Variational Autoencoders

Nurul Lubis
Dialog Systems and Machine Learning Group

Content

Heinrich Heine
Universität Düsseldorf

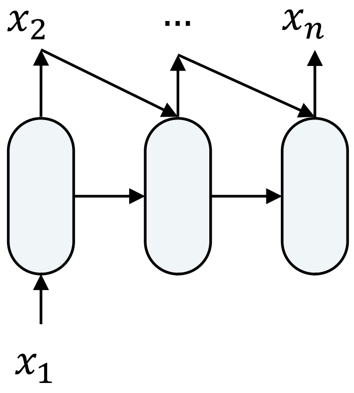
Generative Models

Heinrich Heine Universität Düsseldorf

- Given a set of training data, generate samples that are likely under the distribution
  - E.g. images, sentences
- Likelihood of training data

$$p(x) = \prod_{i=1}^{n} p\left(x_i | x_1, x_2, \dots, x_{i-1}\right)$$

- Model conditional distribution of a point given its context
  - charRNN (Sutskever et al., 2011)
  - LSTM (Graves, 2014)
  - PixelCNN (van den Oord et al., 2016)



Language generation with RNN

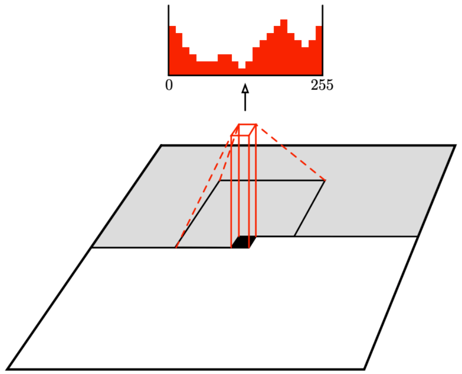


Image generation with PixelCNN (van den Oord et al., 2016)

Generative Models

- Given a set of training data, generate samples that are likely under the distribution
  - E.g. images, sentences
- Likelihood of training data

$$p(x) = \prod_{i=1}^{n} p\left(x_i | x_1, x_2, \ldots, x_{i-1}\right)$$

- Model conditional distribution of a point given its context
  - charRNN (Sutskever et al., 2011)
  - LSTM (Graves, 2014)
  - PixelCNN (van den Oord et al., 2016)

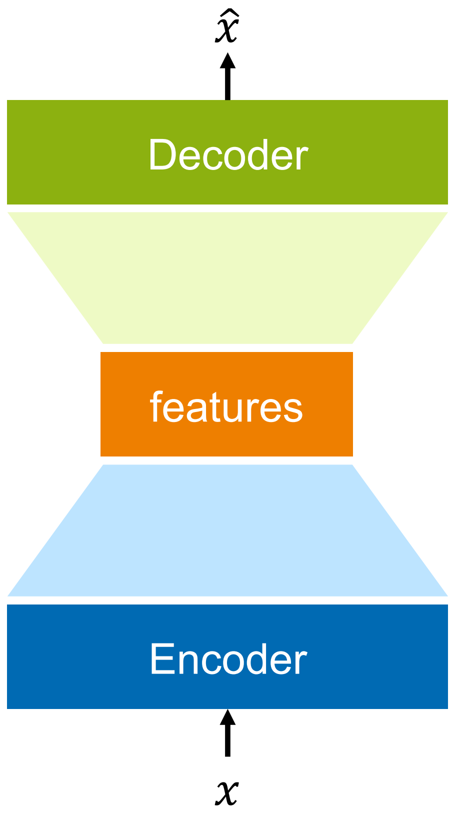
- Pros
  - Easy to optimize
  - Stable
- Cons
  - Sensitive to the choice of context
  - Do not provide rich code of the samples

Latent variable as structure in a generative process

Heinrich Heine Universität Düsseldorf

- Generation process could benefit from structure and hierarchy
  - When we write a digit, we decide beforehand which number to write
  - When we say something, we have an intent in mind to begin with
- Variational autoencoder (VAE) does this via the **latent variable** z in the model
  - Latent: unobserved
  - Tries to capture underlying structure of data
  - Makes a decision before performing the generation

Primer: Autoencoders

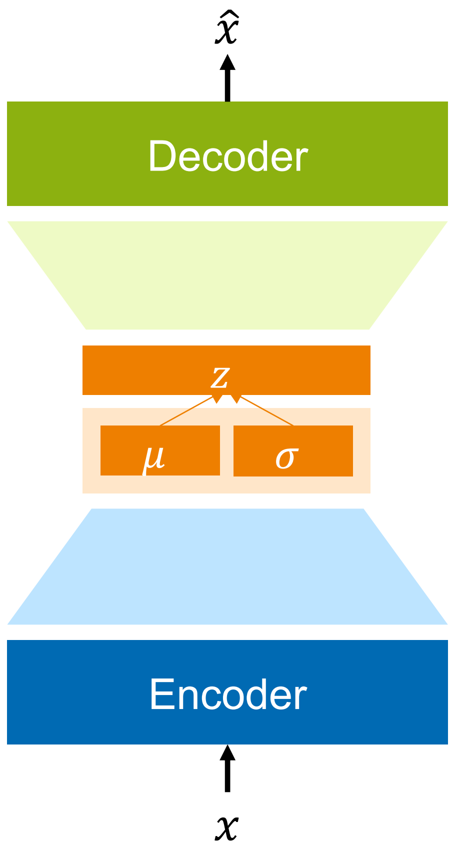


- **Unsupervisedly** learn condensed representation of data through autoencoding task
  - Encode the input into lower-dimensional latent features
  - These features should allow reconstruction of the input
  - Optimize model to minimize reconstruction loss, e.g.

$$L(x, \hat{x}) = \|x - \hat{x}\|^2$$

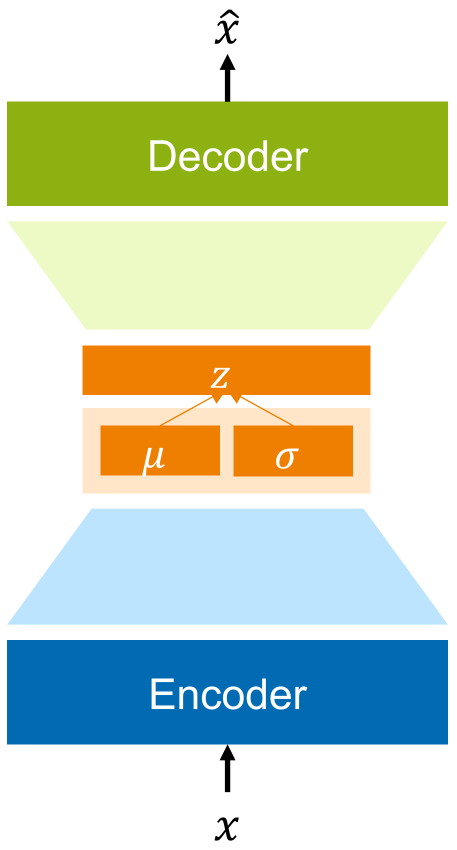
- AE gives features for reconstructing the data
  - The bottleneck forces the model to learn rich important features of the input by ignoring noise in the data
  - However, mapping between input and features are deterministic
    - Feature extraction
  - Can we modify the model such that we can generate more data from it?

Variational Autoencoders



- Instead of deterministic mapping, VAE models the **distribution** of the latent variables

Variational Autoencoders



- Decoder generates new data conditioned on z, i.e. $p_\theta(x|z)$, such that the new data resembles our training data
- a.k.a generation network

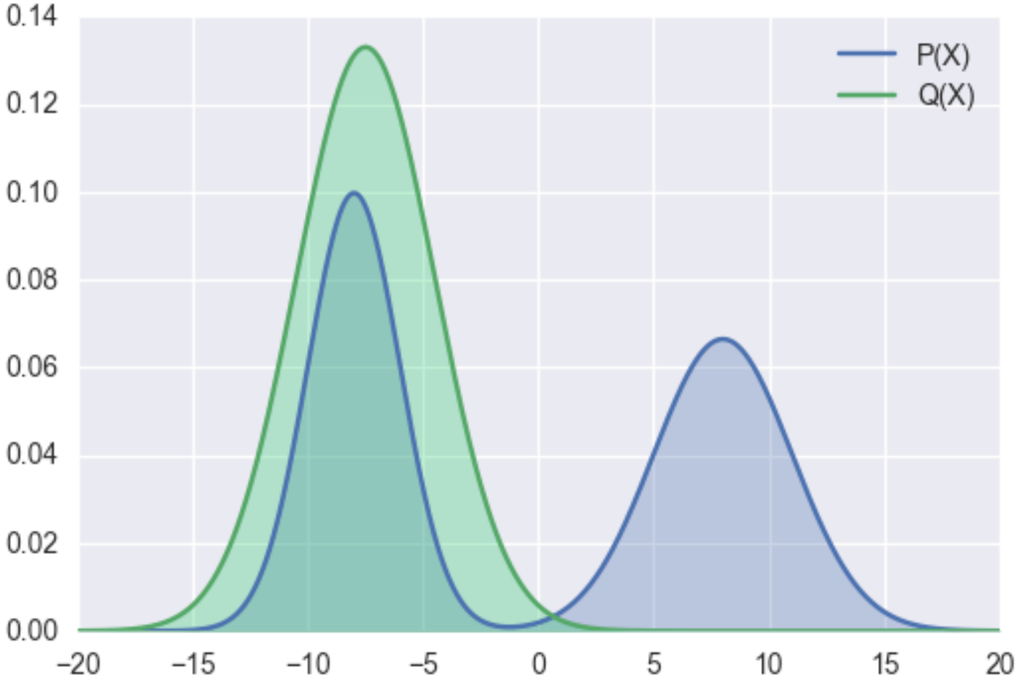
- Distribution of latent variable z
  - True posterior: $p_\theta(z|x)$ not known
  - Prior: $p_\theta(z)$, initial assumption about how z is distributed

- Encoder maps input x to a **distribution** $q_\phi(z|x)$
  - In case of gaussian, the encoder outputs vectors of means and std. dev from which we sample z
- a.k.a recognition network or inference network

Kullback-Leibler divergence

- A measure of distance between two probability distribution
  - Cross entropy minus entropy
  - $D_{KL}\big(Q(x) \parallel P(x)\big) = H(Q,P) - H(Q)$
    - $H(Q,P) = E_{x\sim Q} - \log P(x)$
    - $H(Q) = E_{x\sim Q} - \log\big(Q(x)\big)$

$$\boldsymbol{D_{KL}\big(Q(x) \parallel P(x)\big) = E_{x\sim Q} \log \frac{Q(x)}{P(x)}}$$



Source: https://wiseodd.github.io/techblog/2016/12/21/forward-reverse-kl/

VAE loss function: ELBO

$$\log p\left(x^{(i)}\right) = \mathrm{E}_{z\sim q_\phi(z|x)}\ \log p_\theta\left(x^{(i)}\right)$$

$$\mathrm{E}_z \log \frac{p_\theta\left(x^{(i)}|z\right)p_\theta(z)}{p_\theta\left(z|x^{(i)}\right)}$$

$$\mathrm{E}_z \log \frac{p_\theta\left(x^{(i)}|z\right)p_\theta(z)}{p_\theta\left(z|x^{(i)}\right)}\frac{q_\phi(z|x^{(i)})}{q_\phi(z|x^{(i)})}$$

$$\mathrm{E}_z \log p_\theta\left(x^{(i)}|z\right) - \mathrm{E}_z\ \log \frac{q_\phi(z|x^{(i)})}{p_\theta(z)} + \mathrm{E}_z\ \log \frac{q_\phi(z|x^{(i)})}{p_\theta(z|x^{(i)})}$$

$$\mathrm{E}_z \log p_\theta\left(x^{(i)}|z\right) - D_{\mathrm{KL}}\left(q_\phi\left(z|x^{(i)}\right) \parallel p_\theta(z)\right) + D_{\mathrm{KL}}\left(q_\phi\left(z|x^{(i)}\right) \parallel p_\theta\left(z|x^{(i)}\right)\right)$$

- Taking expectation
- Bayes' rule
- Multiply with constant
- Log rule
- KL terms

VAE loss function : ELBO

Heinrich Heine
Universität Düsseldorf

$$\log p\left(x^{(i)}\right) = \mathrm{E}_{z\sim q_\phi(z|x)}\ \log p_\theta\left(x^{(i)}\right)$$

$$\mathrm{E}_z \log \frac{p_\theta(x^{(i)}|z)p_\theta(z)}{p_\theta(z|x^{(i)})}$$

$$\mathrm{E}_z \log \frac{p_\theta(x^{(i)}|z)p_\theta(z)}{p_\theta(z|x^{(i)})} \frac{q_\phi(z|x^{(i)})}{q_\phi(z|x^{(i)})}$$

$$\mathrm{E}_z \log p_\theta\left(x^{(i)}|z\right) - \mathrm{E}_z \log \frac{q_\phi(z|x^{(i)})}{p_\theta(z)} + \mathrm{E}_z \log \frac{q_\phi(z|x^{(i)})}{p_\theta(z|x^{(i)})}$$

$$\mathrm{E}_z \log p_\theta\left(x^{(i)}|z\right) - D_{\mathrm{KL}}\left(q_\phi\left(z|x^{(i)}\right) \parallel p_\theta(z)\right) + D_{\mathrm{KL}}\left(q_\phi\left(z|x^{(i)}\right) \parallel p_\theta\left(z|x^{(i)}\right)\right)$$

decoder

encoder

z prior

z posterior, not known and intractable!

By definition, $D_{\mathrm{KL}} \geq 0$

- Taking expectation
- Bayes' rule
- Multiply with constant
- Log rule
- KL terms

VAE loss function: ELBO

$$\log p\left(x^{(i)}\right) = \mathrm{E}_{z\sim q_\phi(z|x)}\ \log p_\theta\left(x^{(i)}\right)$$

$$\mathrm{E}_z \log \frac{p_\theta\left(x^{(i)}|z\right)p_\theta(z)}{p_\theta\left(z|x^{(i)}\right)}$$

$$\mathrm{E}_z \log \frac{p_\theta\left(x^{(i)}|z\right)p_\theta(z)}{p_\theta\left(z|x^{(i)}\right)} \frac{q_\phi\left(z|x^{(i)}\right)}{q_\phi\left(z|x^{(i)}\right)}$$

$$\mathrm{E}_z \log p_\theta\left(x^{(i)}|z\right) - \mathrm{E}_z \log \frac{q_\phi\left(z|x^{(i)}\right)}{p_\theta(z)} + \mathrm{E}_z \log \frac{q_\phi\left(z|x^{(i)}\right)}{p_\theta\left(z|x^{(i)}\right)}$$

$$\log p\left(x^{(i)}\right) \geq \mathrm{E}_z \log p_\theta\left(x^{(i)}|z\right) - D_{\mathrm{KL}}\left(q_\phi\left(z|x^{(i)}\right) \parallel p_\theta(z)\right)$$

Evidence lowerbound (ELBO)

$\mathcal{L}(x^{(i)}, \theta, \phi)$

$$\theta^*, \phi^* = \arg\max \sum_{i=1}^{N} \mathcal{L}(x^{(i)}, \theta, \phi)$$

- Taking expectation
- Bayes' rule
- Multiply with constant
- Log rule
- KL terms

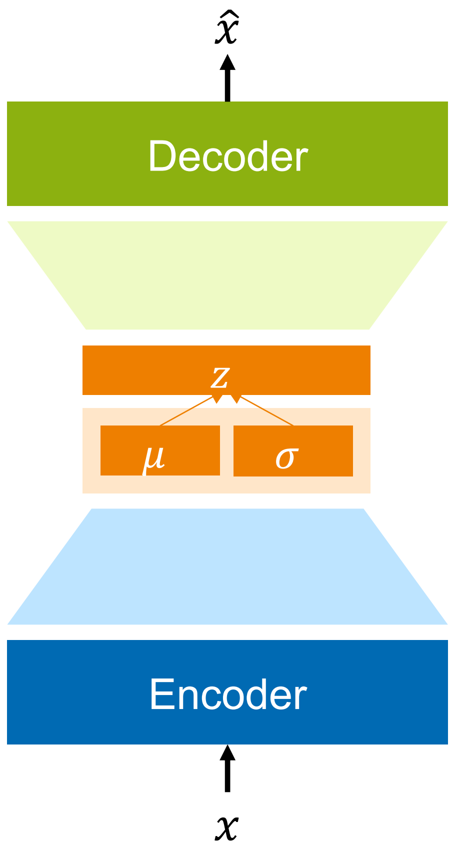
Prior

Heinrich Heine Universität Düsseldorf

$$\mathcal{L}(x^{(i)}, \theta, \phi) = \mathrm{E}_z \log p_\theta(x^{(i)}|z) - D_{\mathrm{KL}}\left(q_\phi(z|x^{(i)}) \parallel \boldsymbol{p_\theta}(\boldsymbol{z})\right)$$

- Prior: an assumption about how the latent is distributed
- For Gaussian-distributed latent, typically isotropic normal Gaussian is used as prior
  - Assumes that each latent variable is normally distributed
  - Zero mean, i.e. $\mu = 0$
  - The identity matrix as diagonal covariance matrix, i.e. $\Sigma = \boldsymbol{I}$
- The diagonal covariance pulls the encoded latent space $q_\phi(z|x)$ to have independent components
- $q_\phi(z|x)$ is penalized from diverging too far from this form
  - A form of regularization

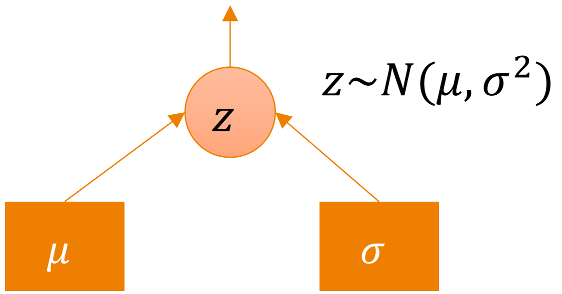
Optimizing VAEs



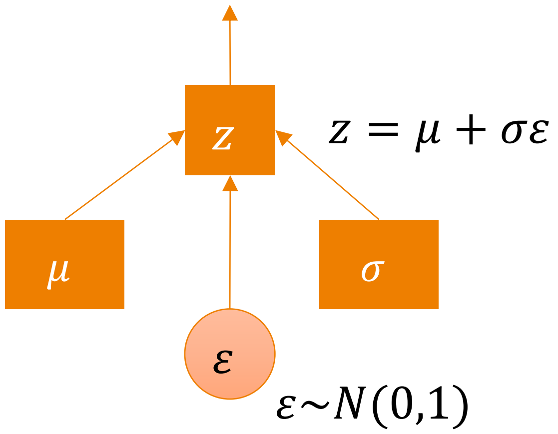
- VAE:
  - ✓ Encoder
  - ✓ Decoder
  - ✓ Loss Function

- Problem: can not backpropagate through stochastic layer
  - Not differentiable
- Solution: reparameterization trick

Reparameterization trick



Without reparameterization trick ☹



With reparameterization trick ☺

- Main idea: all Gaussian distributions are scaled and translated versions of the normal distribution
- To draw from $N(\mu,\sigma^2)$:
  - Draw from $N(0,1)$
  - Scale with σ (multiplication)
  - Translate with μ (addition)
- Shifting the stochasticity in z to a parameter-independent node
  - We do not require any backpropagation through ε
  - Now we can train with standard NN optimization algorithms

Relationship between z and x

(a) Learned Frey Face manifold

(Kingma and Welling, 2014)

- Each dimension of z represent a meaningful characteristic of the data
- Example
  - face rotation (x-axis)
  - smile (y-axis)

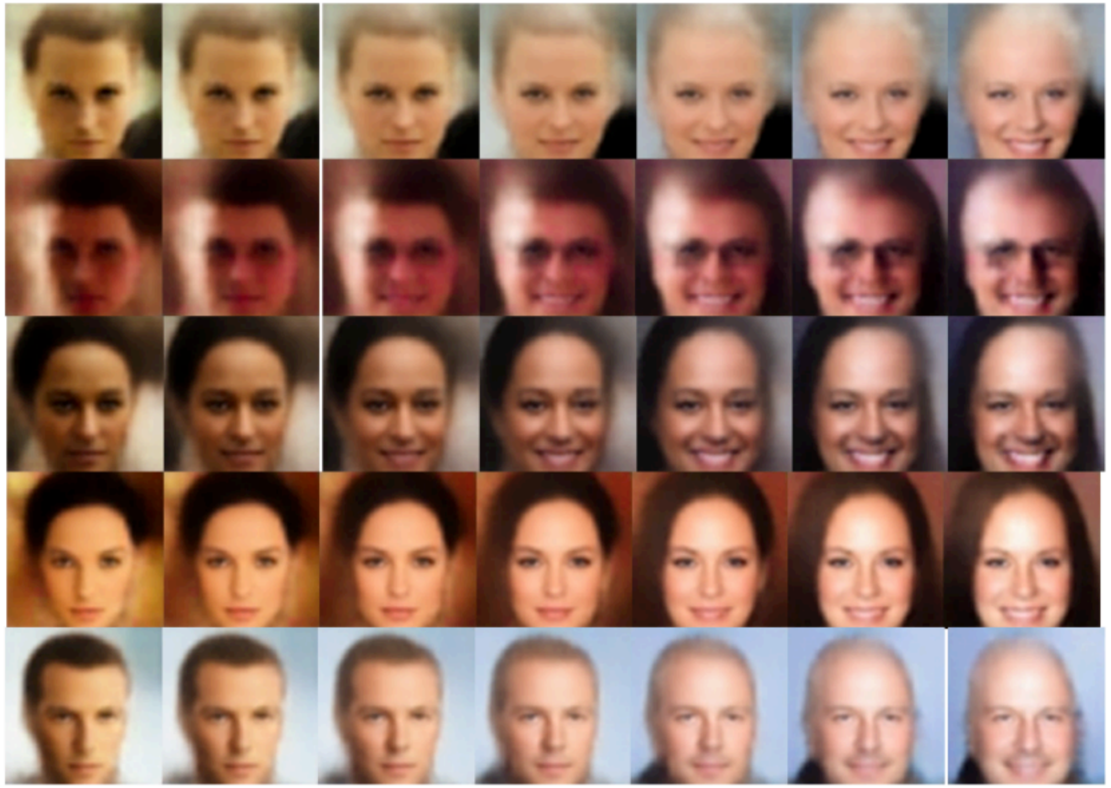
Dimensionality of z

(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

Figure 5: Random samples from learned generative models of MNIST for different dimensionalities of latent space.

(Kingma and Welling, 2014)

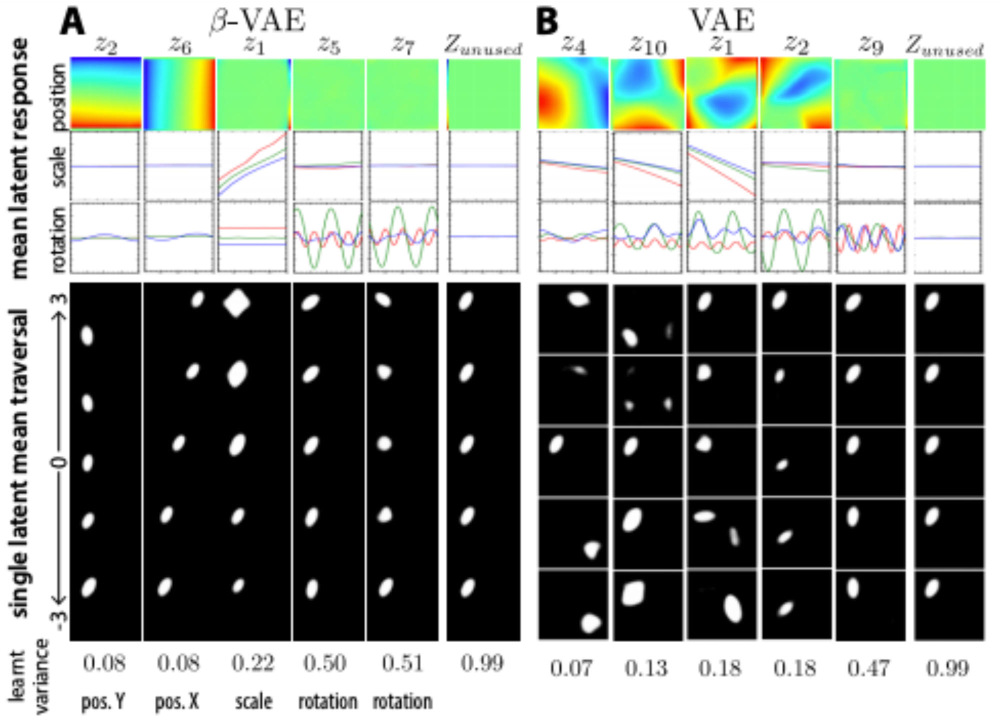
Disentangling the latent space



(Higgins et al., 2017)

- Ideally, we want each latent dimension to encode a single generative factor
- VAE tend to map multiple generative factors into one dimension
- Example: traversing latent dimension which controls smile causes other changes in the generated image
  - Difficult to interpret each dimension
  - Less generative control

Disentangling the latent space



(Higgins et al., 2017)

- β-VAE (Higgins et al., 2017) disentangle the latent dimensions by modifying the objective
  $$\mathrm{E}_z \log p_\theta(x^{(i)}|z) - \beta D_{\mathrm{KL}}\left(q_\phi(z|x^{(i)}) \parallel p_\theta(z)\right)$$
  - Proposal: set $\beta > 1$
- Intuition: KL term can be viewed as the upper limit of the representation capacity of z (Burgess et al., 2018)
  - Setting $\beta > 1$ means increasing the penalty, decreasing channel capacity
  - Decreased capacity encourages condensed representation
    - For some conditionally independent generative factor, best strategy is to keep them separate

Improving representation learning in VAEs

- Active area of research!
- Disentanglement is one of 4 meta-priors (Bengio et al., 2012)

Disentanglement	Hierarchy	Semi-supervised learning	Clustering structure

- A survey paper on representation learning with VAE (Tschannen et al., 2018)

Conditional VAE (CVAE)

- During generative process with VAE, z is sampled from the prior
  - Not possible to specify what kind of sample to generate
- CVAE models data and its latent conditioned on some random variables (Sohn et al., 2015)
- VAE objective:

$$\mathcal{L}\left(x^{(i)}, \theta, \phi\right) = \mathrm{E}_z \log p_\theta\left(x^{(i)} | z\right) - D_{\mathrm{KL}}\left(q_\phi\left(z \middle| x^{(i)}\right) \parallel p_\theta(z)\right)$$

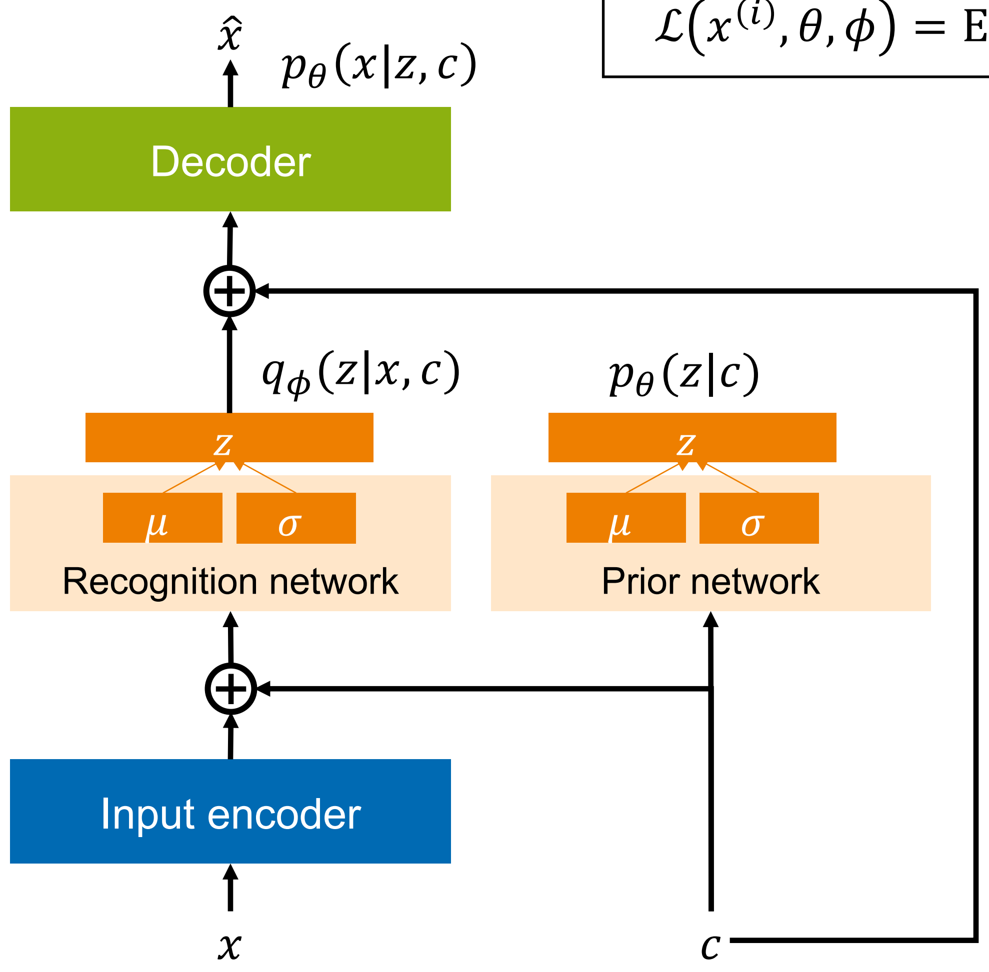
- CVAE objective:

$$\mathcal{L}\left(x^{(i)}, \theta, \phi\right) = \mathrm{E}_z \log p_\theta\left(x^{(i)} | z, c^{(i)}\right) - D_{\mathrm{KL}}\left(q_\phi\left(z \middle| x^{(i)}, c^{(i)}\right) \parallel p_\theta\left(z \middle| c^{(i)}\right)\right)$$

  - The latent distribution is also conditioned on input observation, e.g. labels
  - CVAE has an additional network, called **prior network** which models z conditioned on c, i.e. $p_\theta\left(z | c^{(i)}\right)$

CVAE

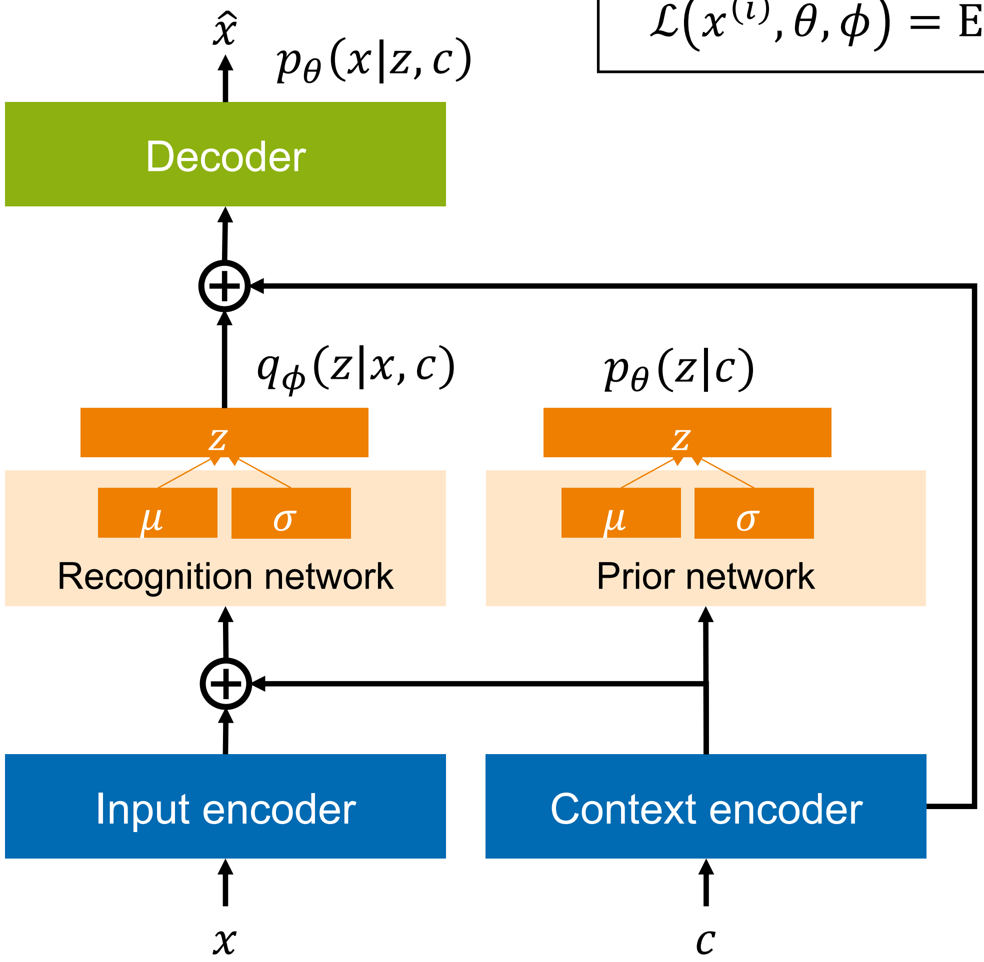
$$\mathcal{L}(x^{(i)}, \theta, \phi) = \mathrm{E}_z \log p_\theta(x^{(i)}|z, c^{(i)}) - D_{\mathrm{KL}}\left(q_\phi(z|x^{(i)}, c^{(i)}) \parallel p_\theta(z|c^{(i)})\right)$$



- During training, minimize $D_{\mathrm{KL}}\left(q_\phi(z|x,c) \parallel p_\theta(z|c)\right)$
  - The distance between recog. and prior network distributions

CVAE

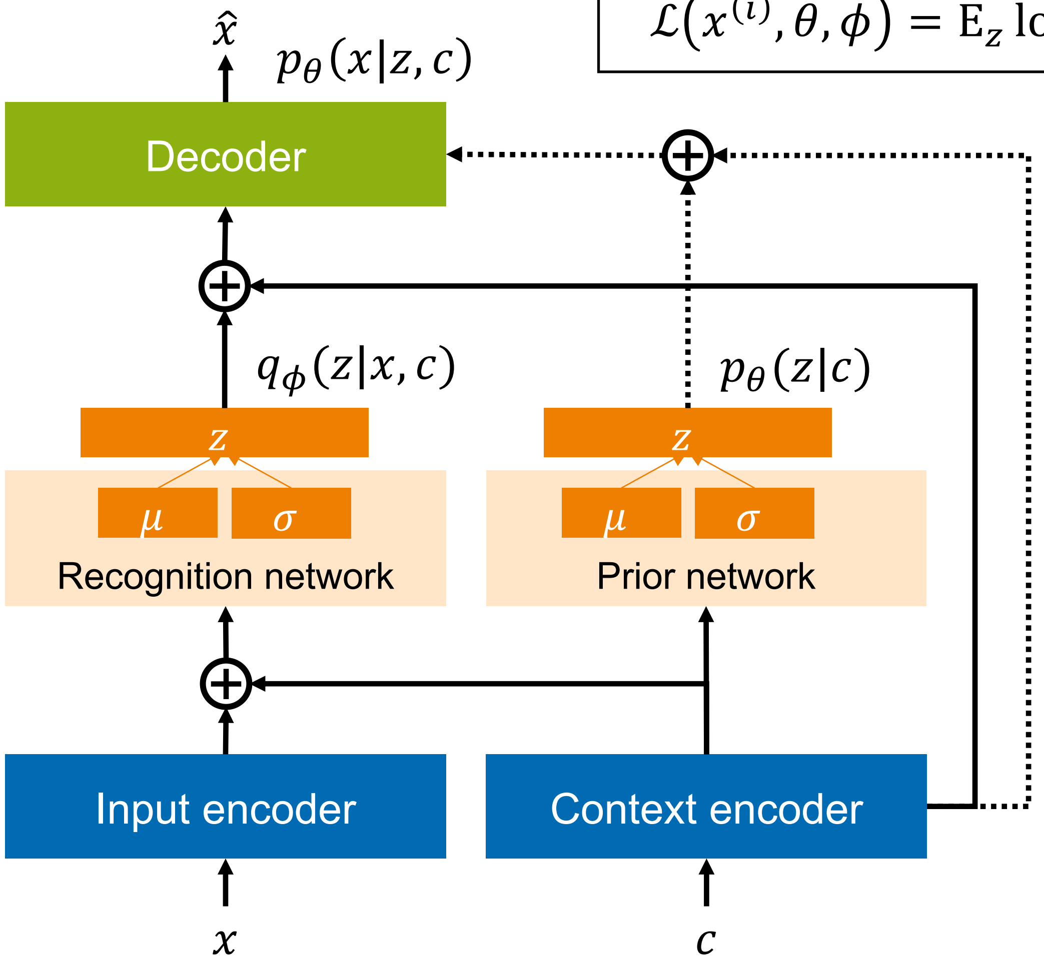
$$\mathcal{L}(x^{(i)},\theta,\phi) = \mathrm{E}_z \log p_\theta(x^{(i)}|z,c^{(i)}) - D_{\mathrm{KL}}\left(q_\phi(z|x^{(i)},c^{(i)}) \parallel p_\theta(z|c^{(i)})\right)$$



- During training, minimize $D_{\mathrm{KL}}\left(q_\phi(z|x,c) \parallel p_\theta(z|c)\right)$
  - The distance between recog. and prior network distributions

CVAE

$$\mathcal{L}(x^{(i)}, \theta, \phi) = \mathrm{E}_z \log p_\theta(x^{(i)} | z, c^{(i)}) - D_{\mathrm{KL}}\left(q_\phi(z | x^{(i)}, c^{(i)}) \parallel p_\theta(z | c^{(i)})\right)$$



- During training, minimize $D_{\mathrm{KL}}\left(q_\phi(z|x,c) \parallel p_\theta(z|c)\right)$
  - The distance between recog. and prior network distributions
- During generation, sample z via $p_\theta(z|c)$

Application in NLP

Heinrich Heine
Universität Düsseldorf

- Sentence generation from continuous latent space (Bowman et al., 2015)
  - Sequential generation does not capture higher level concept e.g. topic and intent
  - Latent variables provide this concept
- Unlike images, decoder outputs discrete tokens

AE

i went to the store to buy some groceries .
i store to buy some groceries .
i were to buy any groceries .
horses are to buy any groceries .
horses are to buy any animal .
horses the favorite any animal .
horses the favorite favorite animal .
horses are my favorite animal .

VAE

no .
he said .
" no , " he said .
" no , " i said .
" i know , " she said .
" thank you , " she said .
" come with me , " she said .
" talk to me , " she said .
" do n't worry about it , " she said .

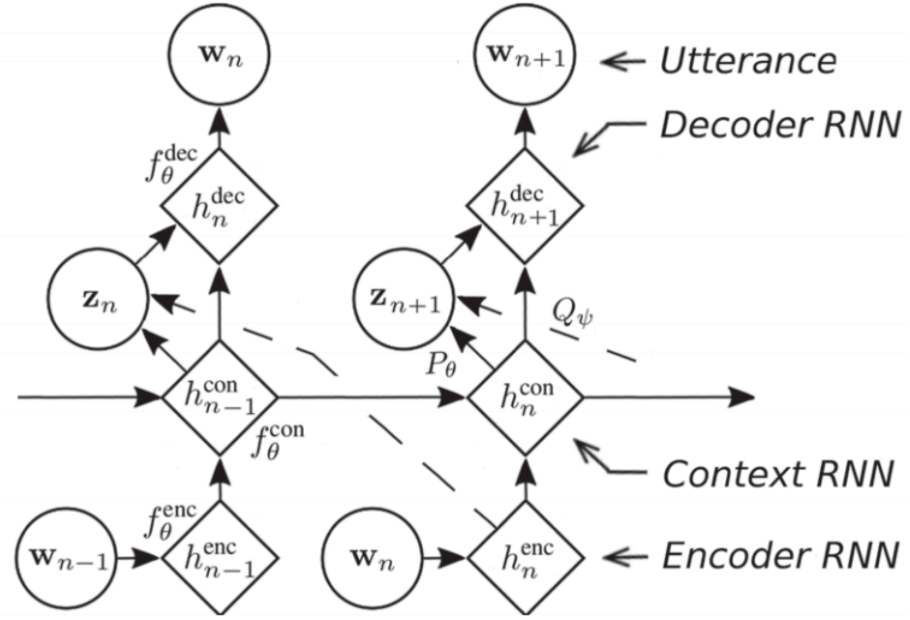
- Challenges
  - The model tends to favor "low hanging fruit" of behaving as a vanilla RNNLM and ignoring the latent variable

$$\mathrm{E}_z \log p_\theta\left(x^{(i)}|z\right) - D_{\mathrm{KL}}\left(q_\phi\left(z|x^{(i)}\right) \parallel p_\theta(z)\right)$$

Simply work on this   Make this zero

- Training strategies
  - *KL annealing* to encourage the model to pass information through z
    - Gradate the KL term weight through training
  - *Word dropout* to encourage the decoder to rely on z
    - Randomly replace words during decoding to <UNK>

Application in dialogue



- Hierarchical Latent Variable Encoder-Decoder (Serban et al., 2017)
- Two hierarchy of sequence:
  - Dialogue as sequence of turns
  - Each turn is a sequence of words
- Maximize likelihood of next turn given dialogue context
- ELBO is modified to include dialogue context

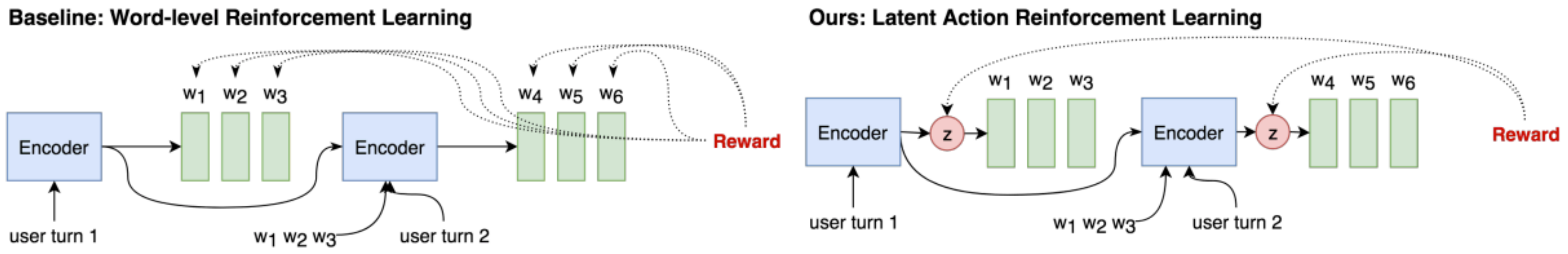
$$\log P_\theta(w_1, \dots w_N) \geq \sum_{n=1}^{N} -D_{KL}\left(Q_\psi(z_n|w_1, \dots, w_n) \parallel P_\theta(z_n|w_{<n})\right) + E_{Q_\psi(z_n|w_1, \dots, w_n)} \log P_\theta(w_n|z_n, w_{<n})$$

Latent is conditioned on previous turns

Generation is conditioned on latent and previous turns

Application in dialogue

Heinrich Heine Universität Düsseldorf



- Latent action reinforcement learning (Zhao et al., 2019)
  - Train a CVAE for dialogue, and perform RL on the latent space
- Shortening the trajectory when performing RL in dialogue
  - Instead of propagating reward to sequence of words $[(w_1, w_2, w_3), (w_4, w_5, w_6)]$, use the latent variable z

Conclusion

- Pros
  - Can generate new data
  - Provides structure in generation
  - Representation learning in latent space
- Cons
  - Requires an assumption about the underlying structure (expressed in the prior)
  - Can not be directly optimized
- Other generative methods?
  - GANs circumvent the explicit definition of density while keeping the ability to sample
    - Trade-off between some pros and cons
- Potentials
  - Analysis and visualization
    - Extract and plot latent structure of data
  - Semi-supervised learning
    - Use unsupervisedly learned representation to support supervised learning (Kingma et al., 2014)
  - Transfer learning
    - Use representation learned from a rich-resource task to complete low-resource tasks (Belhaj et al., 2018)
  - Reinforcement learning
    - Use representation learning for state space abstraction (Higgins et al., 2017)
  - ...and more

Heinrich Heine
Universität Düsseldorf

Thank you

References

- Kingma, D. P. and Welling, M. (2013). Auto-encoding variational Bayes. Proceedings of the 2nd International Conference on Learning Representations.
- Van den Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. (2016, June). Pixel Recurrent Neural Networks. In *International Conference on Machine Learning* (pp. 1747-1756).
- Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., & Graves, A. (2016). Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems* (pp. 4790-4798).
- Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
- Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 1017-1024).
- Sohn, K., Lee, H., & Yan, X. (2015). Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems* (pp. 3483-3491).
- Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., & Lerchner, A. (2018). Understanding disentangling in $\beta $-VAE. *arXiv preprint arXiv:1804.03599*.
- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., ... & Lerchner, A. (2017). beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *Iclr*, *2*(5), 6.
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798-1828.

References

- Tschannen, M., Bachem, O., & Lucic, M. (2018). Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*.
- Bowman, S., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., & Bengio, S. (2016, August). Generating Sentences from a Continuous Space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning* (pp. 10-21).
- Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A., & Bengio, Y. (2017, February). A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence.*
- Zhao, T., Zhao, R., & Eskenazi, M. (2017, July). Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 654-664).
- Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems* (pp. 3581-3589).
- Higgins, I., Pal, A., Rusu, A., Matthey, L., Burgess, C., Pritzel, A., ... & Lerchner, A. (2017, August). Darla: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 1480-1490). JMLR. org.
- Belhaj, M., Protopapas, P., & Pan, W. (2018). Deep variational transfer: Transfer learning through semi-supervised deep generative models. *arXiv preprint arXiv:1812.03123*

Intractability

- Data likelihood

$$p(x) = \int p(x|z)p(z)\,dz$$

- Posterior density

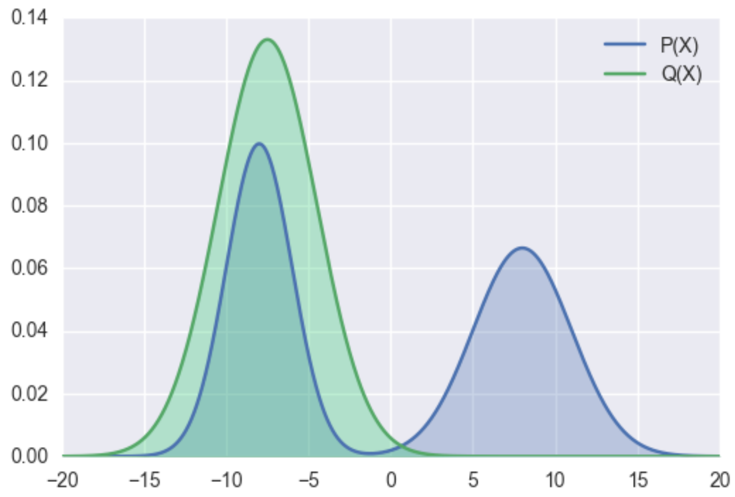
$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

Kullback-Leibler divergence

- A measure of distance between two probability distribution
  - Cross entropy minus entropy
  - $D_{KL}(Q(x) \parallel P(x)) = H(Q,P) - H(Q)$
    - $H(Q,P) = E_{x \sim Q} - \log P(x)$
    - $H(Q) = E_{x \sim Q} - \log(Q(x))$

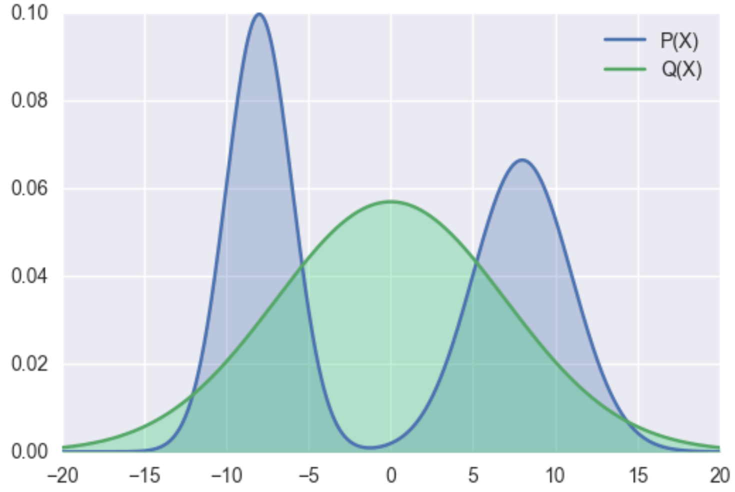
$$\boldsymbol{D_{KL}(Q(x) \parallel P(x)) = E_{x \sim Q} \log \frac{Q(x)}{P(x)}}$$

- Asymmetric!
  - $D_{KL}(Q(x) \parallel P(x)) \neq D_{KL}(P(x) \parallel Q(x))$
  - The first distribution act as “weight”
  - Typical notation: $P(x)$ for true distribution and $Q(x)$ for approximation



$D_{KL}(Q(x) \parallel P(x))$

Reverse KL yields closer distance
- Accepts smaller coverage in favor of good approximation



$D_{KL}(P(x) \parallel Q(x))$

Forward KL yields closer distance
- “zero avoiding”

Source: https://wiseodd.github.io/techblog/2016/12/21/forward-reverse-kl/