北京大學
PEKING UNIVERSITY

Learning Methods

Supervised, Semi-supervised, Weakly-supervised, Unsupervised Learnings

Hao Dong

2019, Peking University

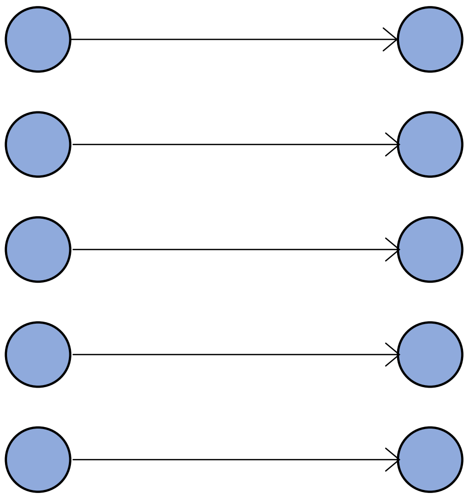
Learning Methods

- Supervised, Semi-supervised, Weakly-supervised, Unsupervised Learnings
- Unsupervised Learning
- Semi-supervised Learning
- Weakly-supervised Learning
- Summary

From Data Point of View: Supervised, Unsupervised, Semi-supervised, and Weakly-supervised Learning

From Data Point of View

Data in both input x and output y
with known mapping
(Learn the mapping f)

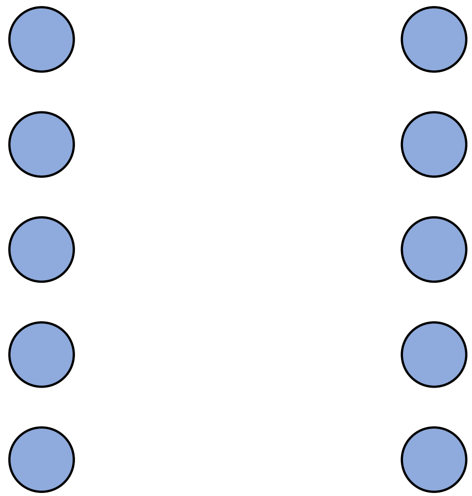


$y = f(x)$

Supervised Learning

- Image classification
- Object detection
- …

Data in both input x and output y
(Learn the mapping f)



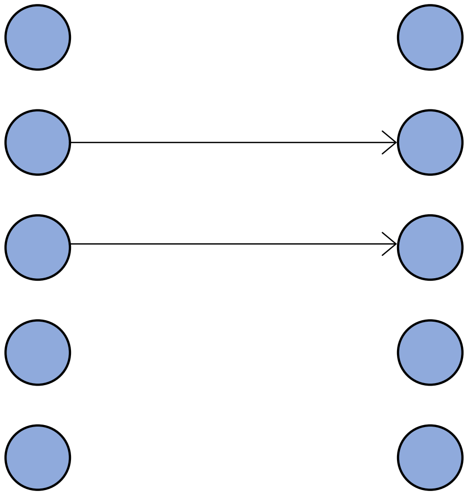
$y = f(x)$

Unsupervised Learning

- Autoencoder
(when output is features)
- GANs
- …

From Data Point of View

Data in both input x and output y
with known partial mapping
(Learn the mapping f)

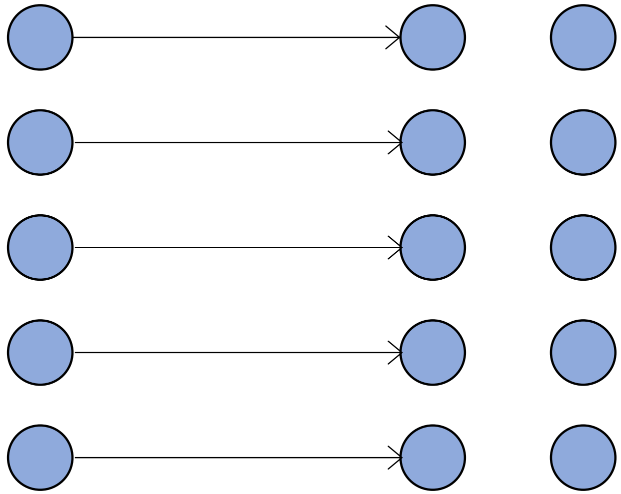


$y = f(x)$

Semi-supervised Learning

- …

Data in both input x and output y
with known mapping for y
(Learn the mapping f for another output y')

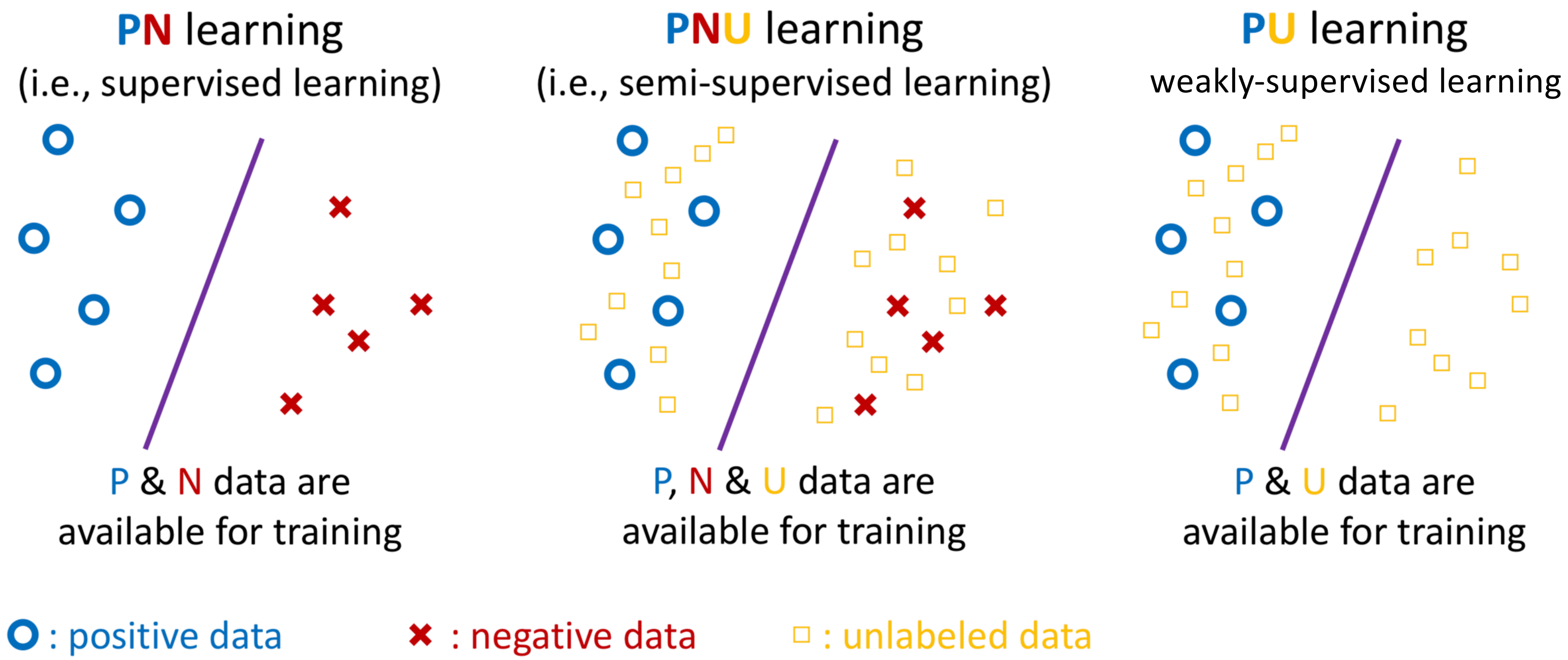


$y' = f(x)$

Weakly-supervised Learning

- Learn segmentation via classification
- …

From Data Point of View



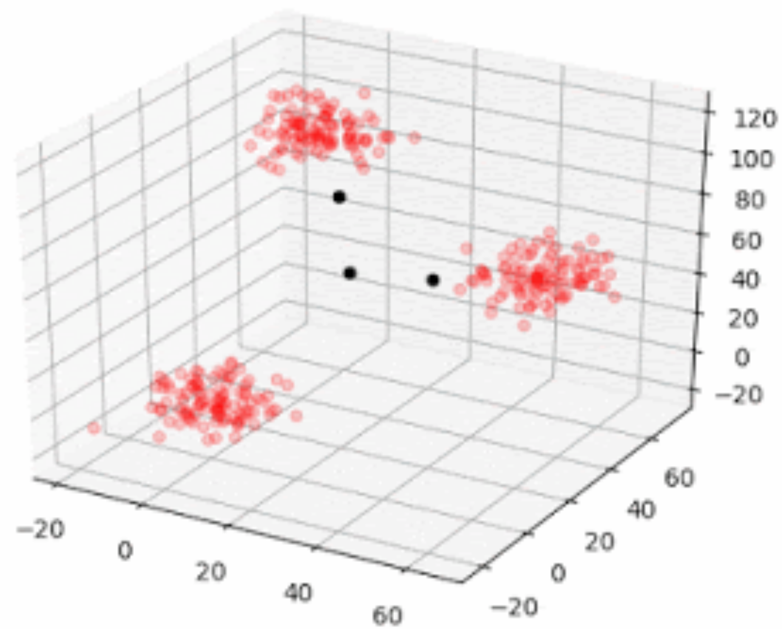
https://niug1984.github.io/paper/niu_tdlw2018.pdf

Unsupervised Learning

Unsupervised Learning

- Unsupervised learning is about problems where we don't have labeled answers, such as clustering, dimensionality reduction, and anomaly detection.

- Clustering: EM
- Dimension Reduction: PCA
- ...



Unsupervised Learning

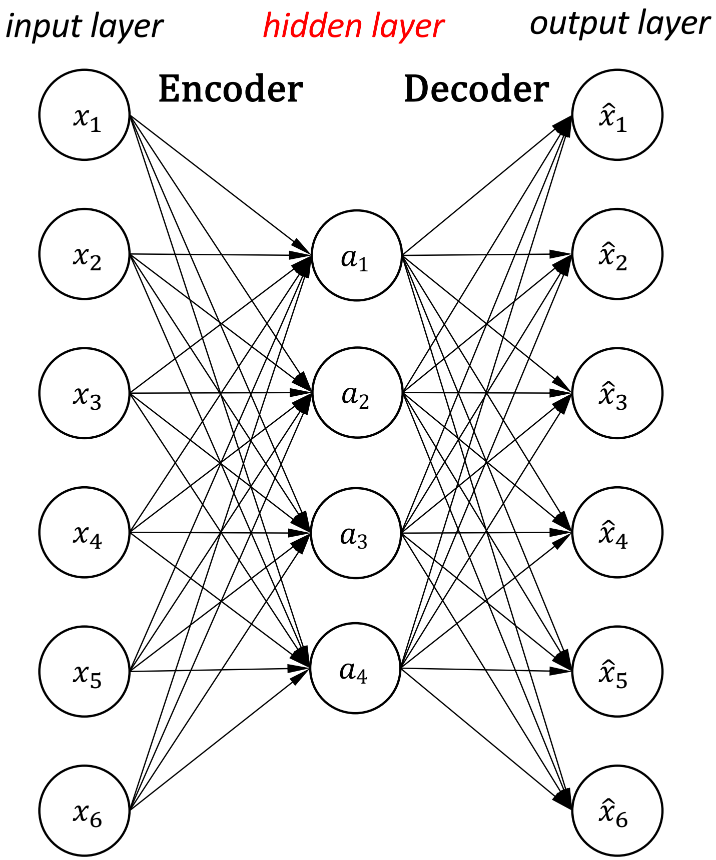
- Autoencoder

    - In practice, it is difficult to obtain a large amount of labeled data, but it is easy to get a large amount of unlabeled data.

    - Learn a good feature extractor using unlabeled data and then learn the classifier using labeled data can improve the performance.

Unsupervised Learning

- Autoencoder



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

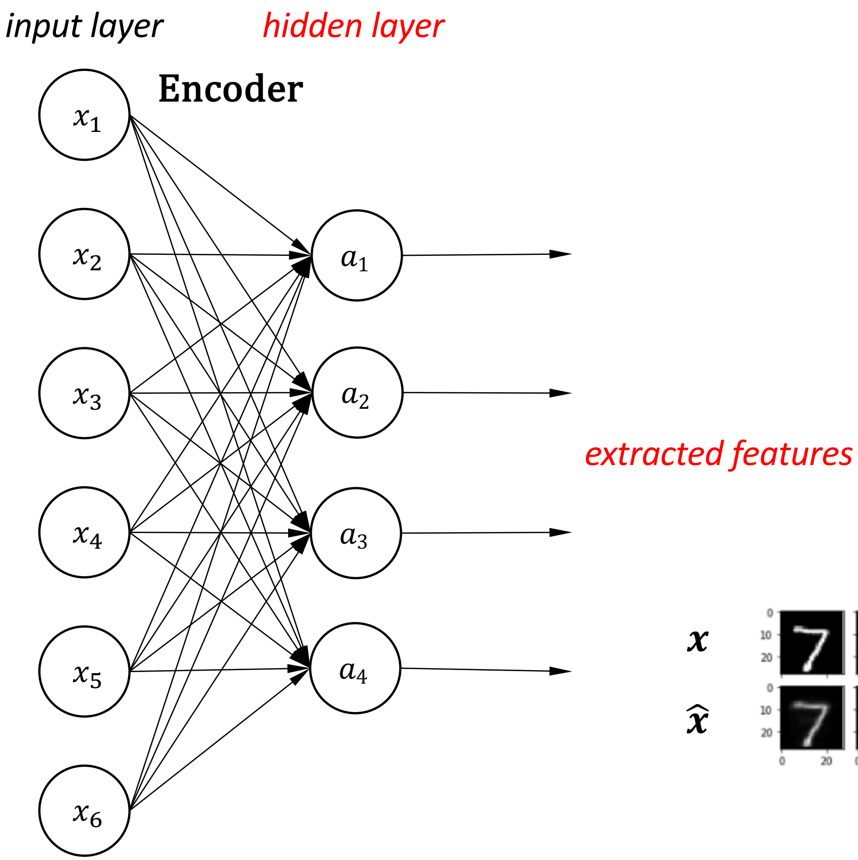
Given M data samples

$$\mathcal{L}_{MSE} = \frac{1}{M}\sum_{m=1}^{M}\|\hat{\boldsymbol{x}}^m - \boldsymbol{x}^m\|_2^2$$

- 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.

Unsupervised Learning

- Autoencoder

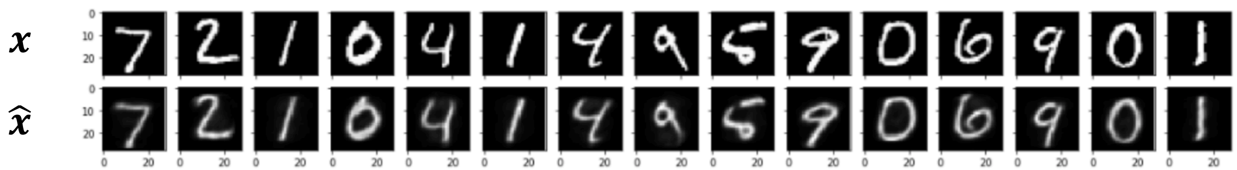


- Autoencoder is an unsupervised learning method if we considered the features as the "output".

- Auto encoder is also a self-taught learning method which is a type of supervised learning where the training labels are determined by the input data.

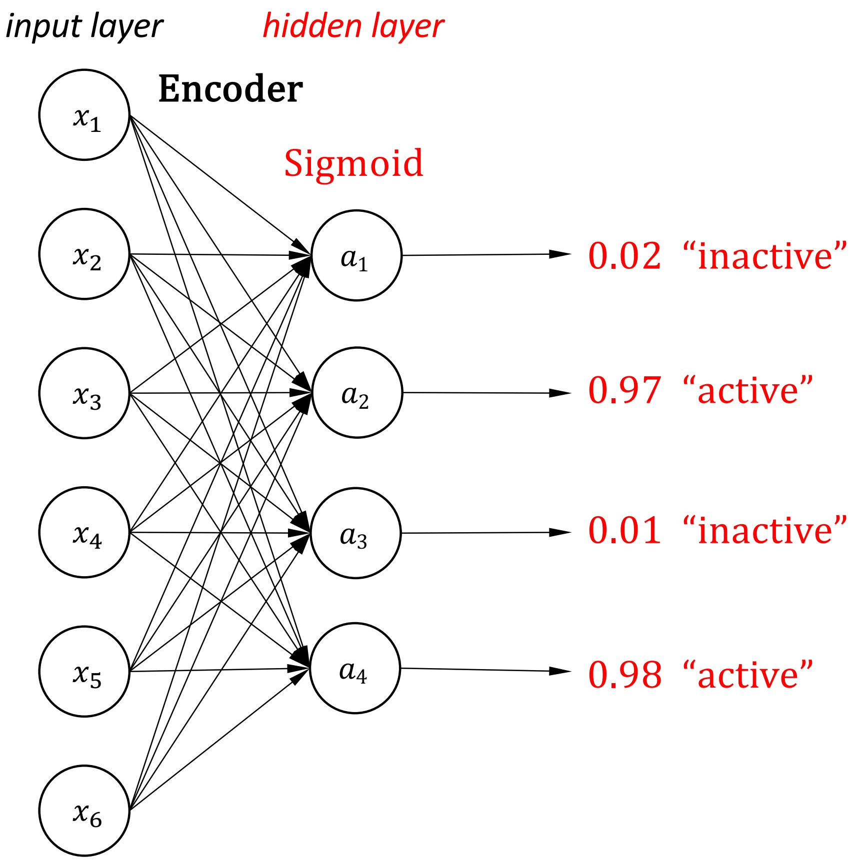
- Word2Vec is another unsupervised, self-taught learning example.

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



Unsupervised Learning

- Sparse Autoencoder

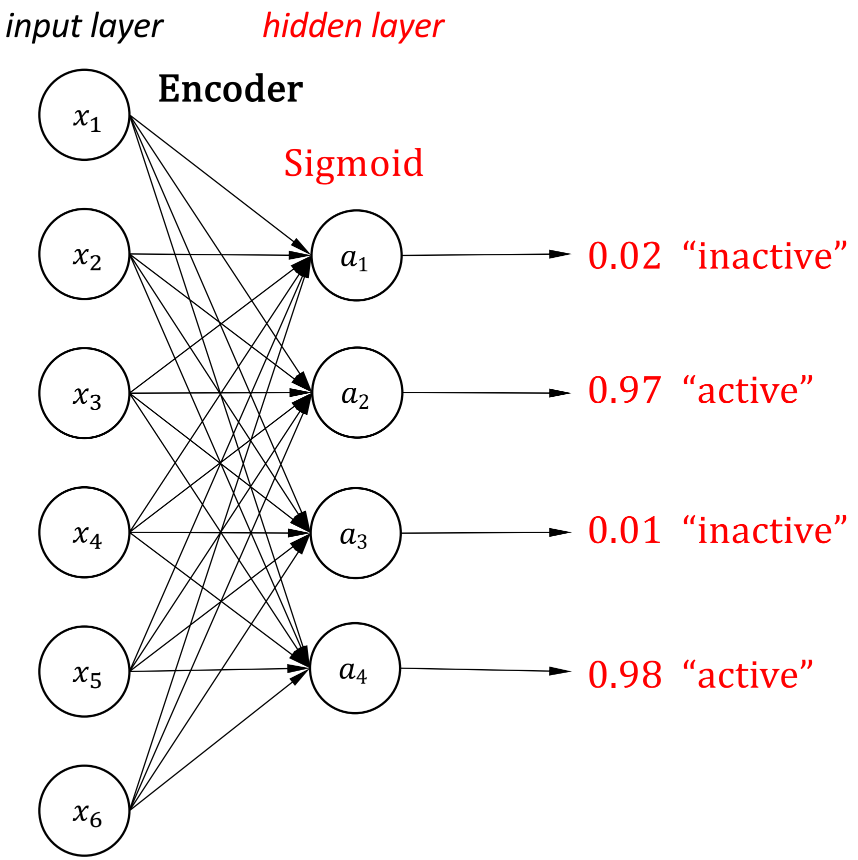


- Even when the number of hidden units is large (perhaps even greater than the number of input pixels), we can still discover interesting structure, by imposing other constraints on the network.

- 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.

Unsupervised Learning

- Sparse Autoencoder



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

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

The 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 output neurons.

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

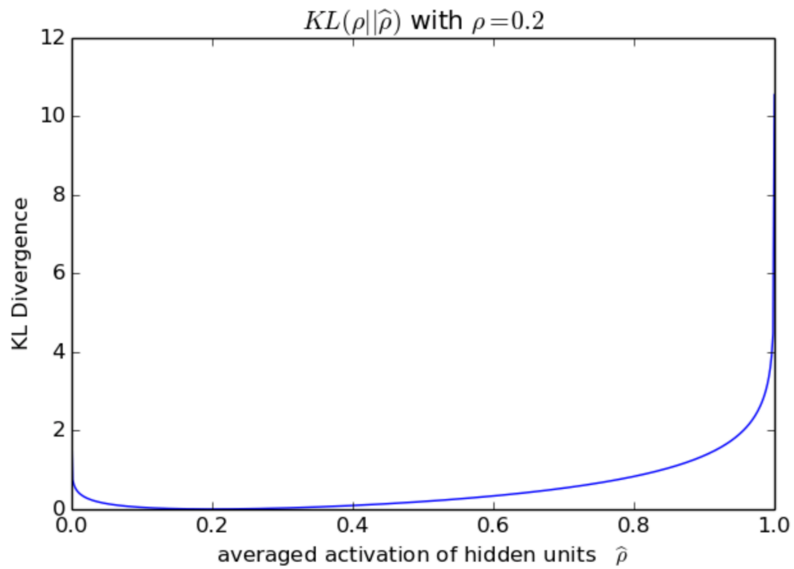
The total loss:

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

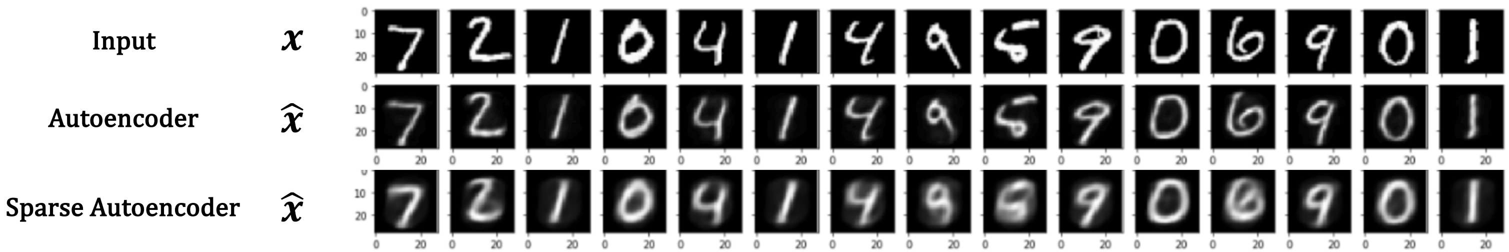
Unsupervised Learning

- Sparse Autoencoder

Smaller ρ == More sparse



Autoencoders for MNIST dataset

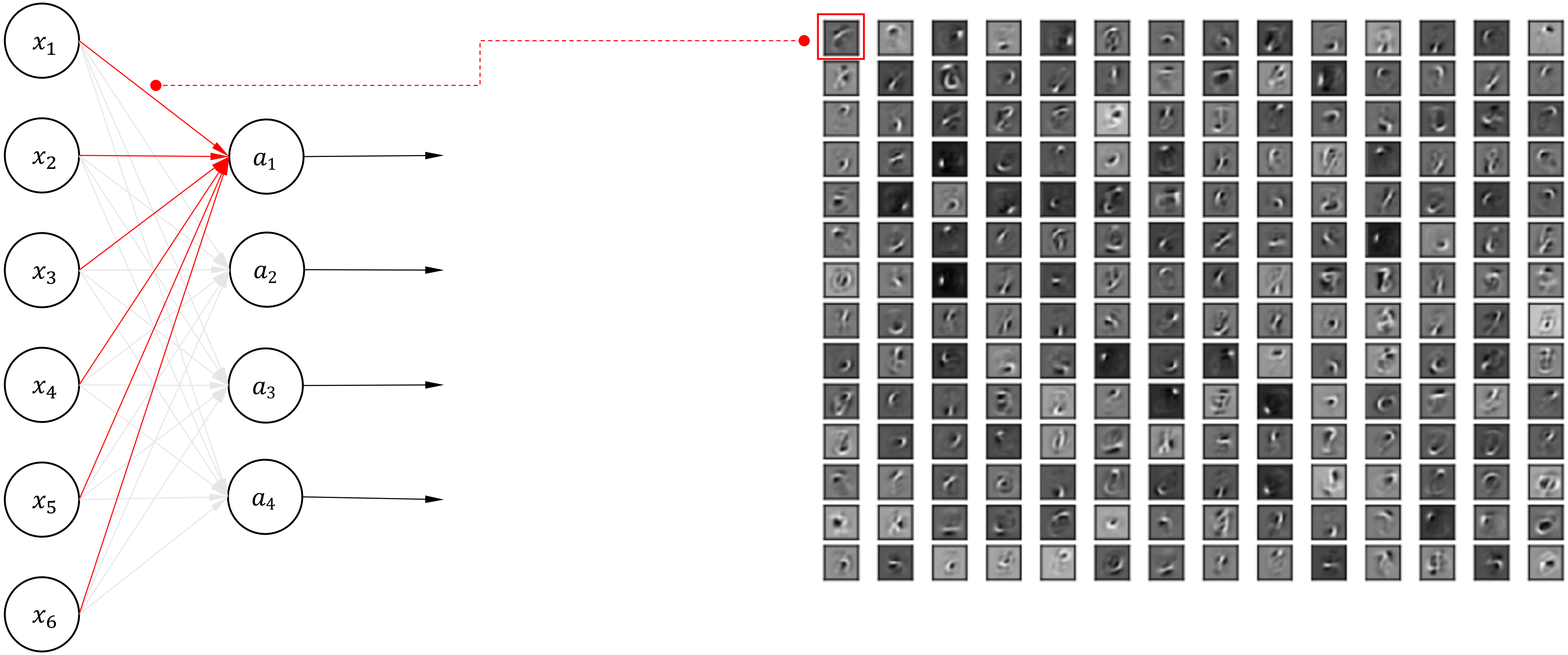


Unsupervised Learning

- Sparse Autoencoder

One neuron == One feature extractor

Visualizing the learned features



Unsupervised Learning

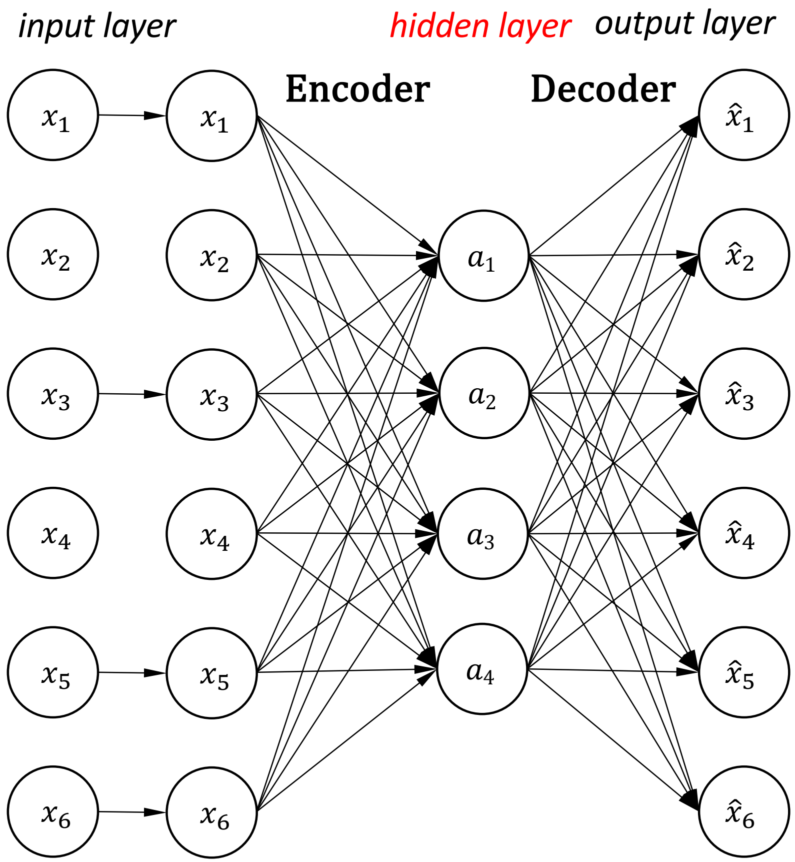
- Sparse Autoencoder

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

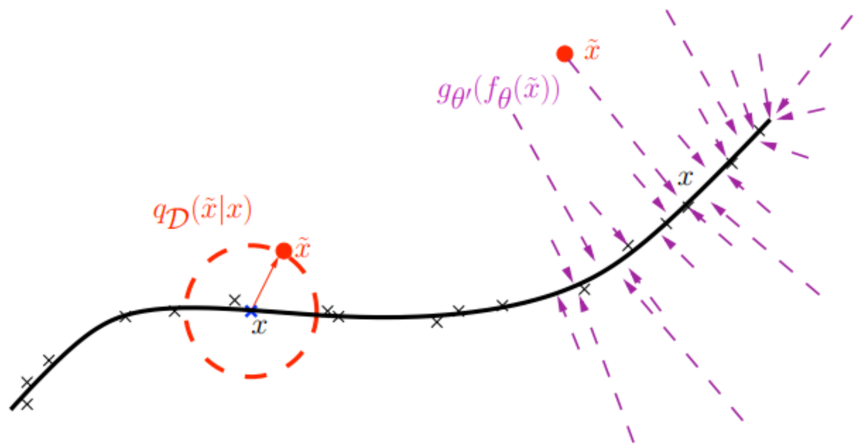
Unsupervised Learning

- Denoising Autoencoder



Applying dropout between the input and the first hidden layer

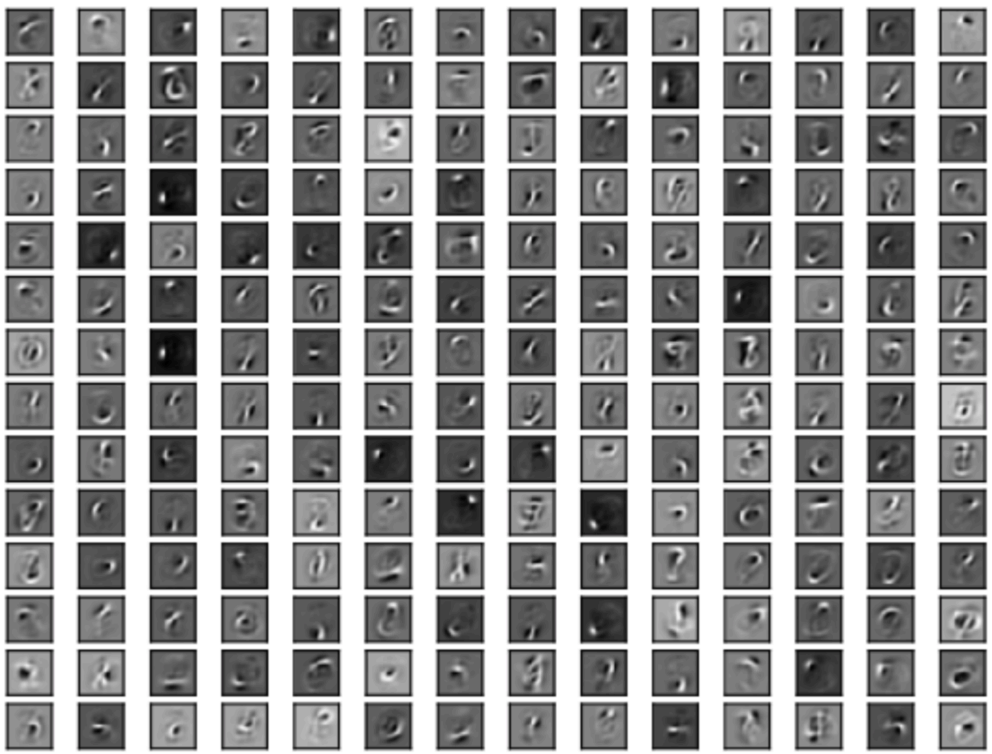
- Improve the robustness



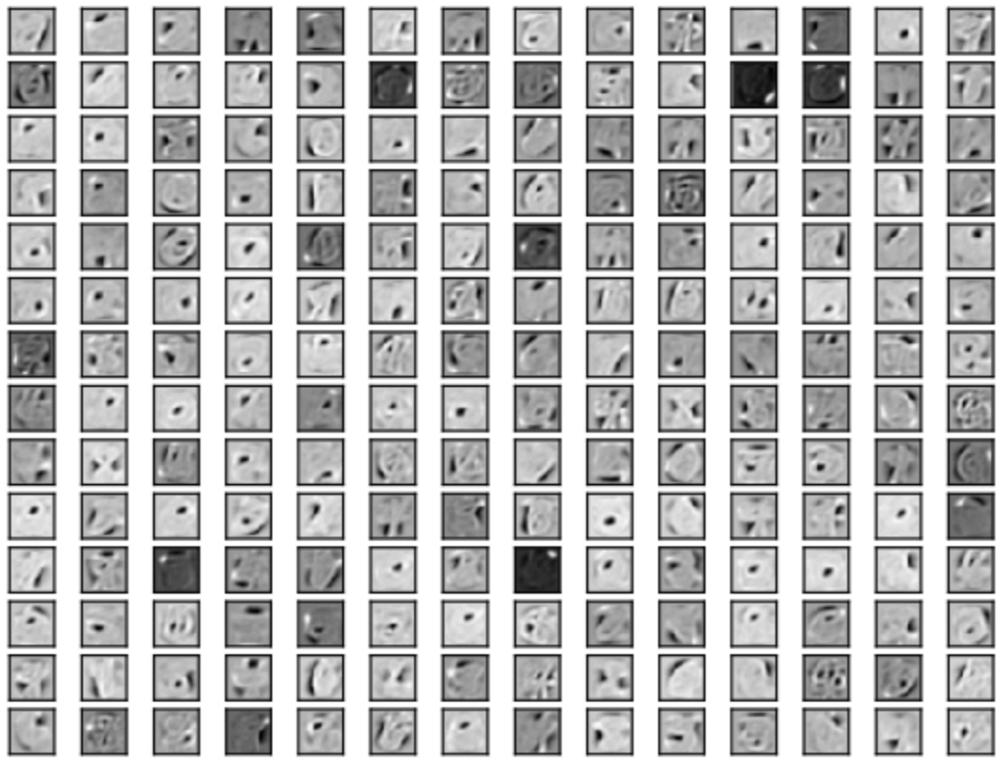
Unsupervised Learning

- Denoising Autoencoder

Features of Sparse Autoencoder



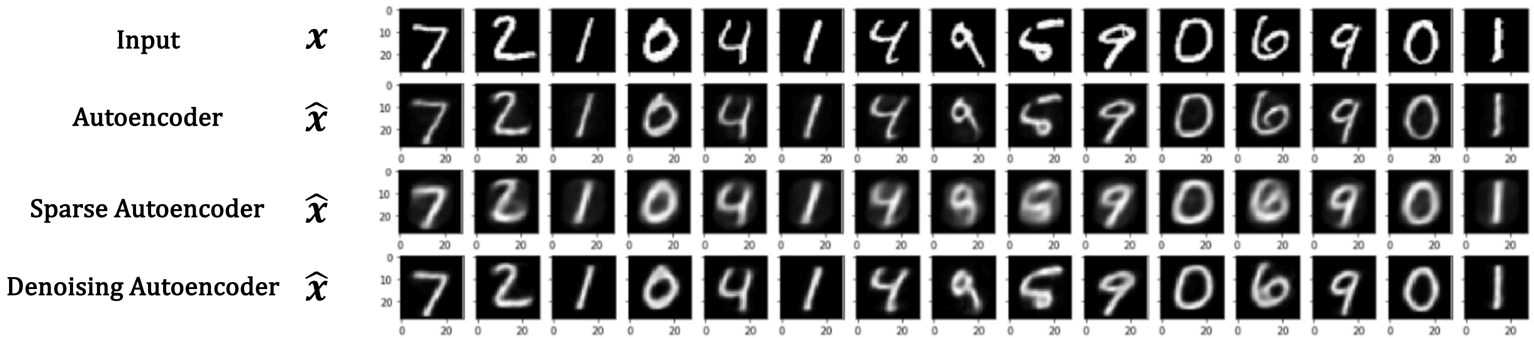
Denoising Autoencoder



Unsupervised Learning

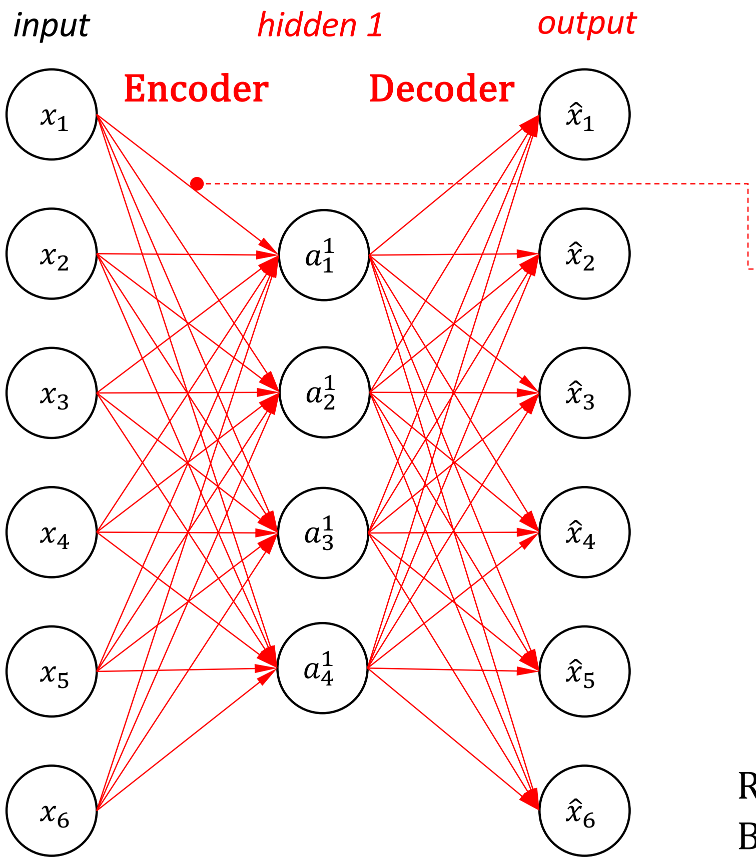
- Denoising Autoencoder

Autoencoders for MNIST dataset



Unsupervised Learning

- Stacked Autoencoder



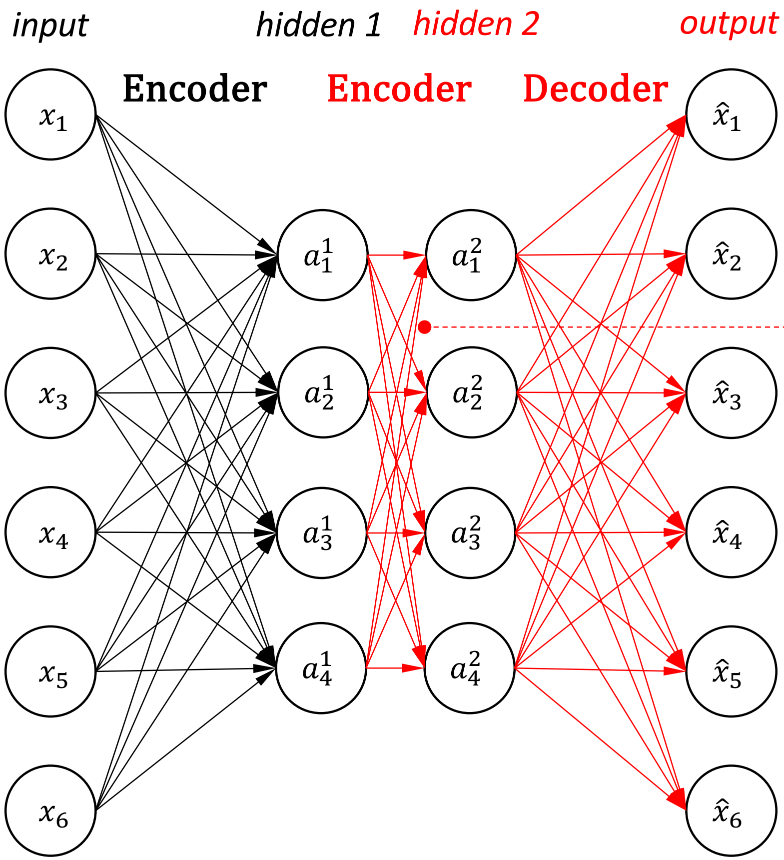
Unsupervised

- The feature extractor for the input data

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

Unsupervised Learning

- Stacked Autoencoder



Unsupervised

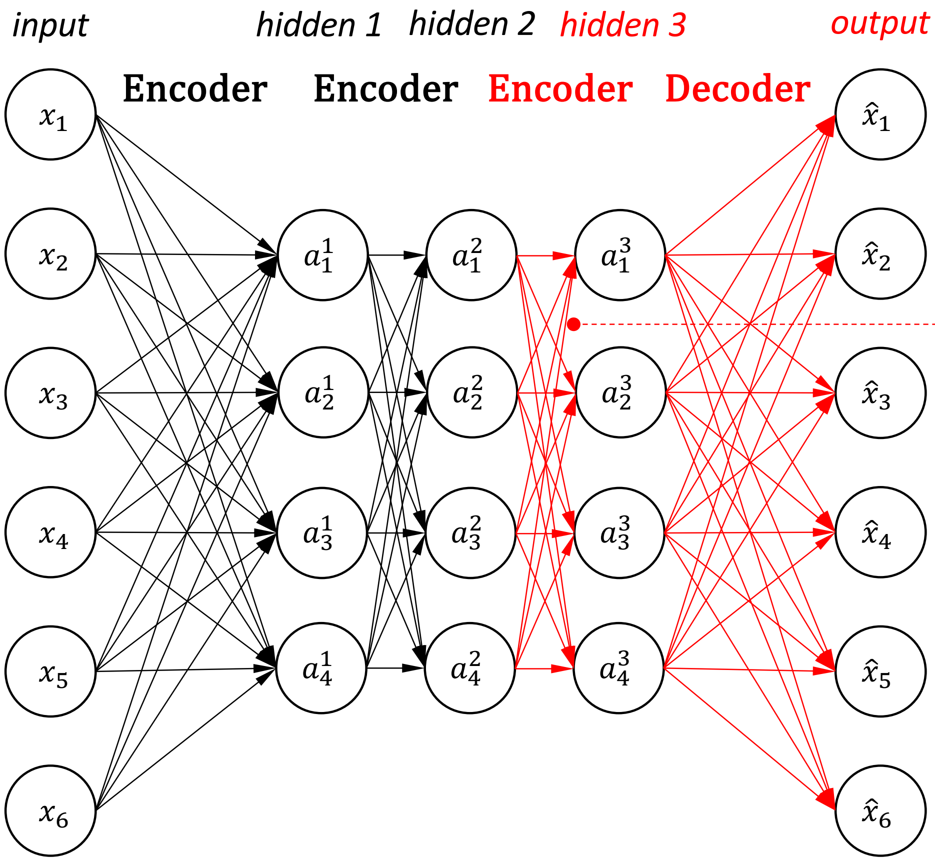
- The feature extractor for the first feature extractor

Red lines indicate the trainable weights

Black lines indicate the fixed/nontrainable weights

Unsupervised Learning

- Stacked Autoencoder



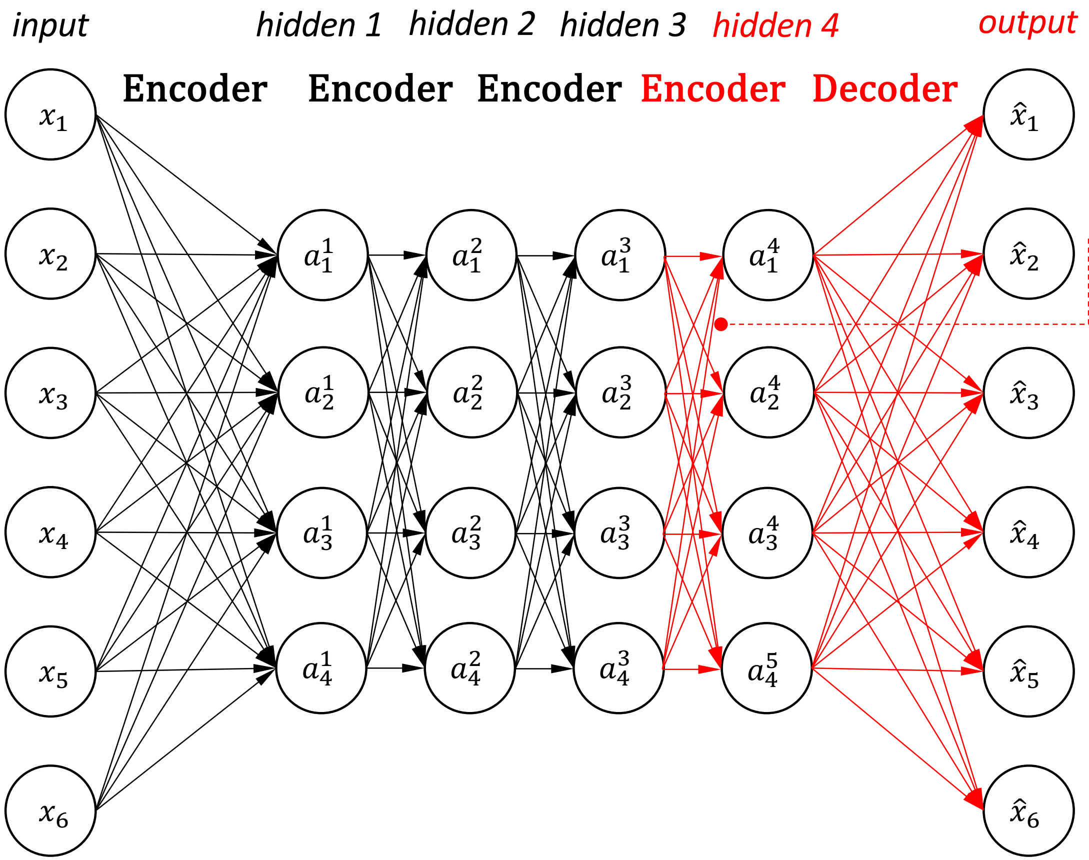
Unsupervised

- The feature extractor for the second feature extractor

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

Unsupervised Learning

- Stacked Autoencoder



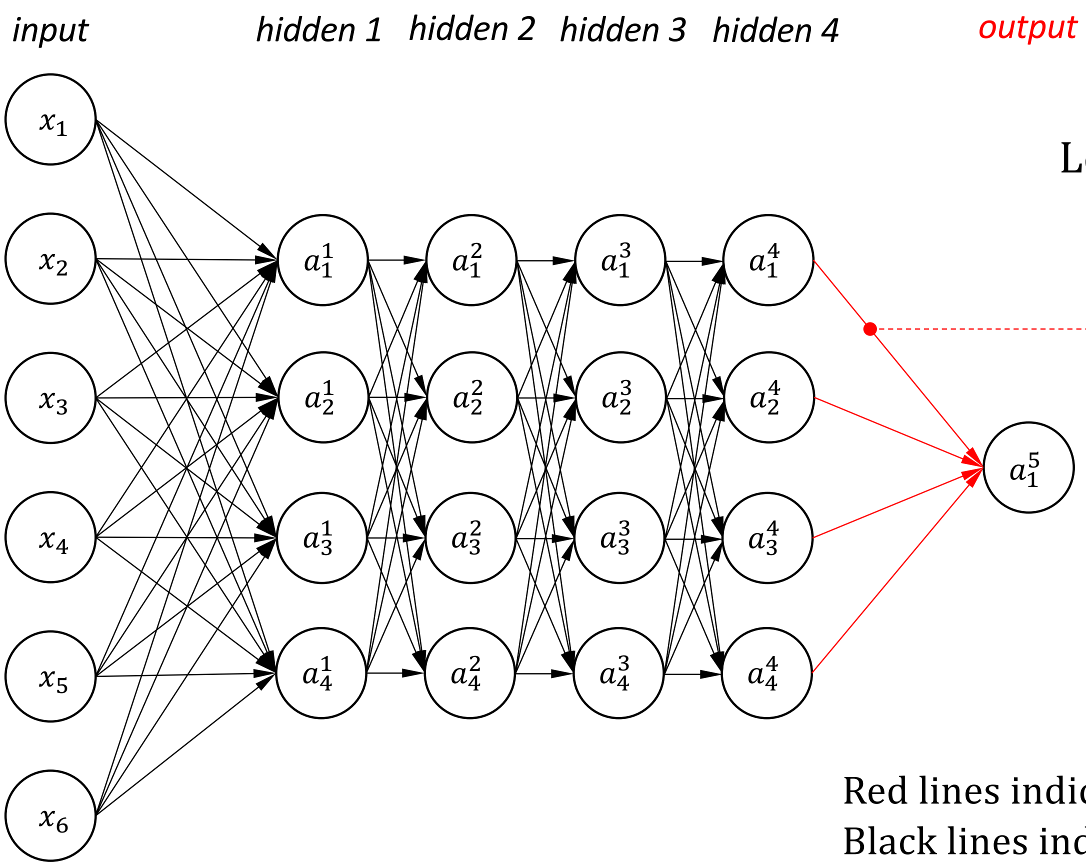
Unsupervised

- The feature extractor for the third feature extracto

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

Unsupervised Learning

- Stacked Autoencoder



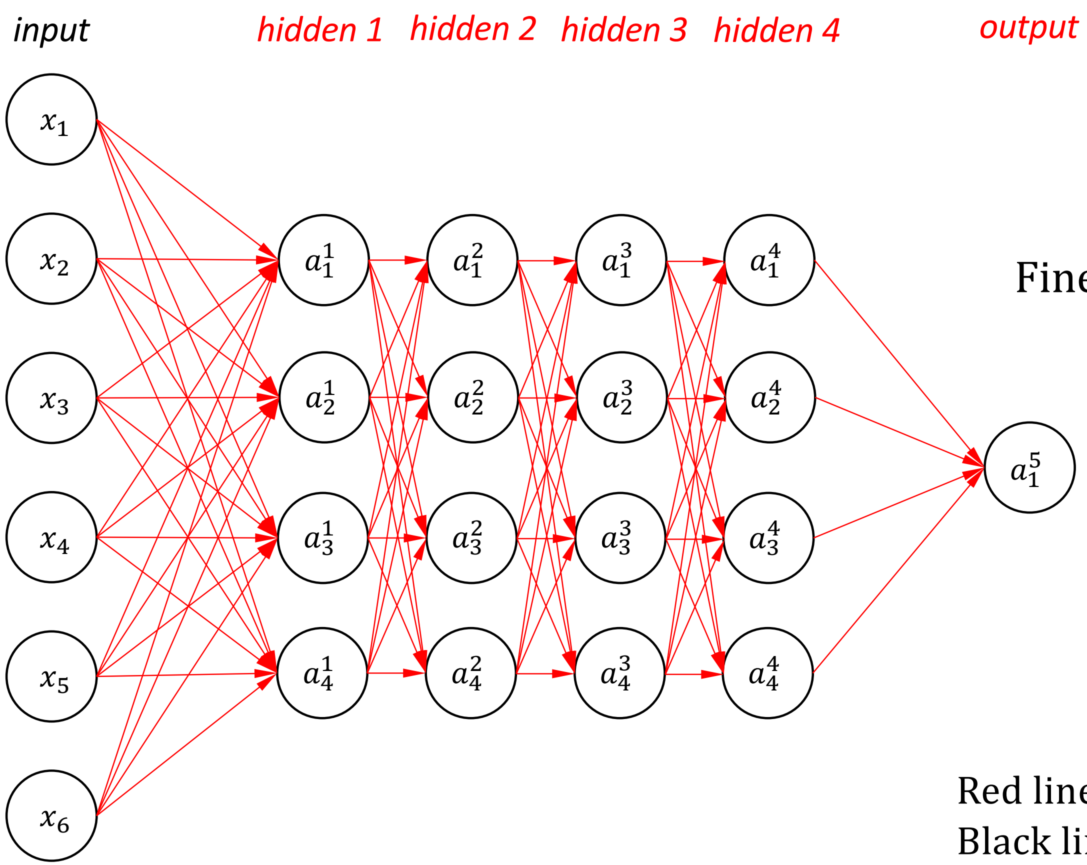
Supervised

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

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

Unsupervised Learning

- Stacked Autoencoder



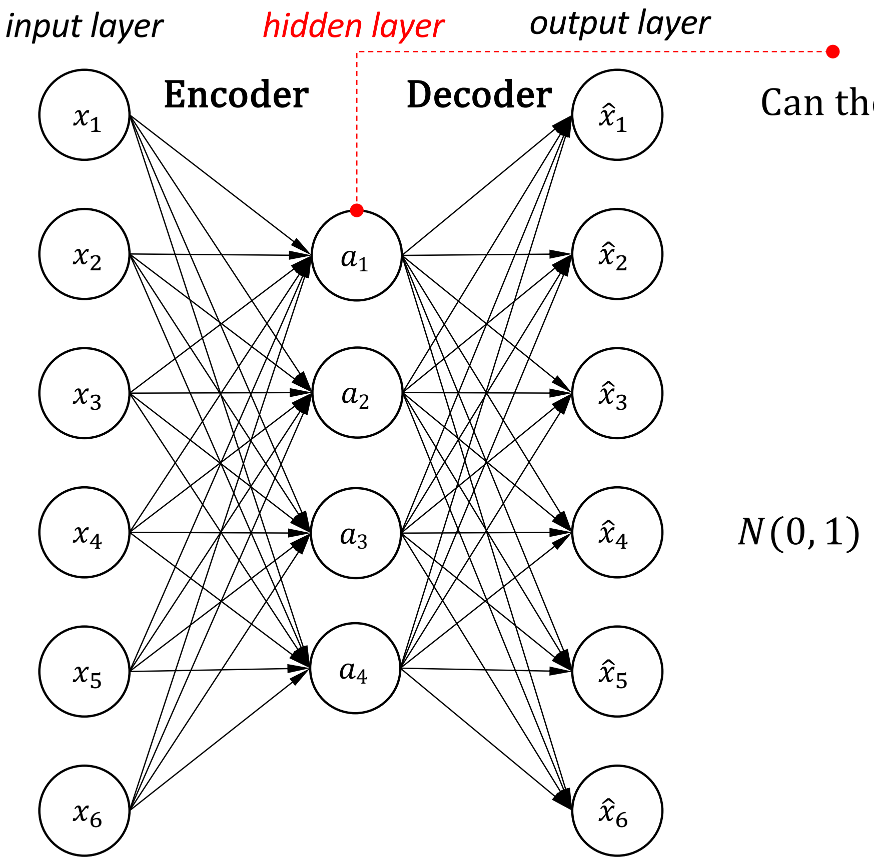
Supervised

Fine-tune the entire model for classification

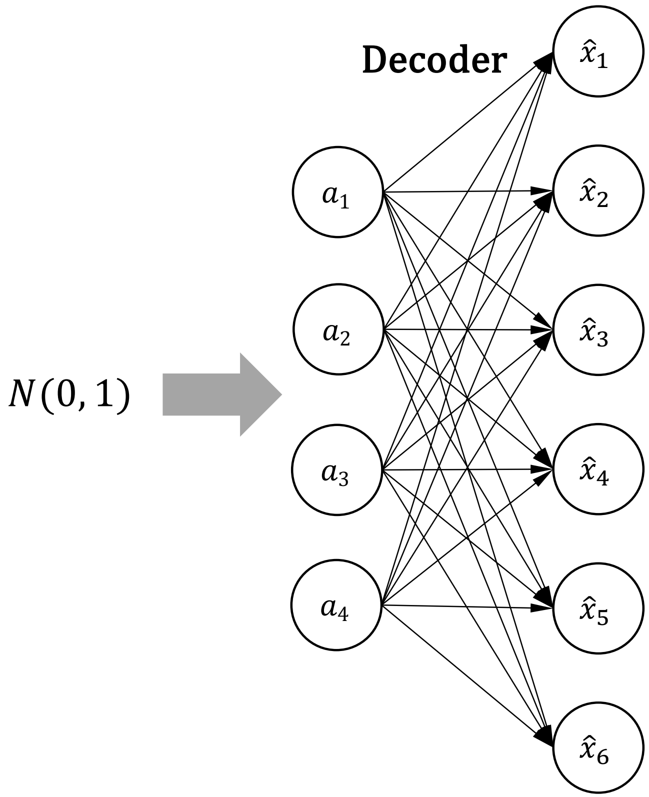
Red lines indicate the trainable weights
Black lines indicate the fixed/nontrainable weights

Unsupervised Learning

- Variational Autoencoder, VAE



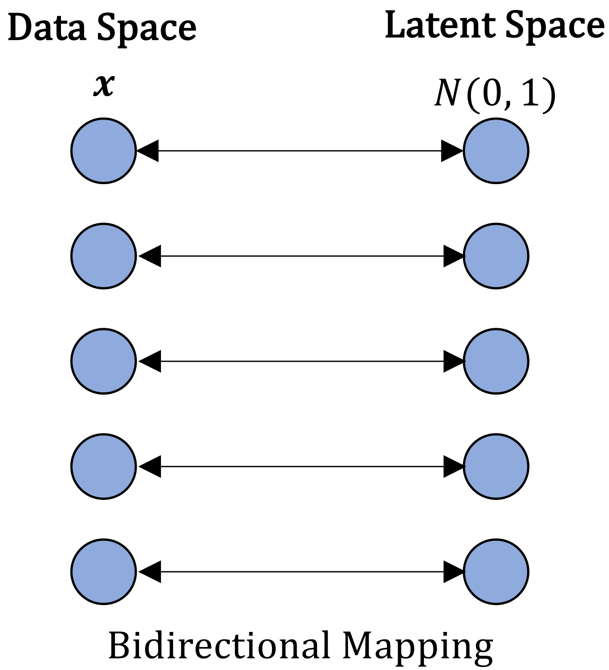
Can the hidden output be a prior distribution, e.g., Normal distribution?



If yes, we can have a Generator that can map $N(0,1)$ to data space

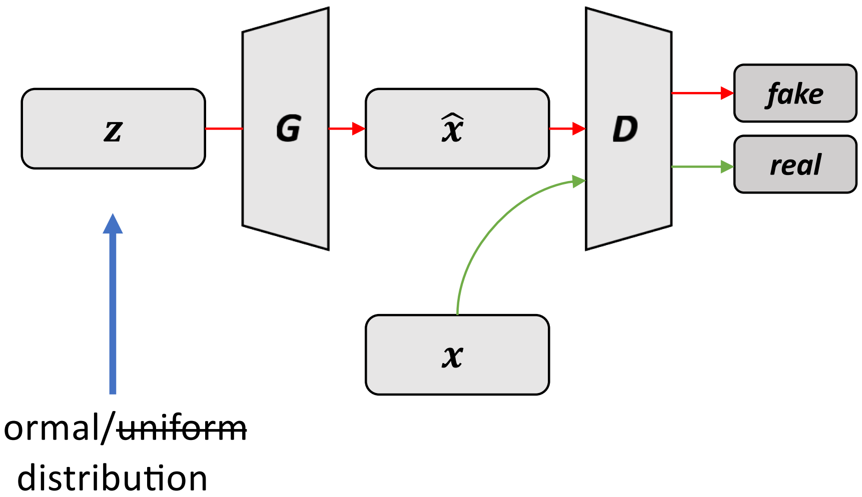
Unsupervised Learning

- Variational Autoencoder, VAE



Unsupervised Learning

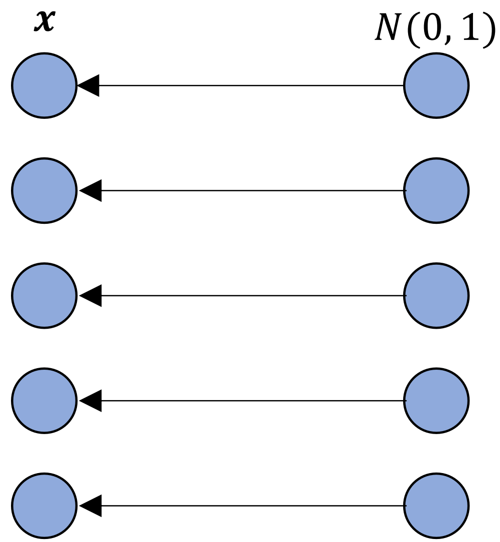
- Generative Adversarial Network, GAN



$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - D(G(\boldsymbol{z}))]$$

$$\mathcal{L}_D = -\boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D(\boldsymbol{x})]} - \boxed{\mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - D(G(\boldsymbol{z}))]}$$

$$\mathcal{L}_G = -\boxed{\mathbb{E}_{\boldsymbol{z}\sim p_z}[\log D(G(\boldsymbol{z}))]}$$



Unidirectional Mapping

Semi-supervised Learning

Semi-supervised Learning

- **Motivation:**
  - Unlabeled data is easy to be obtained
  - Labeled data can be hard to get

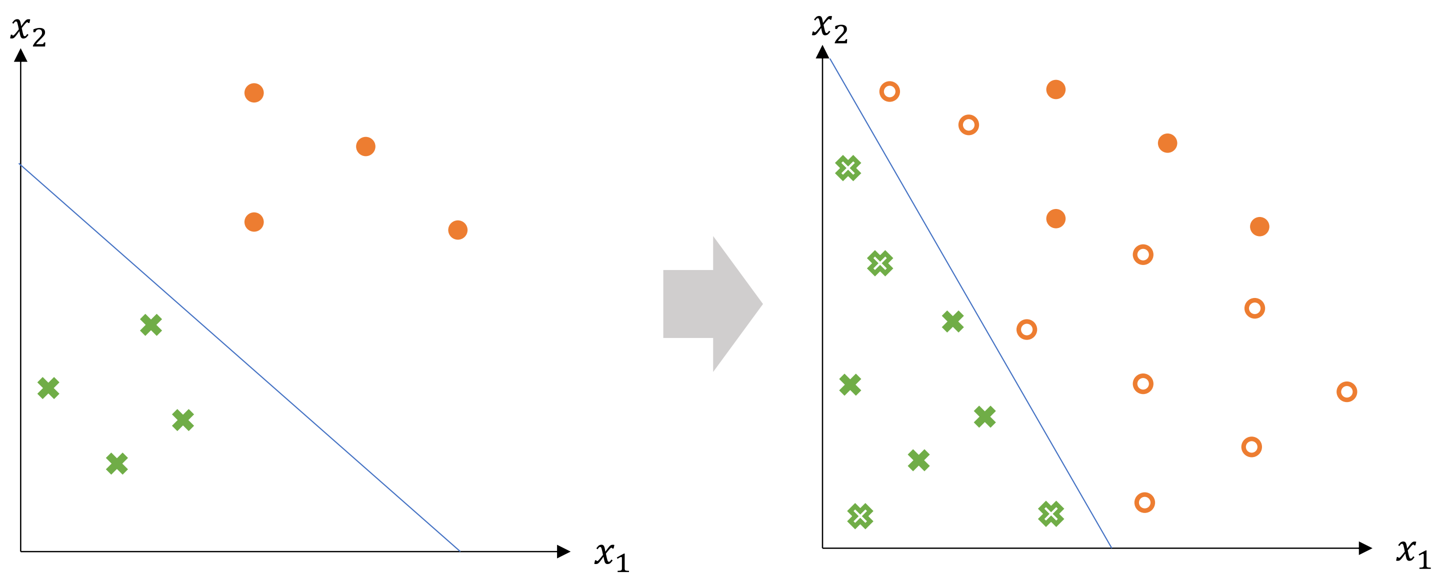
- **Goal:**
  - Semi-supervised learning mixes labeled and labeled data to produce better models.

- **vs. Transductive Learning:**
  - Semi-supervised learning is eventually applied to the testing data
  - Transductive learning is only related to the unlabelled data

Semi-supervised Learning

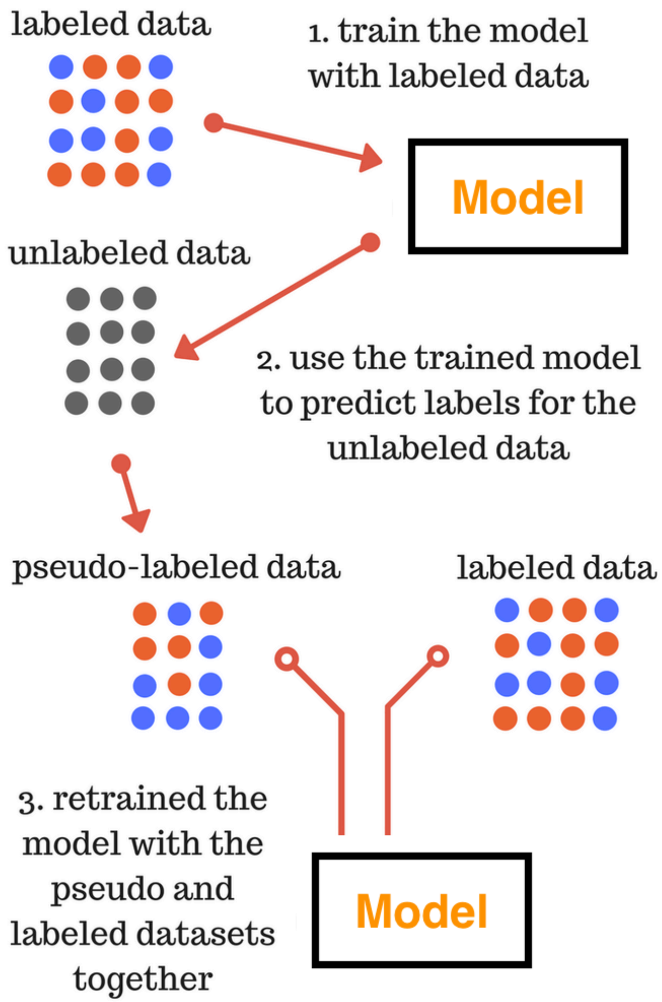
- Unlabeled data can help

Unlabeled data can help to find a better boundary



Semi-supervised Learning

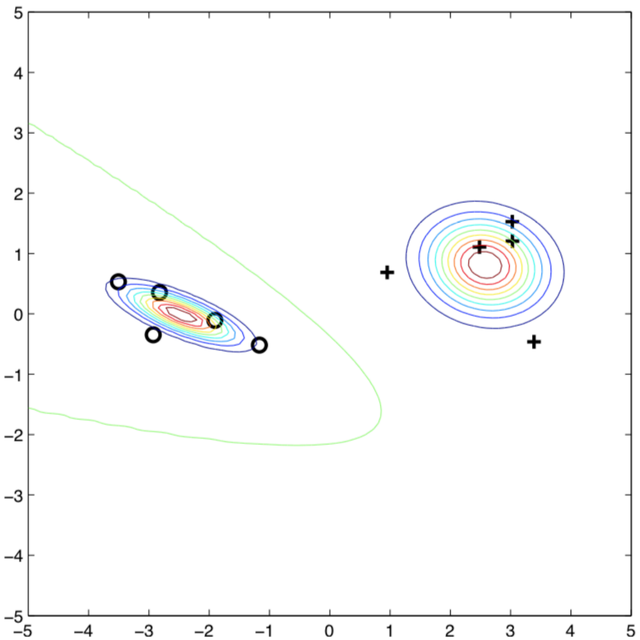
- Pseudo-Labelling

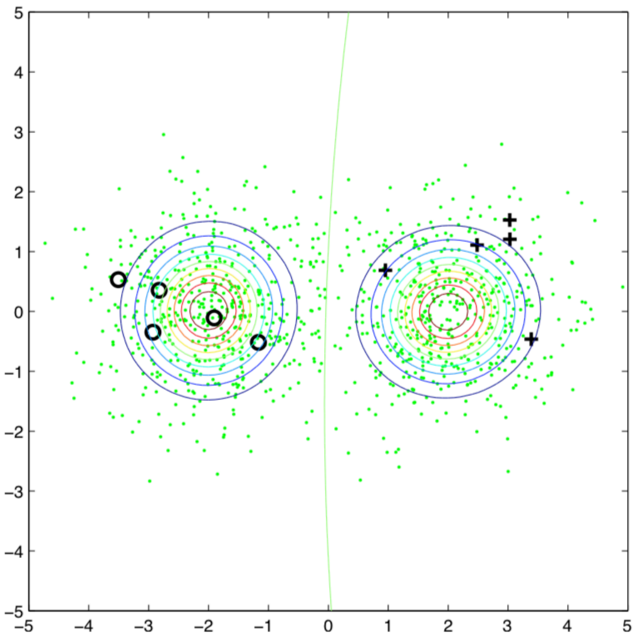


Semi-supervised Learning

- Generative Methods
  - EM with some labelled data

Cluster and then label



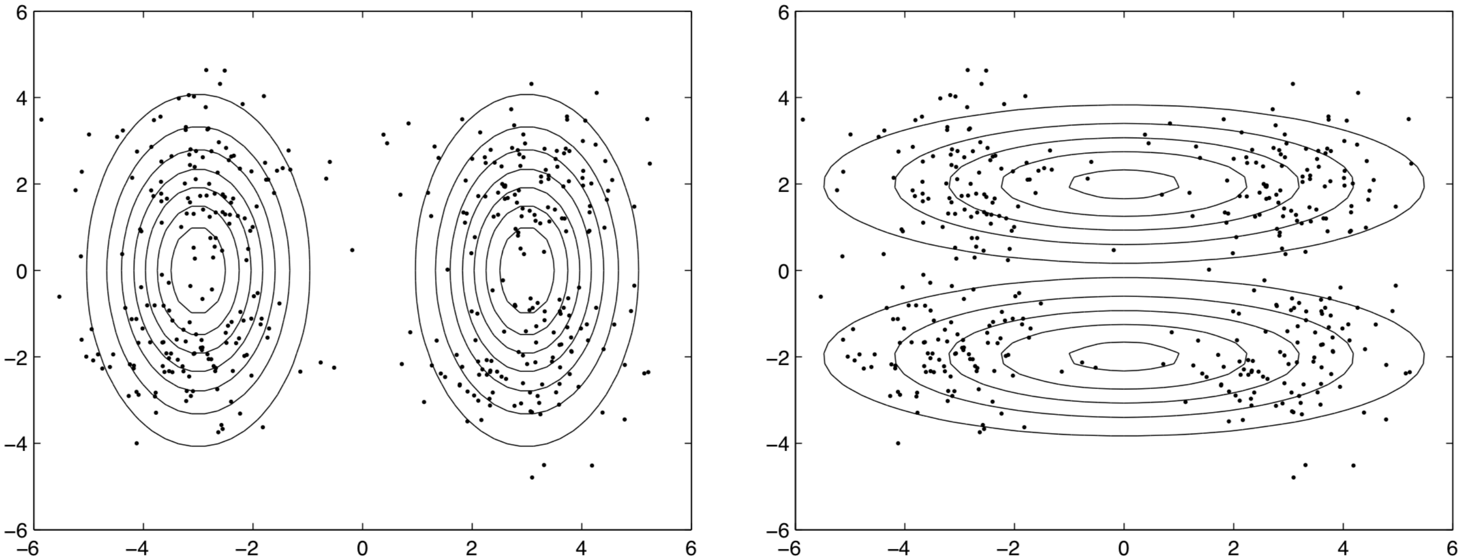


Semi-supervised Learning

- Generative Methods

Unlabeled data may hurt the learning

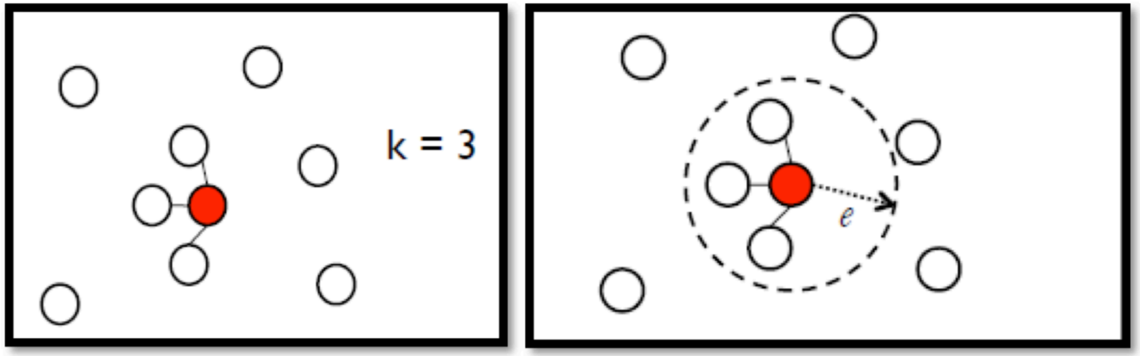
Wrong Correct

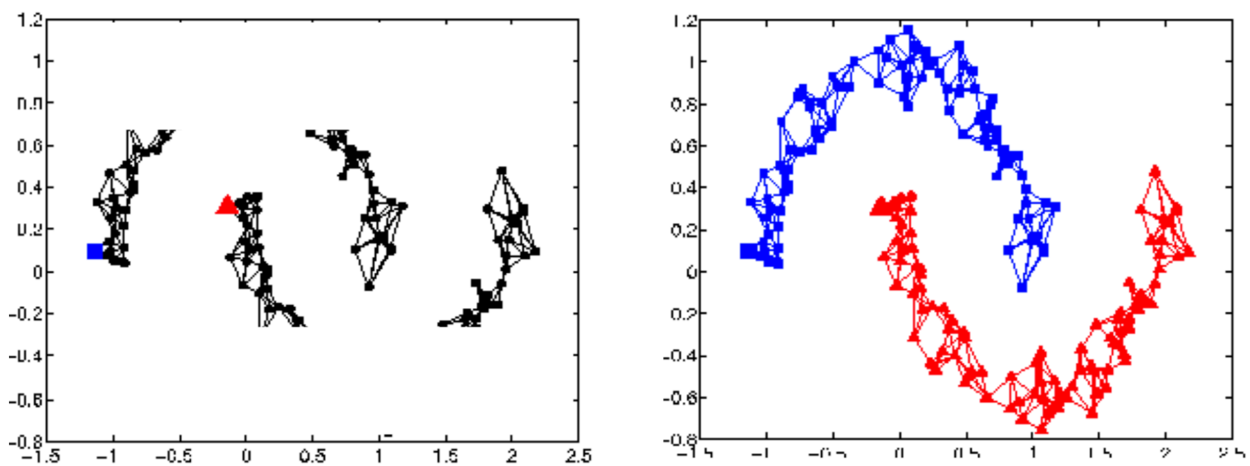


Multi-variables Gaussian model

Semi-supervised Learning

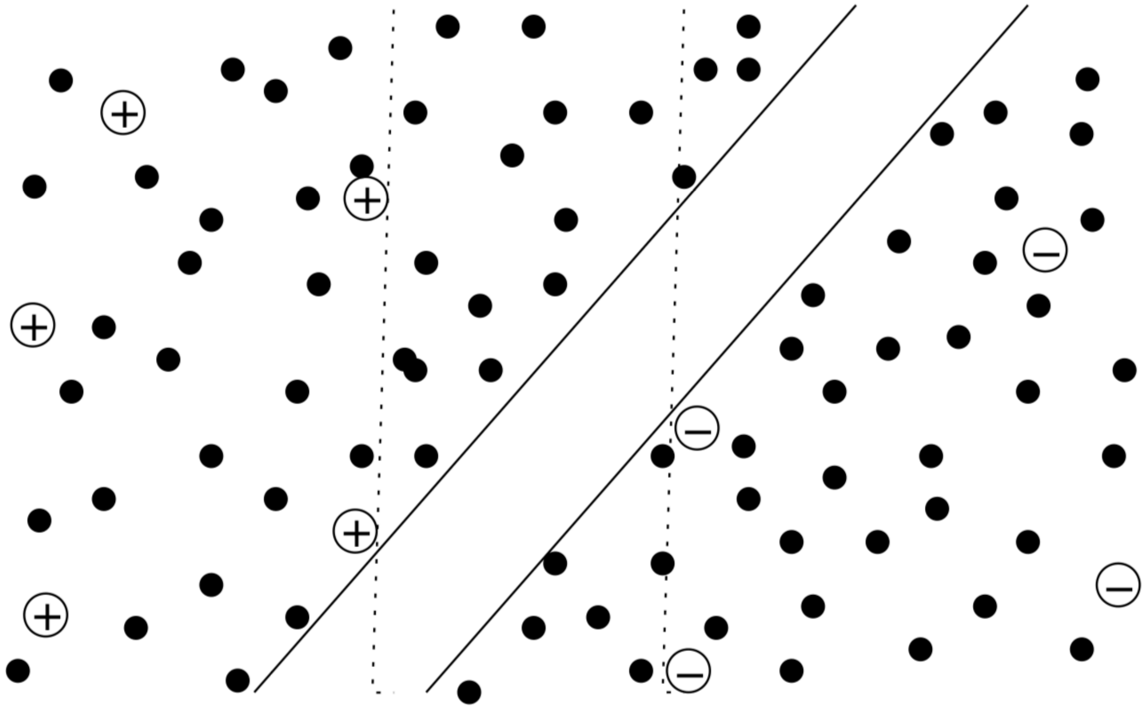
- Graph-based Methods
  1. Define the similarity $s(x_i, x_j)$
  2. Add edges
     1. KNN
     2. e-Neighborhood
  3. Edge weight is proportional to $s(x_i, x_j)$
  4. Propagate through the graph





Semi-supervised Learning

- Low-density separation
  - Semi-supervised SVM (S3VM) == Transductive SVM (TSVM)



Weakly-supervised Learning

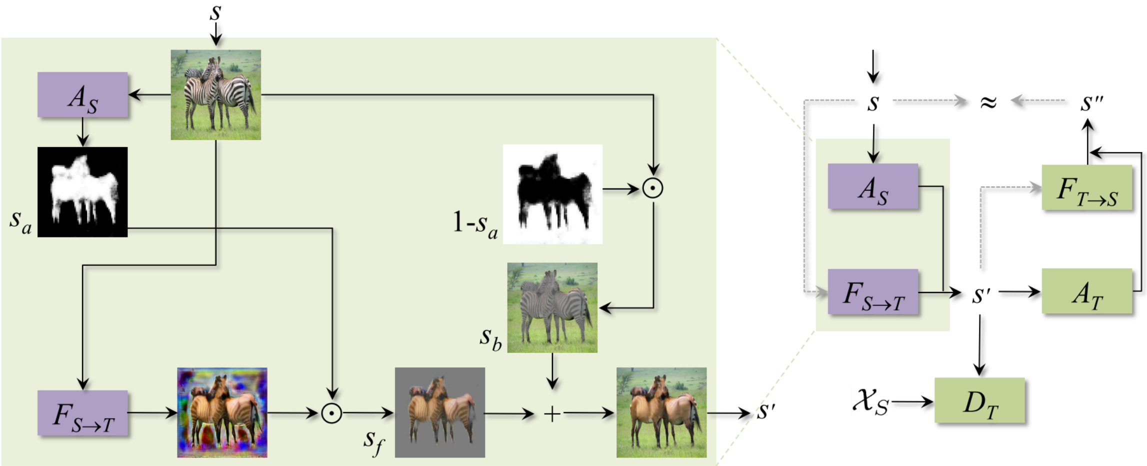
Weakly-supervised Learning

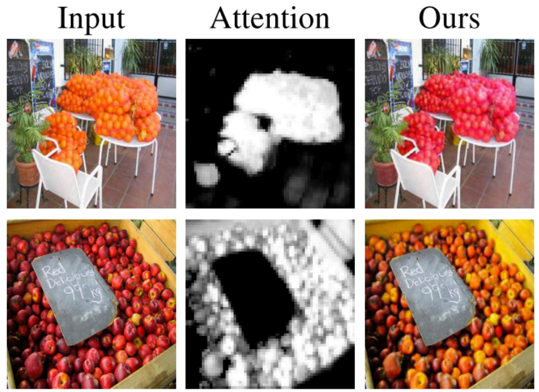
PEKING UNIVERSITY

- Weakly supervised learning is a machine learning framework where the model is trained using examples that are only partially annotated or labeled.

Weakly-supervised Learning

- Attention CycleGAN: Learn the segmentation via synthesis

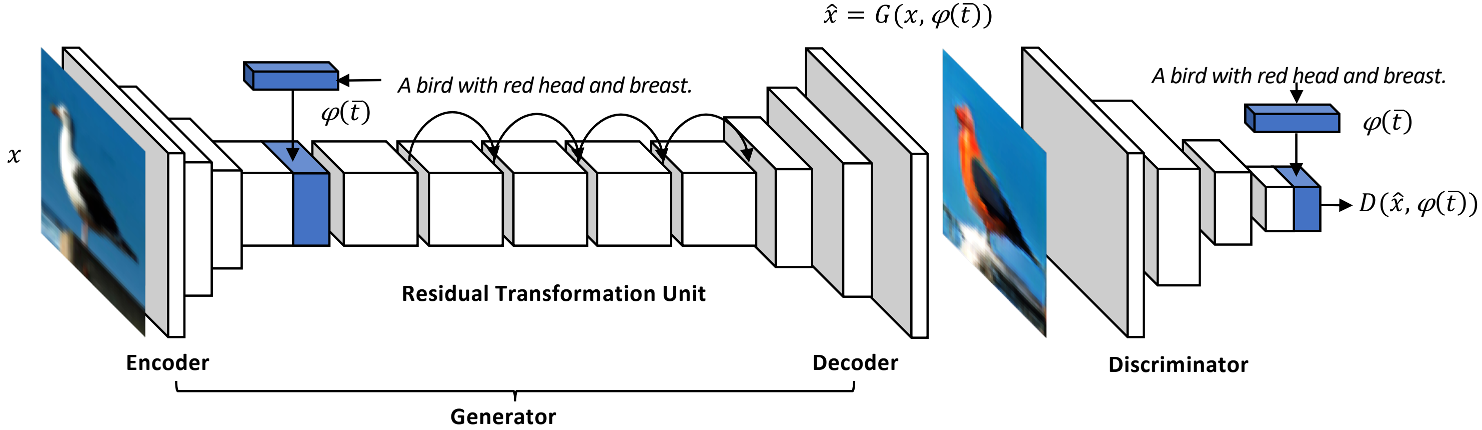


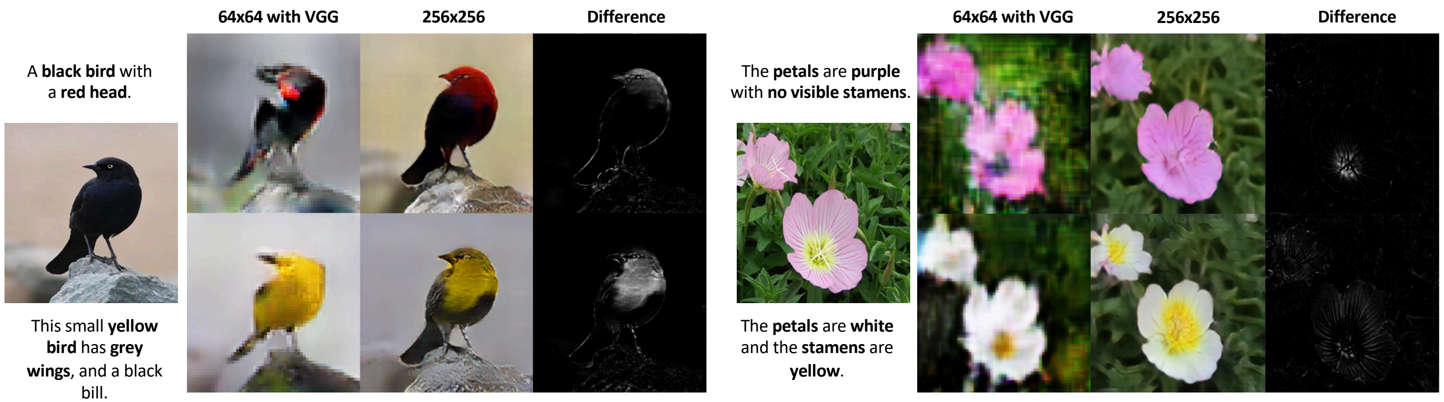


Unsupervised Attention-guided Image-to-Image Translation. *Mejjati, Y. A., Richardt, C., & Cosker, D.* NIPS, 2018

Weakly-supervised Learning

- Semantic Image Synthesis: Learn the segmentation via synthesis





Semantic Image Synthesis via Adversarial Learning. *Dong, H., Yu, S., Wu, C., Guo, Y.* 2017. ICCV

Weakly-supervised Learning

- More and more …

Summary

Learning Methods

- Supervised, Semi-supervised, Weakly-supervised, Unsupervised Learnings
- Unsupervised Learning
- Semi-supervised Learning
- Weakly-supervised Learning

Learning Methods

- Exercise 1:
  - Implement Sparse Autoencoder on MNIST and visualize the learned features.

- Exercise 2:
  - Explain Variational Autoencoder in mathematical way
  - Implement it on MNIST (Optional)

- Exercise 3: (Optional)
  - Choice an application and implement it

Link: https://github.com/zsdonghao/deep-learning-note/

Questions?