From Autoencoder to Variational Autoencoder

Hao Dong

Peking University

From Autoencoder to Variational Autoencoder

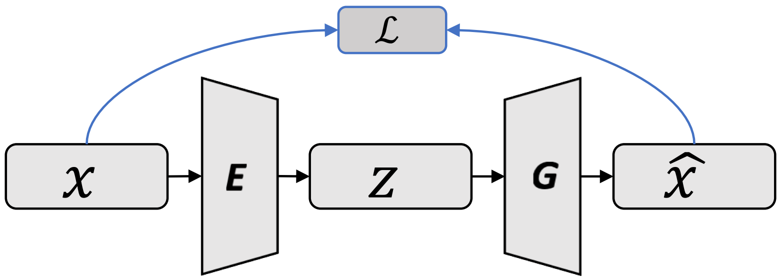
Feature Representation
- Vanilla Autoencoder
- Denoising Autoencoder
- Sparse Autoencoder
- Contractive Autoencoder
- Stacked Autoencoder

Distribution Representation
- Variational Autoencoder (VAE)

- Vanilla Autoencoder
- Denoising Autoencoder
- Sparse Autoencoder
- Contractive Autoencoder
- Stacked Autoencoder
- Variational Autoencoder (VAE)

Vanilla Autoencoder

- **What is it?**



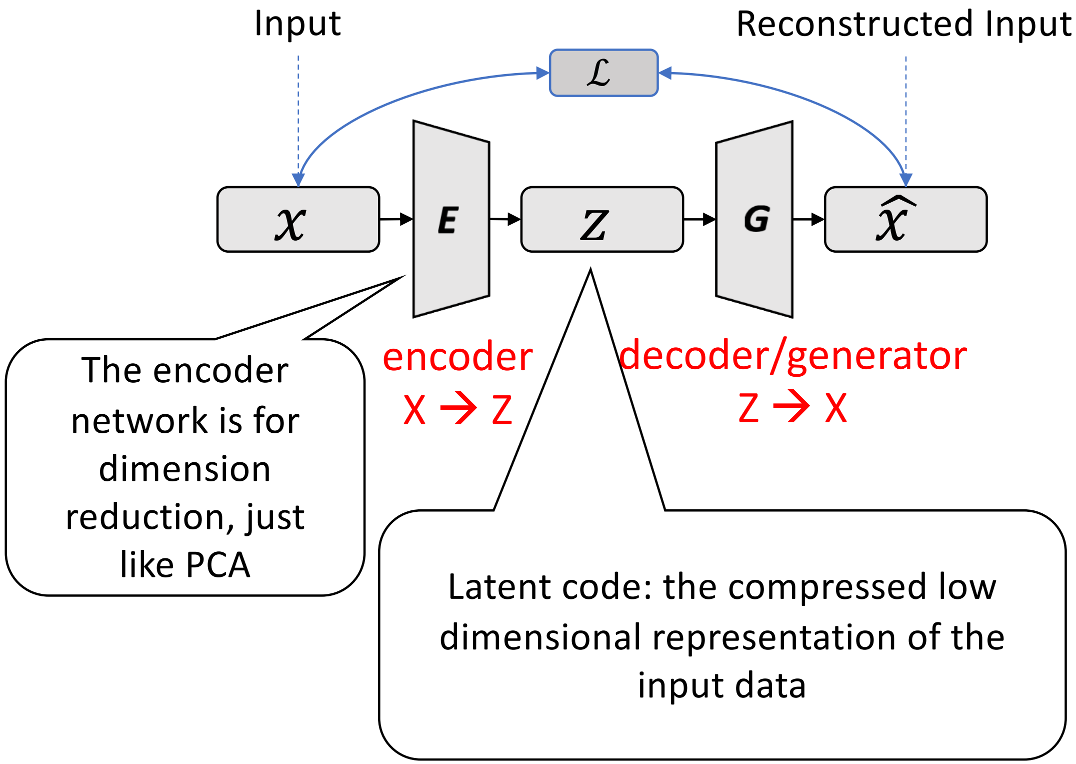
Reconstruct high-dimensional data using a neural network model with a narrow bottleneck layer.

The bottleneck layer captures the compressed latent coding, so the nice by-product is dimension reduction.

The low-dimensional representation can be used as the representation of the data in various applications, e.g., image retrieval, data compression …

Vanilla Autoencoder

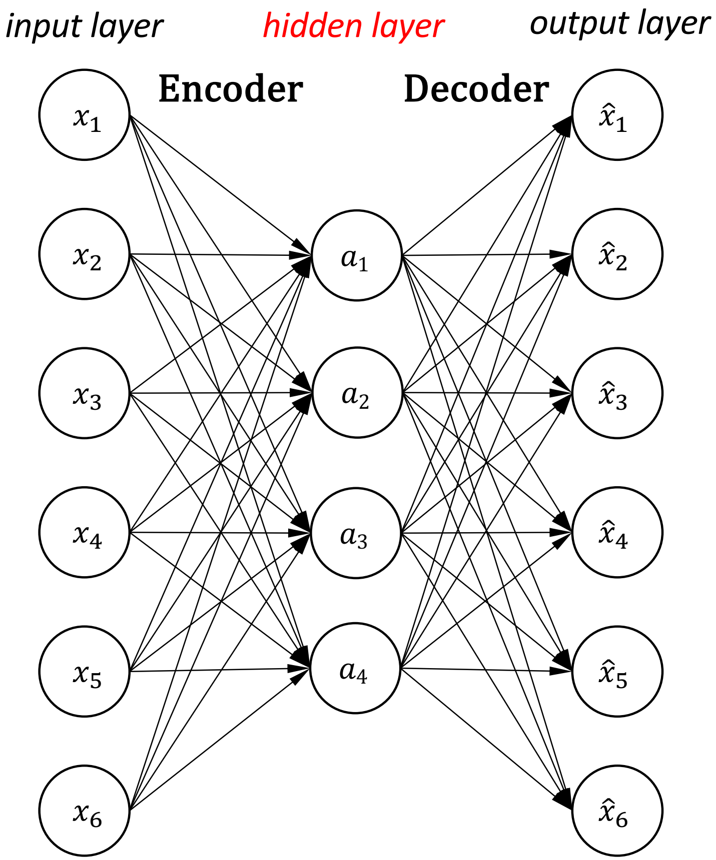
- **How it works?**



Ideally the input and reconstruction are identical

Vanilla Autoencoder

- **Training**



- The hidden units are usually less than the number of inputs
- Dimension reduction --- Representation learning

The distance between two data can be measure by
Mean Squared Error (MSE):

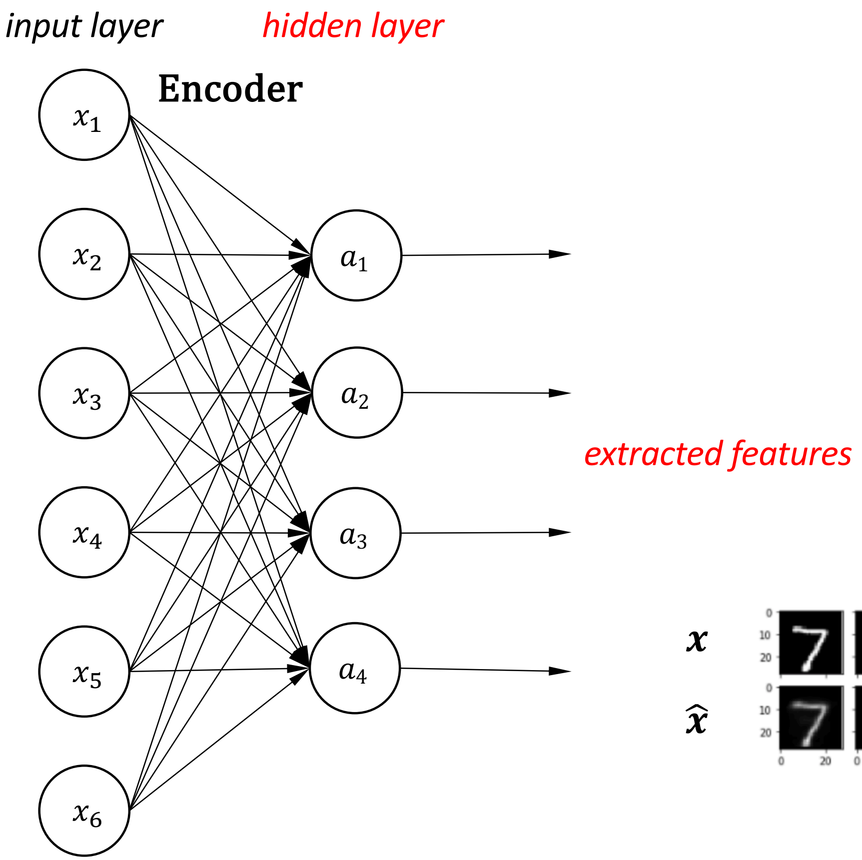
$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(x^i - G(E(x^i))^2$$

where n is the number of variables

- It is trying to learn an approximation to the identity function so that the input is “compress” to the “compressed” features, discovering interesting structure about the data.

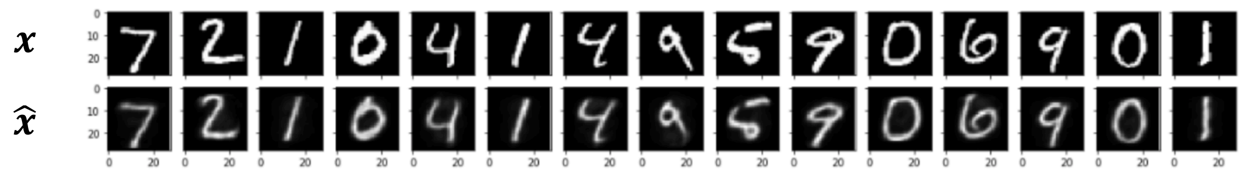
Vanilla Autoencoder

- **Testing/Inferencing**



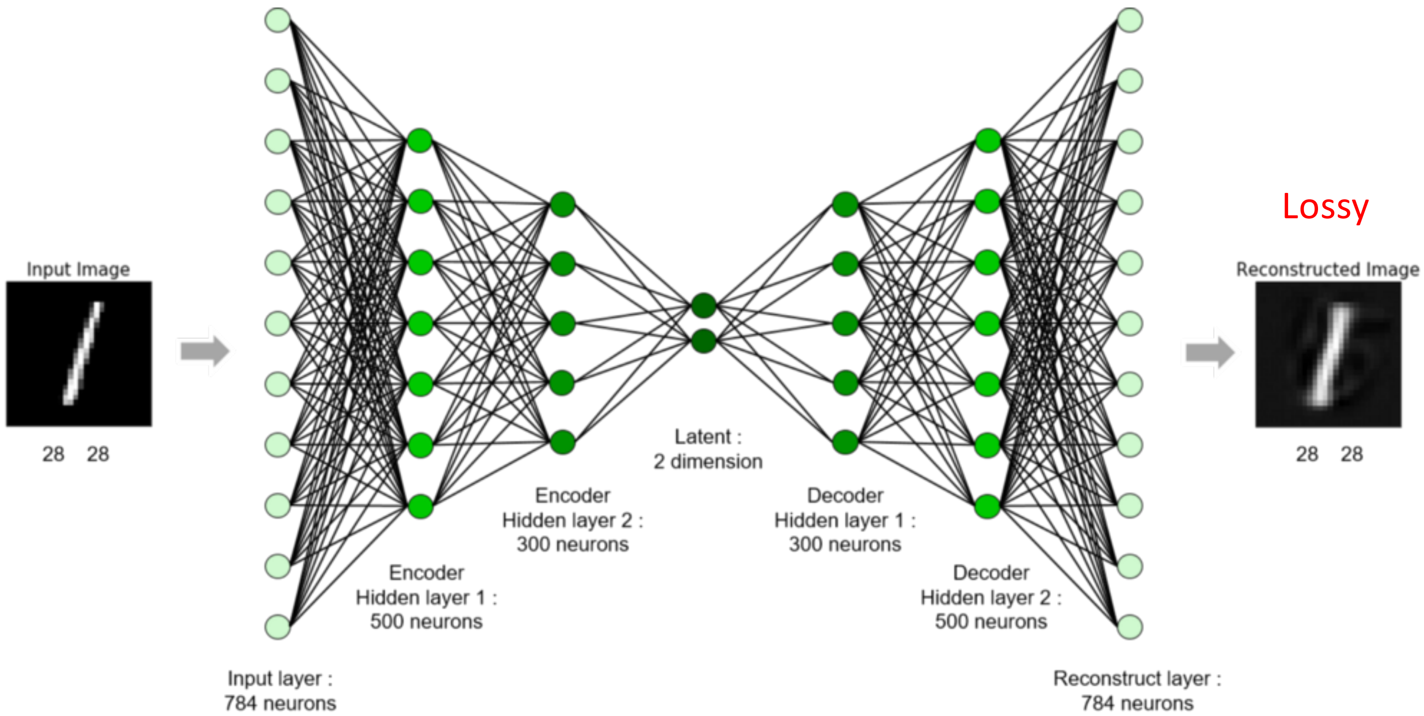
- Autoencoder is an unsupervised learning method if we considered the latent code as the "output".
- Autoencoder is also a self-supervised (self-taught) learning method which is a type of supervised learning where the training labels are determined by the input data.
- Word2Vec (from RNN lecture) is another unsupervised, self-taught learning example.

Autoencoder for MNIST dataset (28×28×1, 784 pixels)



Vanilla Autoencoder

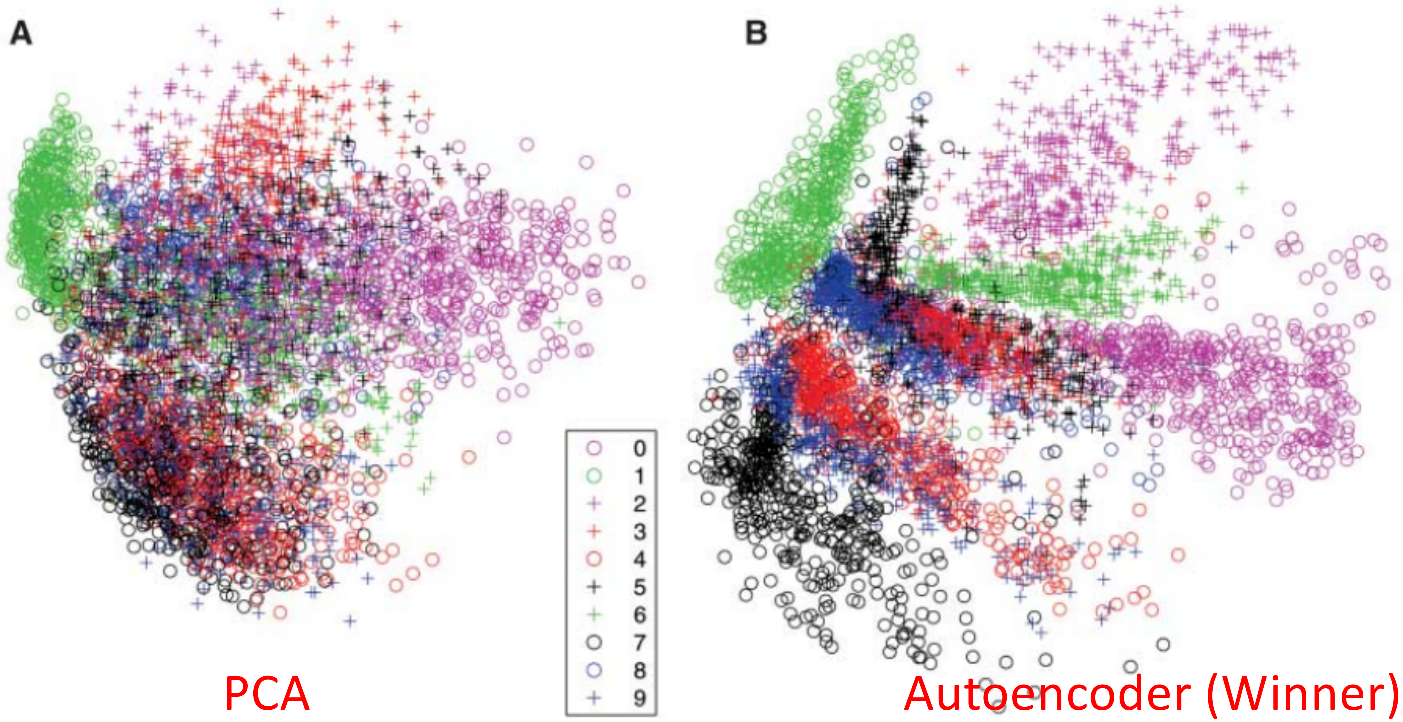
- **Example:**
  - Compress MNIST (28x28x1) to the latent code with only 2 variables



Vanilla Autoencoder

- **Power of Latent Representation**
    - t-SNE visualization on MNIST: PCA vs. Autoencoder

Fig. 3. (**A**) The two-dimensional codes for 500 digits of each class produced by taking the first two principal components of all 60,000 training images. (**B**) The two-dimensional codes found by a 784-1000-500-250-2 autoencoder. For an alternative visualization, see (*8*).



2006 Science paper by Hinton and Salakhutdinov

Vanilla Autoencoder

- **Discussion**
  - Hidden layer is overcomplete if greater than the input layer

Vanilla Autoencoder

- **Discussion**
  - Hidden layer is overcomplete if greater than the input layer
    - No compression
    - No guarantee that the hidden units extract meaningful feature

- Vanilla Autoencoder
- Denoising Autoencoder
- Sparse Autoencoder
- Contractive Autoencoder
- Stacked Autoencoder
- Variational Autoencoder (VAE)

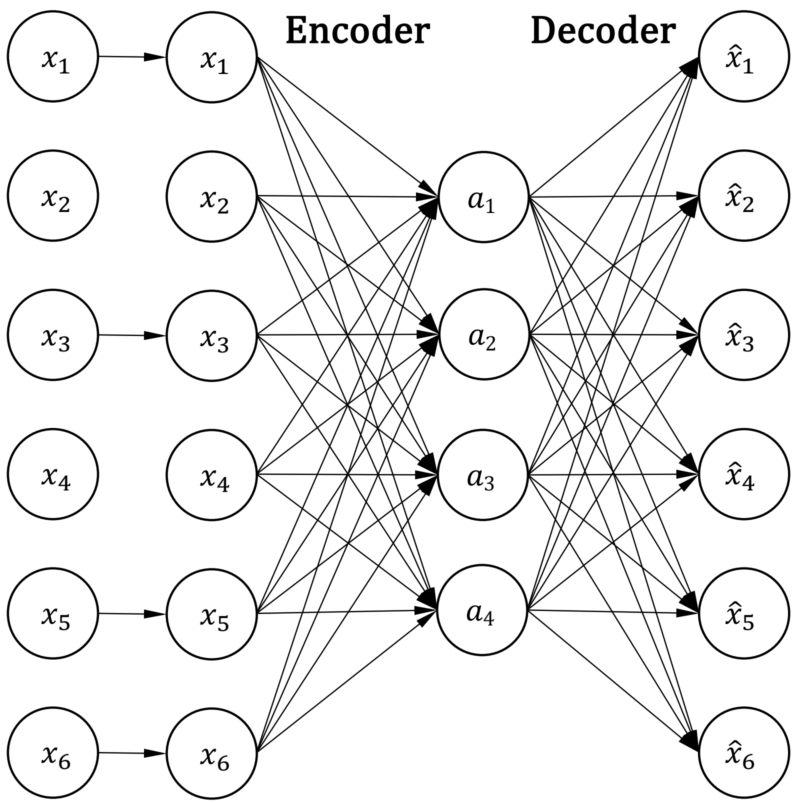
PEKING UNIVERSITY

Denoising Autoencoder (DAE)

- **Why?**
  - **Avoid overfitting**
  - **Learn robust representations**

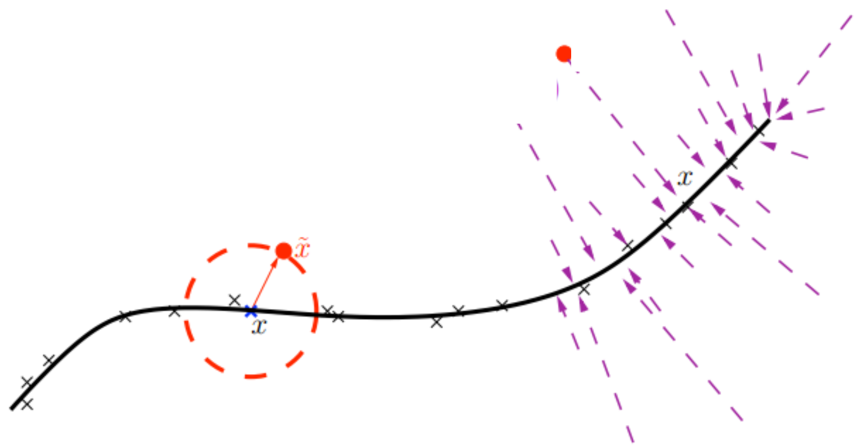
Denoising Autoencoder

- **Architecture**



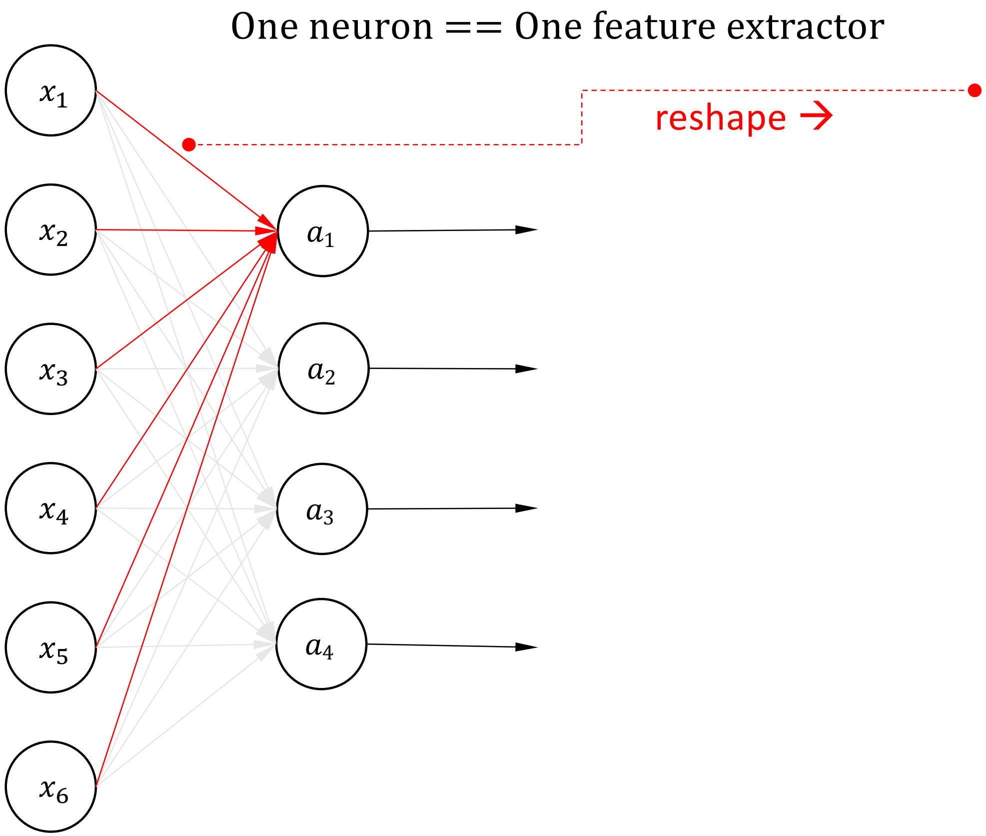
Applying dropout between the input and the first hidden layer

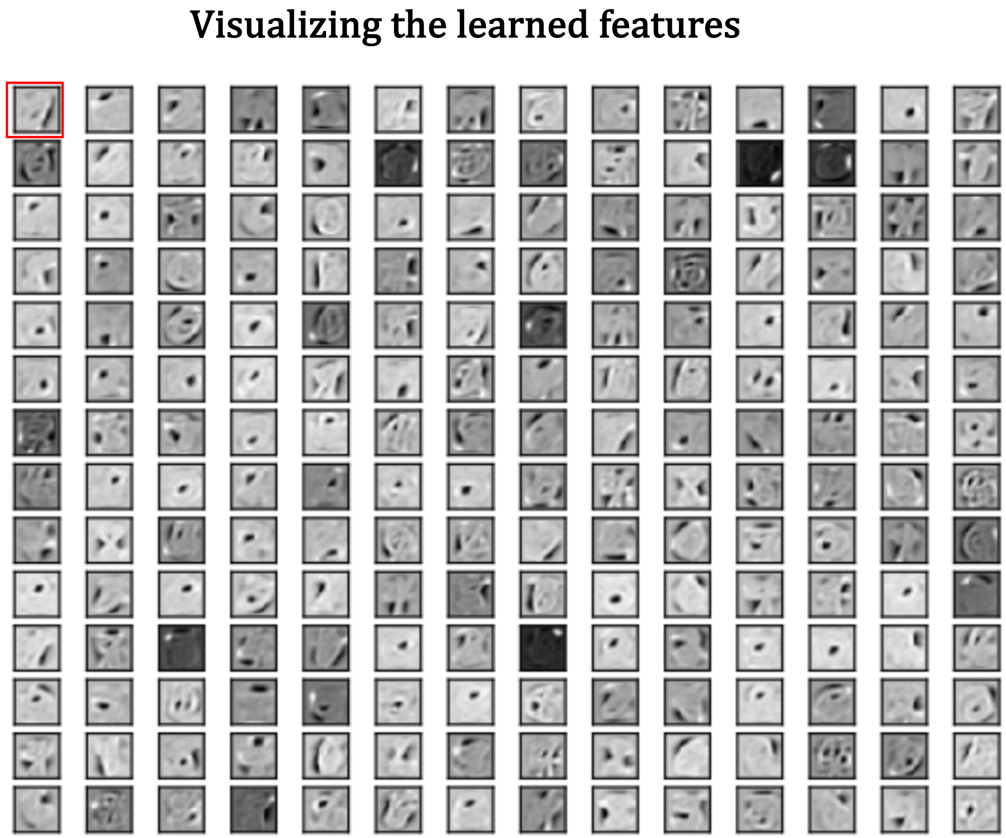
- Improve the robustness



Denoising Autoencoder

- **Feature Visualization**





Denoising Autoencoder

- **Denoising Autoencoder & Dropout**

Denoising autoencoder was proposed in 2008, 4 years before the dropout paper (Hinton, et al. 2012).

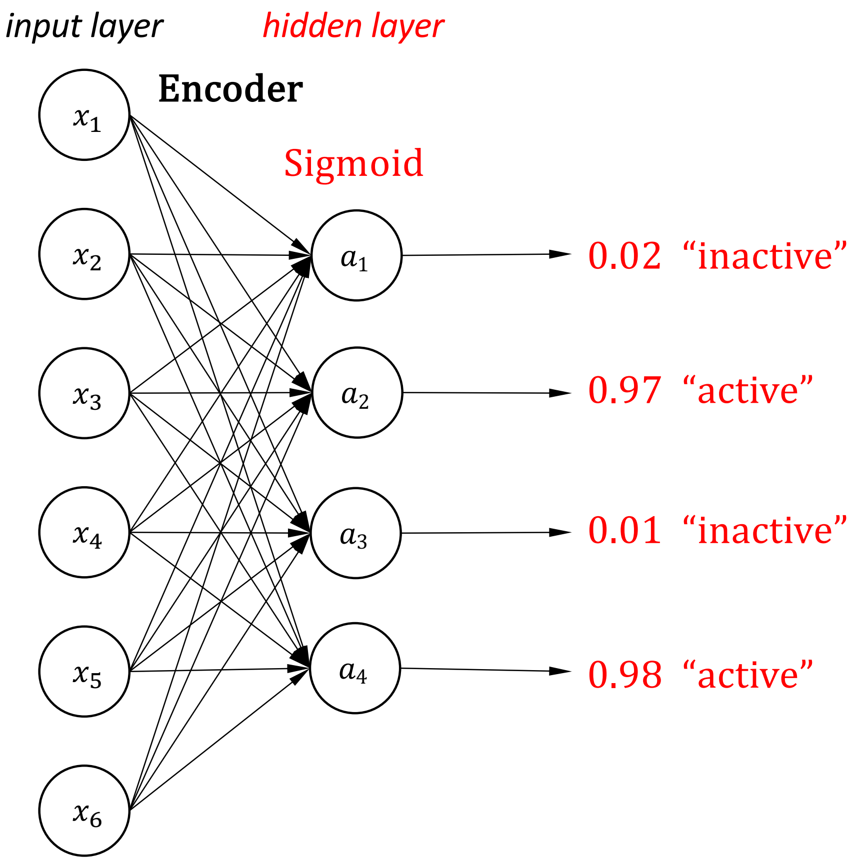
Denoising autoencoder can be seem as applying dropout between the input and the first layer.

Denoising autoencoder can be seem as one type of data augmentation on the input.

- Vanilla Autoencoder
- Denoising Autoencoder
- **Sparse Autoencoder**
- Contractive Autoencoder
- Stacked Autoencoder
- Variational Autoencoder (VAE)

Sparse Autoencoder

- Why?



- Even when the number of hidden units is large (perhaps even greater than the number of input pixels), we can still discover interesting structure, by imposing other constraints on the network.

- In particular, if we impose a "'sparsity"' constraint on the hidden units, then the autoencoder will still discover interesting structure in the data, even if the number of hidden units is large.

Sparse Autoencoder

PEKING UNIVERSITY

- Recap: KL Divergence

$$KL\Big(p(x)\Big\|q(x)\Big) = \int p(x)\ln\frac{p(x)}{q(x)}dx = \mathbb{E}_{x\sim p(x)}\left[\ln\frac{p(x)}{q(x)}\right]$$

Smaller == Closer

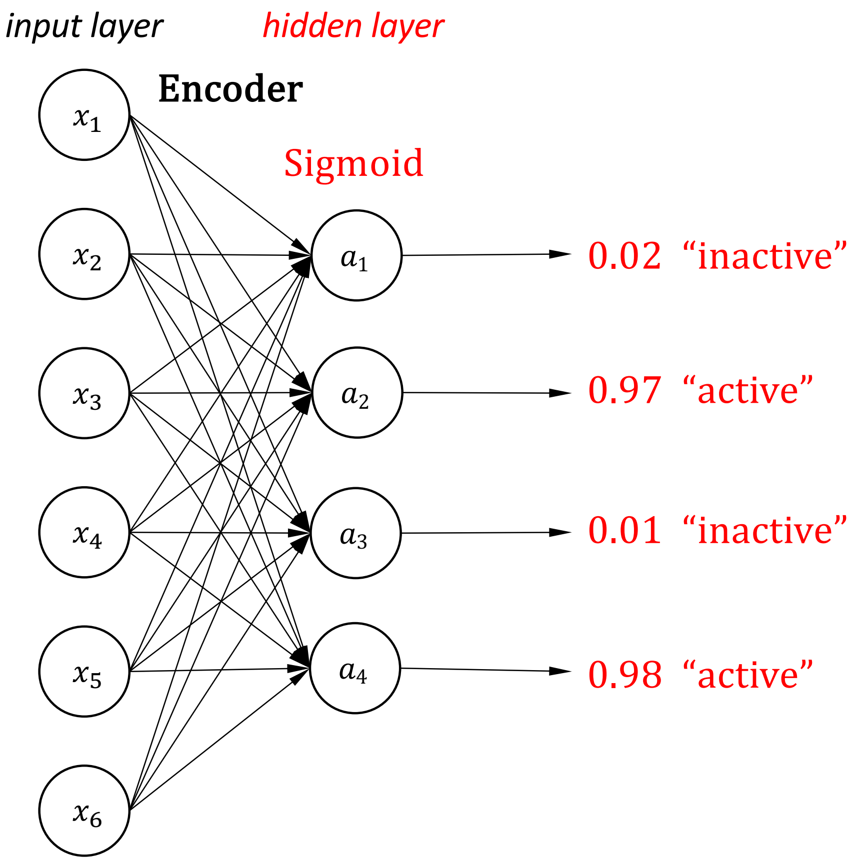
$$KL\Big(p(x)\Big\|q(x)\Big) = 0 \Leftrightarrow p(x) = q(x)$$

$$KL\Big(p(x)\Big\|q(x)\Big) = 0 \Leftrightarrow p(x) = q(x)$$

$$D_B\Big(p(x), q(x)\Big) = -\ln\int\sqrt{p(x)q(x)}dx$$

Sparse Autoencoder

- Sparsity Regularization



The number of hidden units can be greater than the number of input variables.

Given M data samples (batch size) and Sigmoid activation function, the active ratio of a neuron a_j:

$$\hat{\rho}_j = \frac{1}{M}\sum_{m=1}^{M} a_j$$

To make the output "sparse", we would like to enforce the following constraint, where ρ is a "sparsity parameter", such as 0.2 (20% of the neurons)

$$\hat{\rho}_j = \rho$$

The penalty term is as follow, where s is the number of activation outputs.

$$\mathcal{L}_{\rho} = \sum_{j=1}^{s} KL(\rho||\hat{\rho}_j)$$
$$= \sum_{j=1}^{s}(\rho\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j})$$

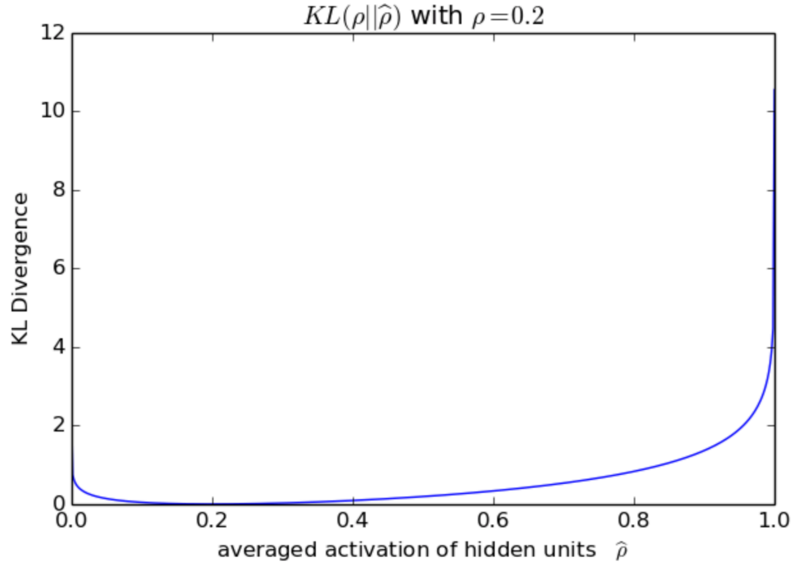
The total loss:

$$\mathcal{L}_{total} = \mathcal{L}_{MSE} + \lambda\mathcal{L}_{\rho}$$

Sparse Autoencoder

- Sparsity Regularization

Smaller ρ == More sparse

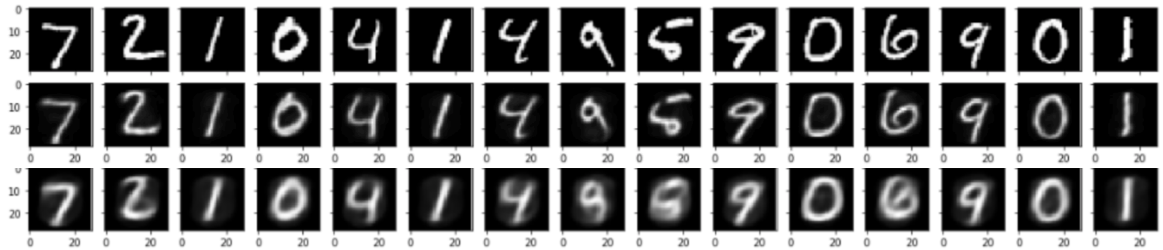


Autoencoders for MNIST dataset

Input x

Autoencoder $\hat{x}$

Sparse Autoencoder $\hat{x}$



Sparse Autoencoder

- **Different regularization loss**

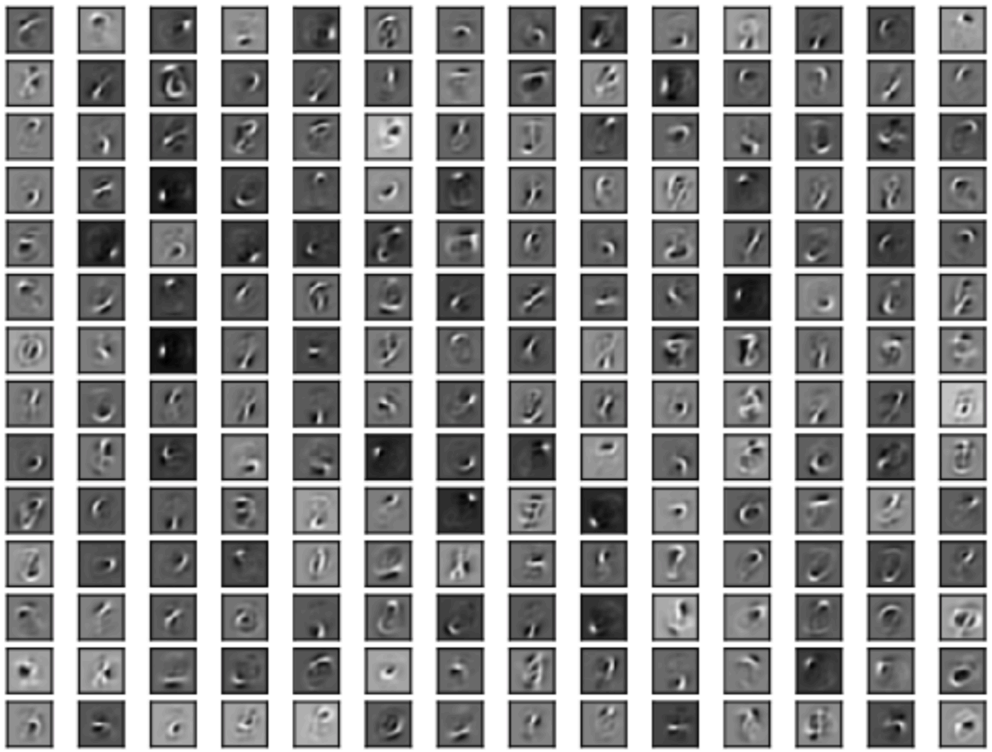
Method	Hidden Activation	Reconstruction Activation	Loss Function
Method 1	Sigmoid	Sigmoid	$\mathcal{L}_{total} = \mathcal{L}_{MSE} + \mathcal{L}_{\rho}$
Method 2	ReLU	Softplus	$\mathcal{L}_{total} = \mathcal{L}_{MSE} + \|\boldsymbol{a}\|$

$\mathcal{L}_1$ on the hidden activation output

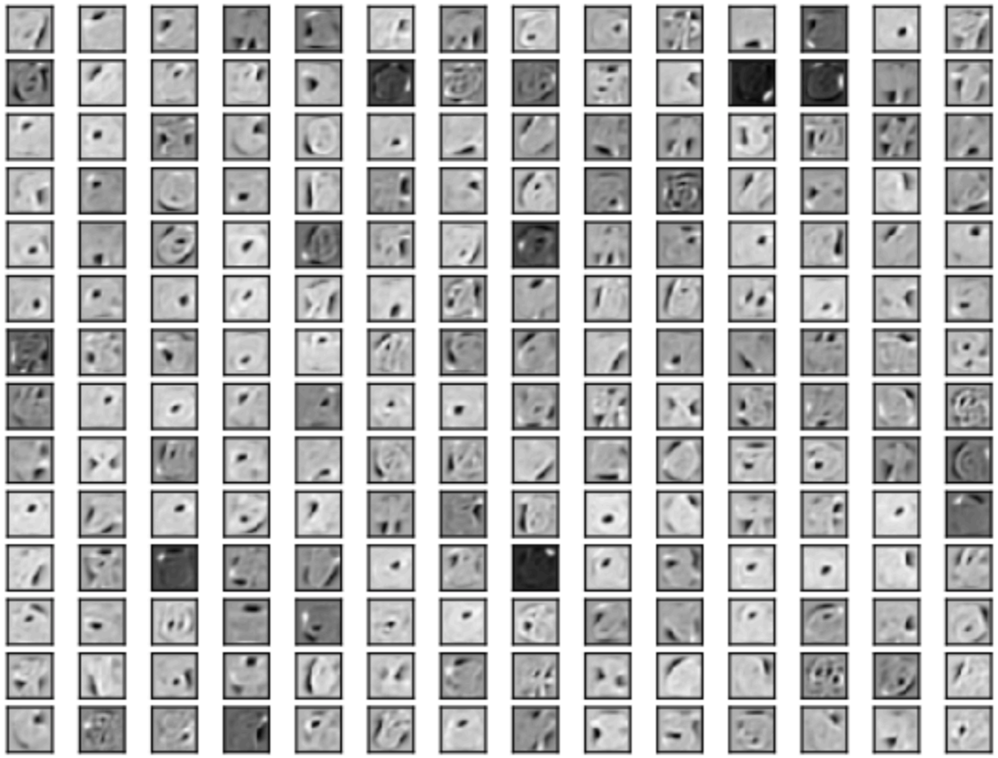
Sparse Autoencoder

- **Sparse Autoencoder vs. Denoising Autoencoder**

Feature Extractors of Sparse Autoencoder



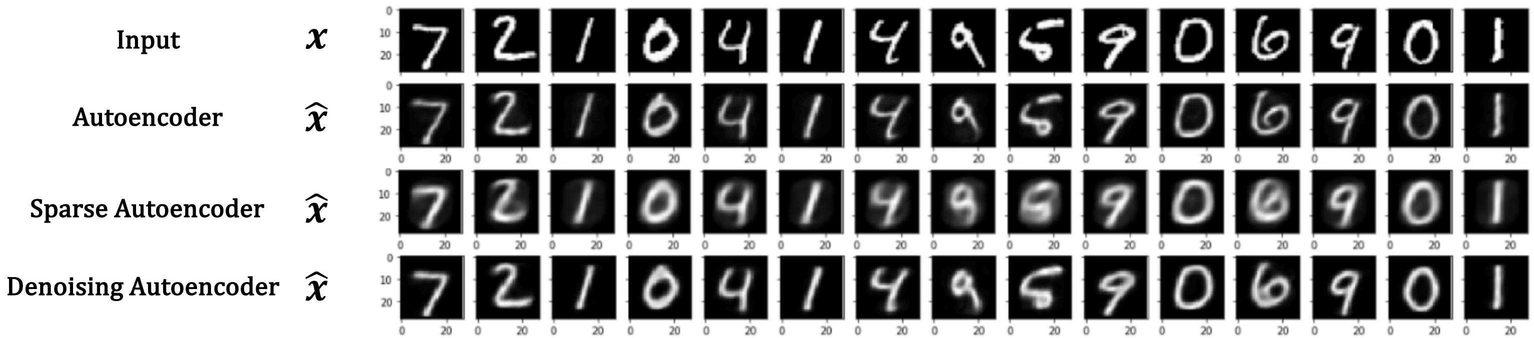
Feature Extractors of Denoising Autoencoder



Sparse Autoencoder

- **Autoencoder vs. Denoising Autoencoder vs. Sparse Autoencoder**

Autoencoders for MNIST dataset



- Vanilla Autoencoder
- Denoising Autoencoder
- Sparse Autoencoder
- Contractive Autoencoder
- Stacked Autoencoder
- Variational Autoencoder (VAE)

Contractive Autoencoder

- **Why?**

  - Denoising Autoencoder and Sparse Autoencoder overcome the overcomplete problem via the input and hidden layers.

  - Could we add an explicit term in the loss to avoid uninteresting features?

    We wish the features that ONLY reflect variations observed in the training set

Contractive Autoencoder

- **How**
  - Penalize the representation being too sensitive to the input
  - Improve the robustness to small perturbations
  - Measure the sensitivity by the Frobenius norm of the Jacobian matrix of the encoder activations

Contractive Autoencoder

- **Recap: Jocobian Matrix**

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} z_1 + z_2 \\ 2z_1 \end{bmatrix} = f\left(\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}\right)$$

$$x = f(z) \qquad z = f^{-1}(x) \qquad \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_2/2 \\ x_1 - x_2/2 \end{bmatrix} = f^{-1}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right)$$

input

$$J_f = \begin{bmatrix} \partial x_1/\partial z_1 & \partial x_1/\partial z_2 \\ \partial x_2/\partial z_1 & \partial x_2/\partial z_2 \end{bmatrix}$$

output

$$J_f = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}$$

$$J_{f^{-1}} = \begin{bmatrix} \partial z_1/\partial x_1 & \partial z_1/\partial x_2 \\ \partial z_2/\partial x_1 & \partial z_2/\partial x_2 \end{bmatrix} \qquad J_{f^{-1}} = \begin{bmatrix} 0 & 1/2 \\ 1 & -1/2 \end{bmatrix}$$

$$J_f J_{f^{-1}} = I$$

Contractive Autoencoder

- **Jocobian Matrix**

$$y=\begin{bmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \\ \vdots \\ f_n(x) \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}=\begin{bmatrix} f_1(x_1,x_2,x_3,\cdots x_n) \\ f_2(x_1,x_2,x_3,\cdots x_n) \\ f_3(x_1,x_2,x_3,\cdots x_n) \\ \vdots \\ f_n(x_1,x_2,x_3,\cdots x_n) \end{bmatrix}$$

$$J(x_1,x_2,x_3,\cdots x_n)=\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \cdots & \frac{\partial f_3}{\partial x_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \frac{\partial f_n}{\partial x_3} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

$$y(x)\approx y(p)+J\bullet(x-p)$$

Contractive Autoencoder

- **New Loss**

$$\|J_f(x)\|_F^2 = \sum_{ij}\left(\frac{\partial h_j(x)}{\partial x_i}\right)^2$$

$$\mathcal{J}_{CAE}(\theta) = \sum_{x\in D_n}\left(L(x, g(f(x))) + \lambda\|J_f(x)\|_F^2\right)$$

reconstruction new regularization

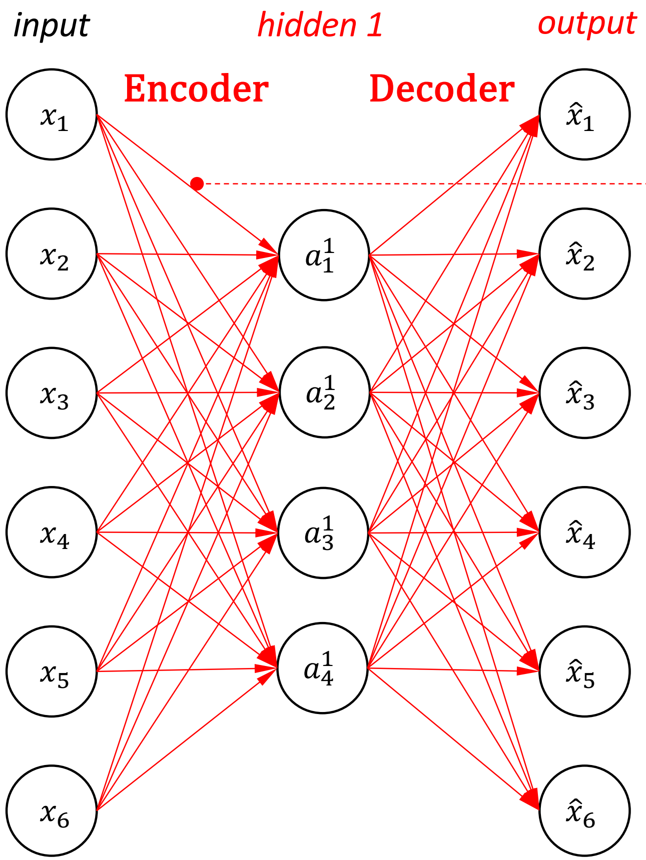
Contractive Autoencoder

- **vs. Denoising Autoencoder**
  - Advantages
    - CAE can better model the distribution of raw data
  - Disadvantages
    - DAE is easier to implement
    - CAE needs second-order optimization (conjugate gradient, LBFGS)

- Vanilla Autoencoder
- Denoising Autoencoder
- Sparse Autoencoder
- Contractive Autoencoder
- Stacked Autoencoder
- Variational Autoencoder (VAE)

Stacked Autoencoder

- **Start from Autoencoder: Learn Feature From Input**



Unsupervised

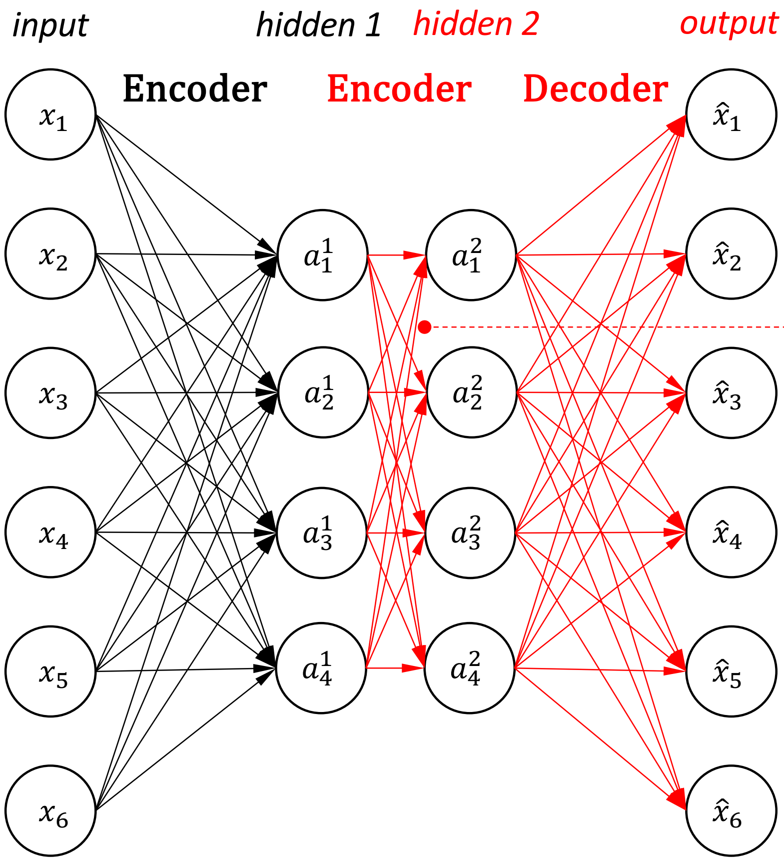
- The feature extractor for the input data

Red color indicates the trainable weights

Red lines indicate the trainable weights
Black lines indicate the fixed/nontrainable weights

Stacked Autoencoder

- **2nd Stage: Learn 2nd Level Feature From 1st Level Feature**



Unsupervised

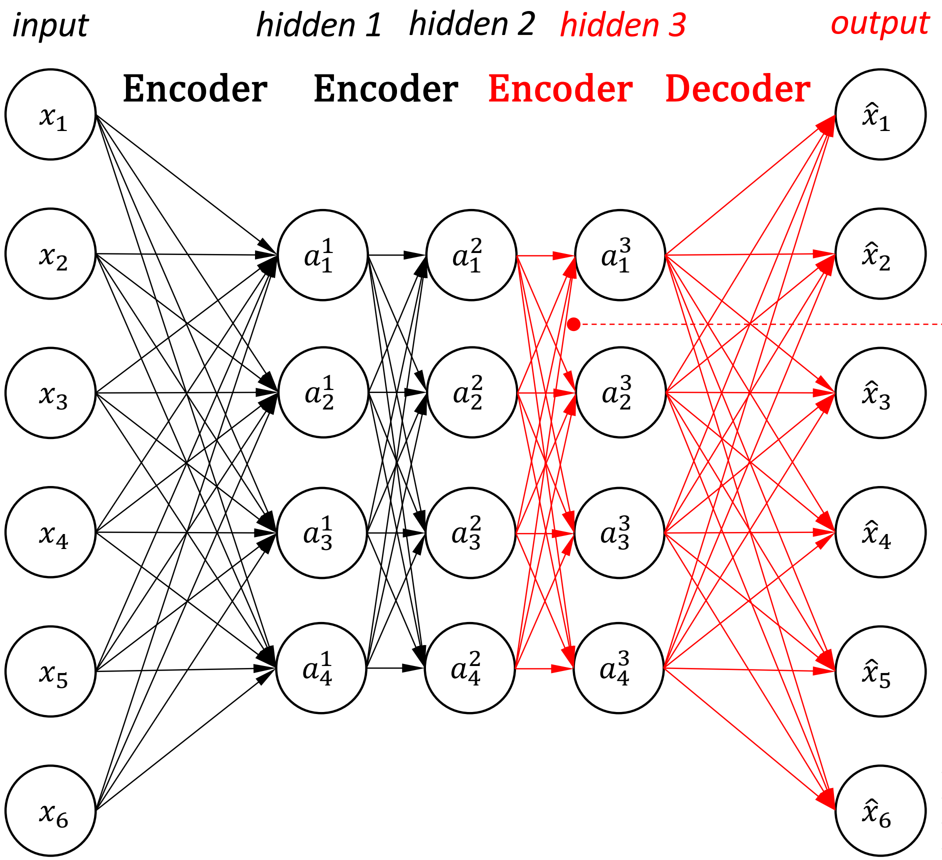
- The feature extractor for the first feature extractor

Red color indicates the trainable weights

Red lines indicate the trainable weights
Black lines indicate the fixed/nontrainable weights

Stacked Autoencoder

- **3rd Stage: Learn 3rd Level Feature From 2nd Level Feature**



Unsupervised

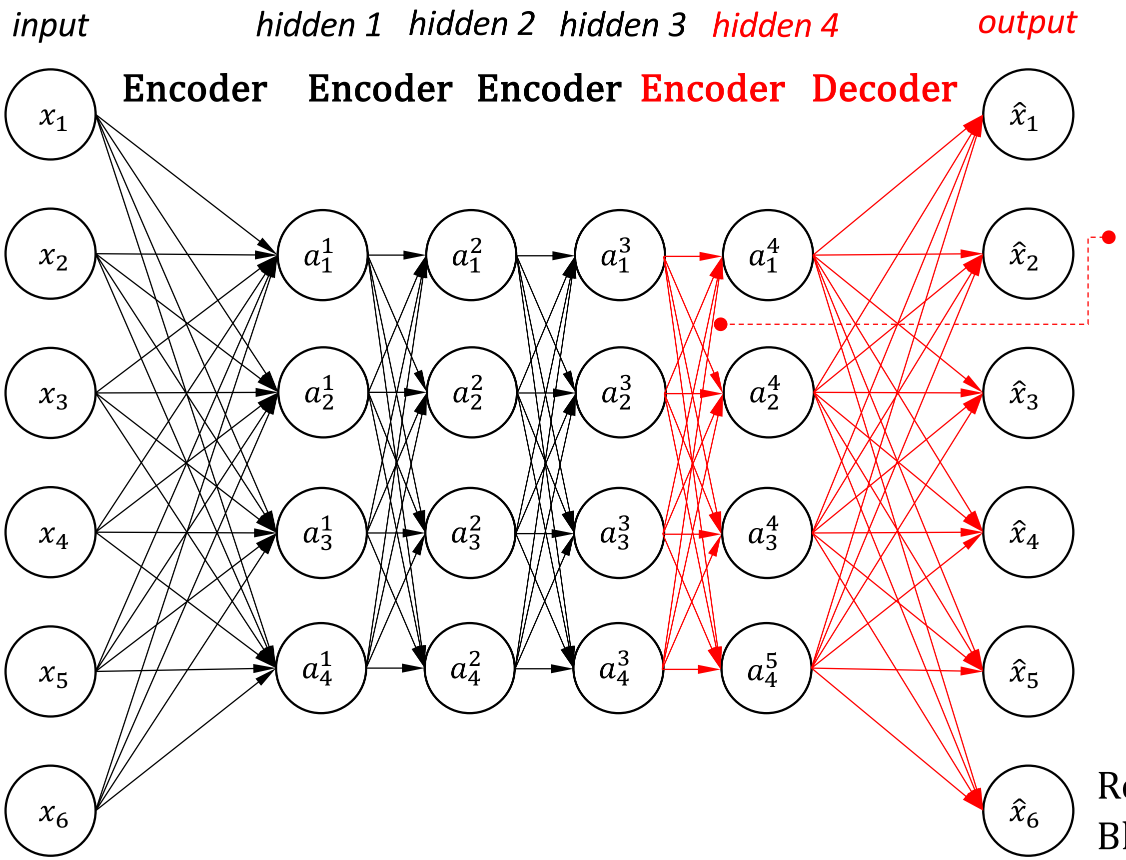
- The feature extractor for the second feature extractor

Red color indicates the trainable weights

Red lines indicate the trainable weights
Black lines indicate the fixed/nontrainable weights

Stacked Autoencoder

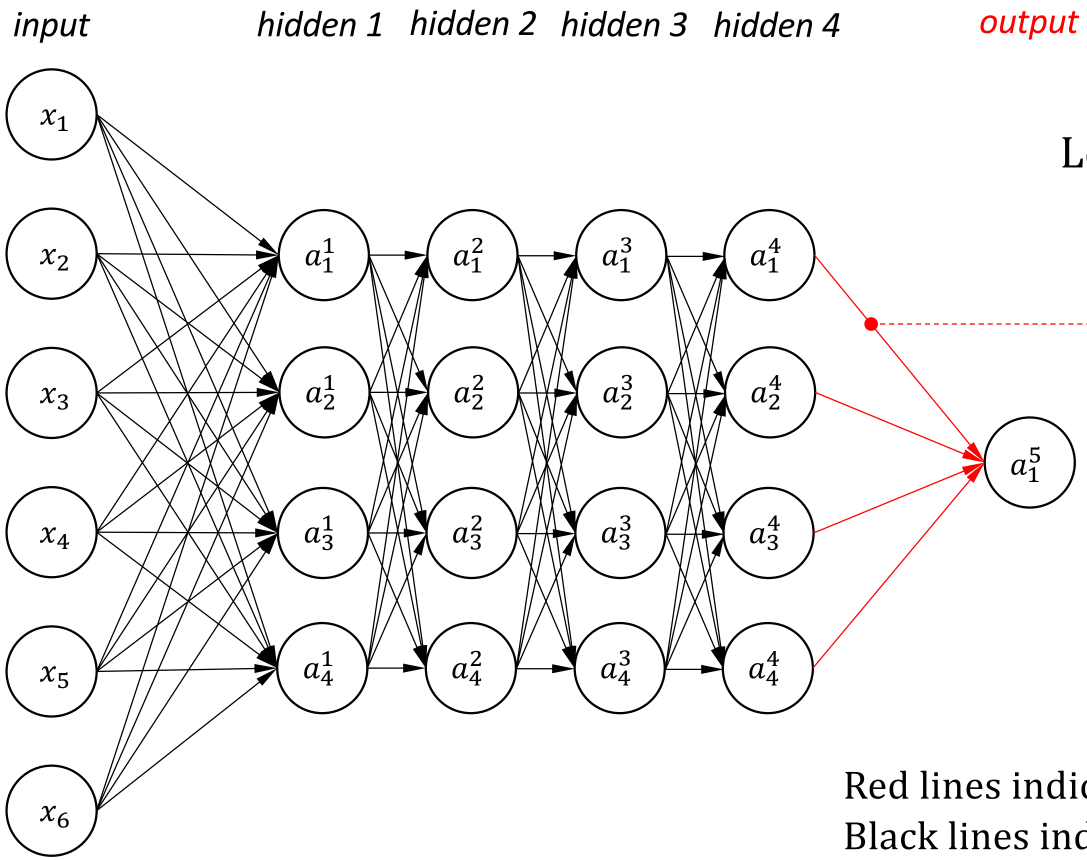
- **4th Stage: Learn 4th Level Feature From 3rd Level Feature**



Red lines indicate the trainable weights
Black lines indicate the fixed/nontrainable weights

Stacked Autoencoder

- **Use the Learned Feature Extractor for Downstream Tasks**



Supervised

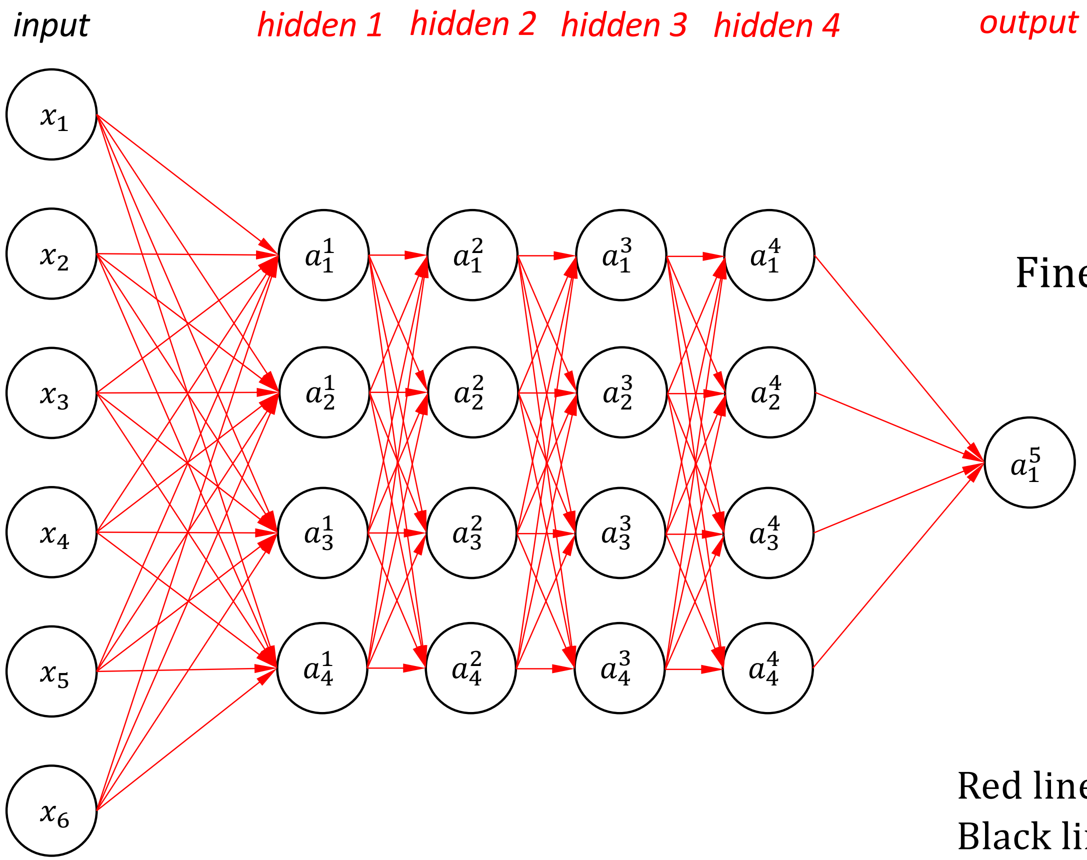
Learn to classify the input data by using the labels and high-level features

Red color indicates the trainable weights

Red lines indicate the trainable weights
Black lines indicate the fixed/nontrainable weights

Stacked Autoencoder

- **Fine-tuning**



Supervised

Fine-tune the entire model for classification

Red color indicates the trainable weights

Red lines indicate the trainable weights
Black lines indicate the fixed/nontrainable weights

Stacked Autoencoder

- **Discussion**
  - Advantages
    - …
  - Disadvantages
    - …

PEKING UNIVERSITY

- Vanilla Autoencoder
- Denoising Autoencoder
- Sparse Autoencoder
- Contractive Autoencoder
- Stacked Autoencoder
- Variational Autoencoder (VAE)
  - From Neural Network Perspective
  - From Probability Model Perspective

Before we start

- **Question?**
  - Are the previous Autoencoders generative model?
  - Recap: We want to learn a probability distribution $p(x)$ over x

○ **Generation (sampling):** $\mathbf{x}_{new} \sim p(\mathrm{x})$
(NO, The compressed latent codes of autoencoders are not prior distributions, autoencoder cannot learn to represent the data distribution)

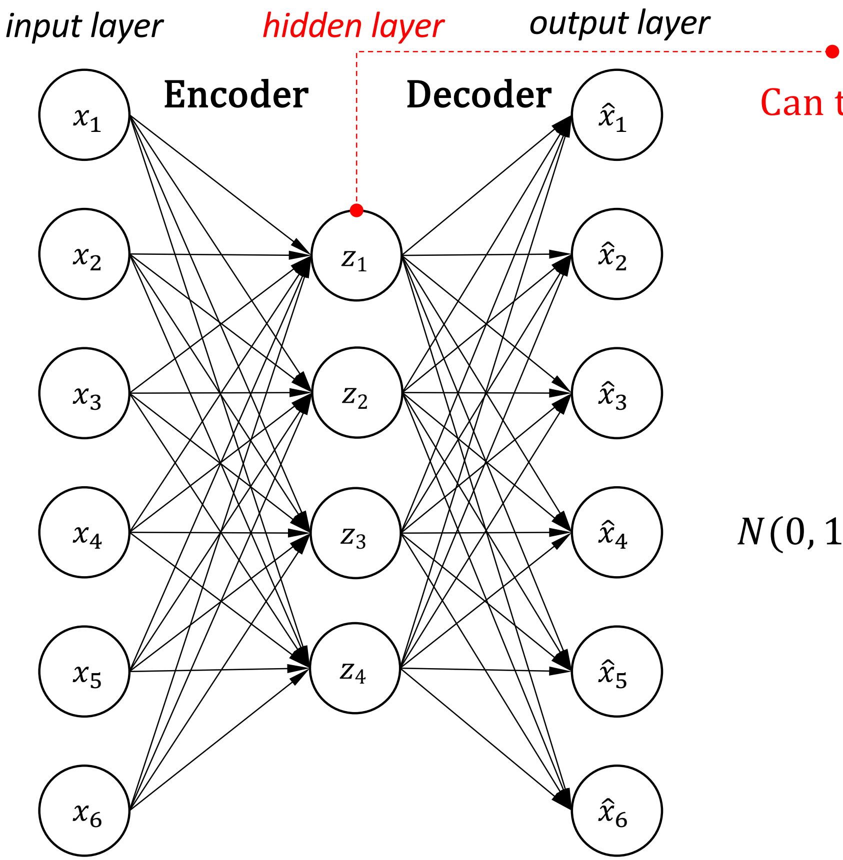
○ **Density Estimation:** $p(\mathrm{x})$ high if $\mathbf{x}$ looks like a real data
NO

○ **Unsupervised Representation Learning:**
Discovering the underlying structure from the data distribution (e.g., ears, nose, eyes …)
(YES, Autoencoders learn the feature representation)

- Vanilla Autoencoder
- Denoising Autoencoder
- Sparse Autoencoder
- Contractive Autoencoder
- Stacked Autoencoder
- Variational Autoencoder (VAE)
  - From Neural Network Perspective
  - From Probability Model Perspective

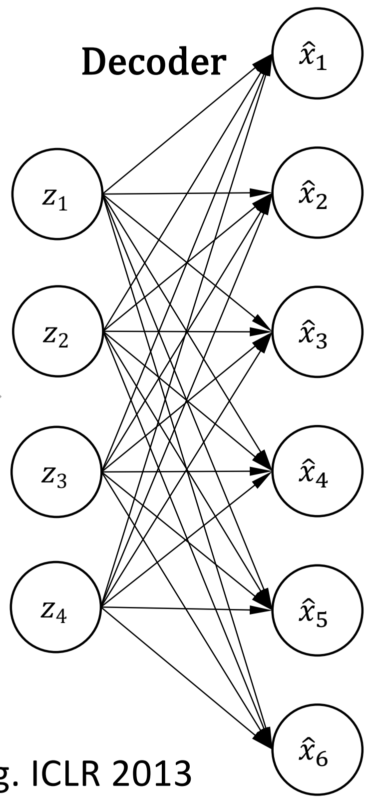
Variational Autoencoder

- **How to perform generation (sampling)?**



Can the hidden output be a prior distribution, e.g., Normal distribution?

$N(0,1)$



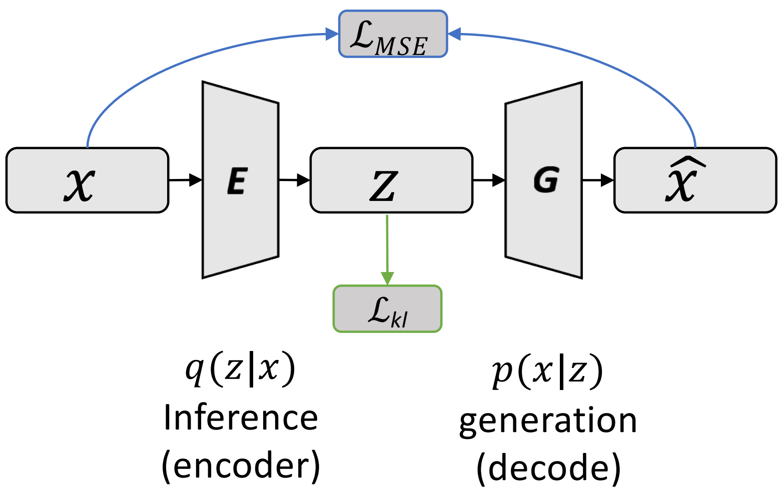
$$p(X) = \sum_Z p(X|Z)p(Z)$$

Decoder(Generator) maps $N(0,1)$ to data space

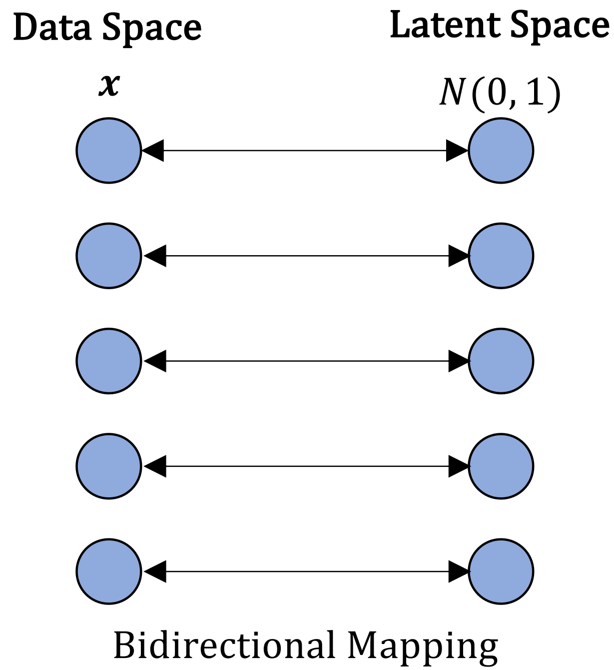
Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Quick Overview**



$$\mathcal{L}_{total} = \mathcal{L}_{MSE} + \mathcal{L}_{kl}$$



Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

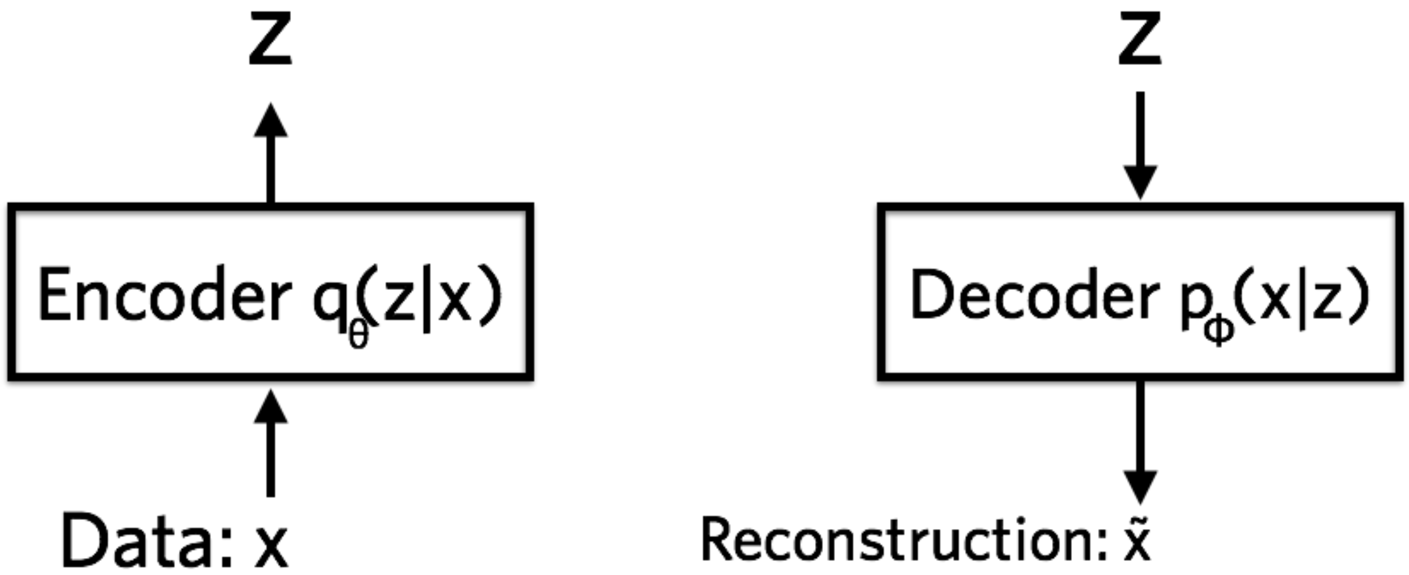
Variational Autoencoder

- **The neural net perspective**
  - A variational autoencoder consists of an encoder, a decoder, and a loss function

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Encoder, Decoder**



Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Loss function**

$$l_i(\theta, \phi) = -\mathbb{E}_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i \mid z)] + \mathbb{KL}(q_\theta(z \mid x_i) \mid\mid p(z))$$

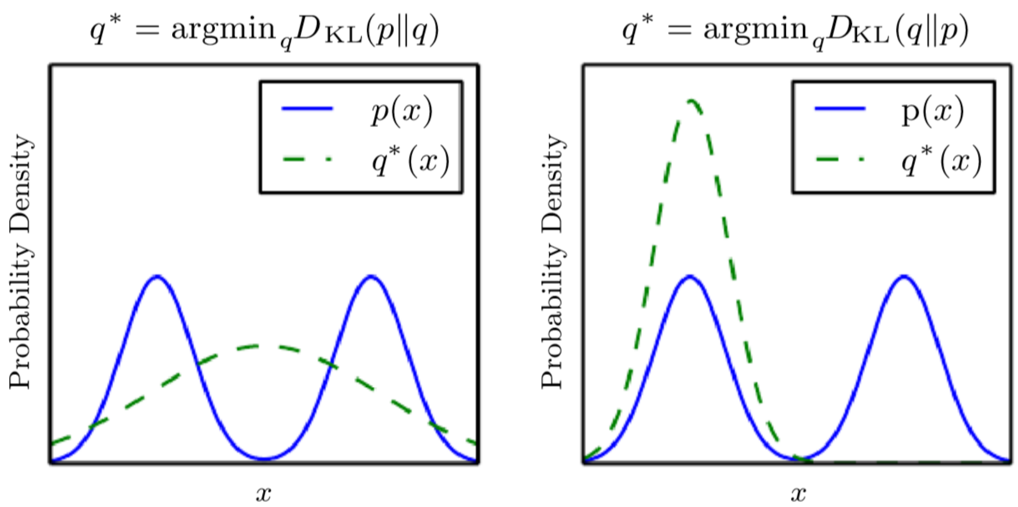
Can be represented by MSE

regularization

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Why KL(Q||P) not KL(P||Q)**



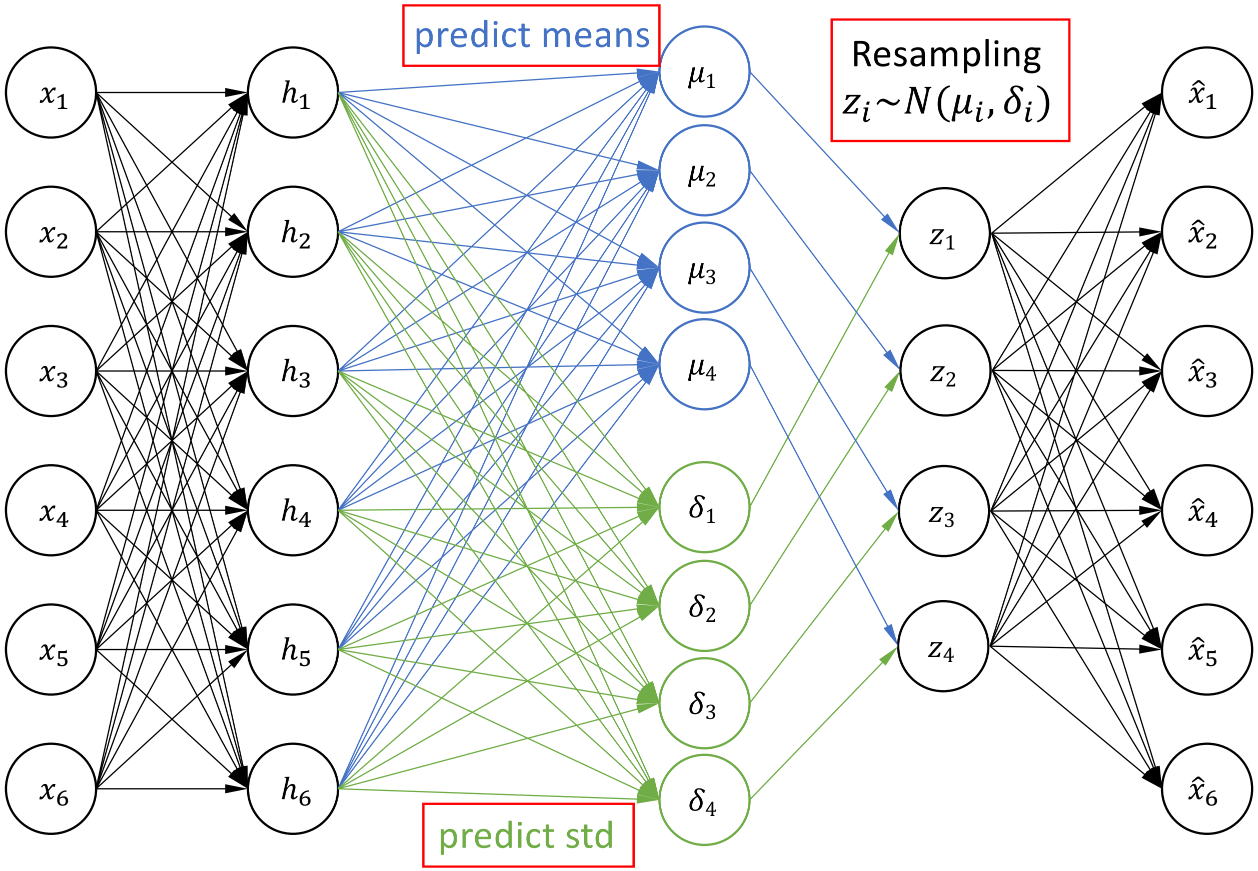
- Which direction of the KL divergence to use?
  - Some applications require an approximation that usually places high probability anywhere that the true distribution places high probability: left one
  - VAE requires an approximation that rarely places high probability anywhere that the true distribution places low probability: right one

If:

$$D_{\text{KL}}(P\|Q) = \mathbb{E}_{\text{x}\sim P}\left[\log \frac{P(x)}{Q(x)}\right] = \mathbb{E}_{\text{x}\sim P}\left[\log P(x) - \log Q(x)\right].$$

Variational Autoencoder

- **Reparameterization Trick**



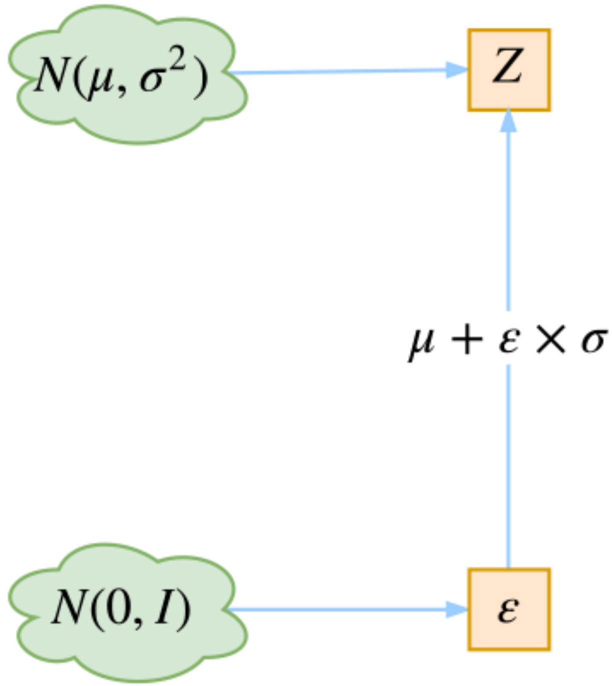
Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

1. Encode the input
2. Predict means
3. Predict standard derivations
4. Use the predicted means and standard derivations to sample new latent variables individually
5. Reconstruct the input

Variational Autoencoder

- **Reparameterization Trick**
  - z ~ N(μ, σ) is not differentiable
  - To make sampling z differentiable
  - z = μ + σ * ϵ   *ϵ ~N(0, 1)*

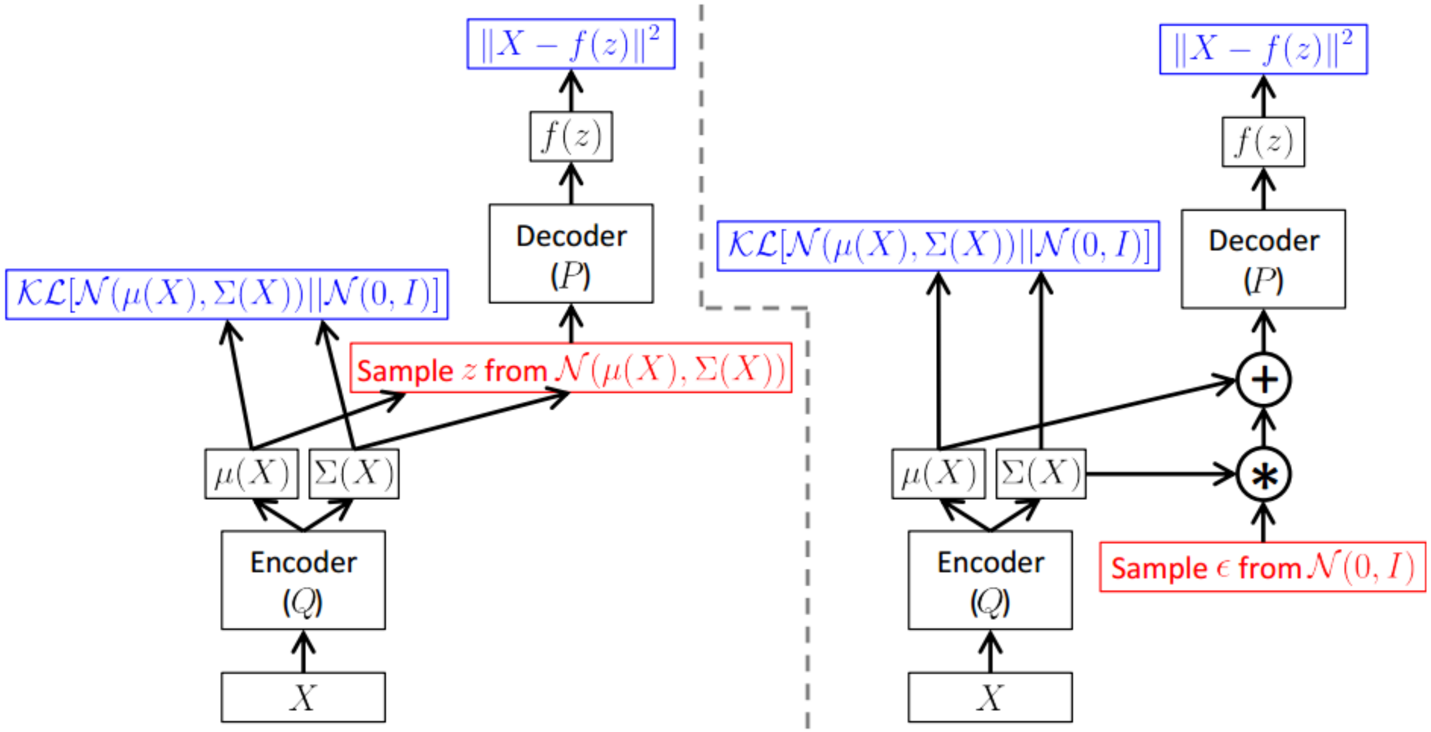
$$\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right)dz$$
$$=\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2\right]d\left(\frac{z-\mu}{\sigma}\right)$$



Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Reparameterization Trick**



$\log P(X|z) - \mathcal{D}\left[Q(z|X)\|P(z)\right]$

$E_{X\sim D}\left[E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z|X)\|P(z)\right]\right]$

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Loss function**

$$\mathbb{KL}(q_\theta(z \mid x_i) \mid\mid p(z))$$

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Where is 'variational'?**

$$\mathbb{KL}(q_\theta(z \mid x_i) \mid\mid p(z))$$

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

- Vanilla Autoencoder
- Denoising Autoencoder
- Sparse Autoencoder
- Contractive Autoencoder
- Stacked Autoencoder
- Variational Autoencoder (VAE)
  - From Neural Network Perspective
  - From Probability Model Perspective

Variational Autoencoder

- **Problem Definition**

Goal: Given $X = \{x_1, x_2, x_3 \ldots, x_n\}$, find $p(X)$ to represent X

How: It is difficult to directly model $p(X)$, so alternatively, we can …

$$p(X) = \sum_{Z} p(X|Z)p(Z)$$

where $p(Z) = N(0,1)$ is a prior/known distribution

i.e., sample X from Z

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **The probability model perspective**
  - P(X) is hard to model

$$p(X) = \sum_{Z} p(X|Z)p(Z)$$

  - Alternatively, we learn the joint distribution of X and Z

$$p(X,Z) = p(Z)p(X|Z)$$

$$p(X) = \sum_{Z} p(X,Z)$$

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Assumption**

$$P(z) = \mathcal{N}(z|0, I)$$

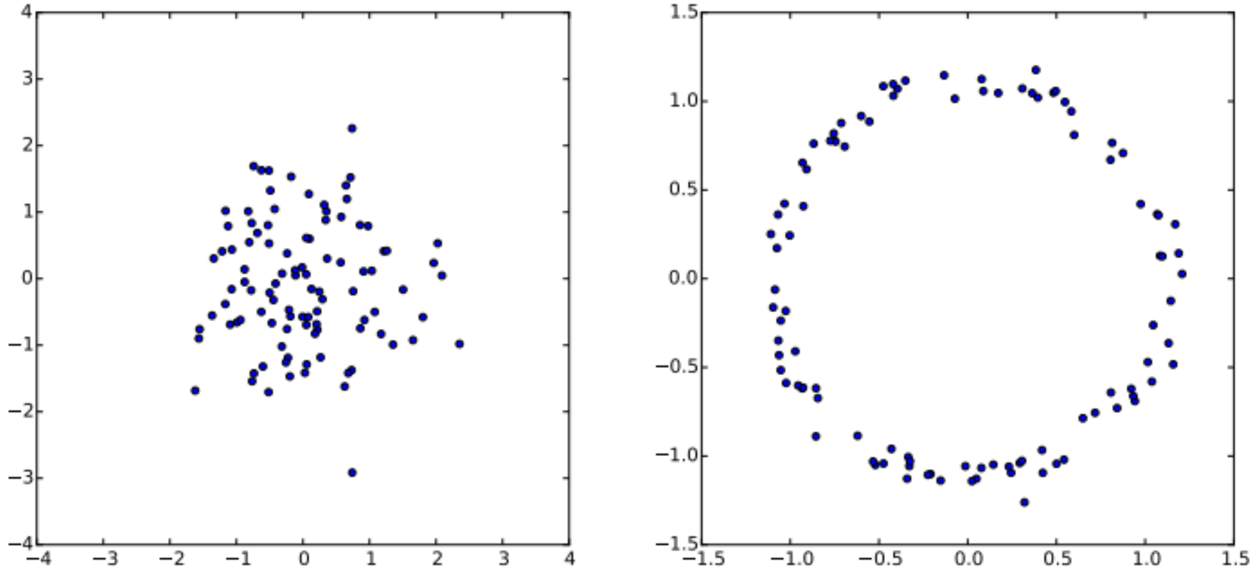
Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Assumption**

$$P(z) = \mathcal{N}(z|0, I)$$

$$g(z) = z/10 + z/||z||$$



Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Monte Carlo?**
  - *n* might need to be extremely large before we have an accurate **estimation of P(X)**

$$P(X) = \int P(X|z;\theta)P(z)dz.$$

$$P(X) \approx \frac{1}{n}\sum_i P(X|z_i)$$

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Monte Carlo?**
  - Pixel difference is different from perceptual difference

(a) (b) (c)

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Monte Carlo?**
  - VAE alters the sampling procedure

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Recap: Variational Inference**
  - VI turns inference into optimization

$$p(x) = \int p(x, z)\mathrm{d}z$$

$$\mathcal{D} = \{q_\theta(z)\}$$

$$\theta^* = \arg\min_\theta \mathsf{KL}(q_\theta(z) \,\|\, p(z|x))$$

approximation ideal

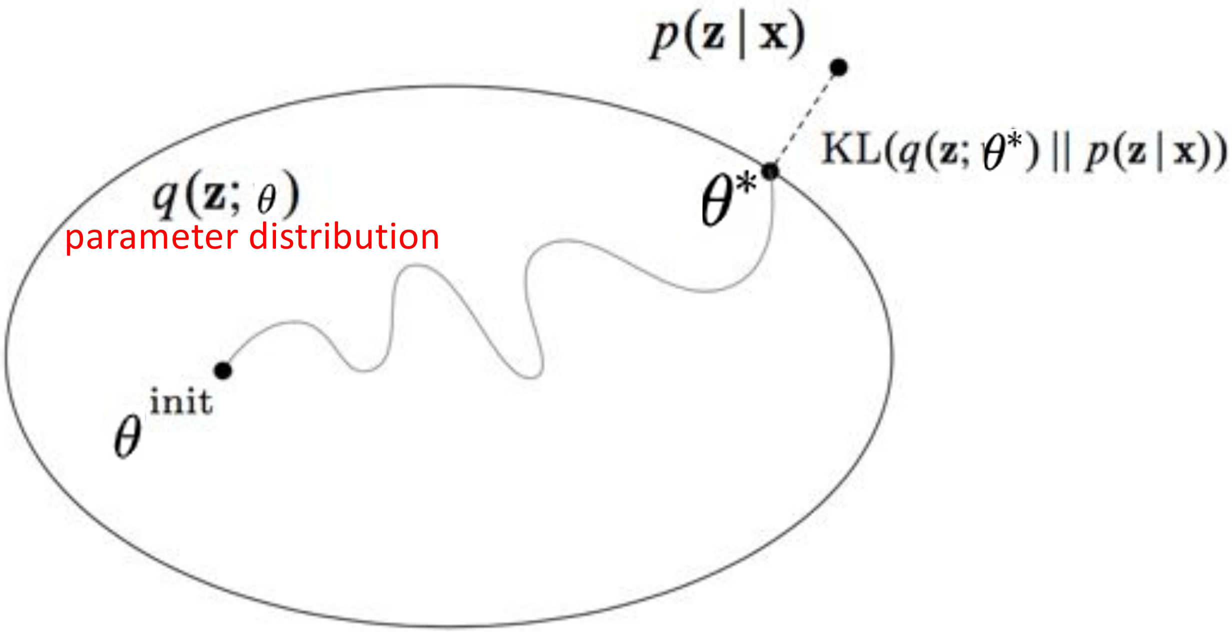
$$p(z|x) = \frac{p(x,z)}{p(x)} \propto p(x,z)$$

$$\theta^* = \arg\max_\theta \mathbb{E}_q[\log p(x,z) - \log q_\theta(z)]$$

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Variational Inference**
  - VI turns inference into optimization



Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Setting up the objective**
  - Maximize P(X)
  - Set Q(z) to be an arbitrary distribution

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

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(z|X)\right]$$

$$\mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log Q(z) - \log P(X|z) - \log P(z)\right] + \log P(X)$$

$$\log P(X) - \mathcal{D}\left[Q(z)\|P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z)\|P(z)\right]$$

$$\boxed{\log P(X) - \mathcal{D}\left[Q(z|X)\|P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z|X)\|P(z)\right]}$$

Goal: maximize this logP(x)

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

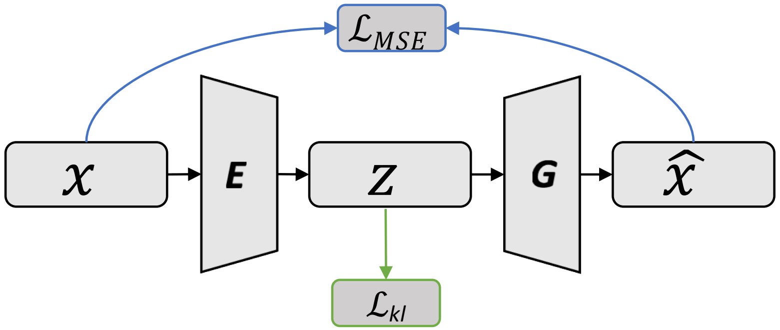
- **Setting up the objective**

$$\log P(X) - \mathcal{D}\left[Q(z|X) \| P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z|X) \| P(z)\right]$$

Goal: maximize this — encoder — ideal — reconstruction/decoder — KLD

difficult to compute

Goal becomes: optimize this



$q(z|x)$
inference

$p(x|z)$
generation

$$\mathcal{L}_{total} = \mathcal{L}_{MSE} + \mathcal{L}_{kl}$$

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Setting up the objective : ELBO**

$$p(z|X) = \frac{p(X,z)}{p(X)}$$

$$\mathbb{KL}(q_\theta(z \mid x) \,||\, p(z \mid x)) =$$

encoder ideal

$$\mathbf{E}_q[\log q_\theta(z \mid x)] - \mathbf{E}_q[\log p(x,z)] + \log p(x)$$

-ELBO

$$q_\theta^*(z \mid x) = \arg\min_\theta \mathbb{KL}(q_\theta(z \mid x) \,||\, p(z \mid x))$$

$$ELBO(\theta) = \mathbf{E}_q[\log p(x,z)] - \mathbf{E}_q[\log q_\theta(z \mid x)]$$

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Setting up the objective : ELBO**

$$ELBO(\theta) = \mathbf{E}_q[\log p(x, z)] - \mathbf{E}_q[\log q_\theta(z \mid x)]$$

$$\log p(x) = ELBO(\theta) + \mathbb{KL}(q_\theta(z \mid x) \mid\mid p(z \mid x))$$

The ELBO for a single datapoint in the variational autoencoder is:

$$ELBO_i(\theta) = \mathbb{E}q_\theta(z \mid x_i)[\log p(x_i \mid z)] - \mathbb{KL}(q_\theta(z \mid x_i) \mid\mid p(z))$$

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Recap: The KL Divergence Loss**

$$KL(\mathcal{N}(\mu,\sigma^2)||\mathcal{N}(0,1))$$

$$= \int \mathcal{N}(\mu,\sigma^2)\, log \frac{\mathcal{N}(\mu,\sigma^2)}{\mathcal{N}(0,1)}\, dx$$

$$= \int \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \left(log \frac{\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}} \right) dx$$

$$= \int \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \log(\frac{1}{\sqrt{\sigma^2}} e^{\frac{x^2-(x-\mu)^2}{2\sigma^2}})dx$$

$$= \frac{1}{2} \int \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \left[-log\sigma^2 + x^2 - \frac{(x-\mu)^2}{\sigma^2} \right] dx$$

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

北京大學 PEKING UNIVERSITY

- **Recap: The KL Divergence Loss**

$$KL(\mathcal{N}(\mu,\sigma^2)||\mathcal{N}(0,1))$$

$$= \frac{1}{2}\int \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \left[-log\sigma^2 + x^2 - \frac{(x-\mu)^2}{\sigma^2}\right] dx$$

$$= \frac{1}{2}(-log\sigma^2 + \mu^2 + \sigma^2 - 1)$$

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Recap: The KL Divergence Loss**

$$\mathcal{D}[\mathcal{N}(\mu_0, \Sigma_0) \| \mathcal{N}(\mu_1, \Sigma_1)] = \frac{1}{2}\left(\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^\top \Sigma_1^{-1}(\mu_1 - \mu_0) - k + \log\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right)\right)$$

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Optimizing the objective**

$$\log P(X) - \mathcal{D}\left[Q(z|X) \| P(z|X)\right] = E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z|X) \| P(z)\right]$$

encoder ideal reconstruction KLD

$$E_{X\sim D}\left[\log P(X) - \mathcal{D}\left[Q(z|X) \| P(z|X)\right]\right] = E_{X\sim D}\left[E_{z\sim Q}\left[\log P(X|z)\right] - \mathcal{D}\left[Q(z|X) \| P(z)\right]\right],$$

dataset dataset

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

北京大學 PEKING UNIVERSITY

- **VAE is a Generative Model**

$p(Z|X)$ is not $N(0,1)$

Can we input $N(0,1)$ to the decoder for sampling?

YES: the goal of KL is to make $p(Z|X)$ to be $N(0,1)$

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **VAE vs. Autoencoder**
  - VAE : distribution representation, $p(z|x)$ is a distribution
  - AE: feature representation, $h = E(x)$ is deterministic

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Variational Autoencoder

- **Challenges**
  - Low quality images
  - …

Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

Summary: Take Home Message

- Autoencoders learn data representation in an unsupervised/ self-supervised way.
- Autoencoders learn data representation but cannot model the data distribution $p(X)$.
- Different with vanilla autoencoder, in sparse autoencoder, the number of hidden units can be greater than the number of input variables.
- VAE
- …
- …
- …
- …
- …
- …

北京大學
PEKING UNIVERSITY

Thanks