CASE SCHOOL OF ENGINEERING
CASE WESTERN RESERVE UNIVERSITY

CSDS 600: Deep Generative Models

From Autoencoder to Variational Autoencoder

Yu Yin (yu.yin@case.edu)

Case Western Reserve University

https://yin-yu.github.io/

Recap

- Autoregressive models:
  - Chain rule based factorization is fully general
  - Compact representation via conditional independence and/or neural parameterizations
  - Examples: FVSBN, NADE, MADE, Pixel RNN, Pixel CNN
- Autoregressive models Pros:
  - Easy to evaluate likelihoods
  - Easy to train
- Autoregressive models Cons:
  - Requires an ordering
  - Generation is sequential (slow)
  - Cannot learn features in an unsupervised way

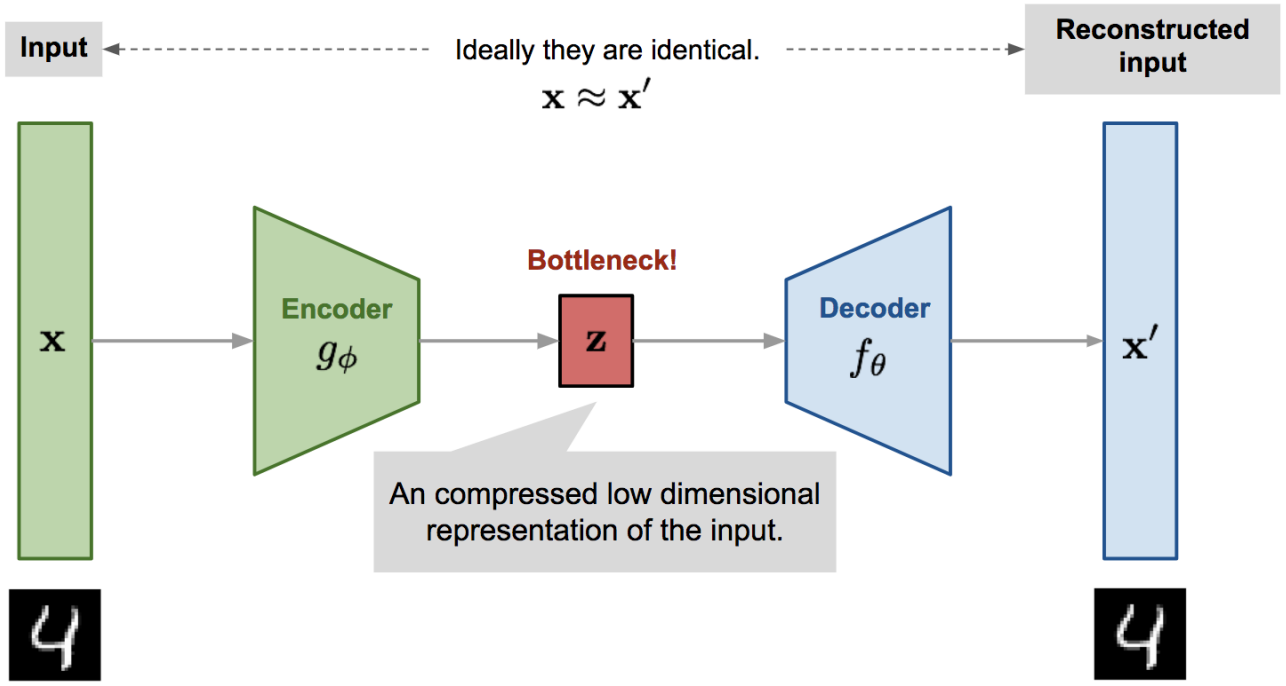
Outline

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

Vanilla Autoencoder

What is it?

- Reconstruct high-dimensional data using a neural network model with a narrow bottleneck layer.
- It consists of two networks:
  - Encoder network: translates the original high-dimension input into the latent low-dimensional code.
  - Decoder network: recovers the data from the code



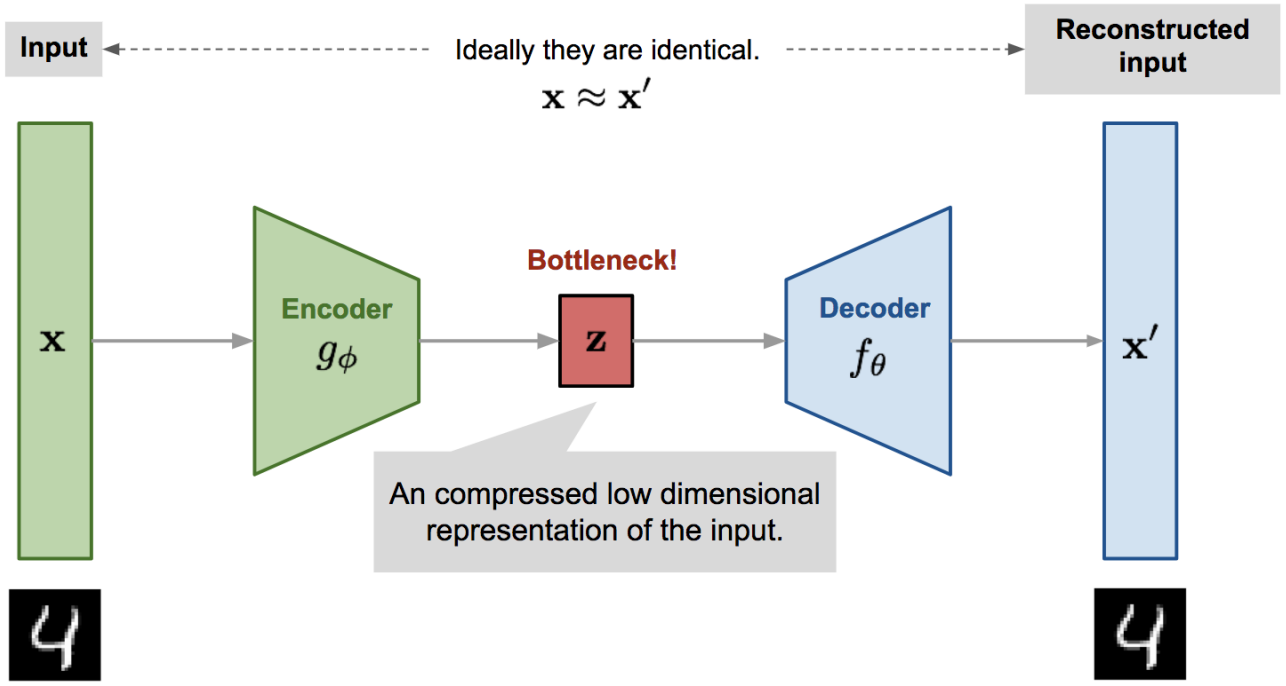
Weng, Lilian, From Autoencoder to Beta-VAE, 2018

Vanilla Autoencoder

How it works?

- The encoder network is for dimension reduction, just like PCA

- Ideally the input and reconstruction are identical

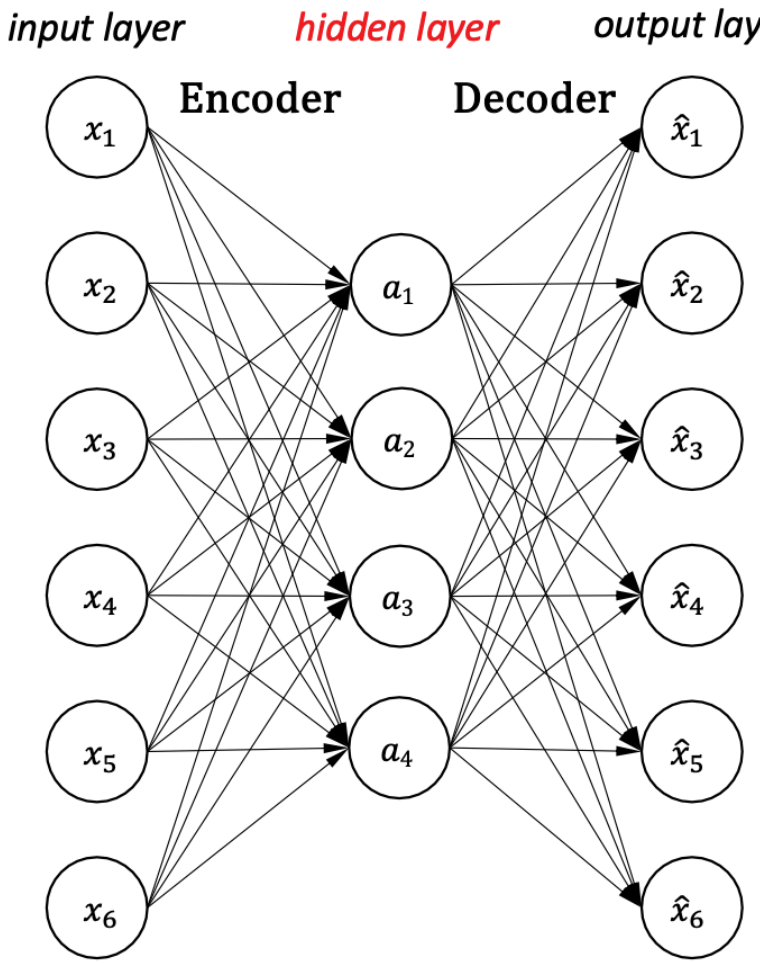


Vanilla Autoencoder

Train

- It is trying to learn an approximation to the identity function so that the input is “compressed” to lower-dimensional features, discovering interesting structure about the data.
- The distance between two data can be measure by Mean Squared Error (MSE):

$$L_{\mathrm{AE}}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}^{(i)} - f_\theta(g_\phi(\mathbf{x}^{(i)})))^2$$



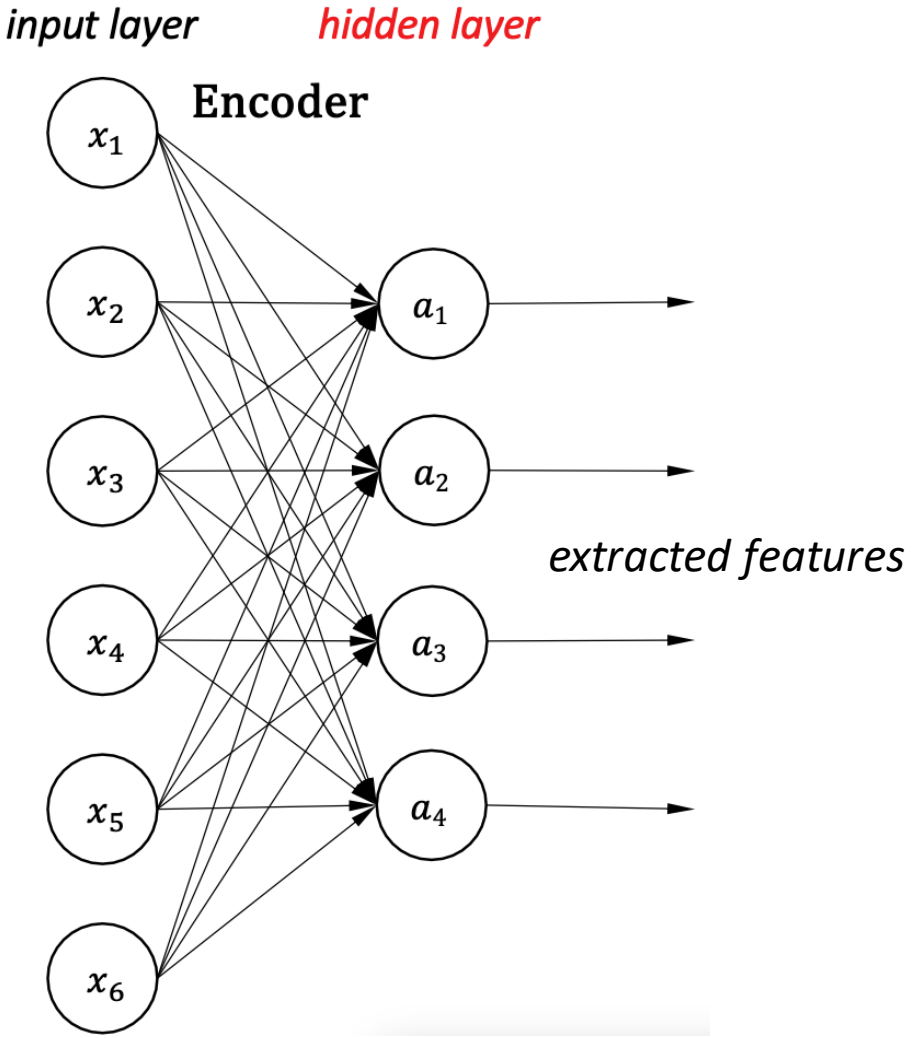
- The hidden units are usually less than the number of inputs
- Dimension reduction -- Representation learning

Hao Dong, ”Deep Generative Models”, 2022

Vanilla Autoencoder

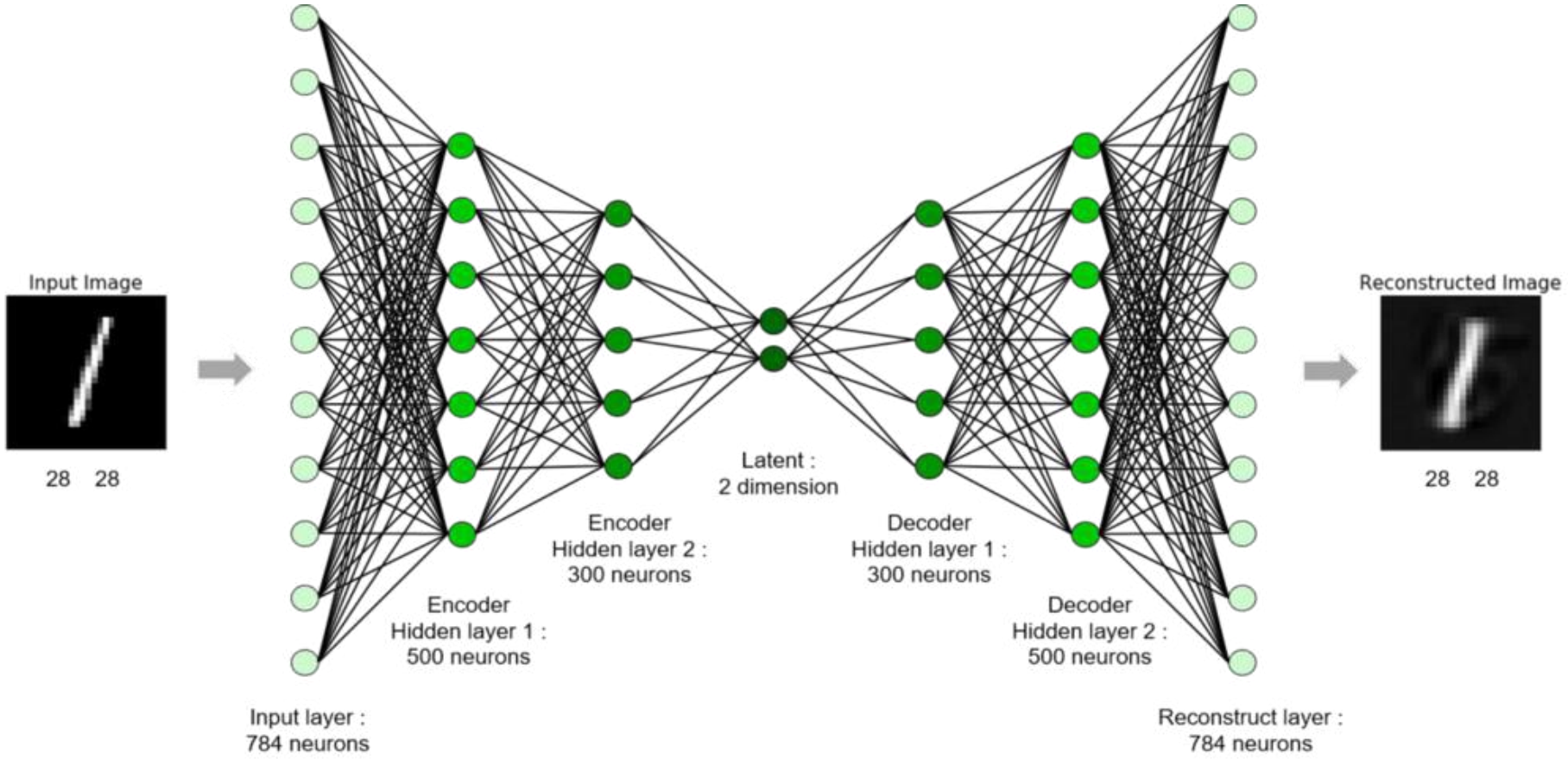
Inference

- 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.



Vanilla Autoencoder: Example

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



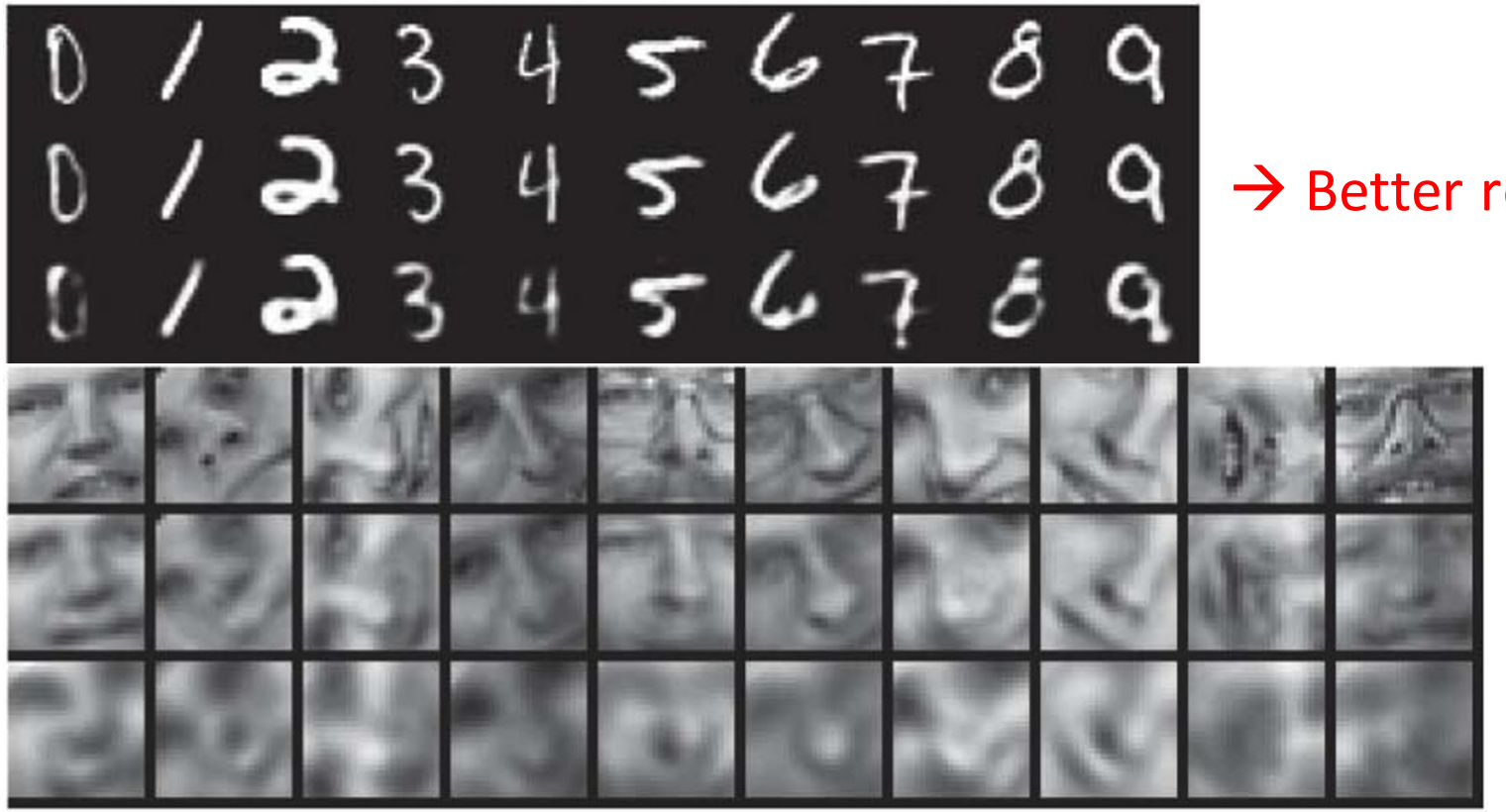
Vanilla Autoencoder

Power of Latent Representation: Reconstruction results

Random samples

Autoencoder

PCA



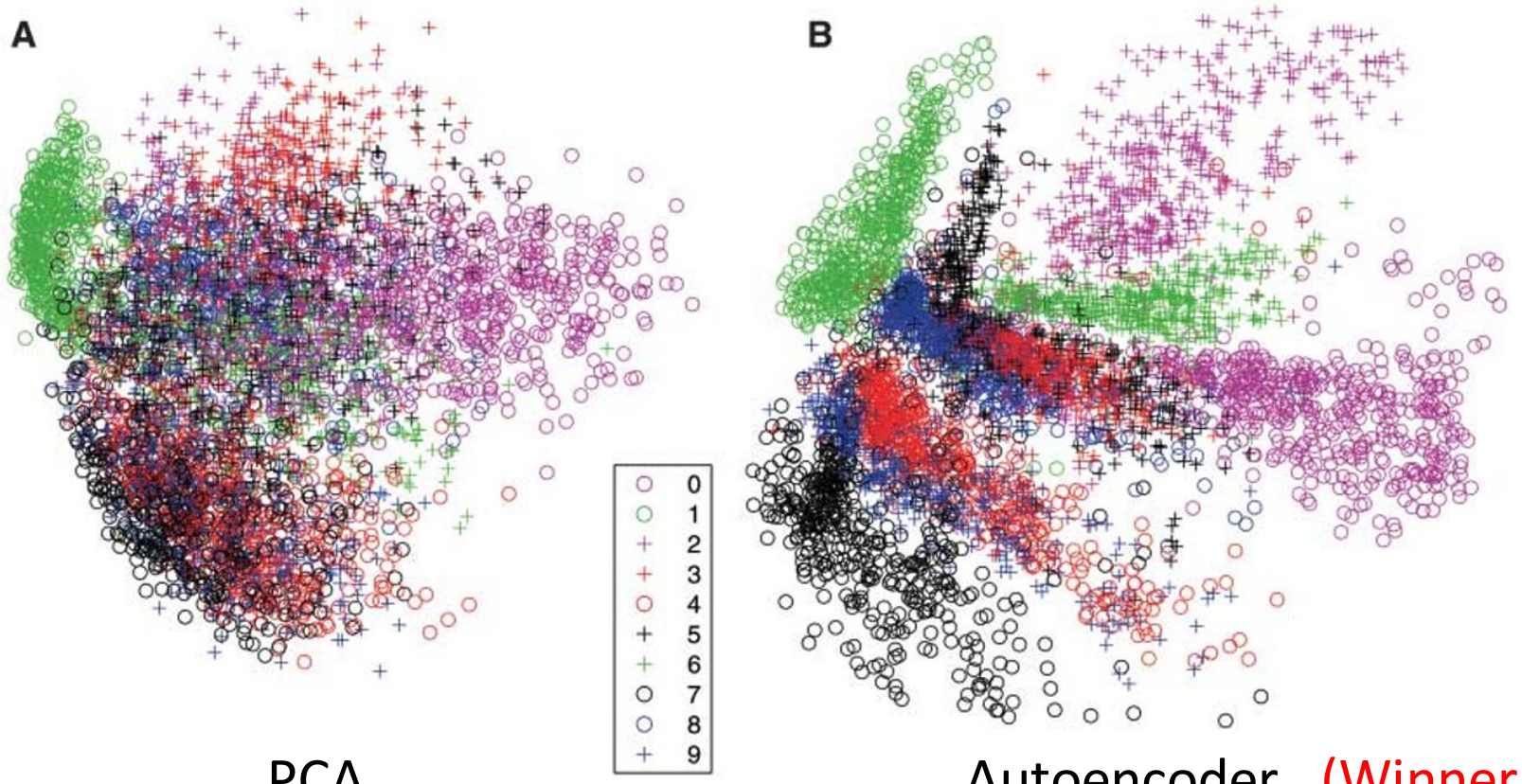
→ Better reconstructions

Hinton and Salakhutdinov , Reducing the Dimensionality of Data with Neural Networks, 2006

Vanilla Autoencoder

Power of Latent Representation: t-SNE visualization on MNIST

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*).



PCA

Autoencoder (Winner)

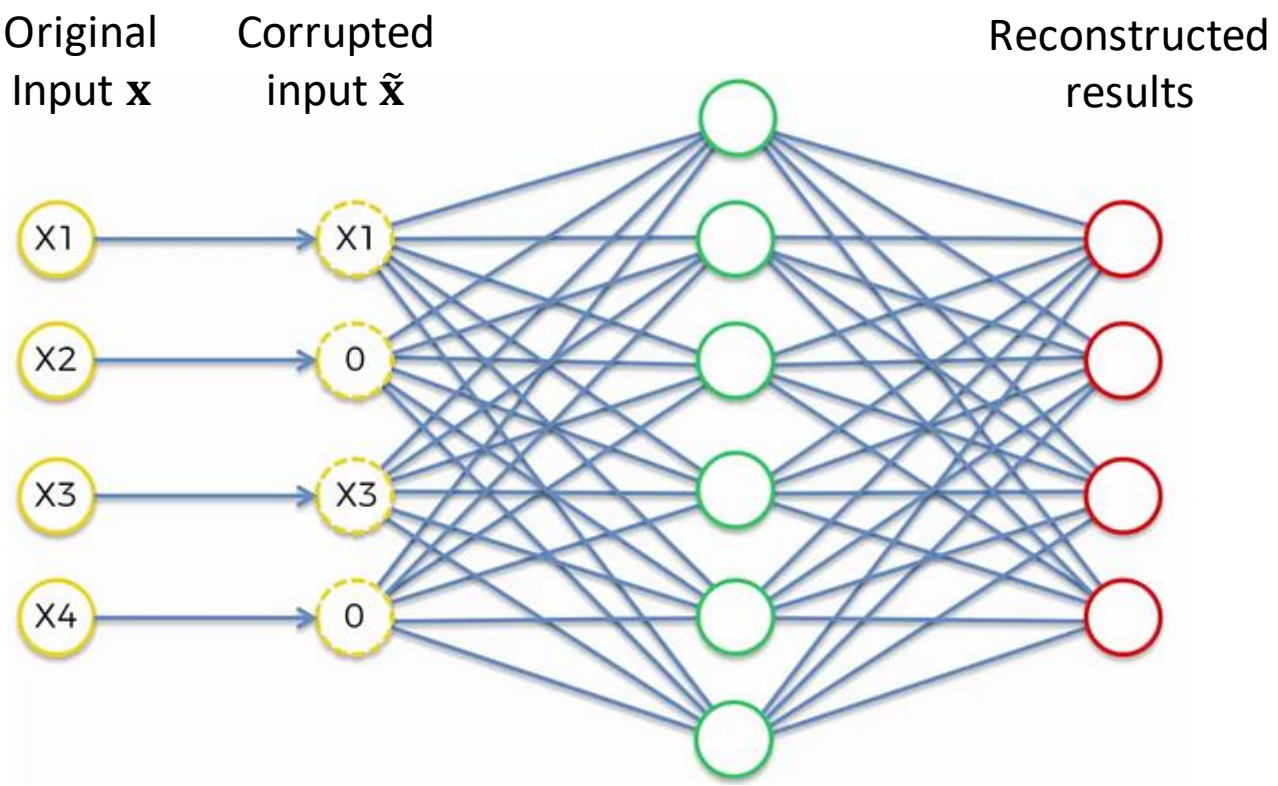
Hinton and Salakhutdinov , Reducing the Dimensionality of Data with Neural Networks, 2006

Denoising Autoencoder (DAE)

Overcomplete representation

- When there are more network parameters than the number of data points:
  - Avoid overfitting
  - Learn robust representations
- Applying dropout between the input and the first hidden layer

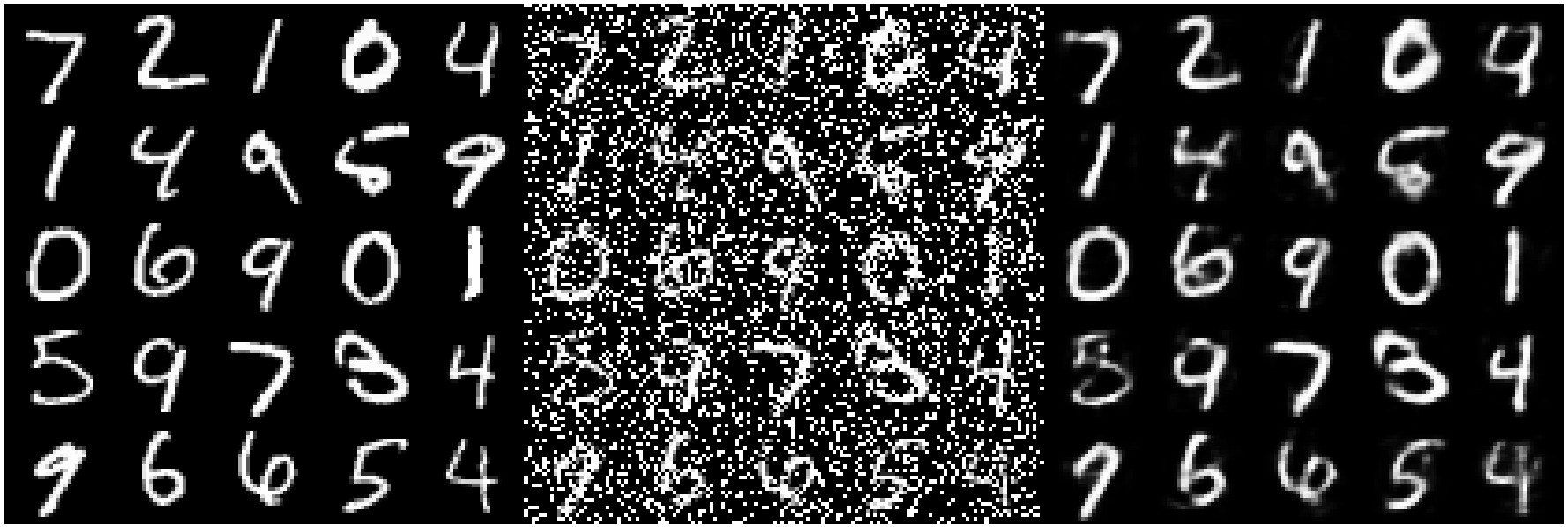
$$L_{\mathrm{DAE}}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}^{(i)} - f_\theta(g_\phi(\tilde{\mathbf{x}}^{(i)})))^2$$



- DAE was proposed 4 years before the dropout paper (Hinton, et al. 2012).
- DAE can be seem as one type of data augmentation on the input.

Vincent et al. 2008 & Kirill Eremenko

Denoising Autoencoder: Example

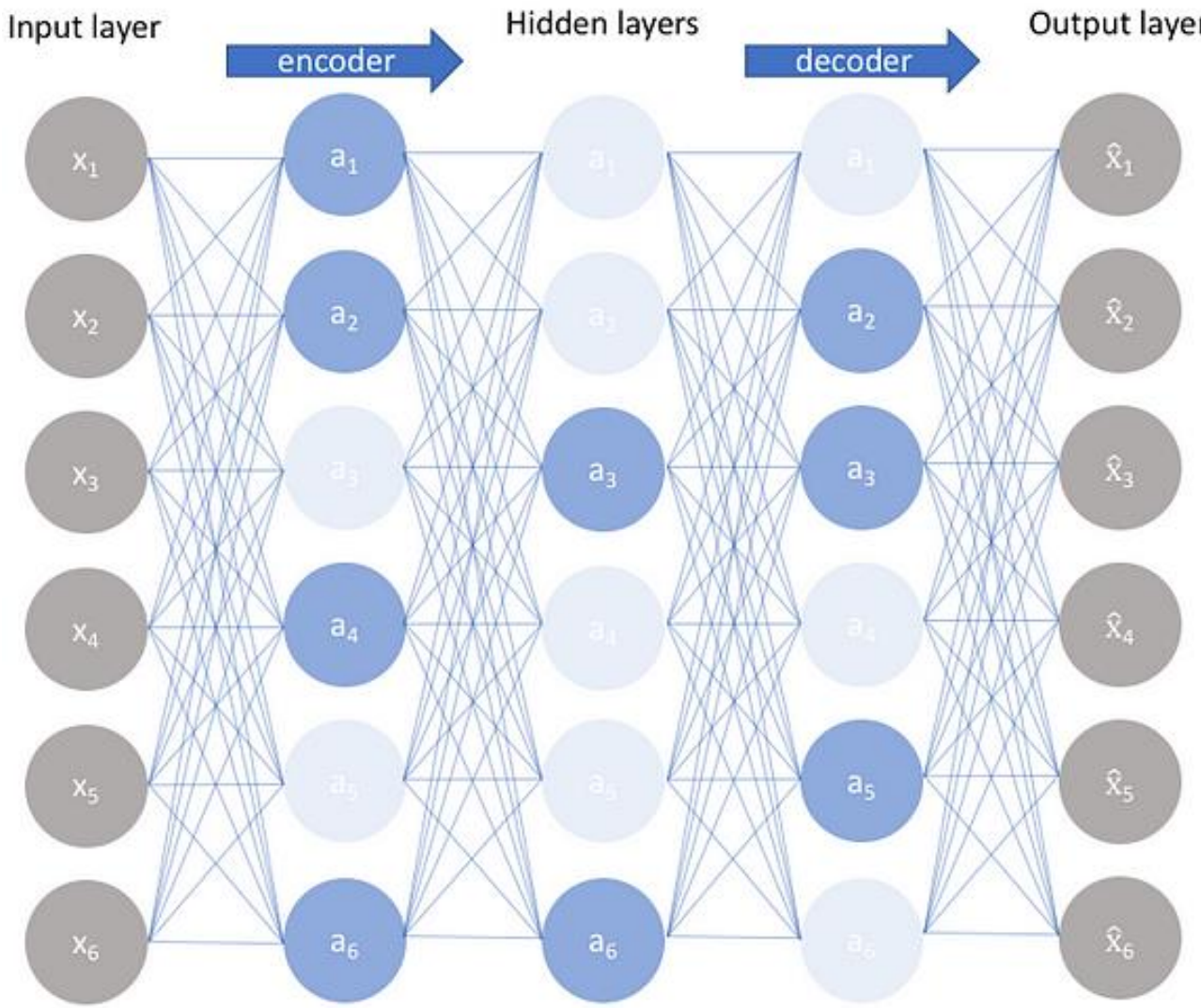


Original input Corrupted data Reconstructed data

http://www.opendeep.org/v0.0.5/docs/tutorial-your-first-model

Sparse Autoencoder

- Imposing a “sparsity” constraint on the hidden units.
- When the number of hidden units is large, we still want to discover interesting structure.
- Sparsity penalty:
  - L1-regularization $\sum_i \left| a_i^{(h)} \right|$
  - KL-divergence $\sum_j KL\left(\rho || \hat{\rho}_j\right)$



Sparse Autoencoder

Recap: KL-Divergence

$$D_{KL}(p||q) = \sum_{i=1}^{N} p(x_i) \cdot (\log p(x_i) - \log q(x_i))$$

OR $$D_{KL}(p||q) = \sum_{i=1}^{N} p(x_i) \cdot log\frac{p(x_i)}{q(x_i)}$$

- Smaller == Closer
- $D_{KL}(p||q) = 0 \iff p = q$

Sparse Autoencoder

Sparsity Regularization using KL-Divergence:

- Given n data samples, and Sigmoid activation function, the active ratio of an input $a_j^{(l)}$:

$$\hat{\rho}_j^{(l)} = \frac{1}{n}\sum_{i=1}^{n}[a_j^{(l)}(\mathbf{x}^{(i)})] \approx \rho$$

($a_j^{(l)}$ is the activation function for the j-th neuron in the l-th hidden layer)

- 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)
- Sparsity loss:

$$\sum_{l=1}^{L}\sum_{j=1}^{s_l} D_{\mathrm{KL}}(\rho \| \hat{\rho}_j^{(l)}) \qquad (\text{Smaller } \rho == \text{Sparser})$$

$$= \sum_{l=1}^{L}\sum_{j=1}^{s_l} \rho \log\frac{\rho}{\hat{\rho}_j^{(l)}} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j^{(l)}}$$

Contractive Autoencoder

Motivation

- 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

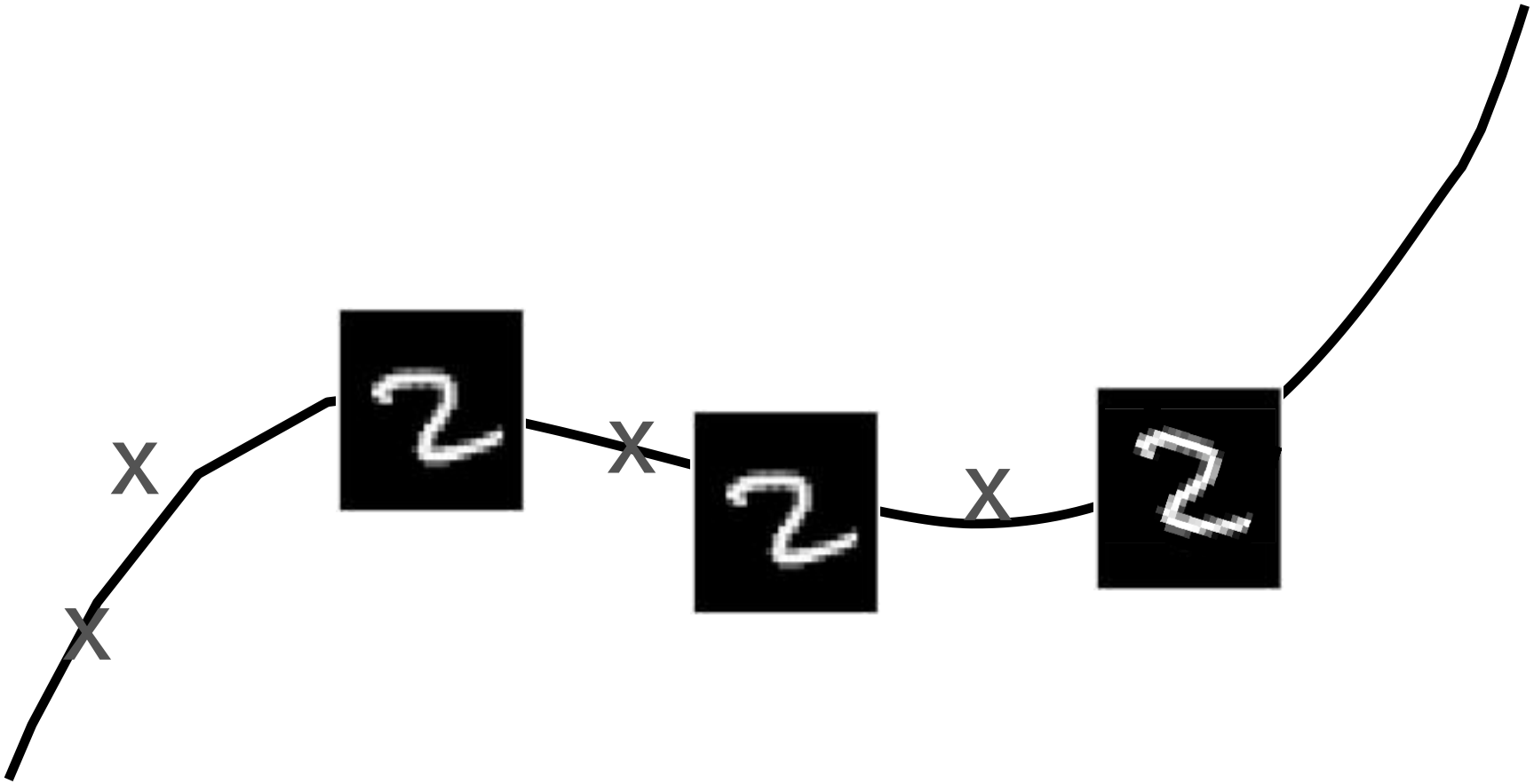
Contractive Autoencoder

How?

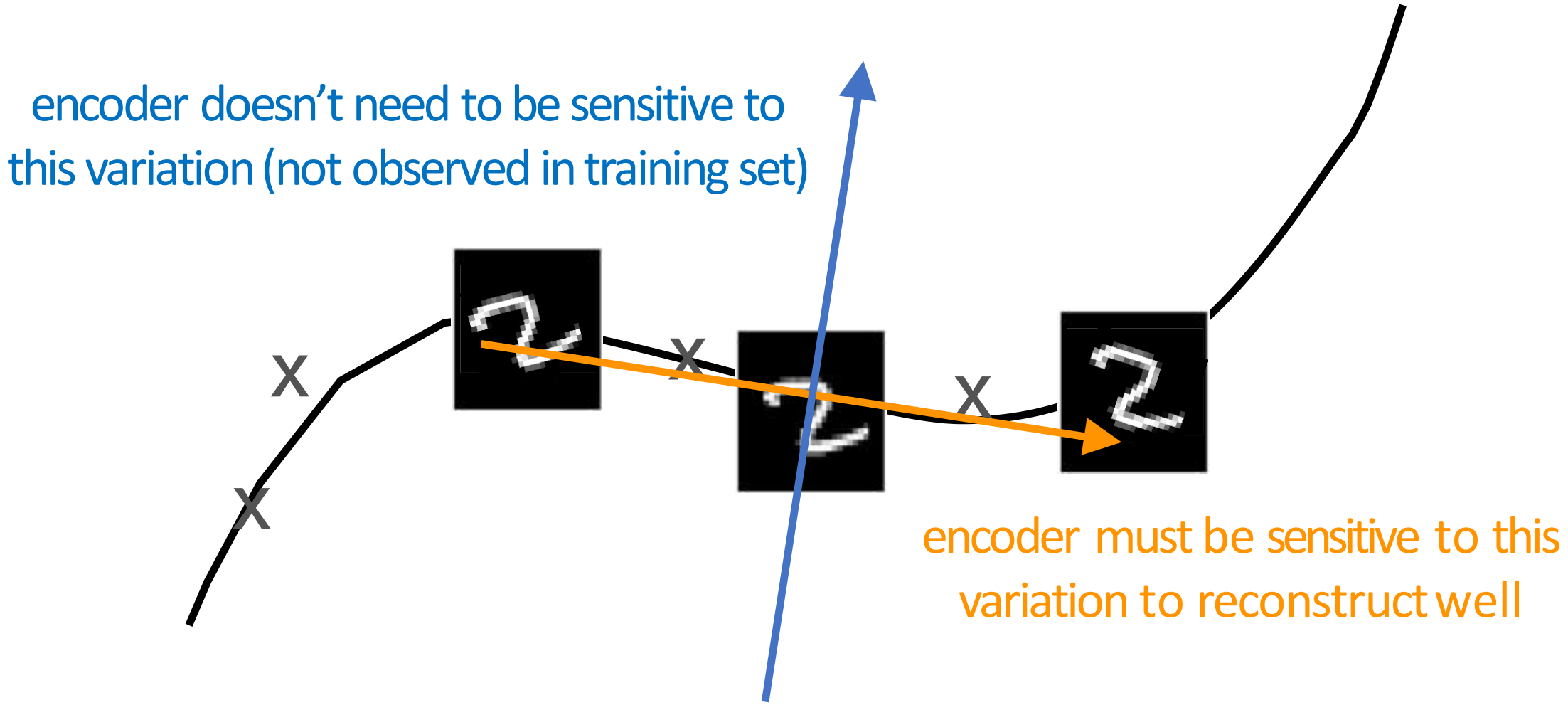
- Penalize the representation being too sensitive to the input
- Improve the robustness to small perturbations
- Measure the sensitivity by the Frobenius norm of the Jacobian matrix of the encoder activations

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

Contractive Autoencoder



Contractive Autoencoder



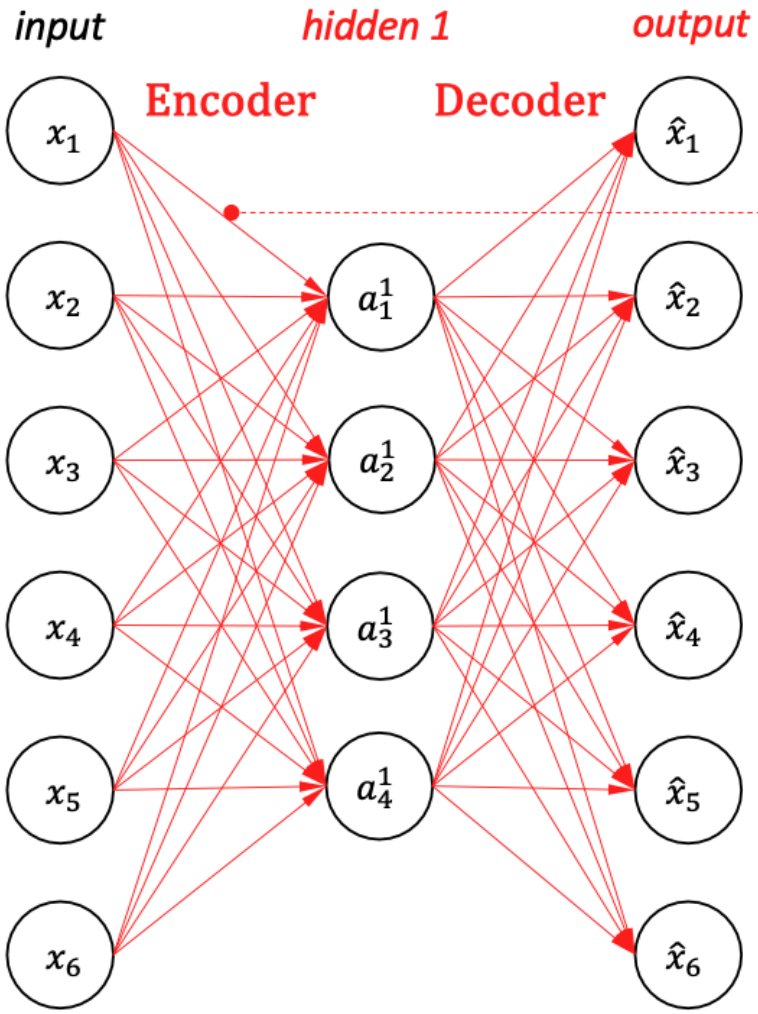
Denoising VS. Contractive Autoencoder

- Denoising autoencoder:
  - simpler to implement
- Contractive autoencoder:
  - gradient is deterministic
  - can better model the distribution of raw data

- To learn more: Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. Salah Rifai, et al., 2011.

Stacked Autoencoder

- Start from Autoencoder: Learn Feature From Input



Unsupervised

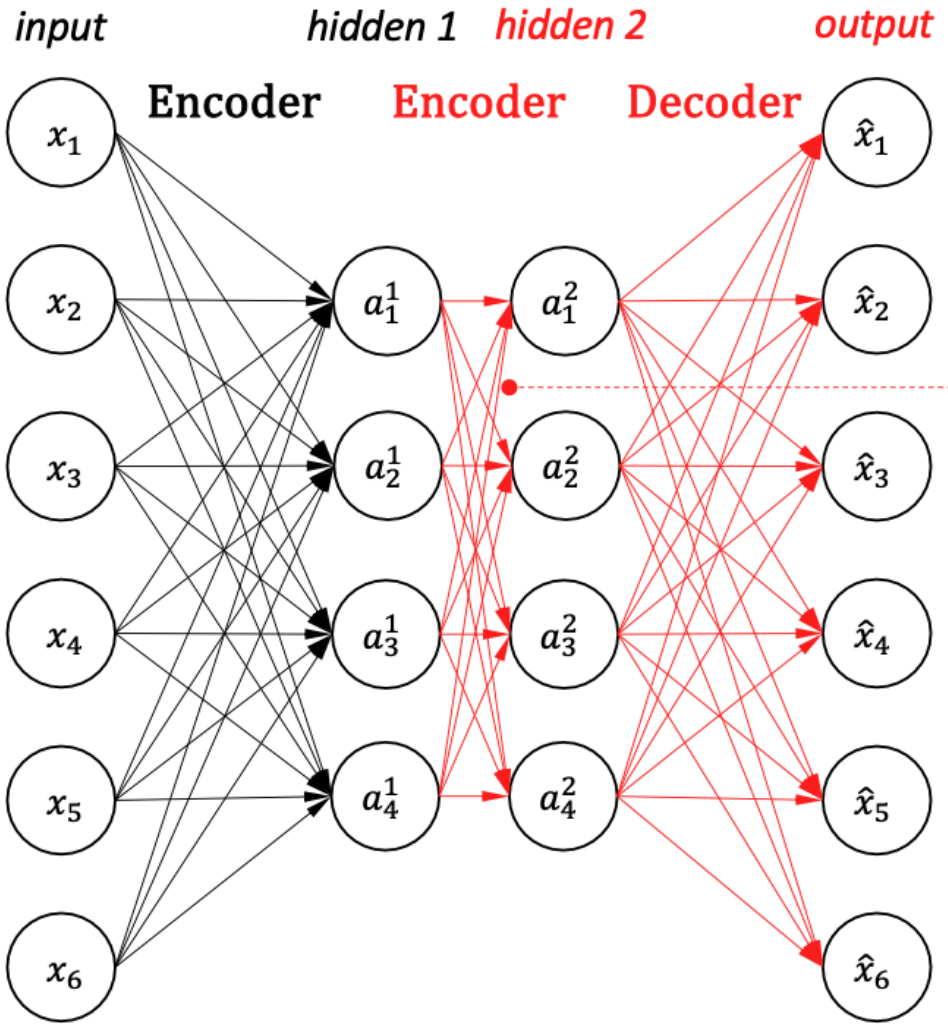
- The feature extractor for the input data

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

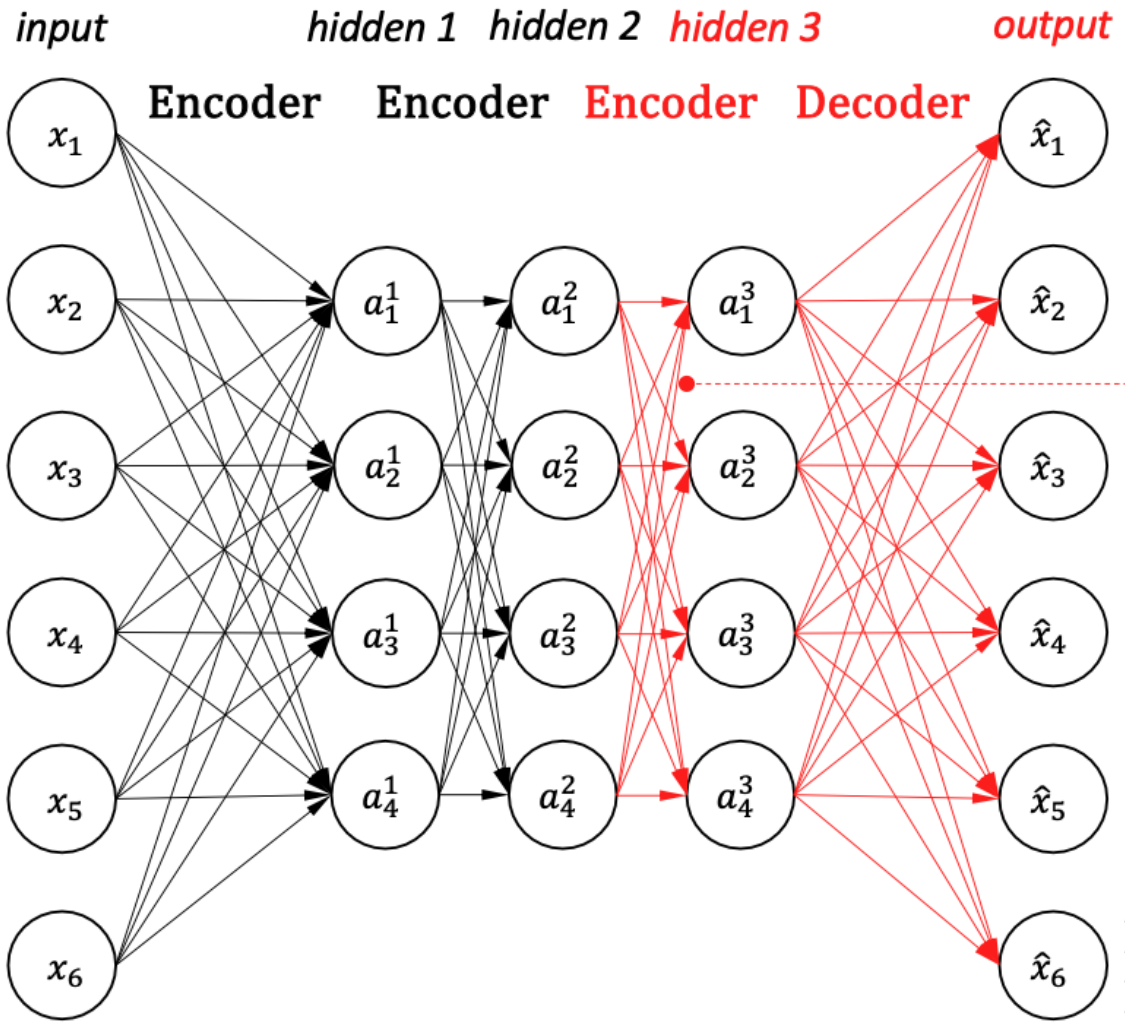
- The feature extractor for the first feature extractor

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

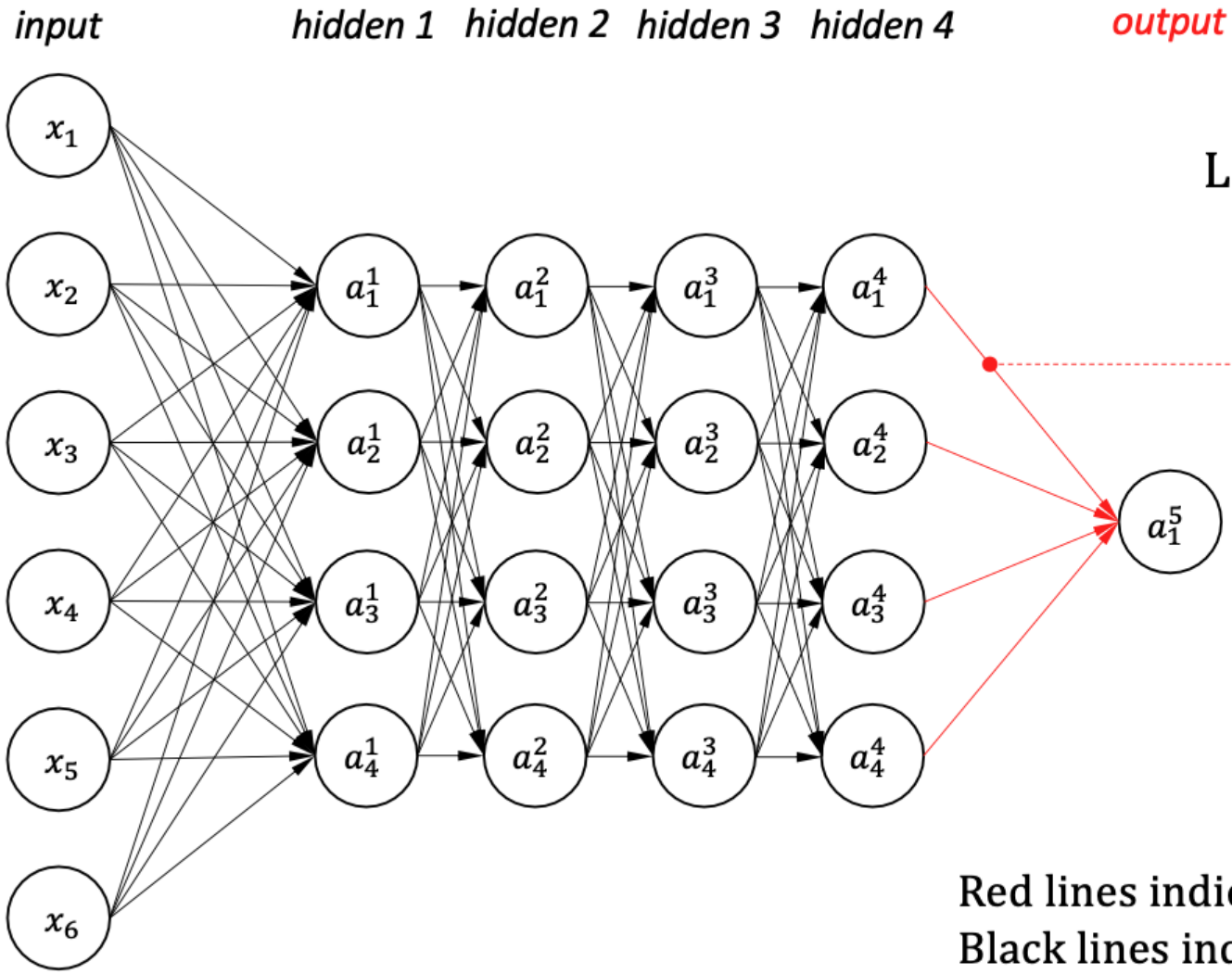
- The feature extractor for the second feature extractor

Red color indicates the trainable weights

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

Stacked Autoencoder

- Use the Learned Feature Extractor for Downstream Tasks



Supervised

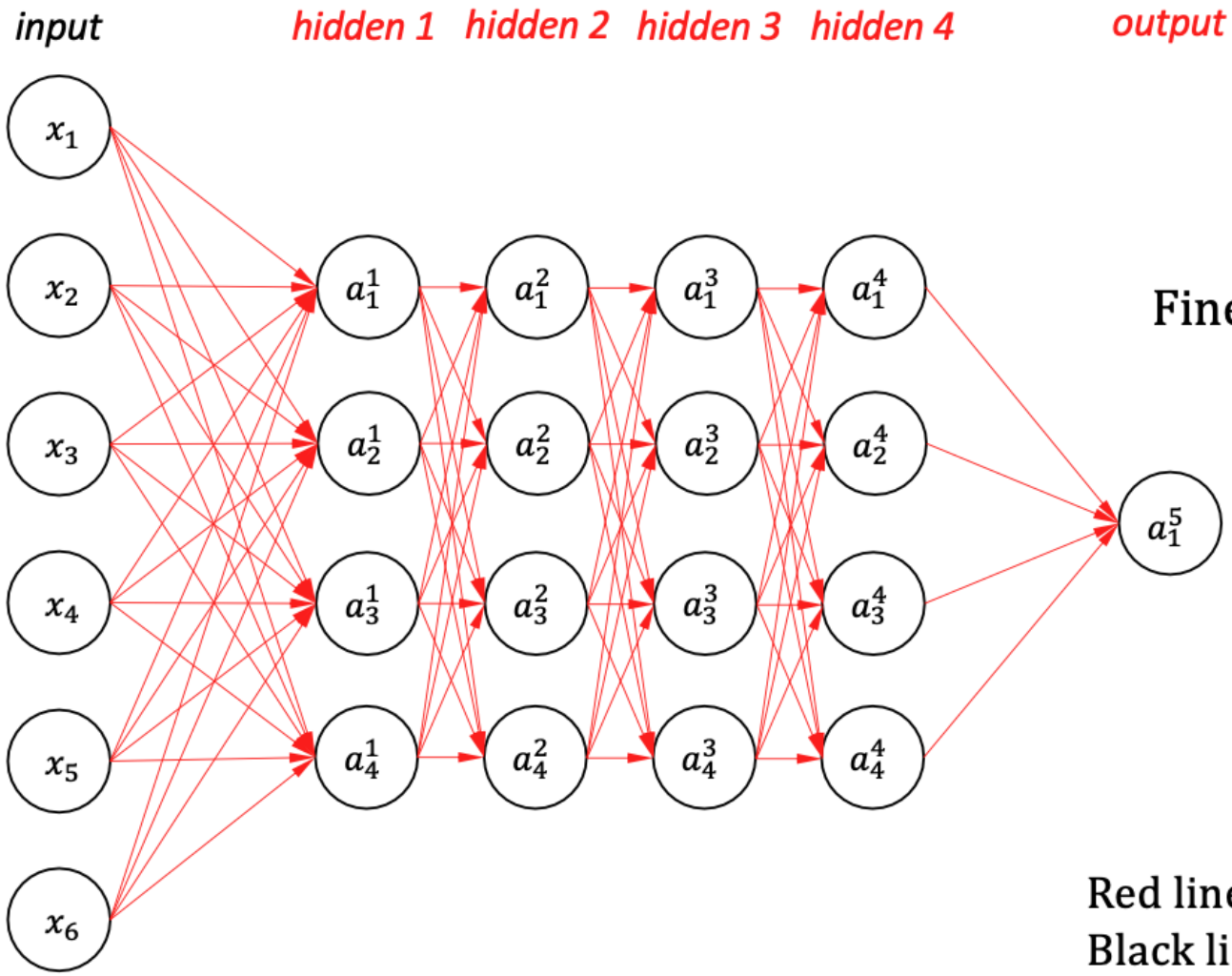
Learn to classify the input data by using the labels and high-level features

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

- Advantages?

- Disadvantages?

VAE: Variational Autoencoder

Before we start

- Questions: Are the previous Autoencoders generative model?

  Recap: We want to learn a probability distribution $p(x)$ over x

  - **Generation (sampling)**: $\mathbf{x} \sim p(\mathrm{x})$

    (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

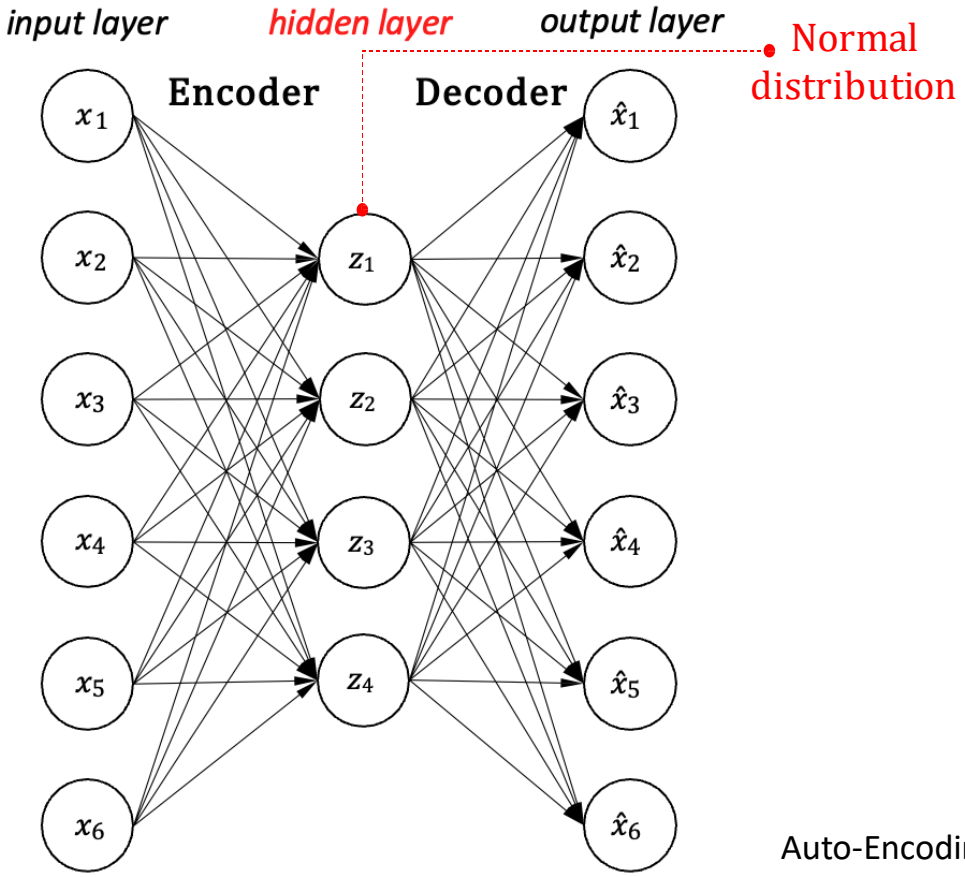
    NO

  - **Unsupervised Representation Learning**: Discovering the underlying structure from the data distribution (e.g., ears, nose, eyes …)

    (YES, Autoencoders learn the feature representation)

VAE: Variational Autoencoder

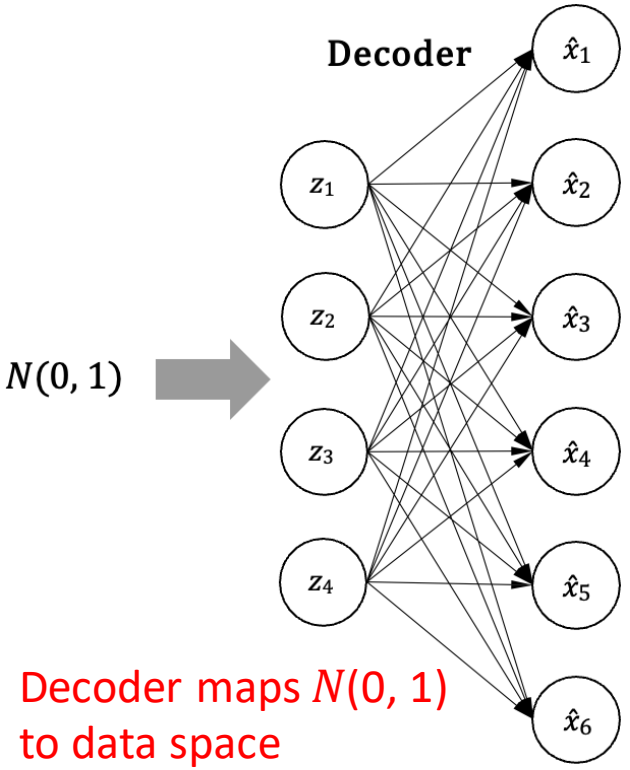
- How to perform generation (sampling)?
- Instead of mapping the input into a fixed vector, we want to map it into a distribution p_θ, *e.g.*, Normal distribution



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

VAE: Variational Autoencoder

- How to perform generation (sampling)?
- Instead of mapping the input into a fixed vector, we want to map it into a distribution p_θ, *e.g.*, Normal distribution
- Assuming that we know the real parameter θ^* for this distribution. In order to generate a sample that looks like a real data point $\mathbf{x}^{(i)}$, we follow these steps:
  - First, sample a $\mathbf{z}^{(i)}$ from a prior distribution $p_{\theta^*}(\mathbf{z})$
  - Then a value $\mathbf{x}^{(i)}$ is generated from a conditional distribution $p_{\theta^*}(\mathbf{x}|\mathbf{z}=\mathbf{z}^{(i)})$.

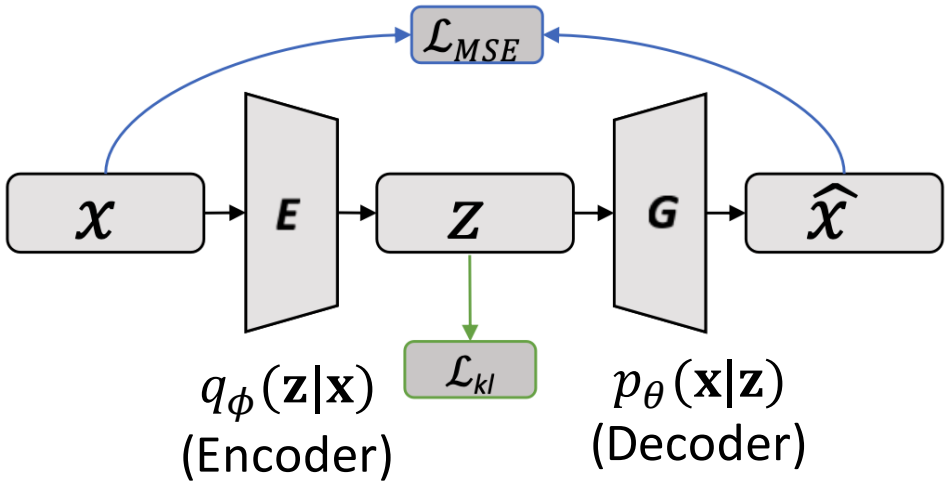


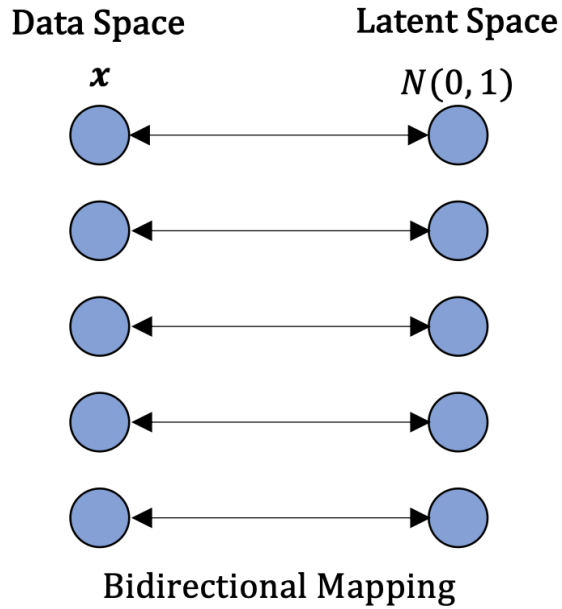
Auto-Encoding Variational Bayes. Diederik P. Kingma, Max Welling. ICLR 2013

VAE: Variational Autoencoder

- From Neural Network Perspective:

  A variational autoencoder consists of an encoder, a decoder, and a loss function





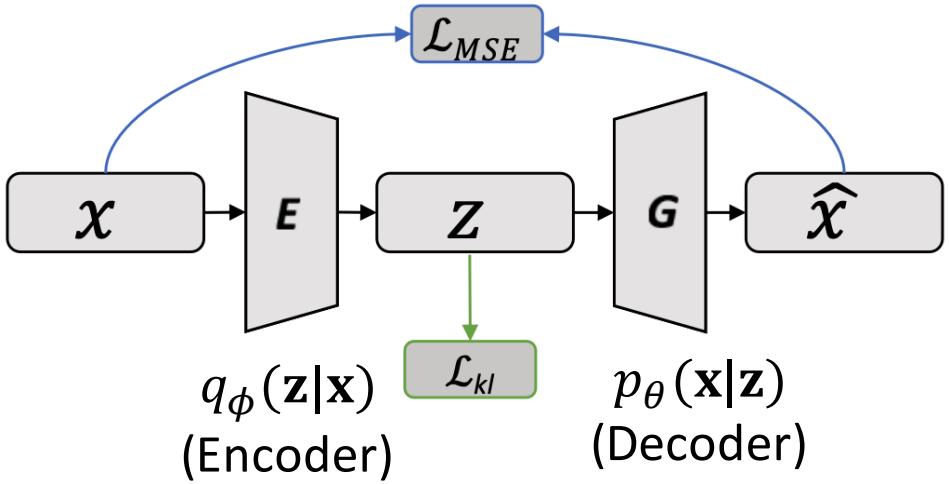
$$\mathcal{L}_{total} = \mathcal{L}_{MSE} + \mathcal{L}_{KL}$$

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

VAE: Variational Autoencoder

- Loss function

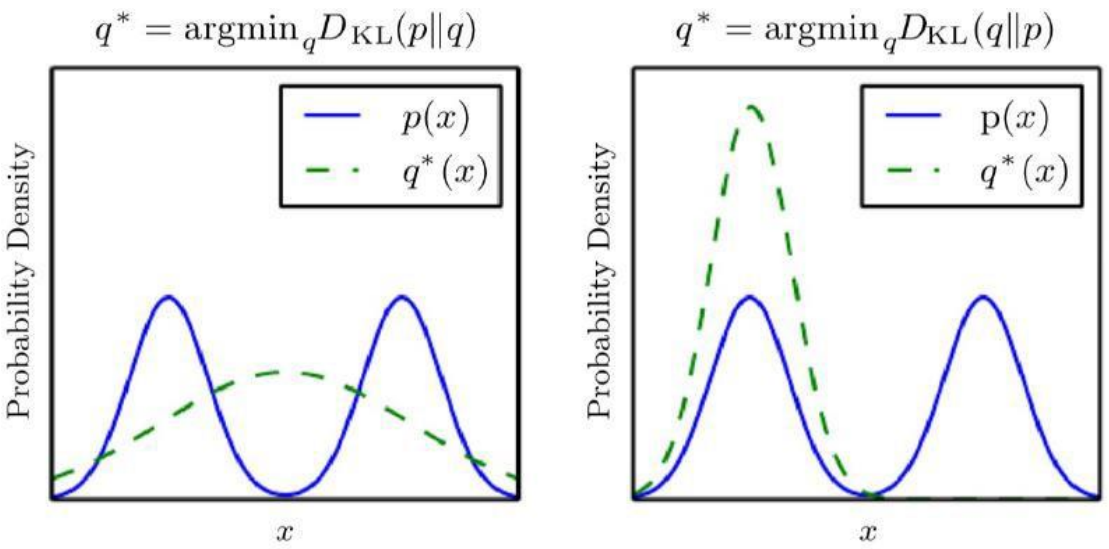
$$L_{\mathrm{VAE}}(\theta,\phi) = -\log p_\theta(\mathbf{x}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x}))$$
$$= \underbrace{-\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\log p_\theta(\mathbf{x}|\mathbf{z})}_{\text{Can be represented by MSE}} + \underbrace{D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}))}_{\text{regularisation}}$$



$$\mathcal{L}_{total} = \mathcal{L}_{MSE} + \mathcal{L}_{KL}$$
$$= \mathcal{L}(\mathbf{x},\hat{\mathbf{x}}) + \mathbb{KL}(q_\phi(\mathbf{z}|\mathbf{x})\,||p_\theta(\mathbf{z}|\mathbf{x}))$$

VAE: Variational Autoencoder

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



- 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:

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

VAE: Variational Autoencoder

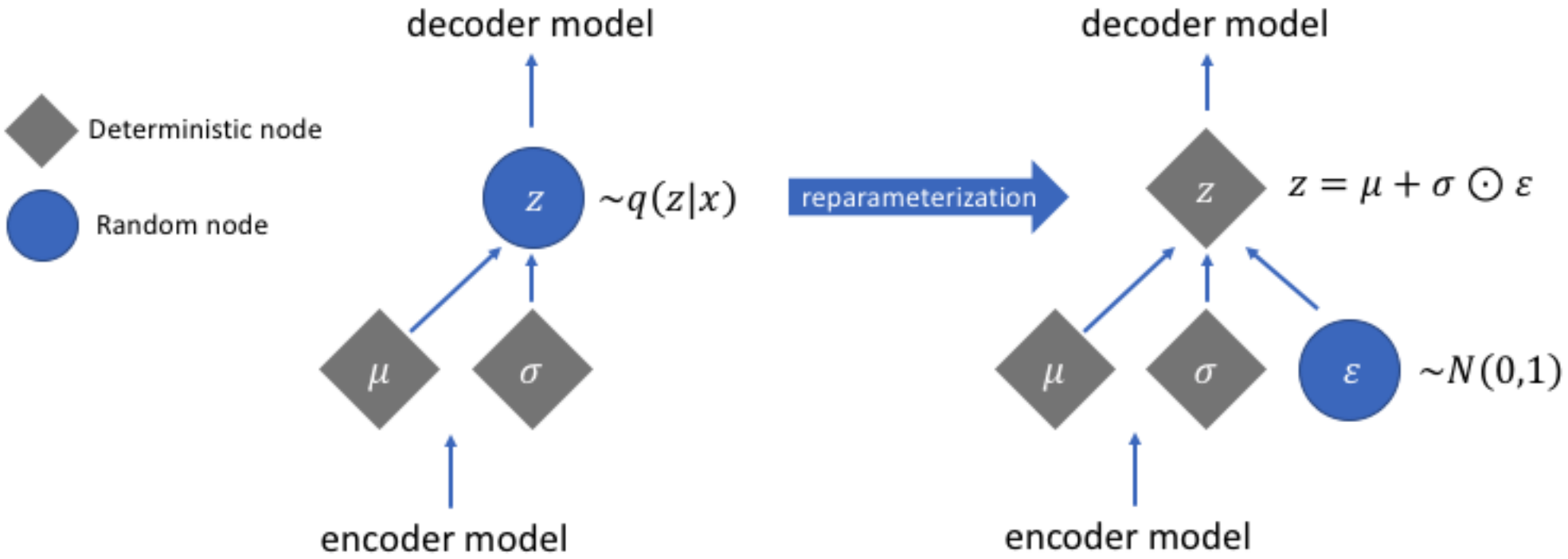
Reparameterization Trick

- The expectation term in the loss function invokes generating samples from $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$
- Sampling is a stochastic process and therefore we cannot backpropagate the gradient.
- Reparameterization trick is introduced to make it trainable

$$
\begin{aligned}
L_{\mathrm{VAE}}(\theta, \phi) &= -\log p_\theta(\mathbf{x}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x})) \\
&= -\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}))
\end{aligned}
$$

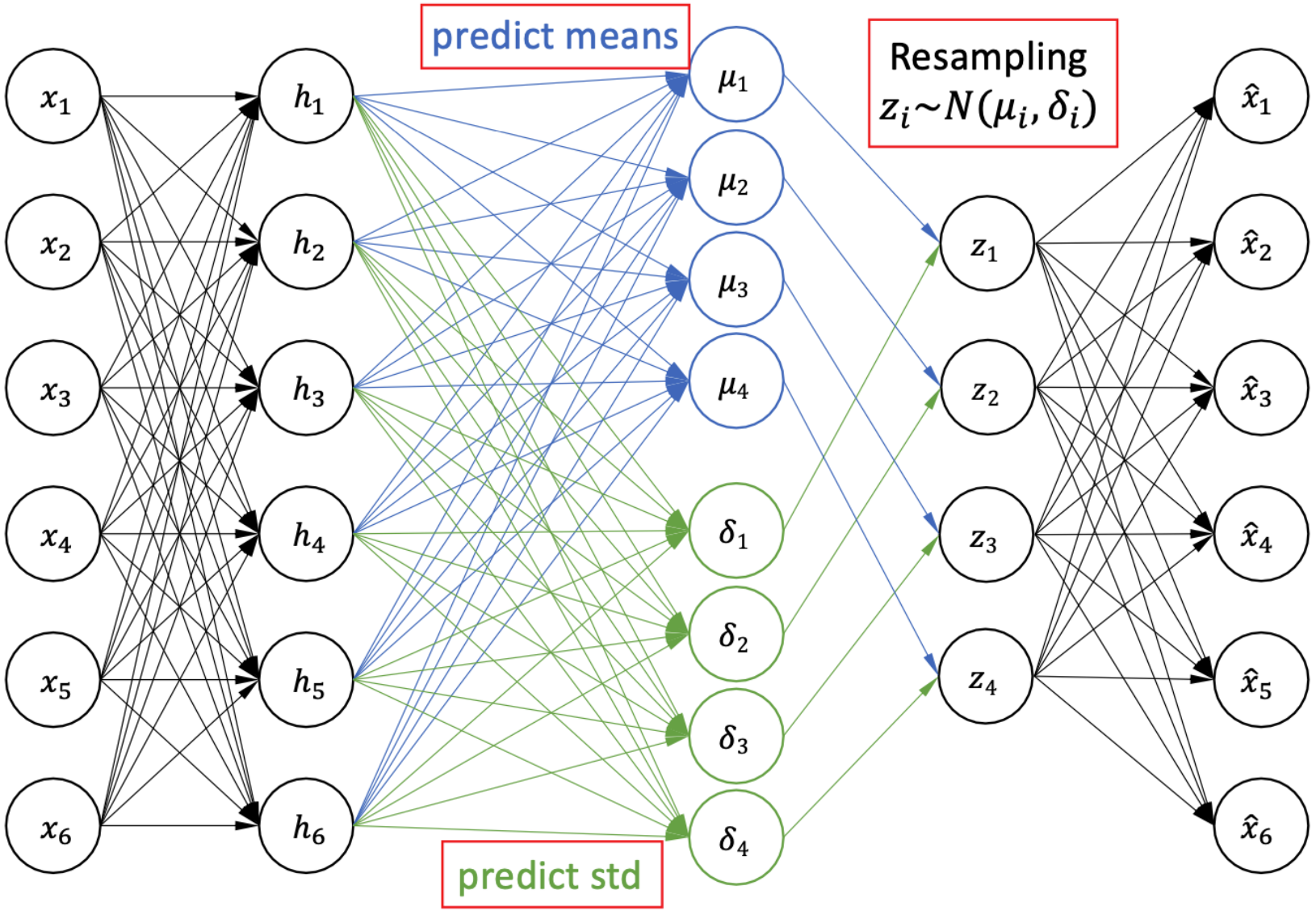
VAE: Variational Autoencoder

Reparameterization Trick



https://andacdemir-94699.medium.com/auto-encoding-variational-bayes-c0a2e0743aac

VAE: Variational Autoencoder



Reparameterization Trick

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

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

Next

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

CASE SCHOOL OF ENGINEERING
CASE WESTERN RESERVE UNIVERSITY

Thank You

- Questions?

- Email: yu.yin@case.edu