Safeway 4 miles away."
"There are no hospitals within 2 miles."
"There is Jing Jing and PF Changs."
…

Inferred action =Request-weather
"What is the weather today?"
"What is the weather like in the city?"
"What's the weather forecast in New York?"
…

Utterances of the same actions could be assigned with the same discrete latent variable c.

DEM-VAE [W. Shi, H. Zhou, N. Miao, **Lei Li**, ICML 2020]

Generate Sensible Dialog Response with DEM-VAE



Responses with different actions are generated by sampling different values of discrete latent variables.

DEM-VAE [W. Shi, H. Zhou, N. Miao, **Lei Li**, ICML 2020]

Summary

- Auto-Encoder: learning representation by reconstruction
- Variational Auto-Encoder: put prior on latent representation and use variational method to train

Next Up

- Industrial talk on computer vision