Variational Autoencoder[1][2]

Presented by Prashnna K. Gyawali, Computational Biomedicine Lab @ RIT

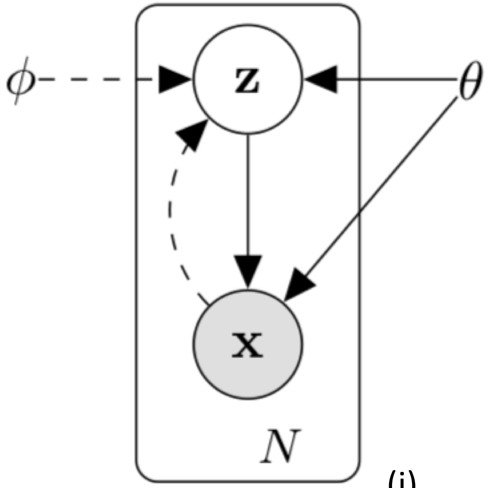
“.. marry ideas from deep neural networks and approximate Bayesian inference to derive a generalized class of deep, directed generative models ...”

[1] Auto-Encoding Variational Bayes. *Kingma and Welling. ICLR 2014.*

[2] Stochastic backpropagation and approximate inference in deep generative models. *Rezende et. al., ICML 2014.*

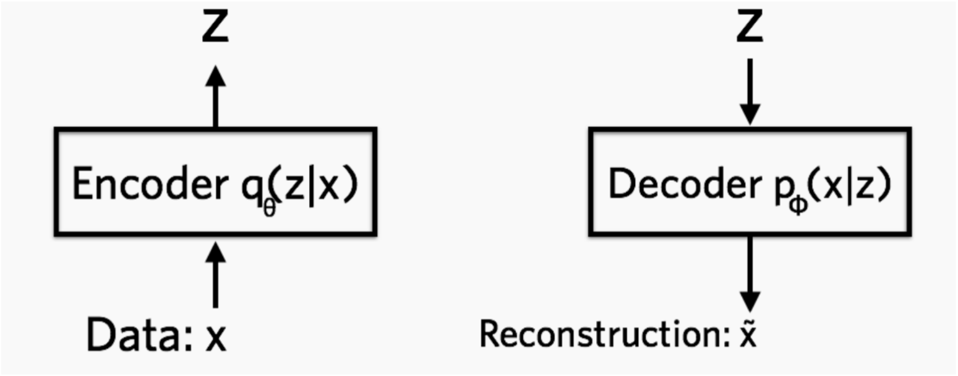
Perspective

- Probabilistic



(i)

- Deep nets



(ii)

Probabilistic model

- Intuition
  - We want to generate **the data: x**
  - What kind of data?
  - Imagination = **latent variable: z**

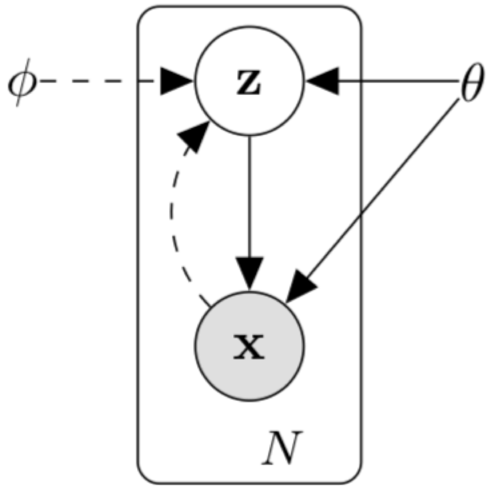
- Joint probability of data x and latent variables z:

$$p(x, z) = p(x|z)p(z)$$

  - For each datapoint *i*,

    Draw latent variables $z_i \sim p(z)$

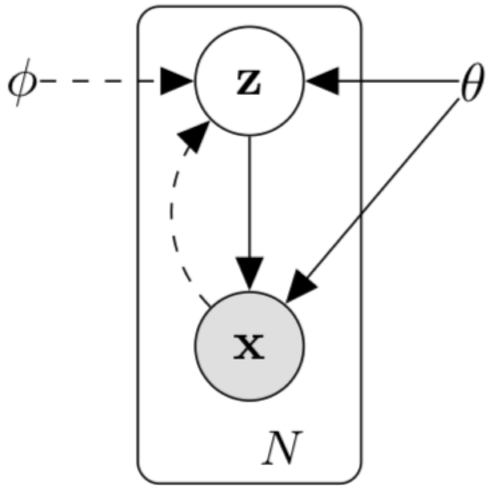
    Draw datapoint $x_i \sim p(x|z)$



Probabilistic model

- Inference
  - Goal: infer good values of the latent variables given observed data
  - Bayes rule:

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

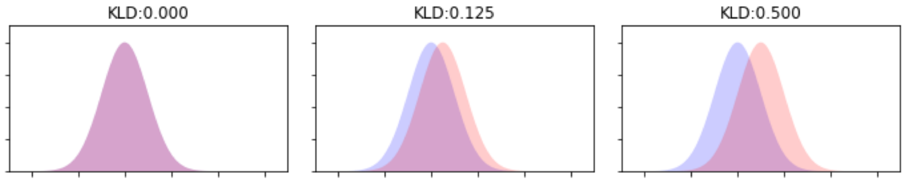


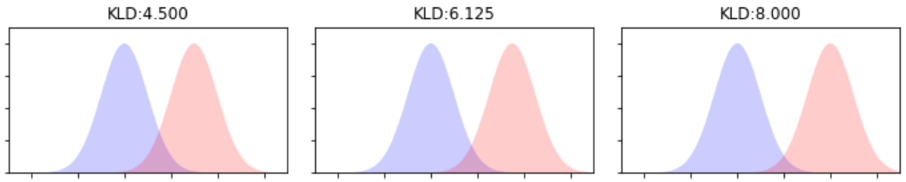
  - Evidence: $p(x) = \int p(x|z)p(z)dz$
    - Problem scenario:
      a. Intractability
      b. A large dataset

Probabilistic model

- Variational Inference
  - Idea: modeling the true distribution $p(z|x)$ using simpler distribution $q(z|x)$ *(proposal distribution)* that is easy to evaluate.

  - How to 'measure' how well well our variational posterior $q(z|x)$ approximates the true posterior?
    - Kullback-Leibler (KL) divergence
      - measures the information lost when using q to approximate p (in units of nats).
      - $D_{KL}(q||p) \;=\; \int q(\cdot)\log\frac{p(\cdot)}{q(\cdot)}dx$





(iii)

Probabilistic model

Integral Problem

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

Proposal distribution

$$p(x) = \int p(x|z)p(z)\boxed{\frac{q(z|x)}{q(z|x)}}dz$$

Taking log

$$\boxed{\log} p(x) = \boxed{\log} \int p(x|z)\frac{p(z)}{q(z|x)}q(z|x)dz$$

Jensen's inequality

$$\log \int p(x)g(x)dx \geq \int p(x) \log g(x)dx$$

$$\boxed{\geq} \int q(z|x) \boxed{\log} (\frac{p(z)}{q(z|x)}p(x|z))\, dz$$

$$= \int q(z|x) \log p(x|z) - \int q(z|x) \log \frac{q(z|x)}{p(z)}$$

Variational lower bound

$$\boxed{\log p(x) \geq E_{q(Z|x)}[\log p(x|z)] - KL[q(z|x)||p(z)]}$$

Probabilistic model

- Variational lower bound (ELBO)

$$\log p(x) \geq ELBO = E_{q(z|x)}[\log p(x|z)] - KL[q(z|x)||p(z)]$$

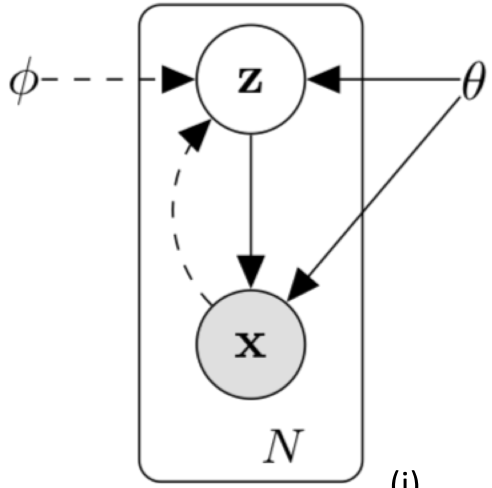
- Note:

a) $\log p(x) = ELBO + KL[q(z|x)||p(z|x)]$

b) KL is always non-negative, so maximizing $ELBO$ is equivalent to minimizing $KL[q(z|x)||p(z|x)]$

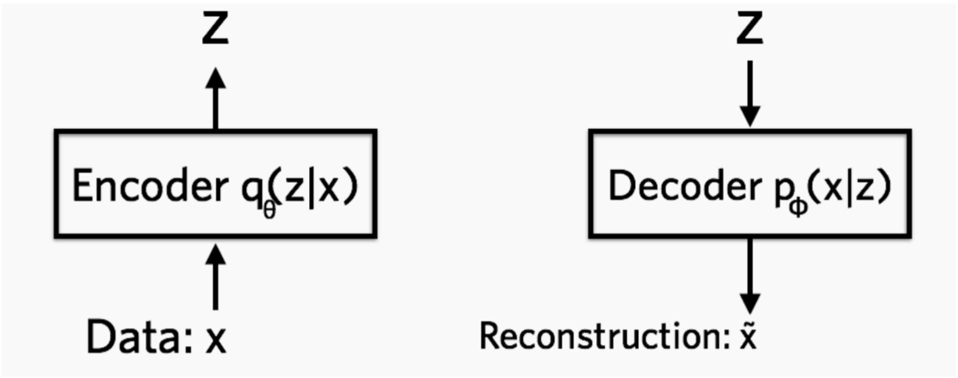
Perspective

- Probabilistic



(i)

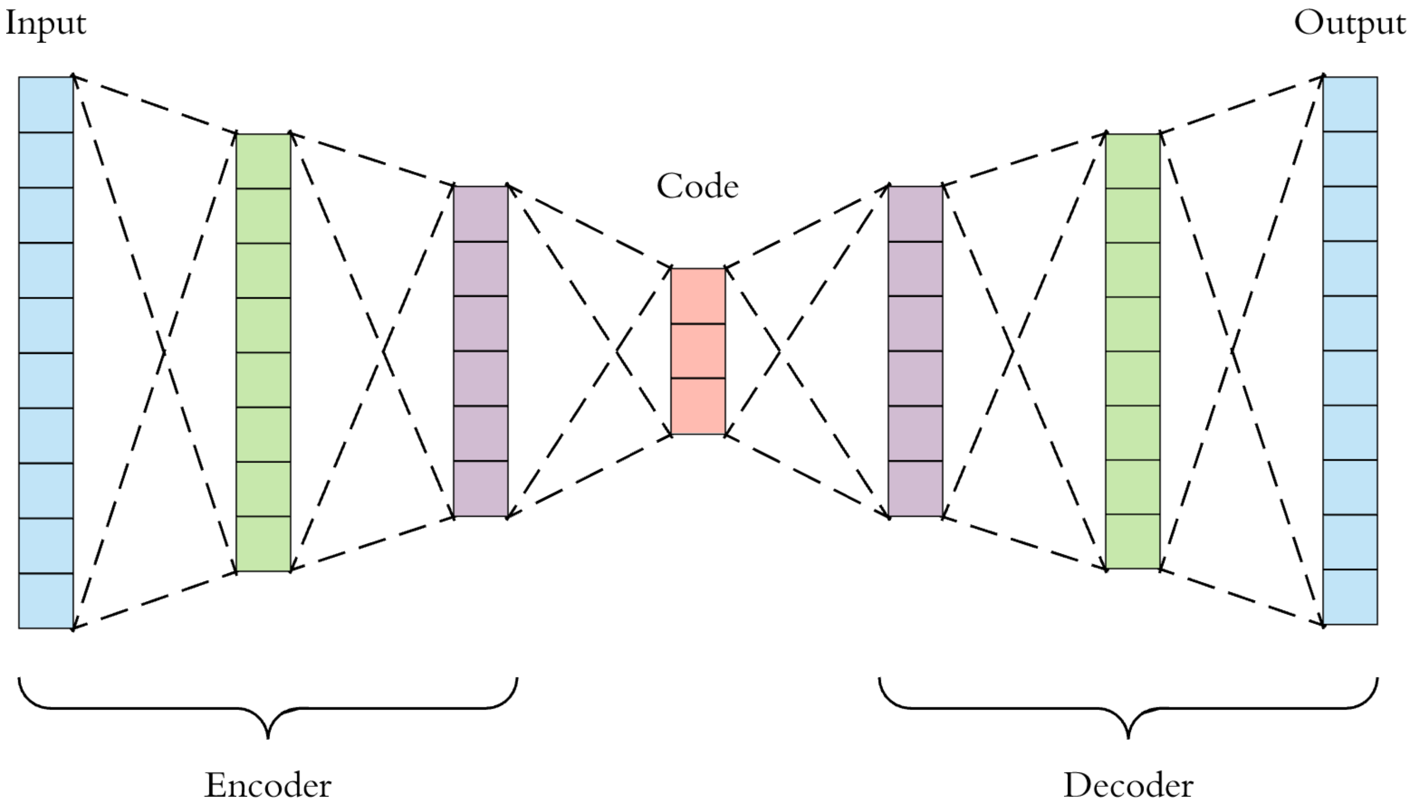
- Deep nets



(ii)

Deep nets

- Autoencoder → Encoder + Decoder

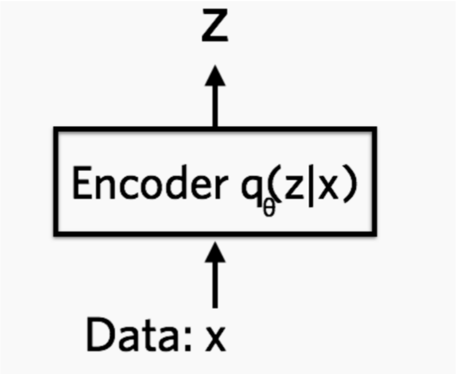


(iv)

Deep nets

$$\log p(x) \geq ELBO = E_{\boxed{q(z|x)}}[\log p(x|z)] - KL[\boxed{q(z|x)}||p(z)]$$

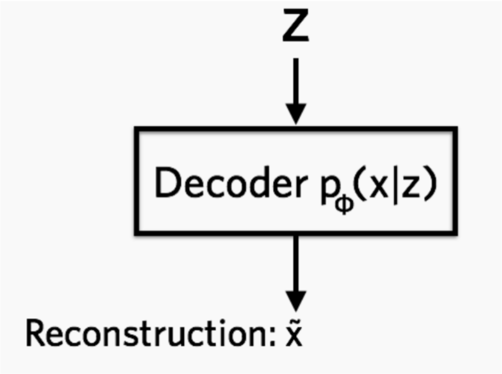
- Encoder
  - encode data $\boldsymbol{x}$ into latent representation $\boldsymbol{z}$
  - inference of $\boldsymbol{z}$ given $\boldsymbol{x}$ i.e. $q(z|x)$



Deep nets

$$\log p(x) \geq ELBO = E_{q(z|x)}[\log \boxed{p(x|z)}] - KL[q(z|x)||p(z)]$$

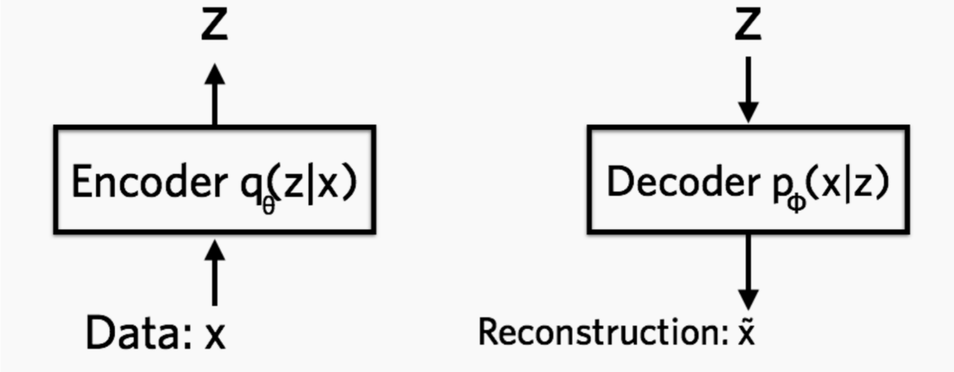
- Decoder
  - decode latent representation $\boldsymbol{z}$ into data $\boldsymbol{x}$
  - generating data $\boldsymbol{x}$ given $\boldsymbol{z}$ i.e. $p(x|z)$



Deep nets

$$\log p(x) \geq ELBO = E_{q(z|x)}[\log p(x|z)] - KL[q(z|x)||p(z)]$$

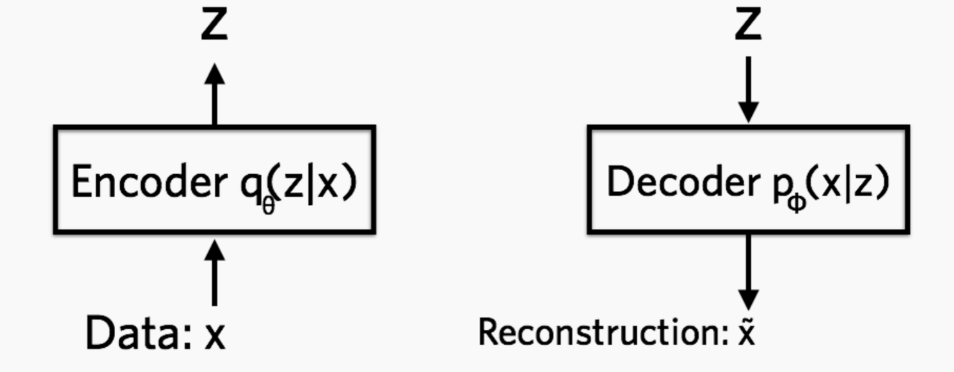
- Variational Autoencoder → Encoder + Decoder
  - Objective: Minimize reconstruction error + Regularizer on latent space



Deep nets

$$\log p(x) \geq ELBO = \boxed{E_{q(z|x)}[\log p(x|z)]} - KL[q(z|x)||p(z)]$$

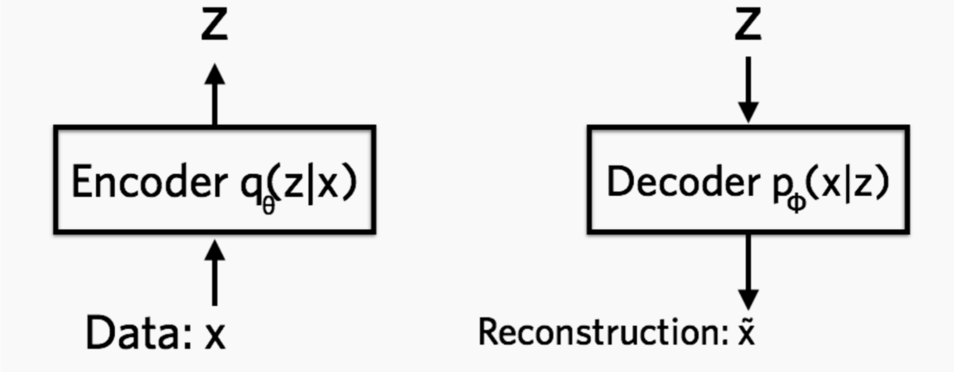
- Variational Autoencoder → Encoder + Decoder
  - Objective: Minimize reconstruction error + Regularizer on latent space



Deep nets

$$\log p(x) \geq ELBO = E_{q(z|x)}[\log p(x|z)] - \boxed{KL[q(z|x)||p(z)]}$$

- Variational Autoencoder → Encoder + Decoder
  - Objective: Minimize reconstruction error + Regularizer on latent space



Optimizing the objective

$$\log p(x) \geq ELBO = E_{q(z|x)}[\log p(x|z)] - KL[q(z|x)||p(z)]$$

- $\boldsymbol{q(z|x)}$: multivariate* Gaussian with a diagonal covariance structure:

$$q(z|x) = \mathcal{N}\,(z; \mu, \sigma^2 I)$$

where μ and σ^2 are the output of the deep nets.

- $\boldsymbol{\log p(x|z)}$: Bernoulli or Gaussian, depending on the data.

- $\boldsymbol{p(z)}$ as multivariate Gaussian (e.g. $\mathcal{N}(0, I)$), the KL divergence from another multivariate Gaussian $q(z|x)$ has differentiable form as

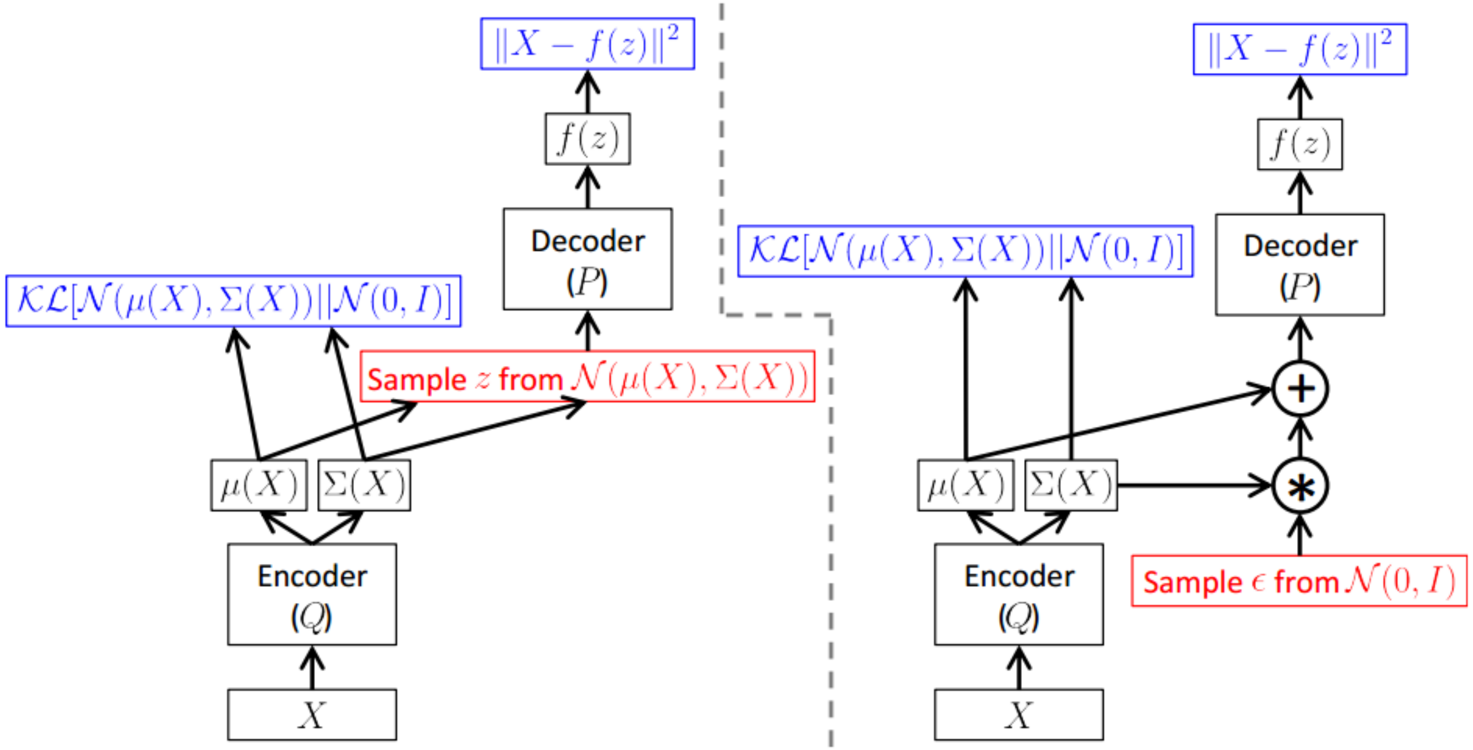
$$D_{\mathrm{KL}}(\mathcal{N}_0 \| \mathcal{N}_1) = \frac{1}{2}\left(\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^\top \Sigma_1^{-1}(\mu_1 - \mu_0) - k + \ln\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right)\right)$$

Dimension of latent space

Optimizing the objective

- Re-parameterization trick:

$$\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma}^2 \odot \boldsymbol{\varepsilon}$$



(v)

Implementation: pseudo-code

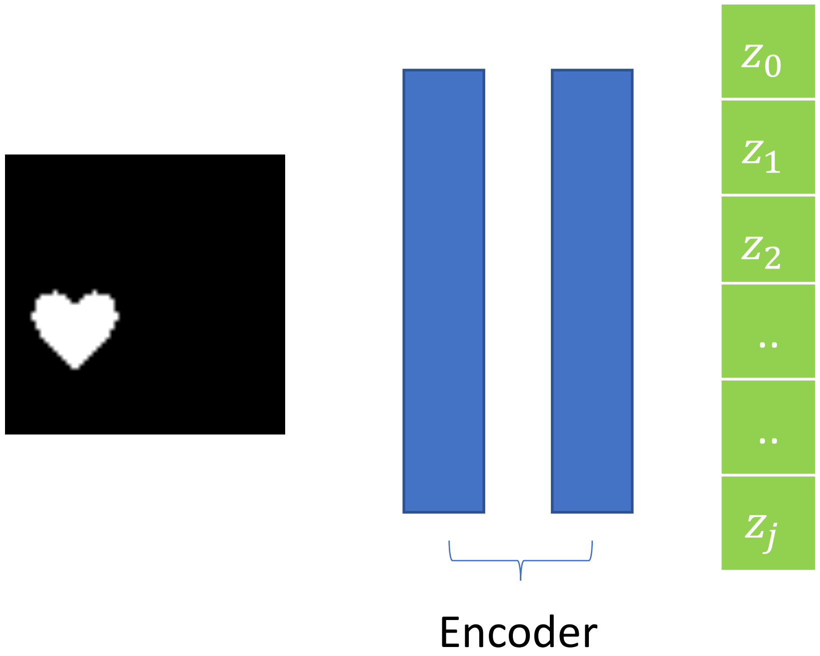
```
loss function {
        recon_loss = batch_size * MSE(input, decoder_output)
        kl_loss = - 0.5 * sum(1 + z_variance – square(z_mean) – exp(z_varaiance))
        total_loss = recon_loss + kl_loss
}

train function {
        input= sample_batch_data()
        z_mean, z_logvar = encoder(input)
        noise = sample_noise()
        z = z_mean + noise * z_variance
        decoder_output = decoder(z)
        loss = loss_fuction(input, decoder_output, z_mean, z_logvar)
        loss.backward()
}
```

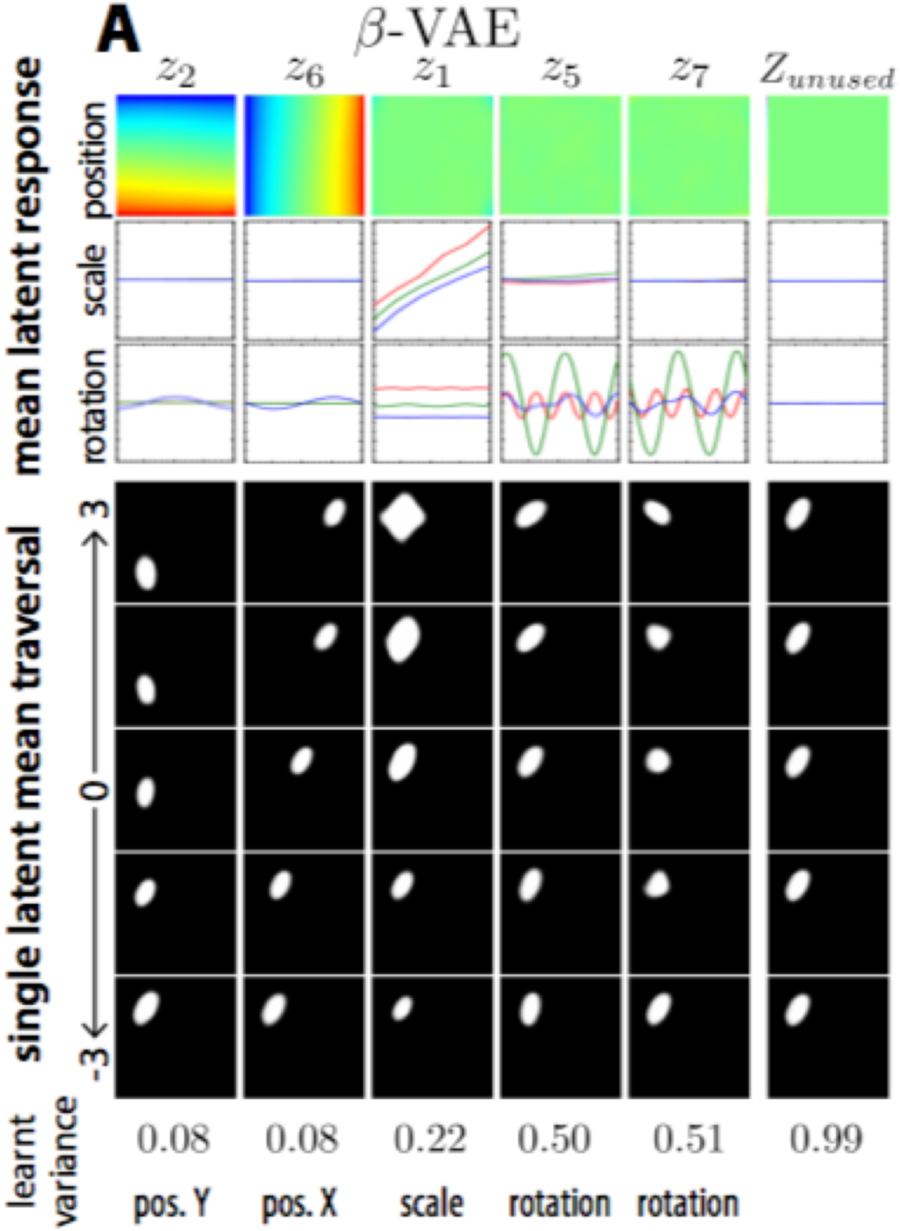
Applications

- Generating new samples (like GAN)
  - e.g. sentenceVAE (Bowman et. al., 2016)
- Semi-supervised learning
  - e.g. M1+M2 model for SSL (Kingma et. al., 2014)
- Disentanglement
  - e.g. betaVAE (Higgins et.al., 2016)
- .....

Applications

- Disentanglement
  - Disentangled representation: where single latent units are sensitive to changes in single generative factors.

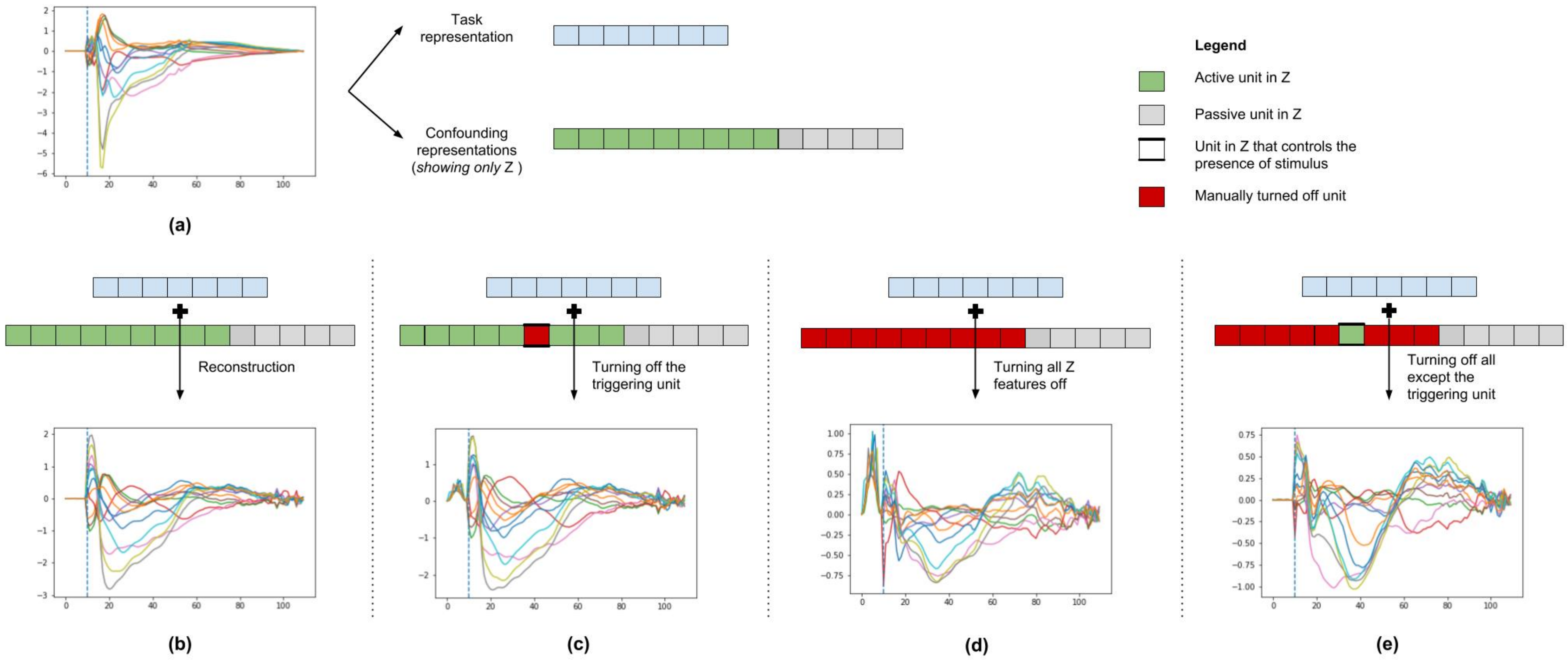




(vi)

Applications

- Disentangling *confounding* factors
  - Integrating Bayesian non-parametric (Indian Buffet Process) into VAE



Picture references

(i) Auto-encoding variational Bayes [Knigma, 2014]

(ii) https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

(iii) http://yusuke-ujitoko.hatenablog.com/entry/2017/05/07/200022

(iv) https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

(v) Tutorial on Variational Autoencoders [Doersch, 2016]

(vi) Beta-VAE [Higgins, 2016]

Thank you.

www.pkgyawali.com pkgyawali@mail.rit.edu @pkgyawali