Chapter Six

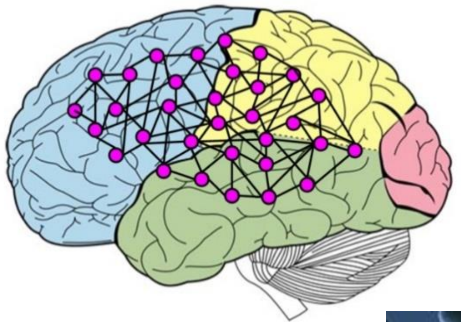
Unsupervised Learning

Zhang Yi, *IEEE Fellow*
Autumn 2020

Outline

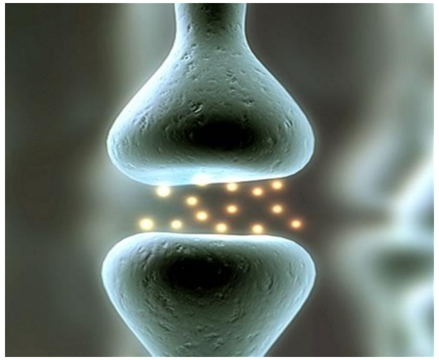
- Learning in Neural Networks
- Supervised Learning
- Unsupervised Learning
- Autoencoder Neural Networks
- Generative Adversarial Networks
- Assignment

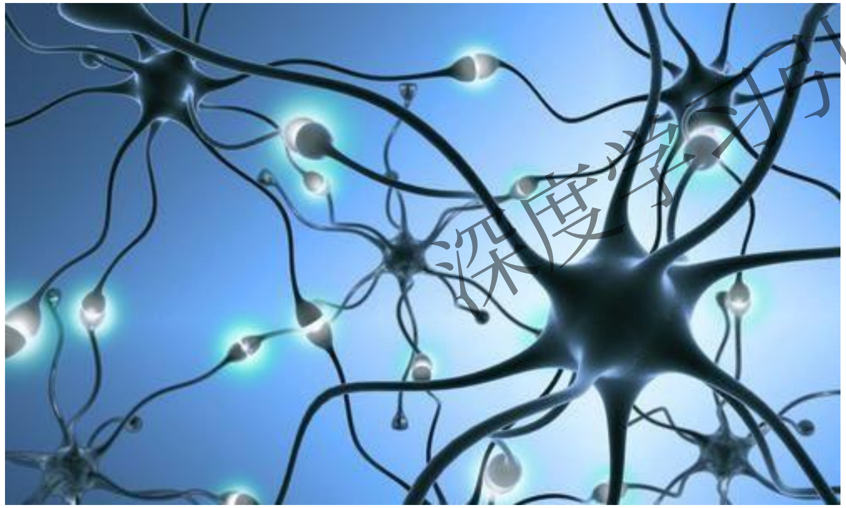
Learning in Neural Networks



Neural Network

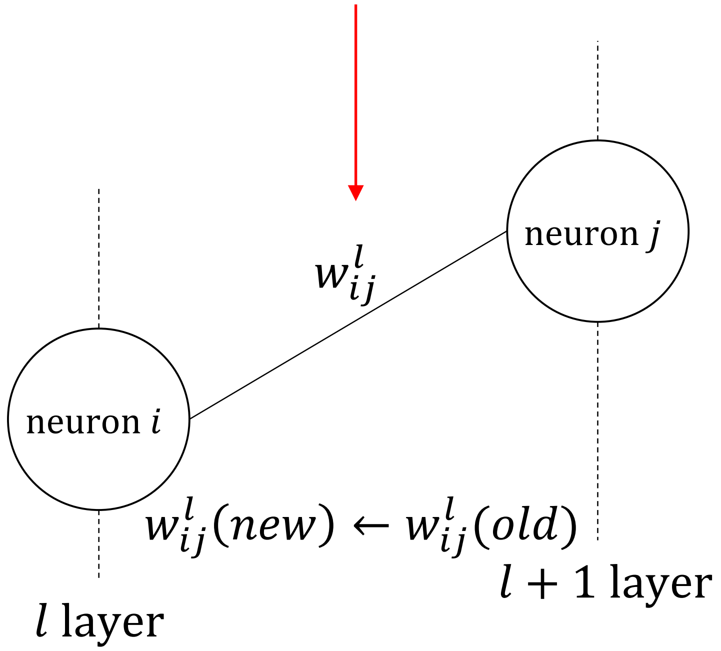
The brain is a learning system. The brain can learn by some supervisor or by itself. Thus, there are Supervised Learning and Unsupervised Learning.





Learning is the changing of connection strength between two connected neurons

The knowledge is stored in connections weight.

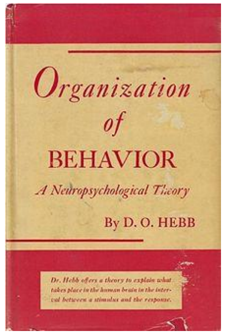


Learning in Neural Networks

Hebb's Postulate

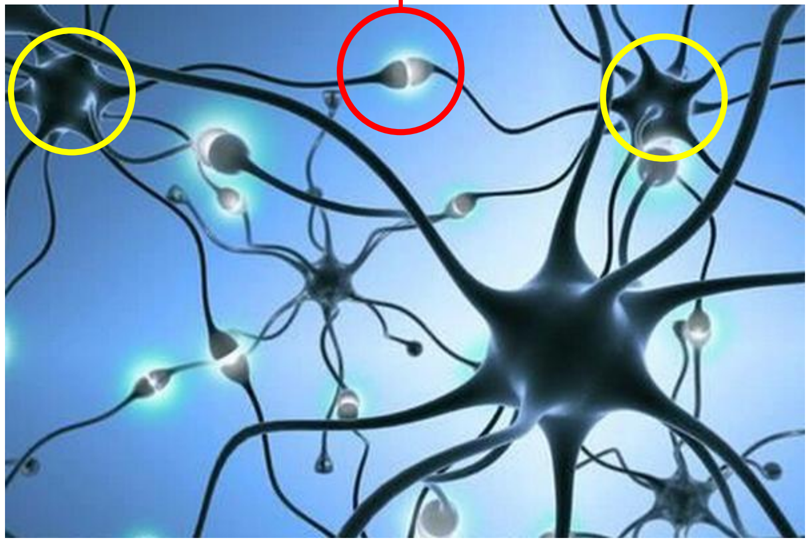
When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.

——D.O Hebb，1949



D. O. Hebb
Father of Cognitive Psychobiology
1904-1985

Synapse



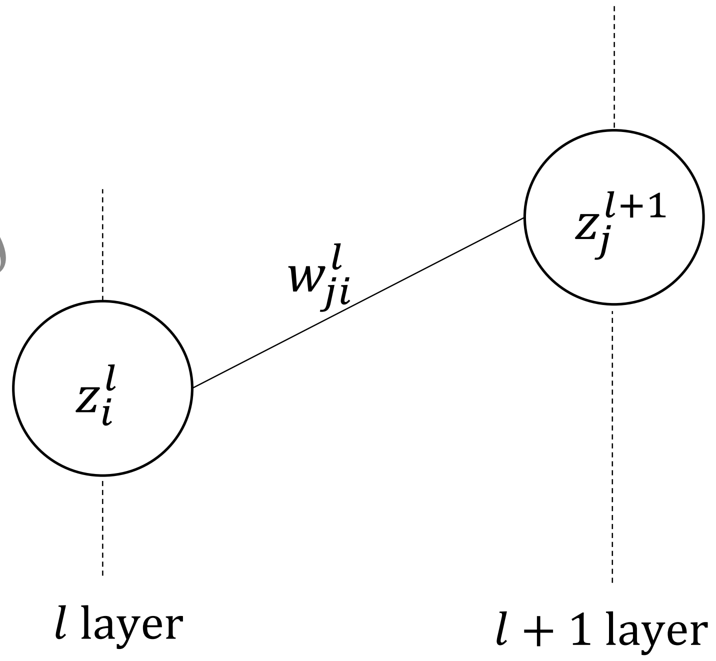
Learning in Neural Networks

When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.

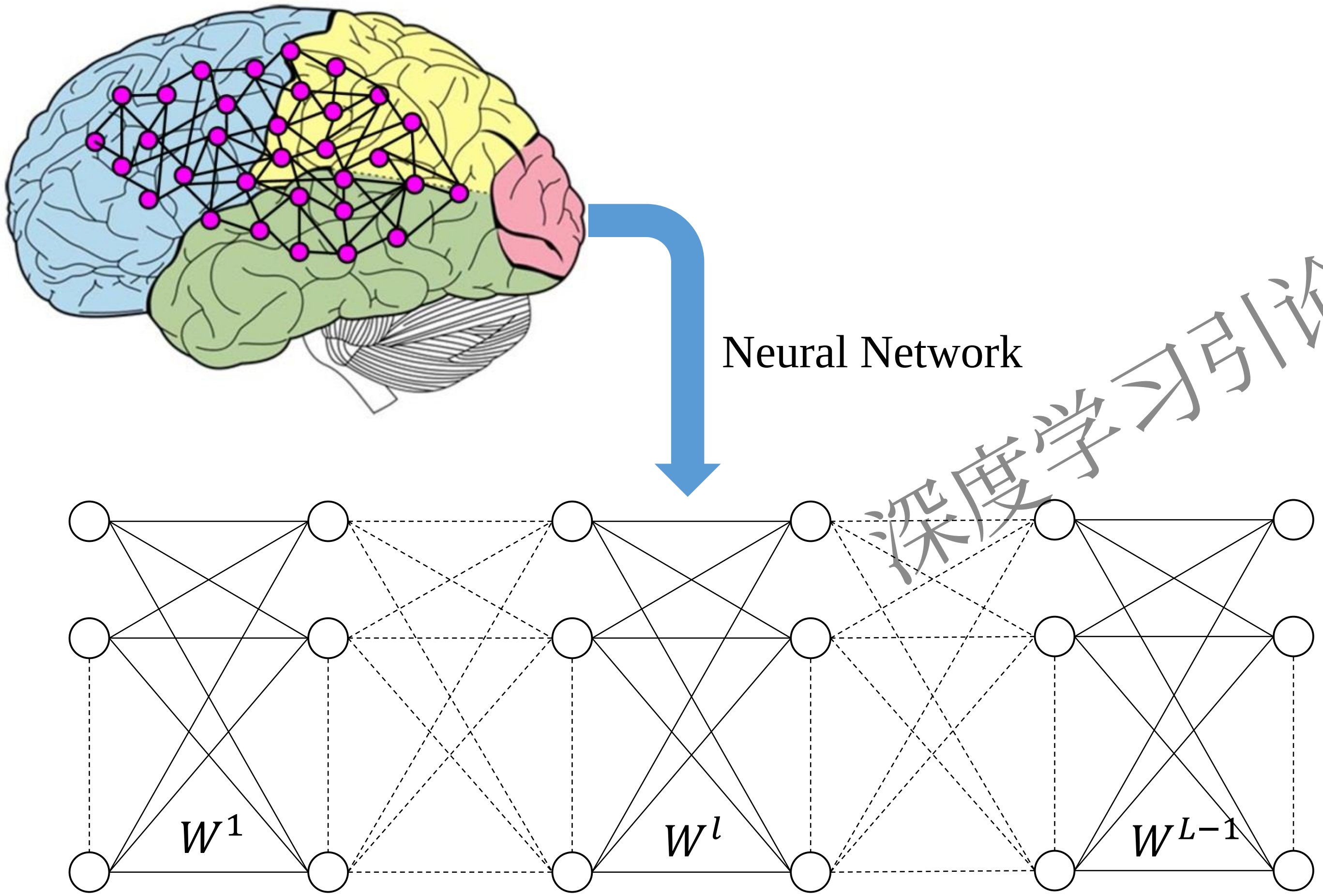
If two neurons of either side of a synapse are activated simultaneously, the strength of the synapse will increase.

Hebbian Learning

$$w_{ji}^{l} \leftarrow w_{ji}^{l} + F\left(z_j^{l+1}, z_i^{l}\right)$$



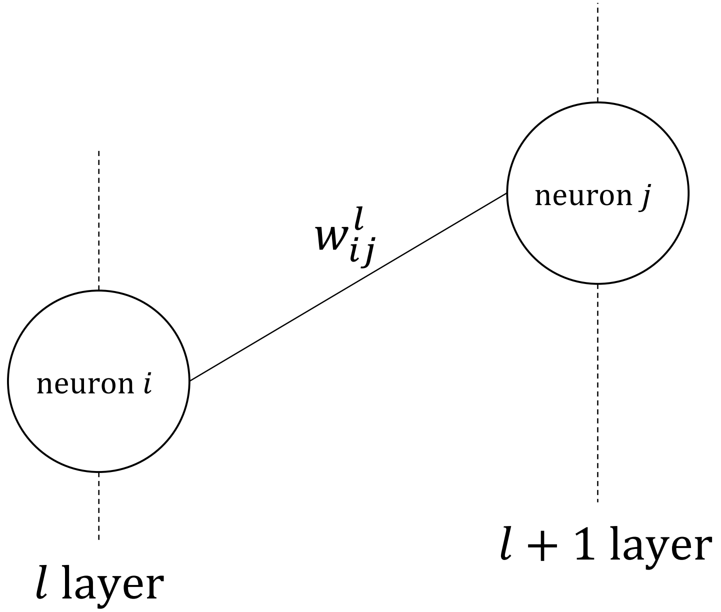
Learning in Neural Networks



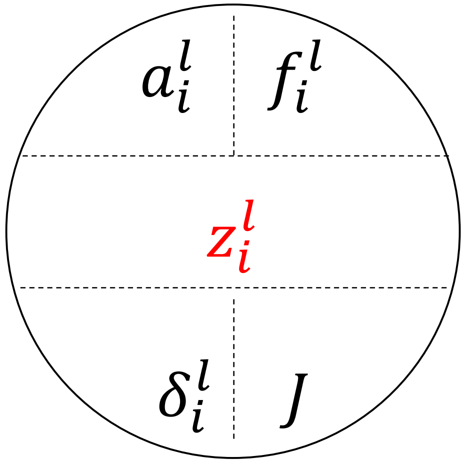
The network stores only knowledge, it does not store original data.

Learning is the updating of connection weight between two connected neurons

$$w_{ij}^l(new) \leftarrow w_{ij}^l(old)$$



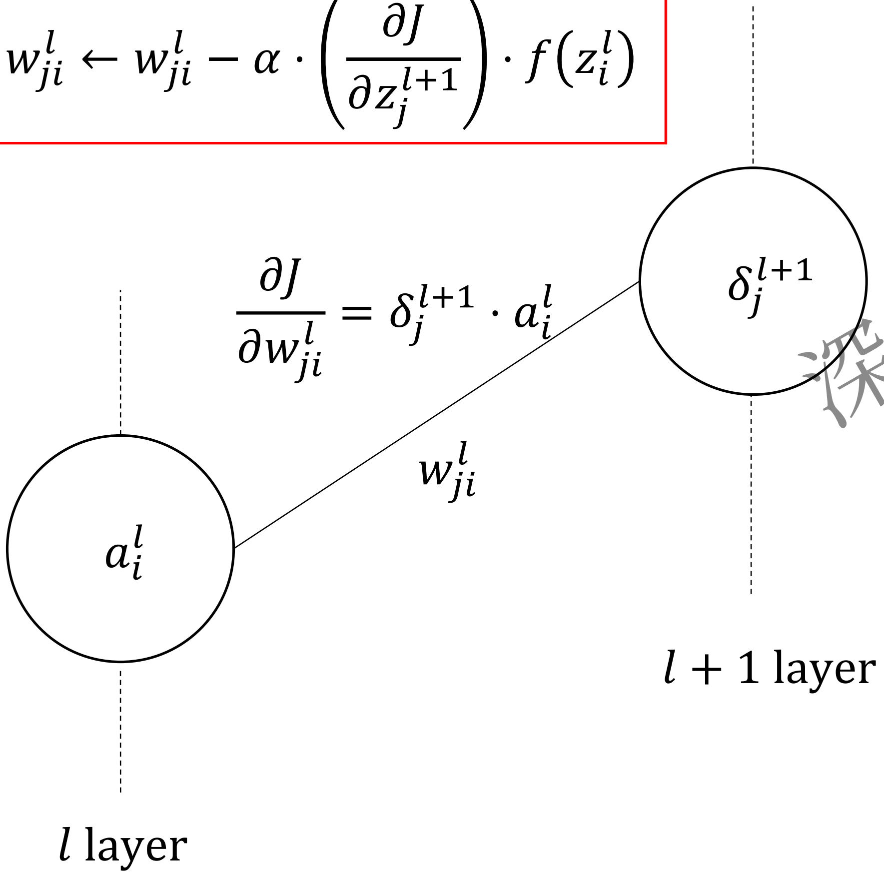
Learning in Neural Networks

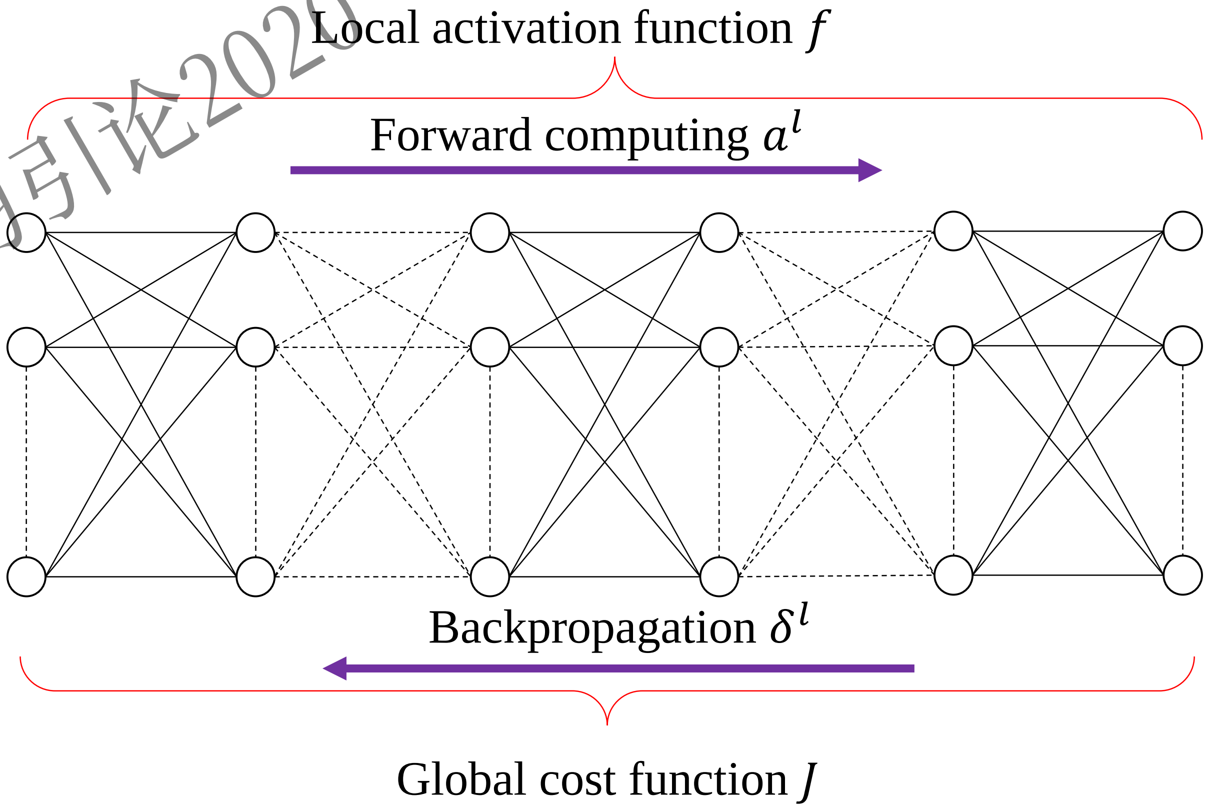


BP Learning Algorithm

$$w_{ij}^l \leftarrow w_{ij}^l - \alpha \cdot (\delta_j^{l+1} \cdot a_i^l)$$

$$w_{ji}^l \leftarrow w_{ji}^l - \alpha \cdot \left(\frac{\partial J}{\partial z_j^{l+1}}\right) \cdot f(z_i^l)$$





On the Network Learning Rule

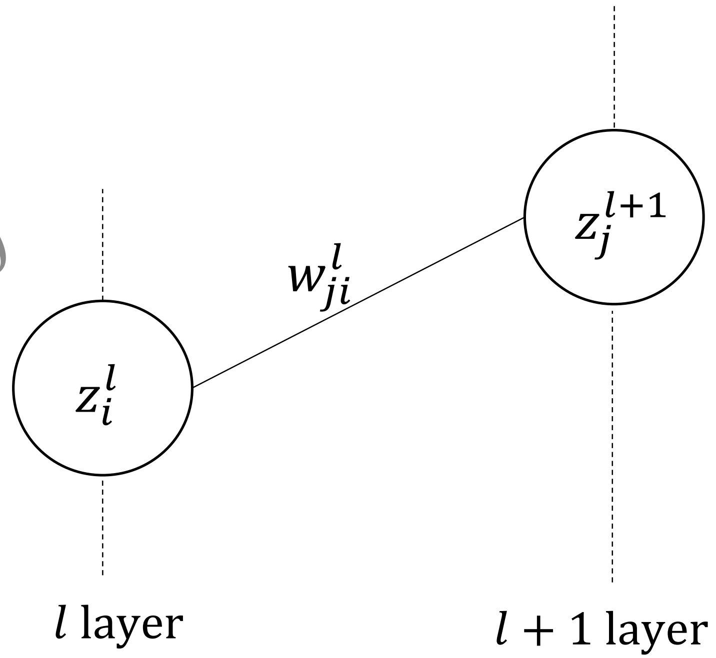
If two neurons of either side of a synapse are activated simultaneously, the strength of the synapse will increase.

Hebbian Learning

$$w_{ji}^{l} \leftarrow w_{ji}^{l} + F\left(z_j^{l+1}, z_i^{l}\right)$$

BP Learning

$$w_{ji}^{l} \leftarrow w_{ji}^{l} - \alpha \cdot \left(\frac{\partial J}{\partial z_j^{l+1}}\right) \cdot f\left(z_i^{l}\right)$$



$$F\left(z_j^{l+1}, z_i^{l}\right) = -\alpha \cdot \left(\frac{\partial J}{\partial z_j^{l+1}}\right) \cdot f\left(z_i^{l}\right)$$

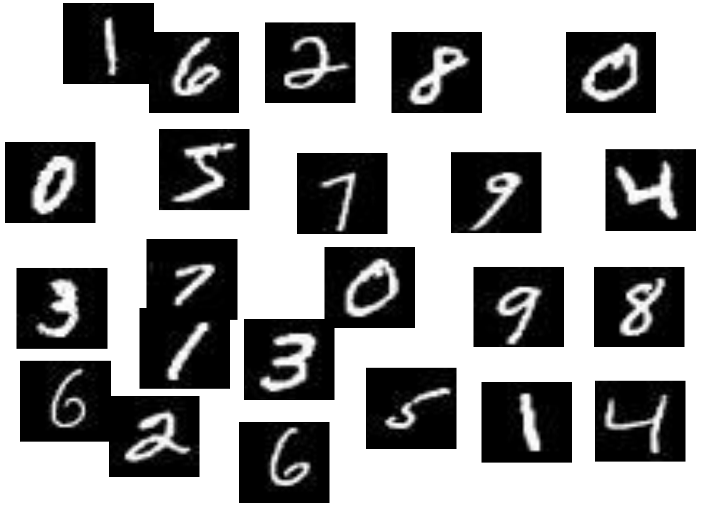
Outline

- Learning in Neural Networks
- Supervised Learning
- Unsupervised Learning
- Autoencoder Neural Networks
- Generative Adversarial Networks
- Assignment

Supervised Learning

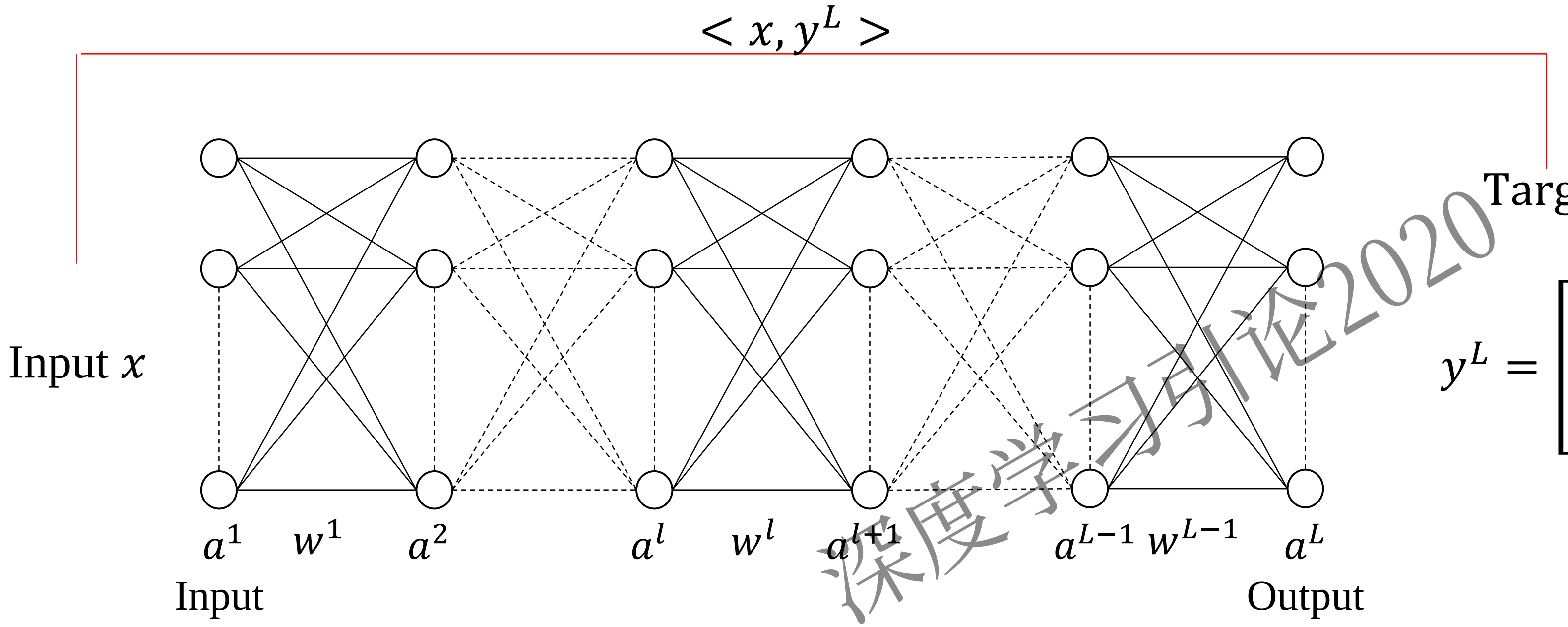
- Supervised Learning
  - Learning with a supervisor
  - The supervisor knows the correct answer
  - Each training sample must contain input and target

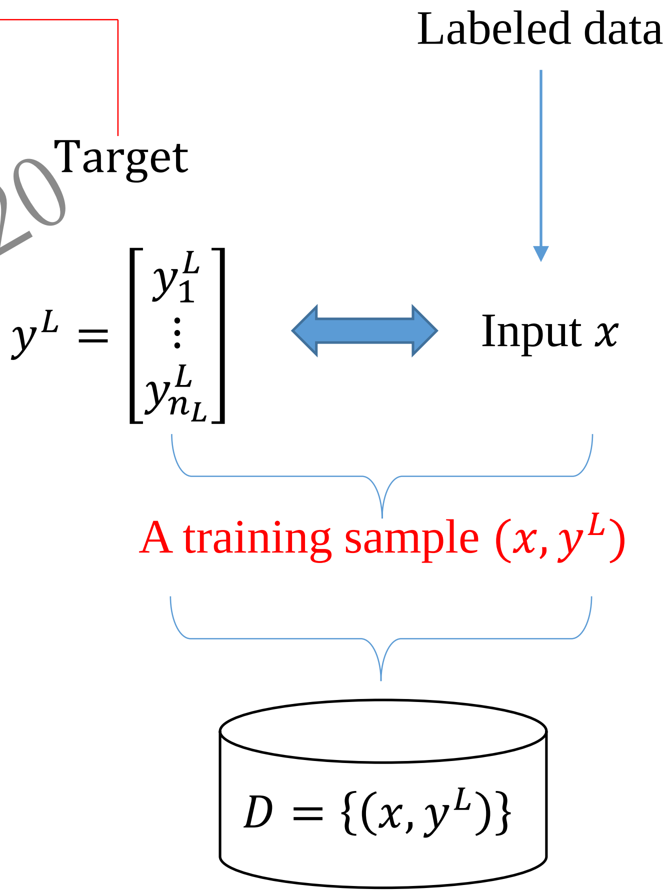
Learn from supervisor



Learn from parent

Supervised Learning





In supervised learning, each training sample contains input and the associated target output.

With only training data is not sufficient, you need a learning algorithm to update the knowledge, i.e., updating the connection weights in the network.

A training data set is a set composed by training samples

Supervised Learning

$$D = \{(x, y^L)\}$$

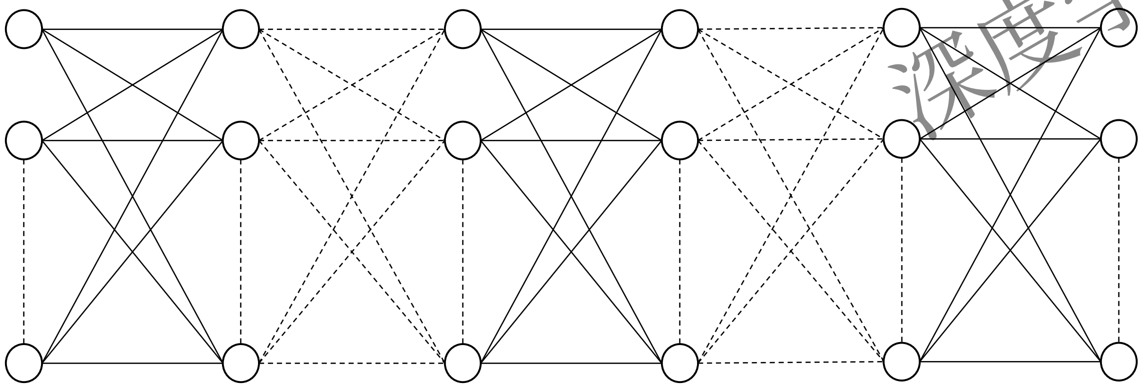
The well known BP algorithm is a supervised learning algorithm.

Network prediction

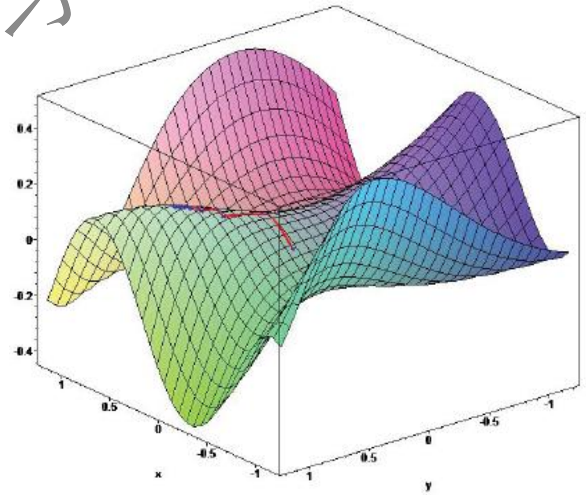
Target

$$a^L = \begin{bmatrix} a_1^L \\ \vdots \\ a_{n_L}^L \end{bmatrix} \qquad y^L = \begin{bmatrix} y_1^L \\ \vdots \\ y_{n_L}^L \end{bmatrix}$$

Forward computing a^l



Backpropagation δ^l



Cost function (Energy Function)

$$J = J(w^1, \cdots, w^{L-1})$$

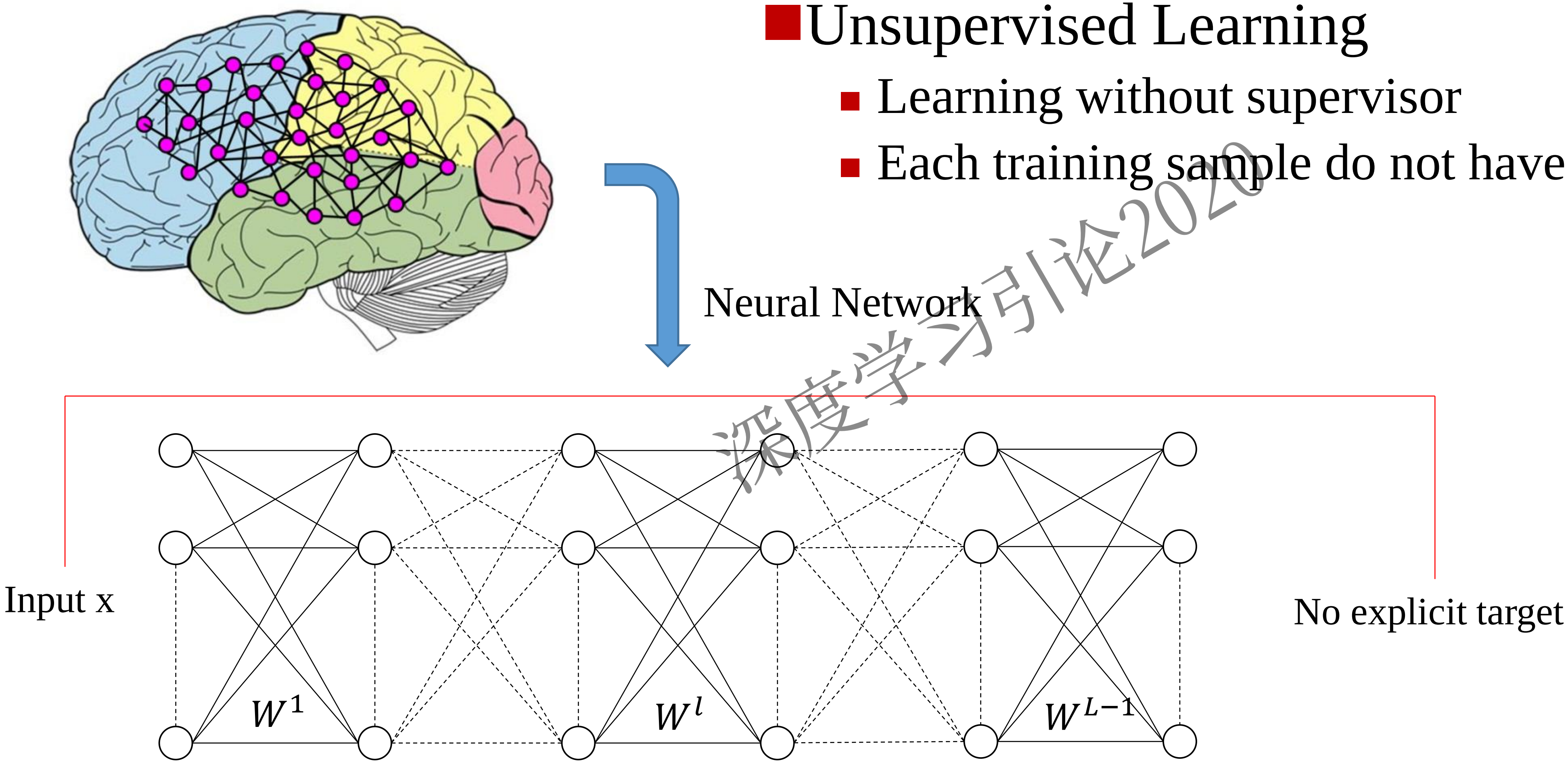
$$\frac{\partial J}{\partial w_{ji}^l} = \delta_j^{l+1} \cdot a_i^l$$

Outline

- Learning in Neural Networks
- Supervised Learning
- Unsupervised Learning
- Autoencoder Neural Networks
- Generative Adversarial Networks
- Assignment

Unsupervised Learning

- Unsupervised Learning
  - Learning without supervisor
  - Each training sample do not have any explicit target



Problem: How can I learn without a supervisor?

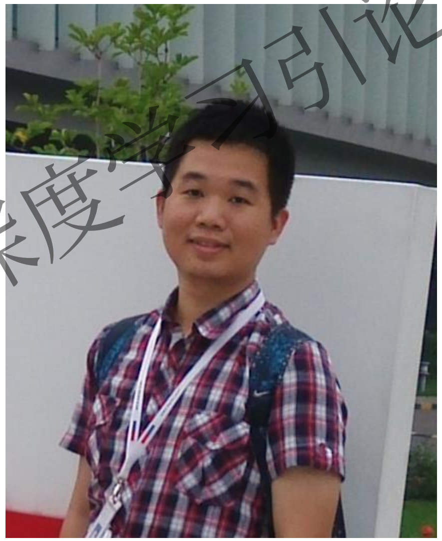
Unsupervised Learning: Feature Extraction

Can we learn something without a supervisor?

What’s this?!

No body can tell the student what this is.

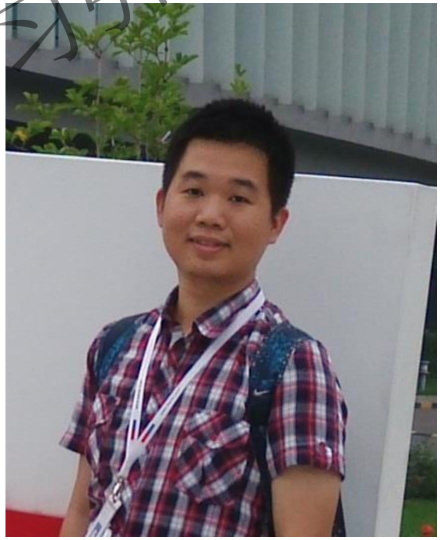
Unsupervised Learning: Feature Extraction

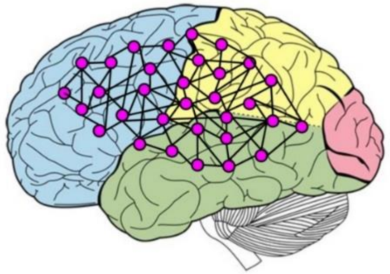


Haha, I saw it yesterday.

Unsupervised Learning: Feature Extraction

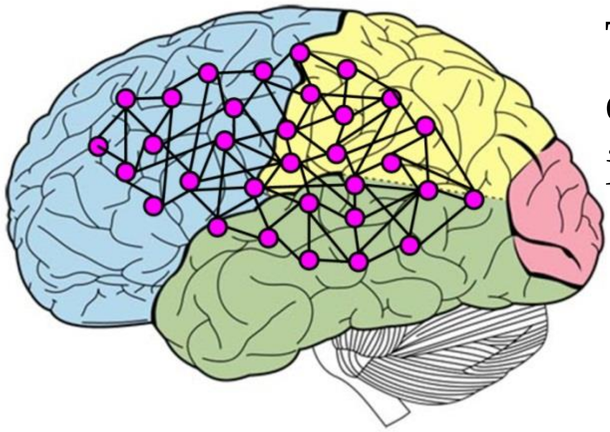
They are not the same one.
The student successfully
learnt the features of this
unknown fruit.



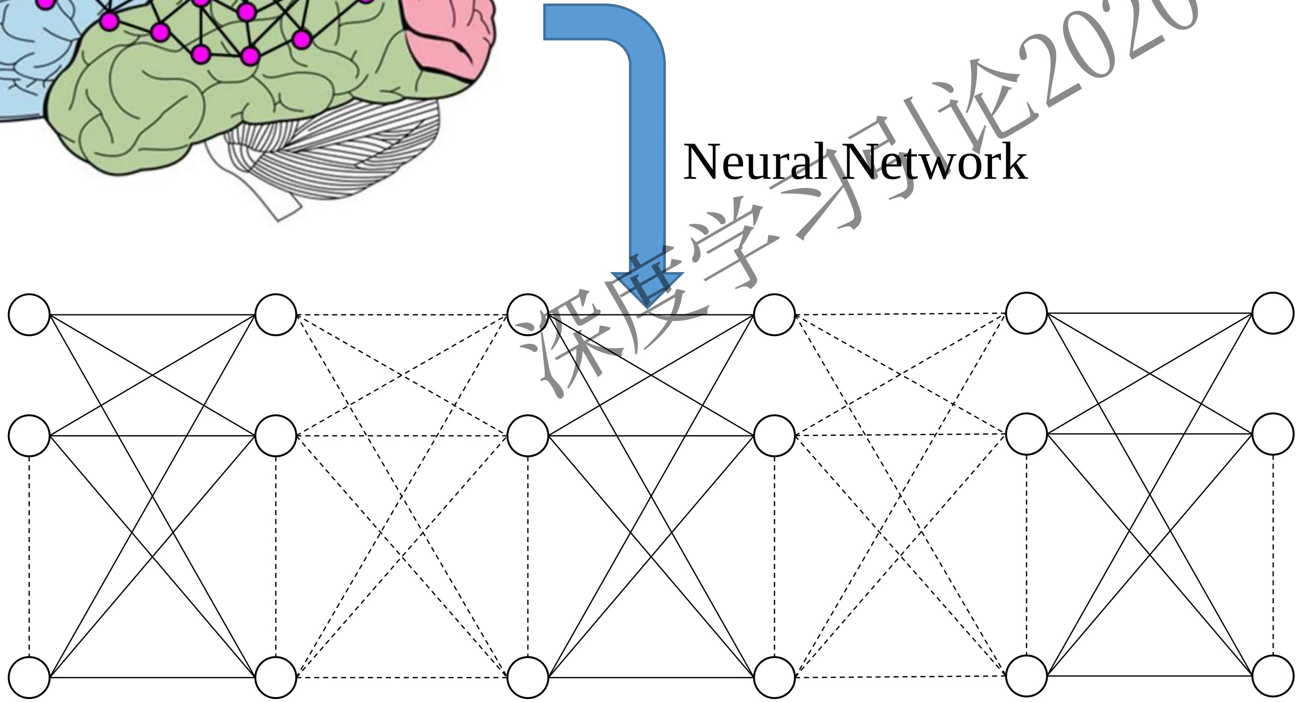


Without a supervisor, how can the student learn that the fruits in these pictures are in fact the same? There must exist some learning mechanism in his brain neural network?

Unsupervised Learning: Feature Extraction



There must exist some learning mechanism can do feature extraction without supervisor in the brain neural network.



Input x

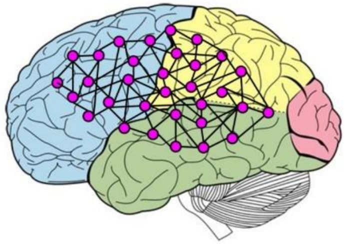
No target

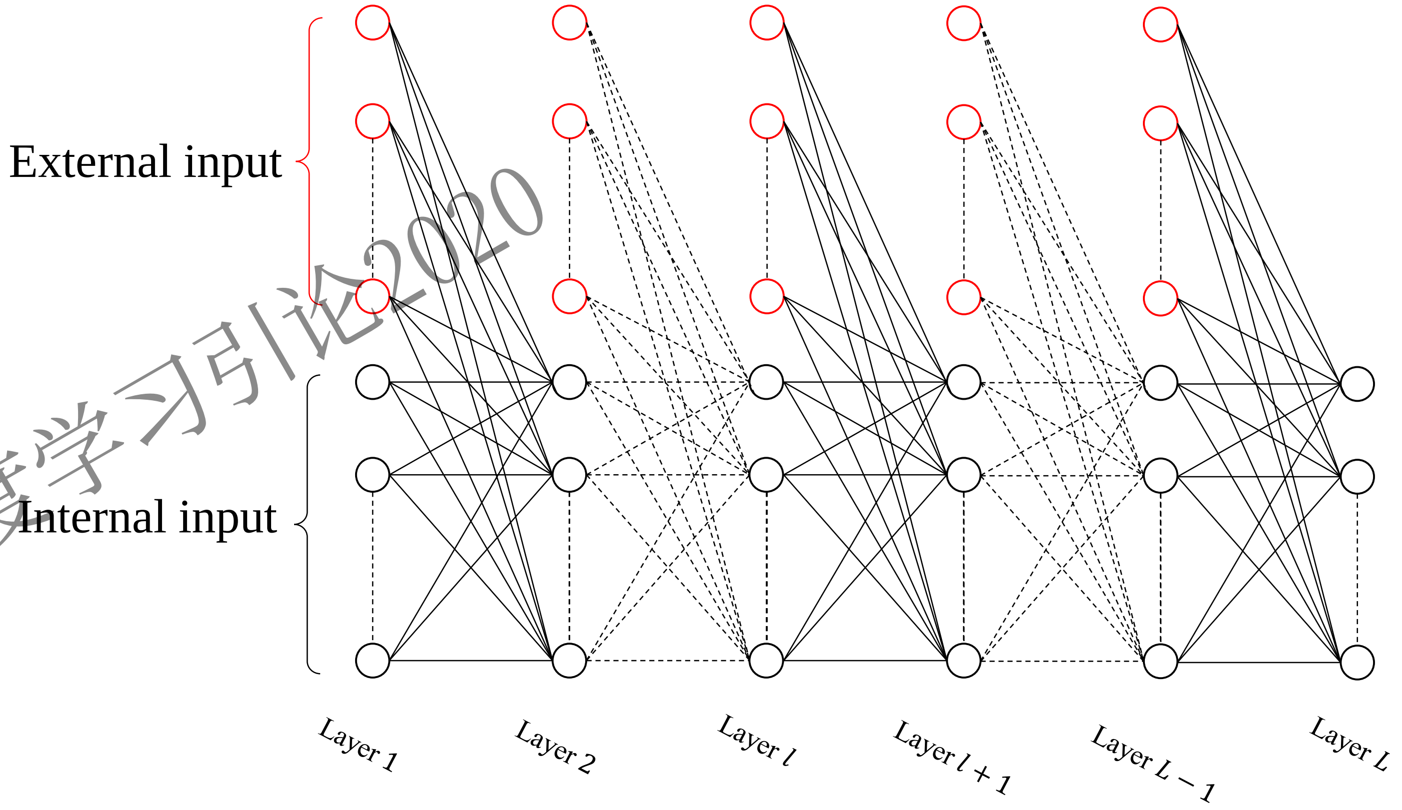
Problem: How to develop algorithms such that an artificial network can learn without any supervisor?

Outline

- Learning in Neural Networks
- Supervised Learning
- Unsupervised Learning
- Autoencoder Neural Networks
  - The Network Structure
  - Compressive Representation
  - Sparse Representation
  - Feature Learning
  - Application to Supervised Learning
- Generative Adversarial Networks
- Assignment

Autoencoder Neural Networks





The well-known network structure

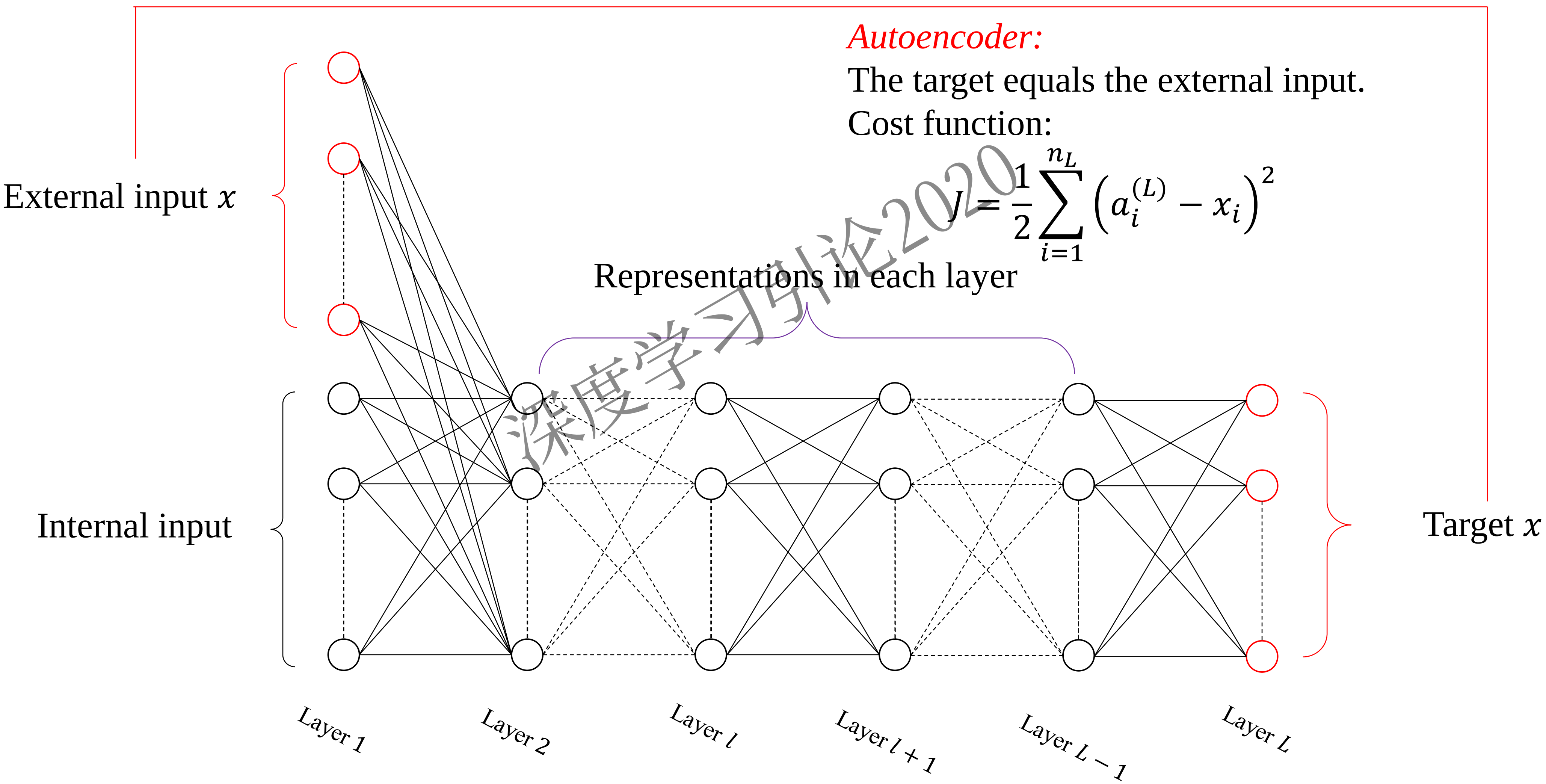
Autoencoder Neural Networks

Autoencoder:

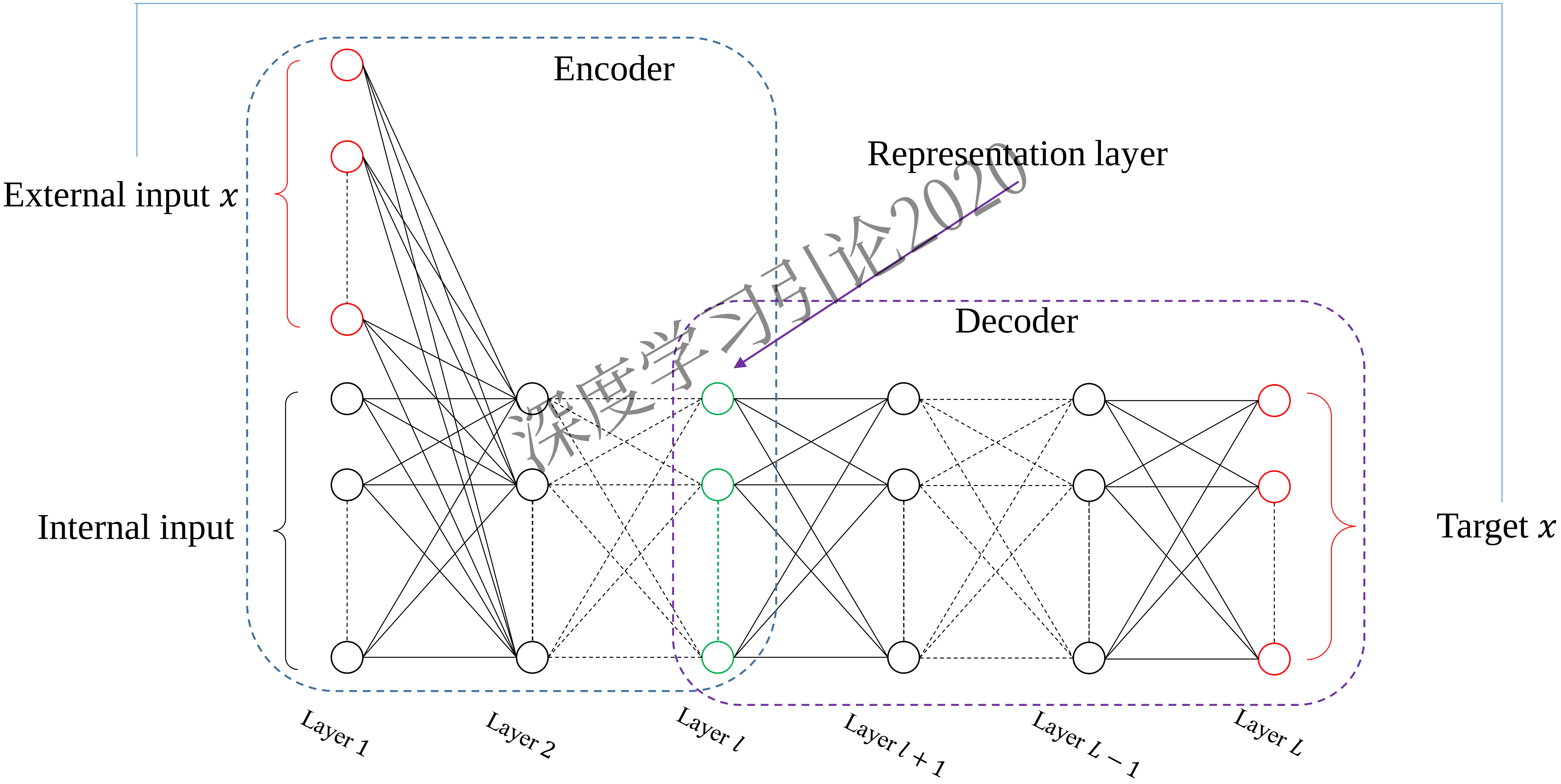
The target equals the external input.

Cost function:

$$J = \frac{1}{2}\sum_{i=1}^{n_L}\left(a_i^{(L)} - x_i\right)^2$$



Autoencoder Neural Networks

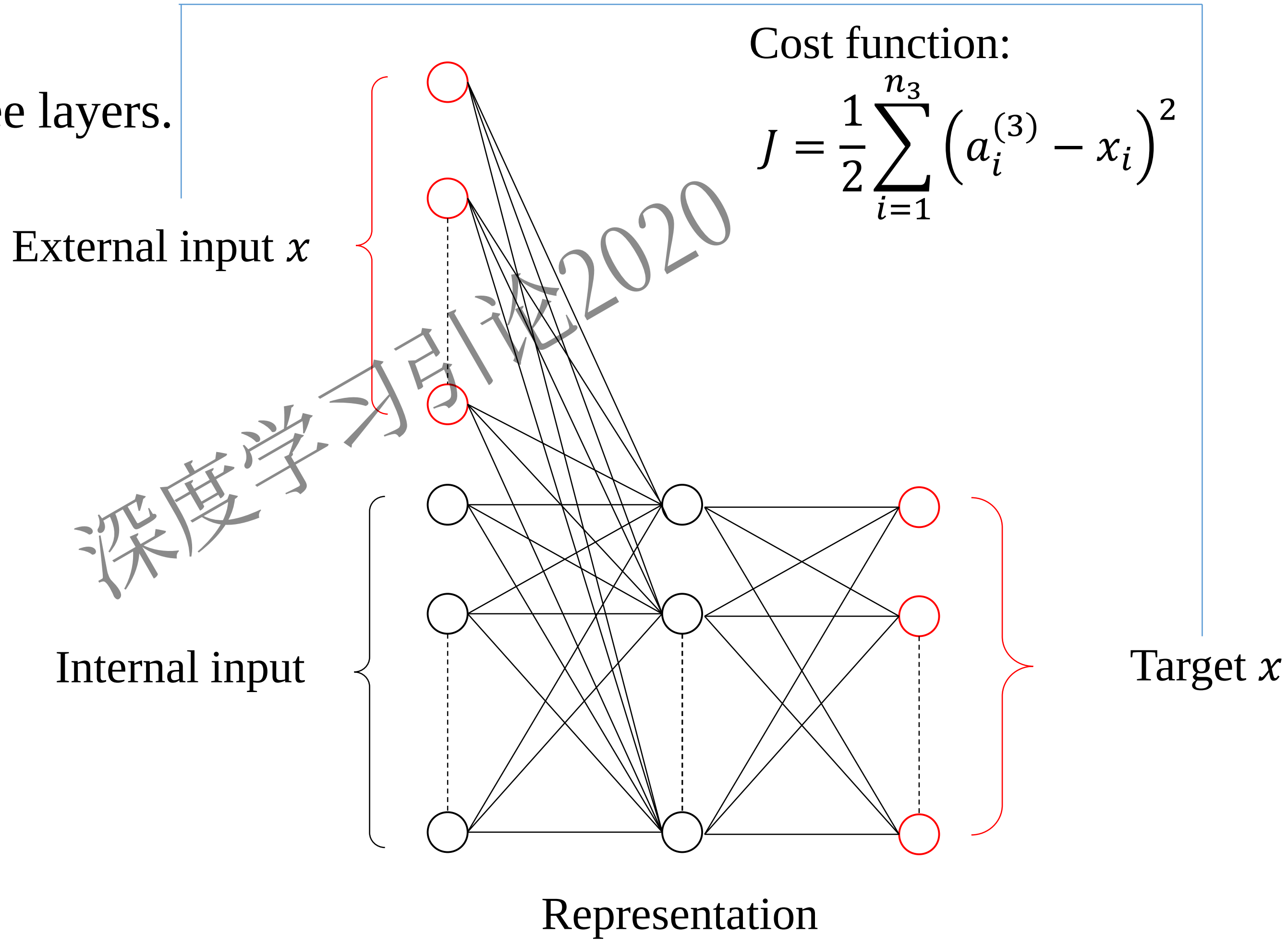


Autoencoder Neural Networks

Simplest Autoencoder:
The simplest autoencoder contains three layers.

$$a_i^{(2)} = f\left(\sum_{j=1}^{n_1} w_{ij}^{(1)} a_j^{(1)}\right)$$

$$a_i^{(3)} = f\left(\sum_{j=1}^{n_2} w_{ij}^{(2)} a_j^{(2)}\right)$$



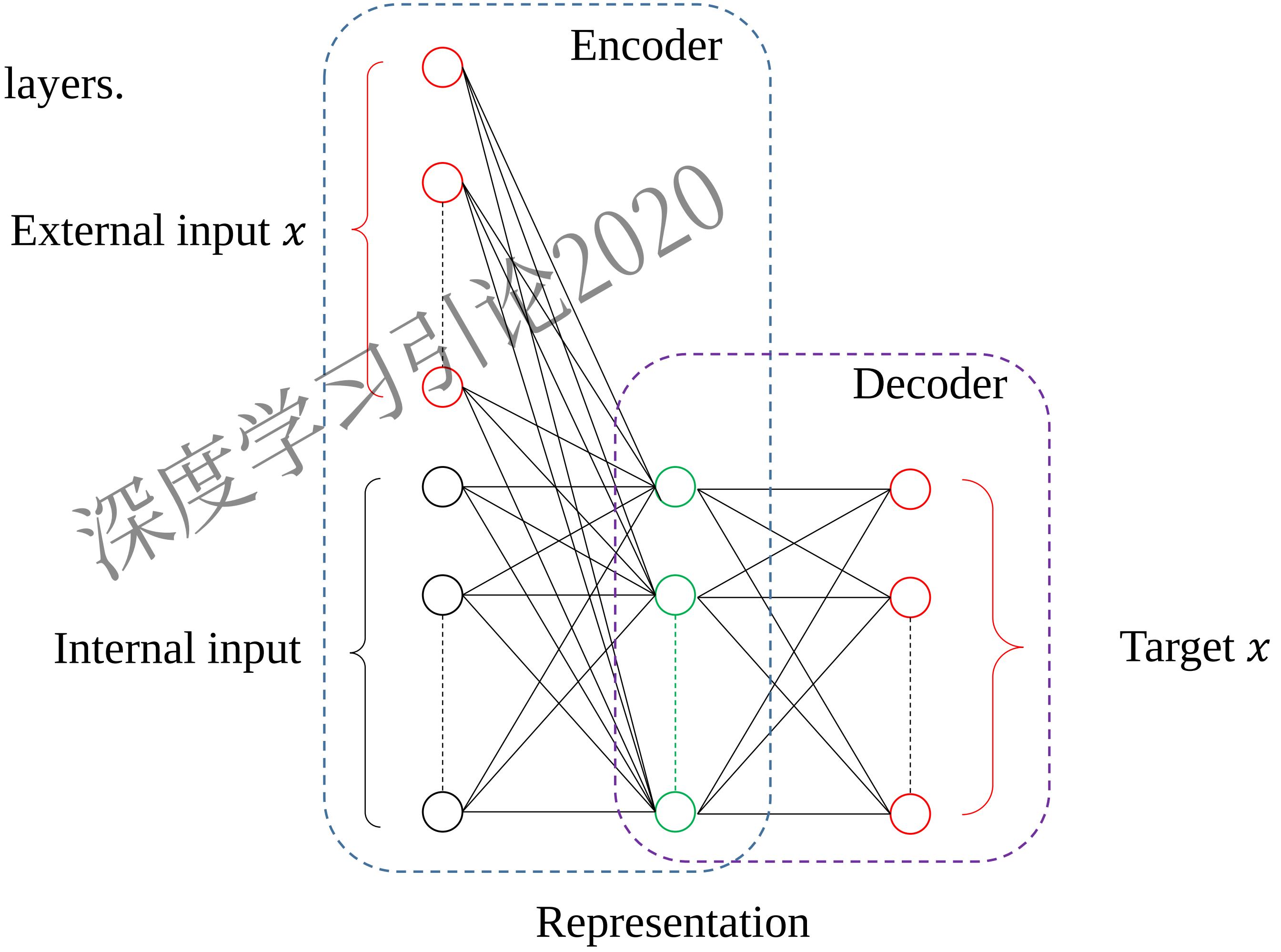
Autoencoder Neural Networks

Simplest Autoencoder:

The simplest autoencoder contains three layers.

$$a_i^{(2)} = f\left(\sum_{j=1}^{n_1} w_{ij}^{(1)} a_j^{(1)}\right)$$

$$a_i^{(3)} = f\left(\sum_{j=1}^{n_2} w_{ij}^{(2)} a_j^{(2)}\right)$$



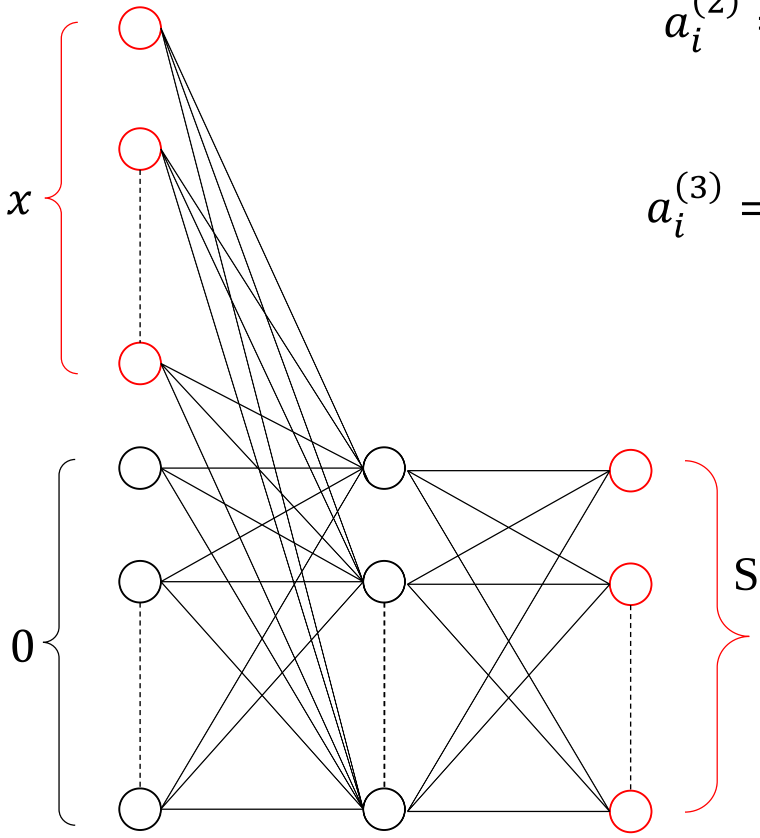
Autoencoder Neural Networks

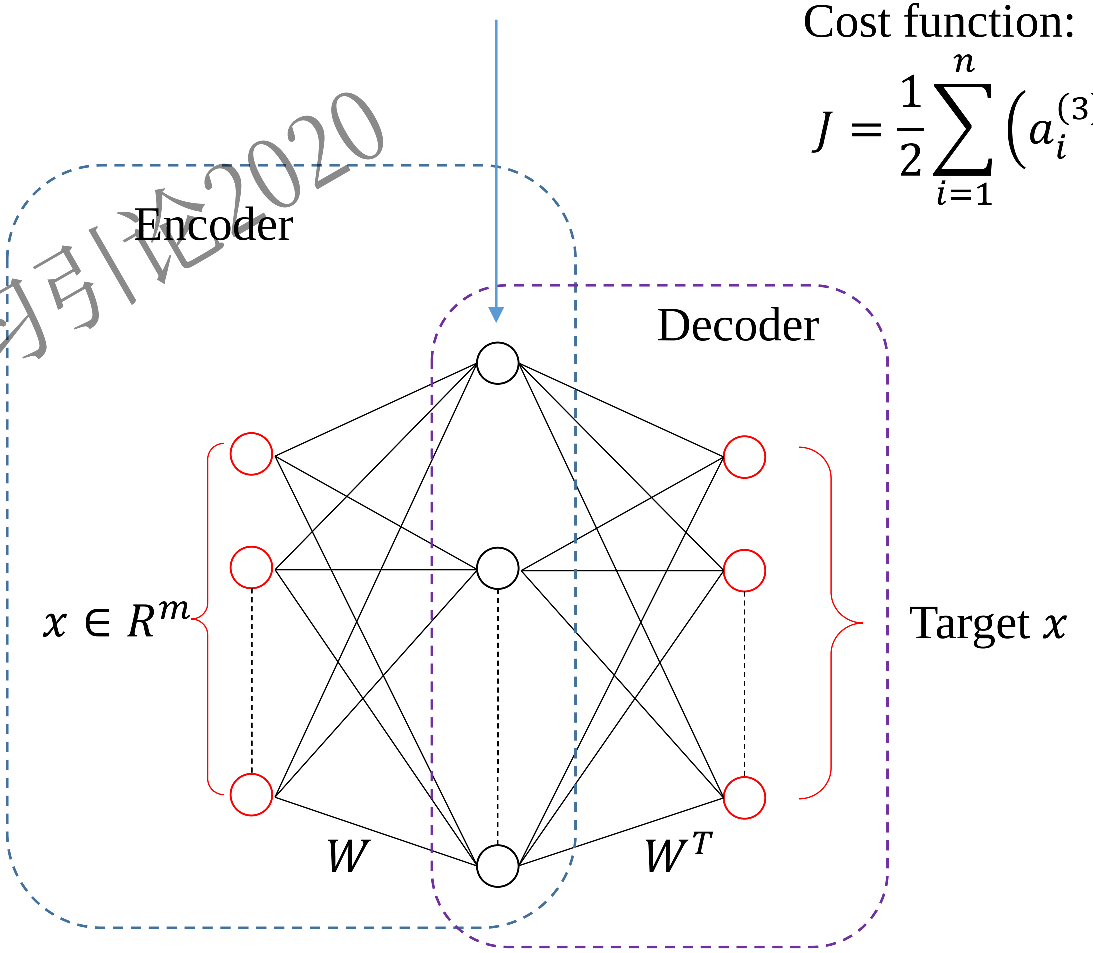
$$a_i^{(2)} = f\left(\sum_{j=1}^{m} w_{ij} x_j\right)$$

$$a_i^{(3)} = f\left(\sum_{j=1}^{n} w_{ji} a_j^{(2)}\right)$$

Cost function:

$$J = \frac{1}{2}\sum_{i=1}^{n}\left(a_i^{(3)} - x_i\right)^2$$



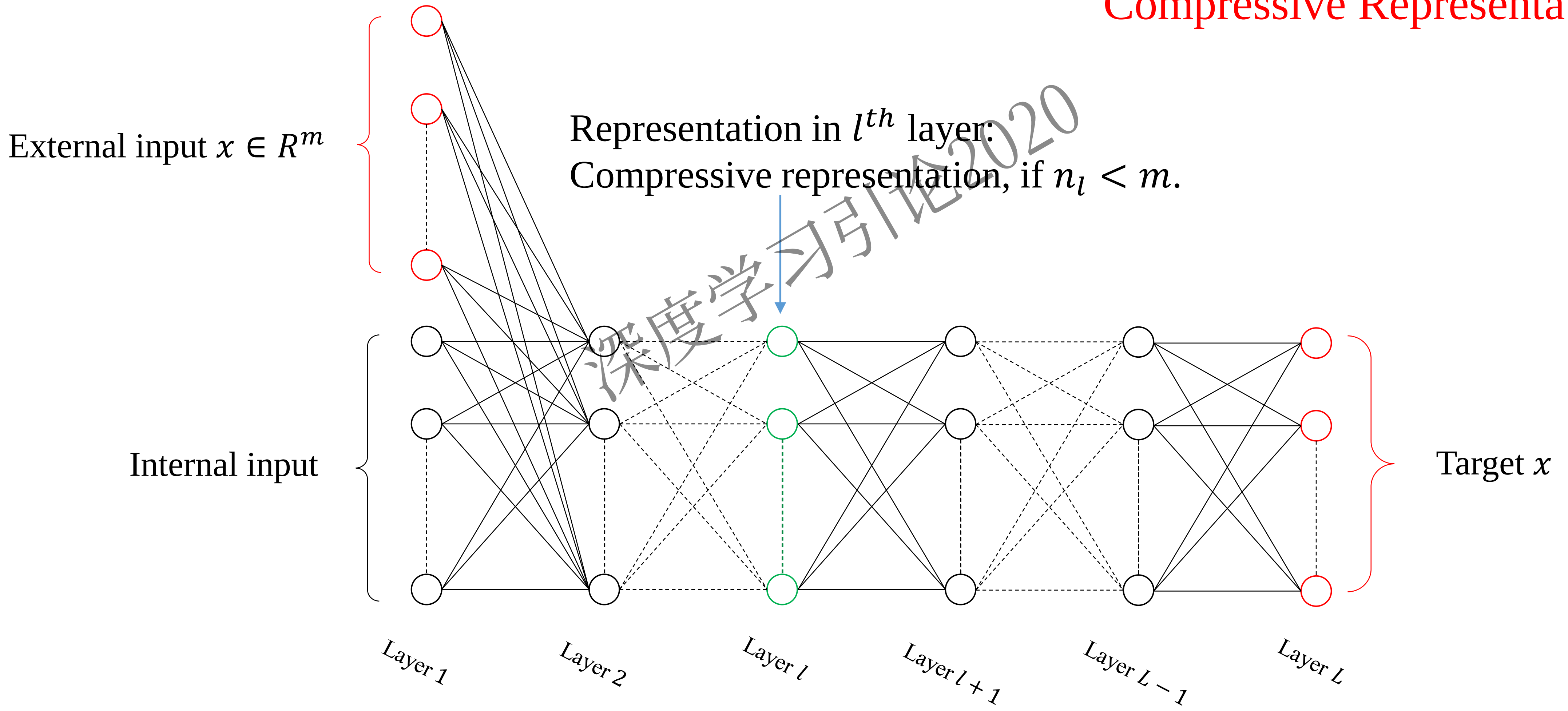


Outline

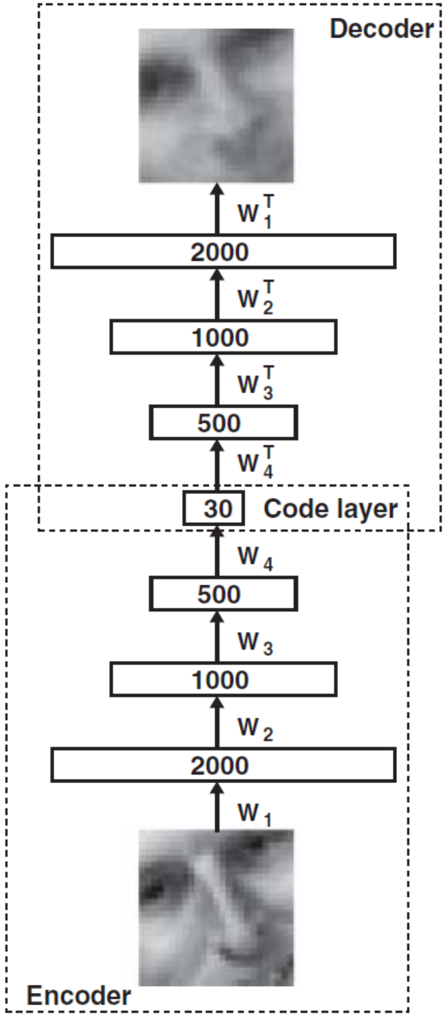
- Learning in Neural Networks
- Supervised Learning
- Unsupervised Learning
- Autoencoder Neural Networks
  - The Network Structure
  - Compressive Representation
  - Sparse Representation
  - Feature Learning
  - Application to Supervised Learning
- Generative Adversarial Networks
- Assignment

Autoencoder Neural Networks

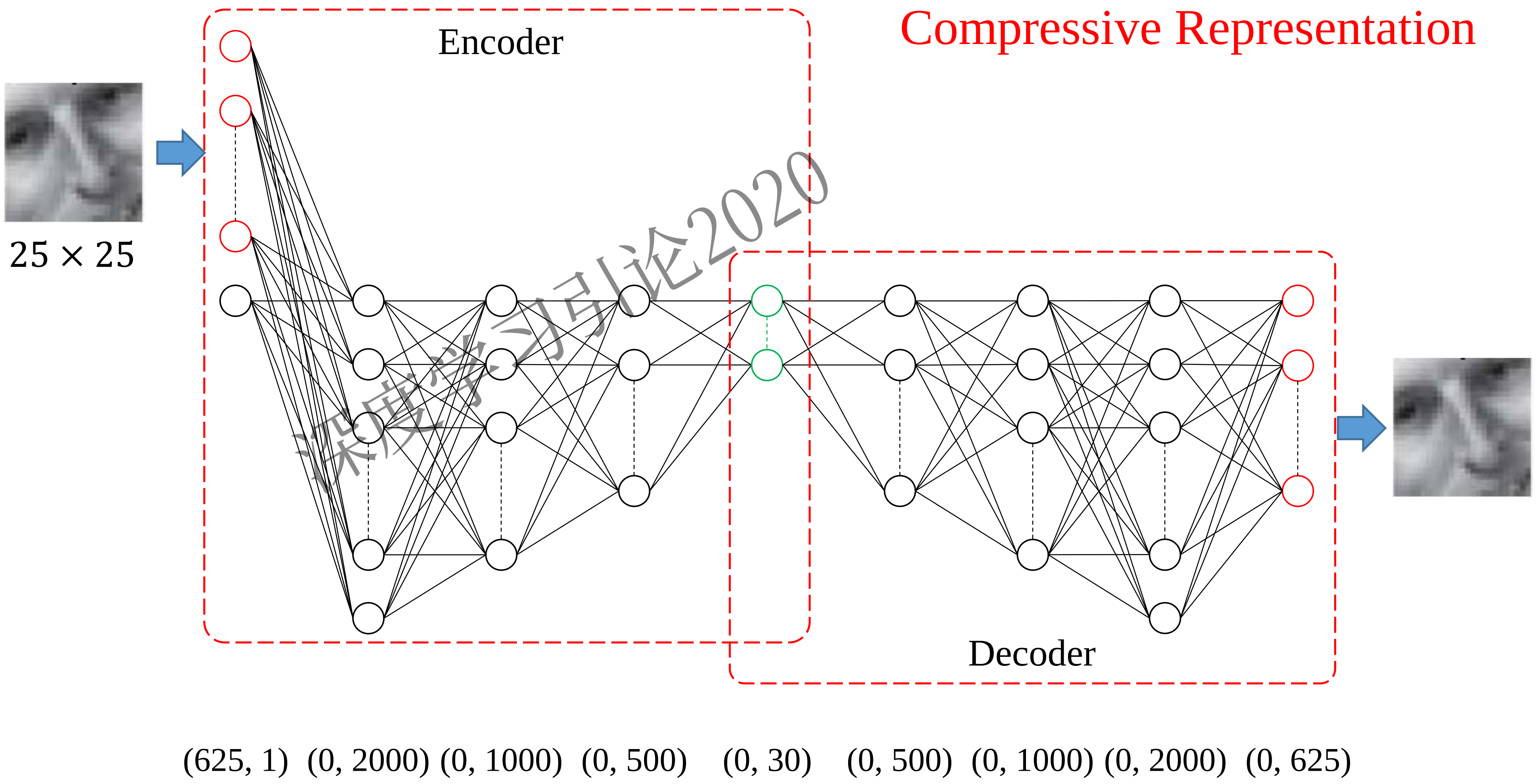
Compressive Representation



Autoencoder Neural Networks



Unrolling

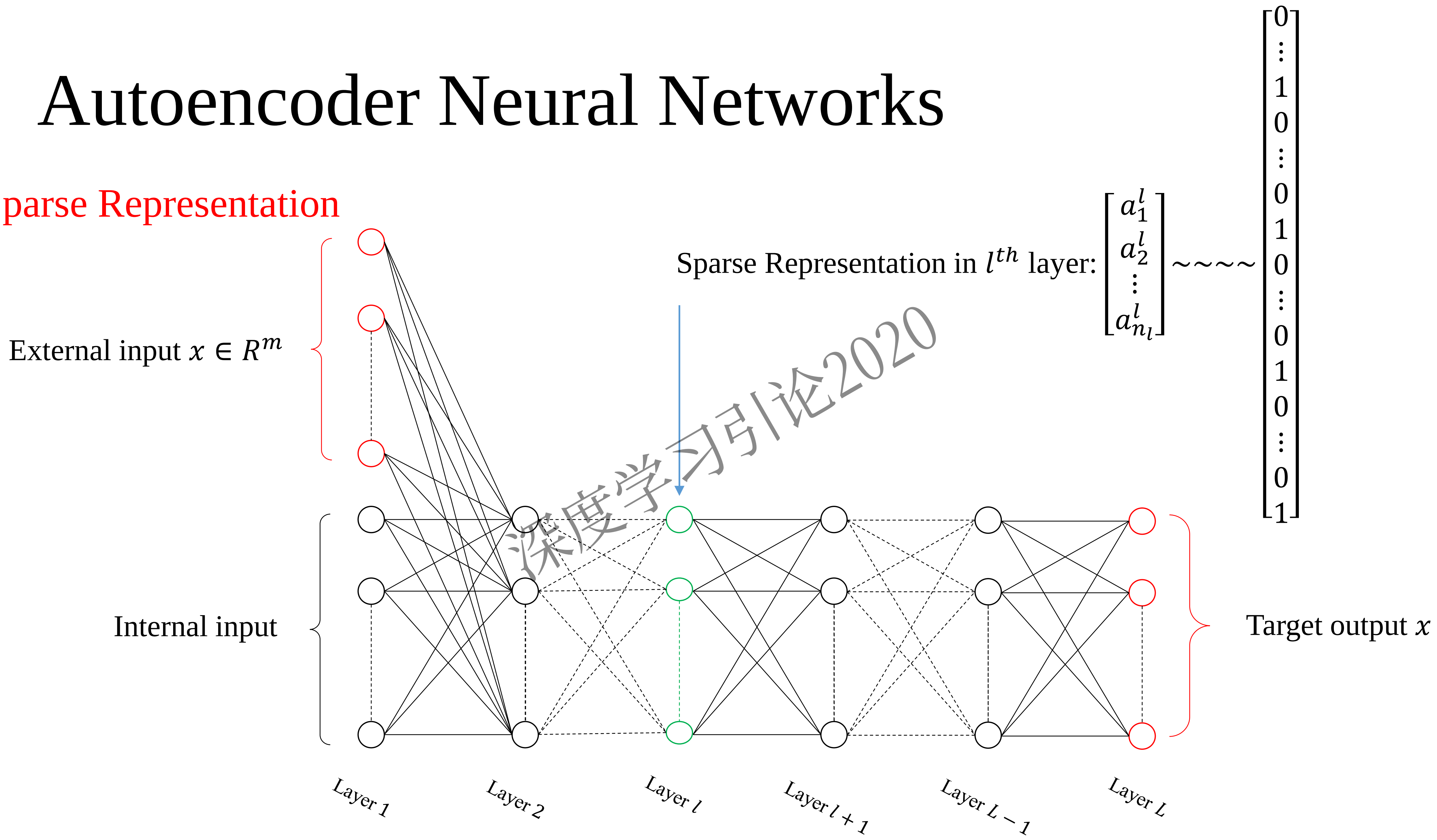


Outline

- Learning in Neural Networks
- Supervised Learning
- Unsupervised Learning
- Autoencoder Neural Networks
  - The Network Structure
  - Compressive Representation
  - Sparse Representation
  - Feature Learning
  - Application to Supervised Learning
- Generative Adversarial Networks
- Assignment

Autoencoder Neural Networks

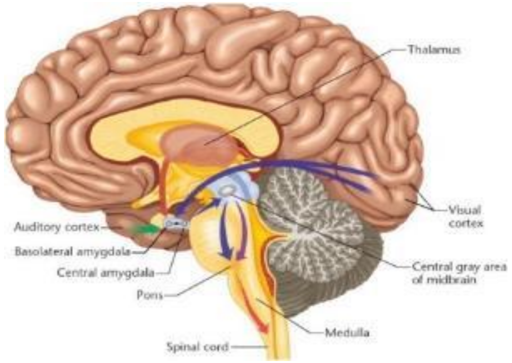
Sparse Representation



Autoencoder Neural Networks

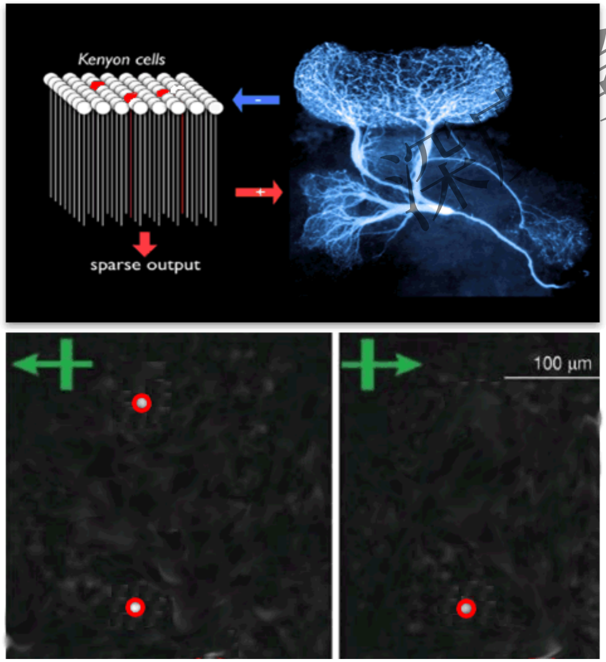
Sparse Representation

The brain represents information in sparsity way.



nature

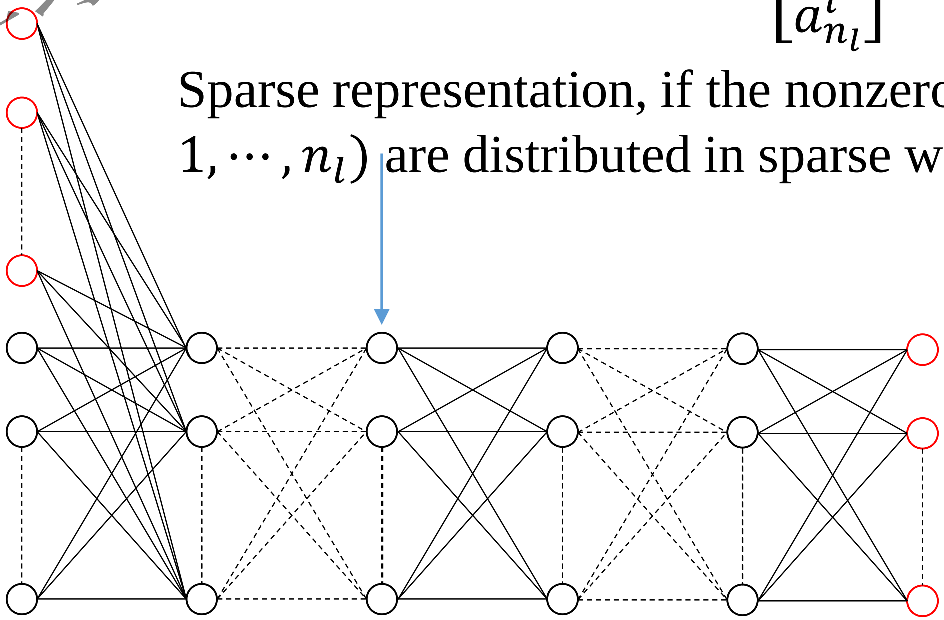
Emergence of simple-cell receptive field properties by learning a sparse code for natural images
Bruno A. Olshausen and David J. Field
Nature **381**, 607 - 609 (13 June 1996);
doi:10.1038/381607a0



Problem:
How to get sparse representation?

Representation in l^{th} layer: $\begin{bmatrix} a_1^l \\ a_2^l \\ \vdots \\ a_{n_l}^l \end{bmatrix}$

Sparse representation, if the nonzero $a_i^l (i = 1, \cdots, n_l)$ are distributed in sparse way.



Autoencoder Neural Networks

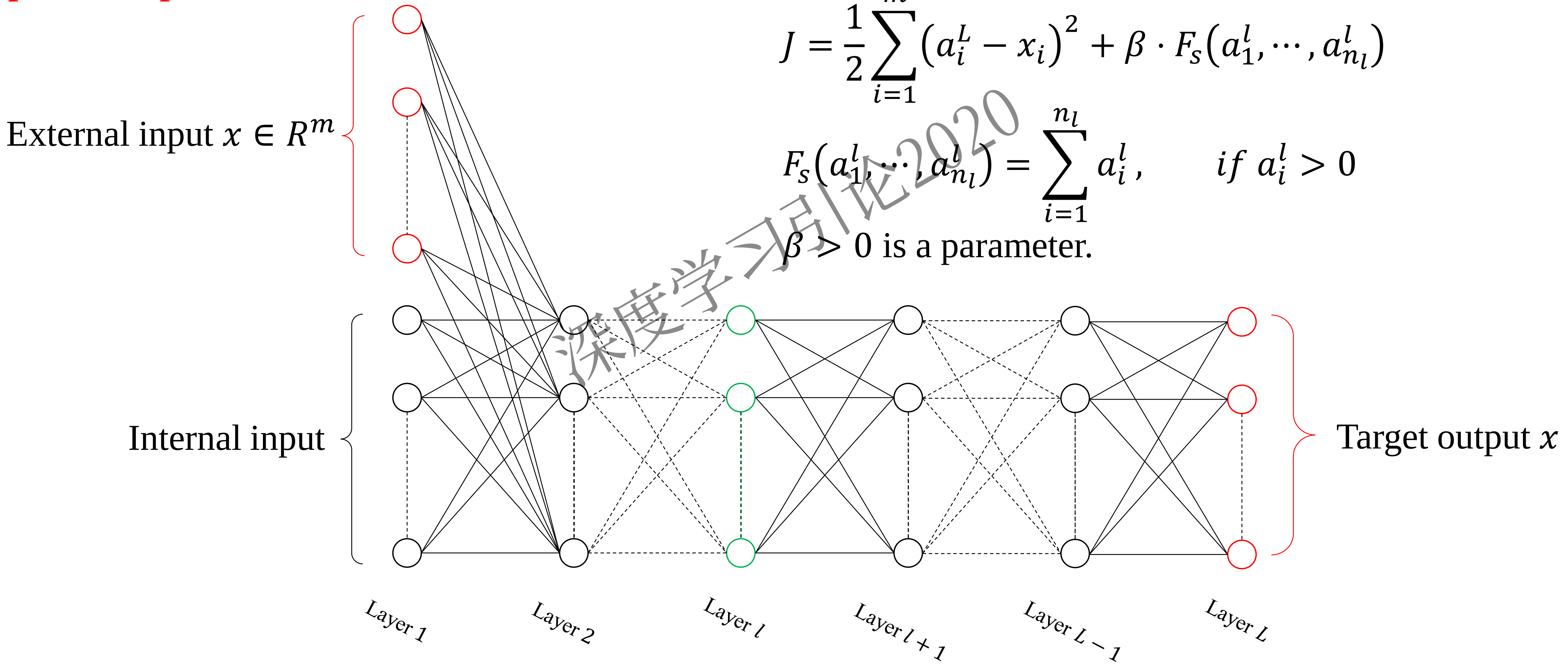
Sparse Representation

- Cost function:

$$J = \frac{1}{2}\sum_{i=1}^{m}\left(a_i^L - x_i\right)^2 + \beta \cdot F_s\left(a_1^l, \cdots, a_{n_l}^l\right)$$

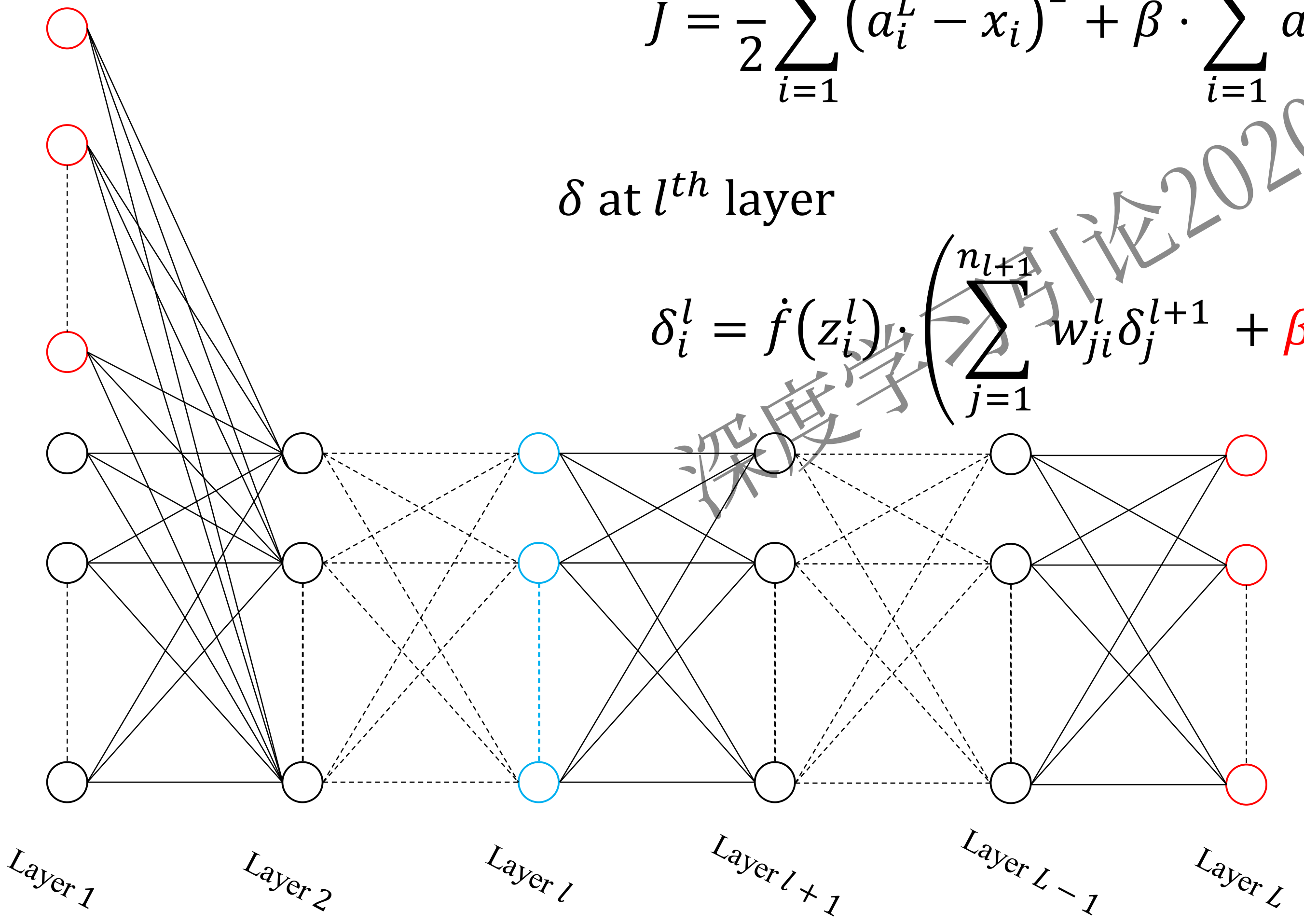
$$F_s\left(a_1^l, \cdots, a_{n_l}^l\right) = \sum_{i=1}^{n_l} a_i^l, \qquad if\ a_i^l > 0$$

$\beta > 0$ is a parameter.



Autoencoder Neural Networks

Sparse Representation

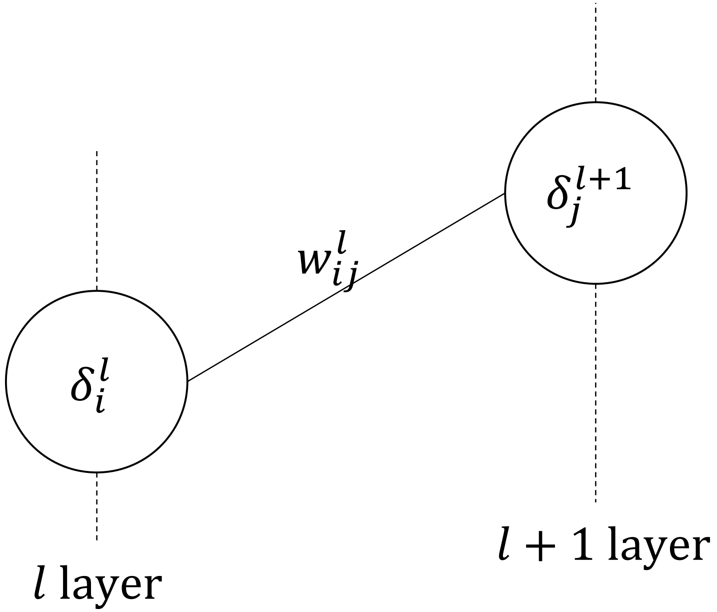


Cost function

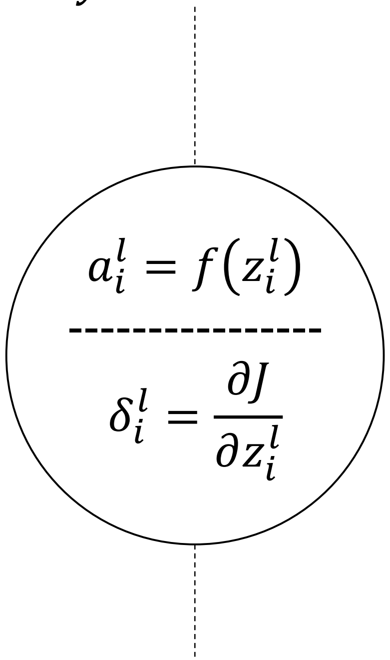
$$J = \frac{1}{2}\sum_{i=1}^{m}\left(a_i^L - x_i\right)^2 + \beta \cdot \sum_{i=1}^{n_l} a_i^l$$

δ at l^{th} layer

$$\delta_i^l = \dot{f}(z_i^l) \cdot \left(\sum_{j=1}^{n_{l+1}} w_{ji}^l \delta_j^{l+1} + \beta\right)$$



l *layer* i^{th} *neuron*

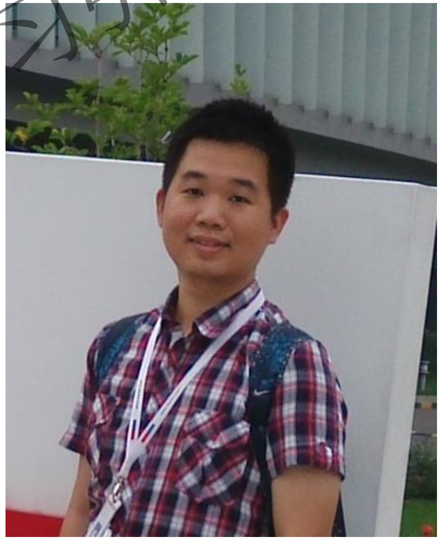


Outline

- Learning in Neural Networks
- Supervised Learning
- Unsupervised Learning
- Autoencoder Neural Networks
  - The Network Structure
  - Compressive Representation
  - Sparse Representation
  - Feature Learning
  - Application to Supervised Learning
- Generative Adversarial Networks
- Assignment

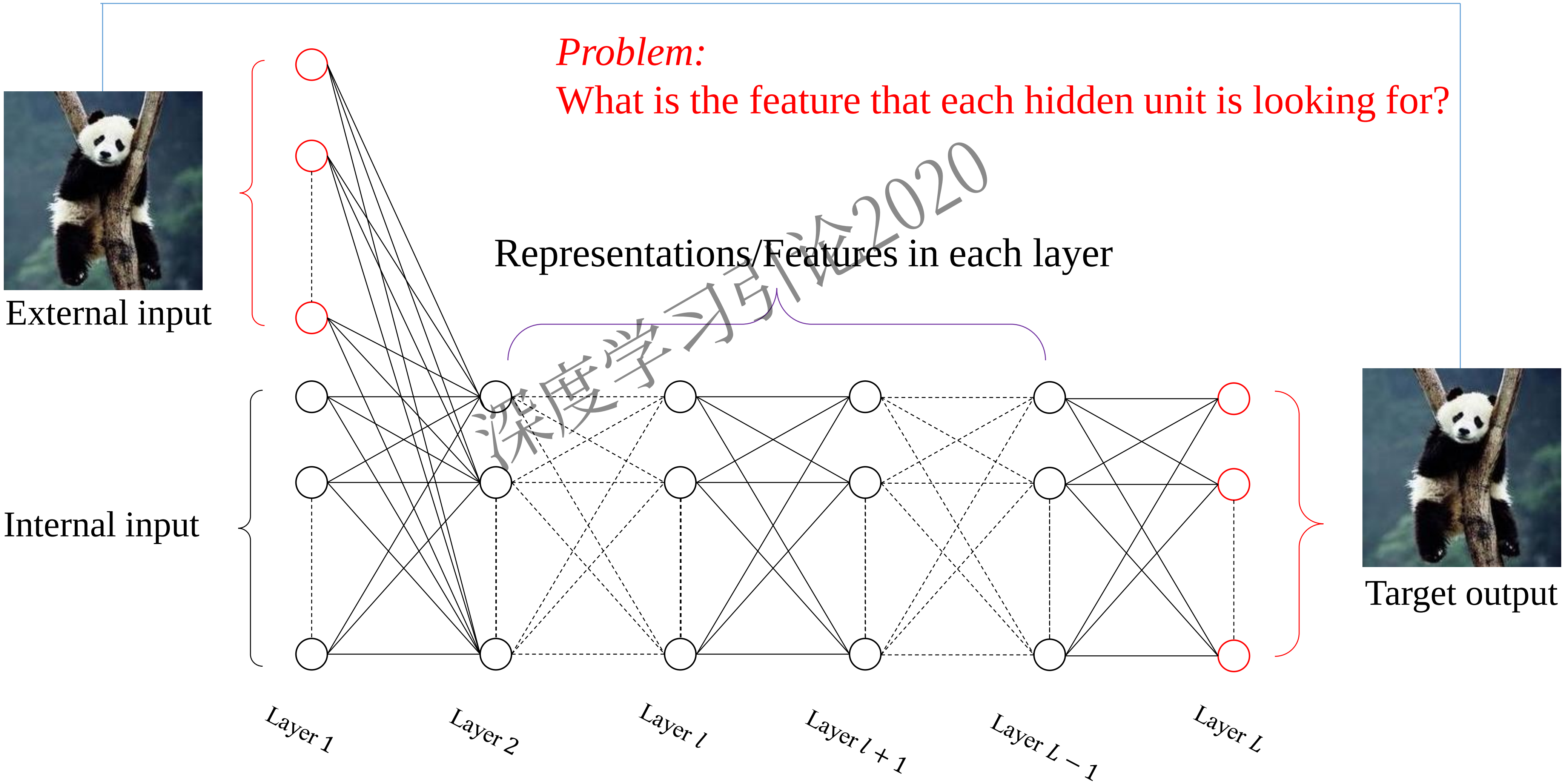
Autoencoder Neural Networks

They are not the same one. The student successfully learnt the features of this unknown fruit.



Without a supervisor, how can the student learn that the fruits in these pictures are in fact the same? There must exist some learning mechanism in his brain neural network?

Autoencoder Neural Networks



Autoencoder Neural Networks

Problem:

What is the feature that hidden neuron i is looking for?

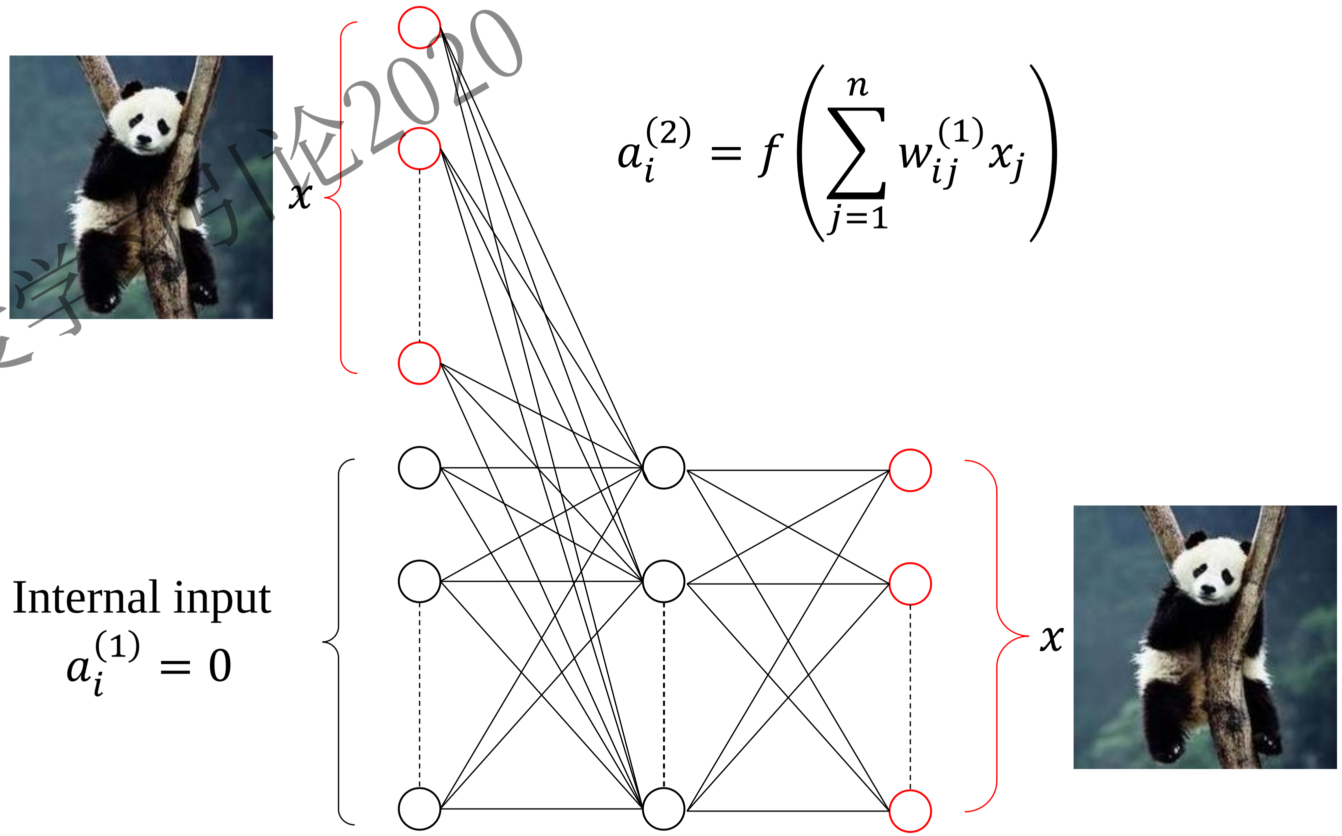
In other words, what input image x cause $a_i^{(2)}$ to be maximally activated?

$$\Updownarrow$$

$$\begin{cases} \max \sum_{i=1}^{n} w_{ij}^{(1)} x_j \\ s.t. \sum_{i=1}^{n} x_i^2 \le 1 \end{cases}$$

$$\Updownarrow$$

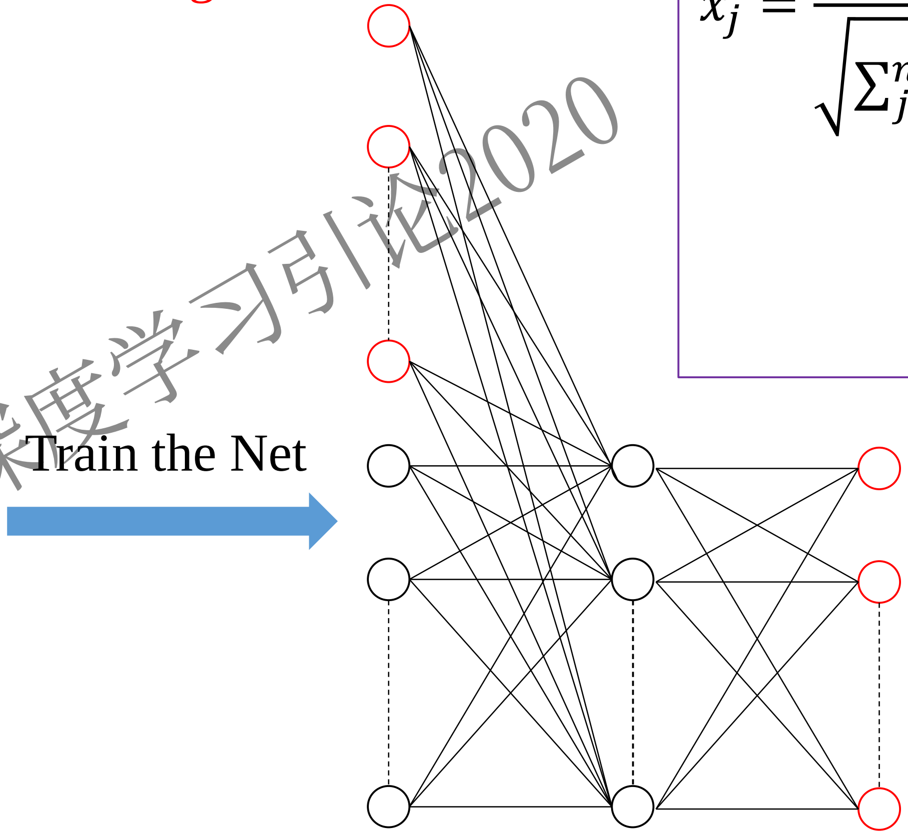
$$x_j = \frac{w_{ij}^{(1)}}{\sqrt{\sum_{j=1}^{n}\left(w_{ij}^{(1)}\right)^2}}, (j = 1, \cdots, n)$$



Exercise: How to solve this problem?

Autoencoder Neural Networks

Problem:
What is the feature that hidden neuron i is looking for?



$$x_j = \frac{w_{ij}^{(1)}}{\sqrt{\sum_{j=1}^{n}\left(w_{ij}^{(1)}\right)^2}}, (j = 1, \cdots, n)$$

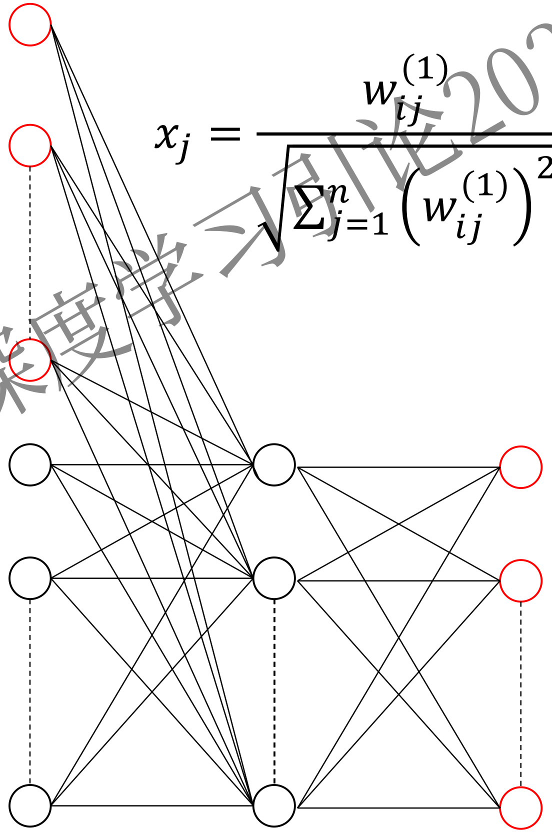
?

Autoencoder Neural Networks

Problem:

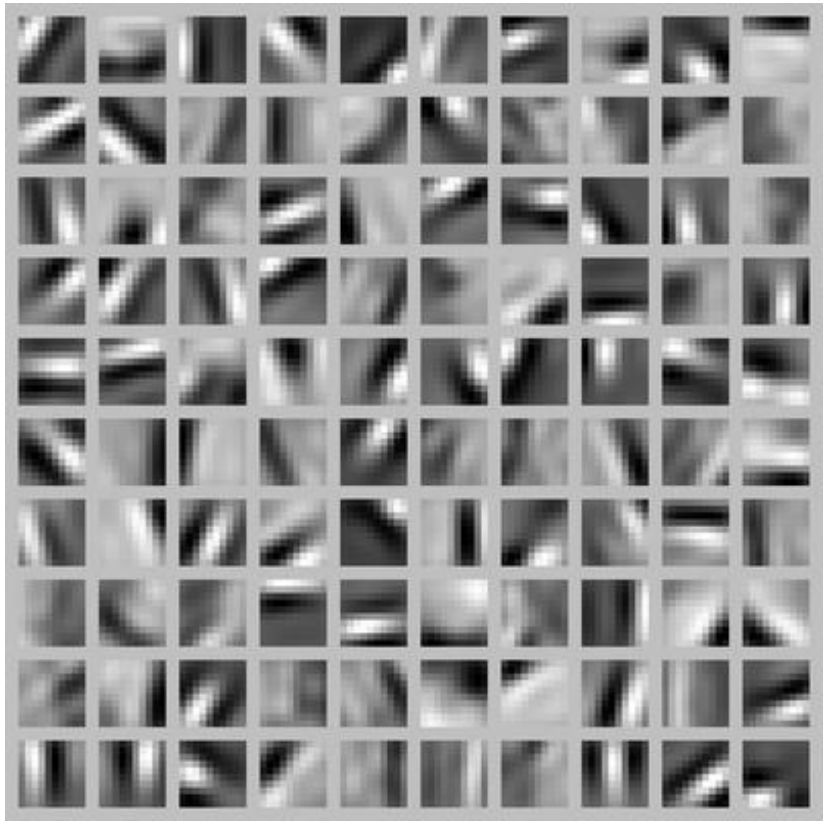
What is the feature that hidden neuron i is looking for?

Edges at different positions and orientations



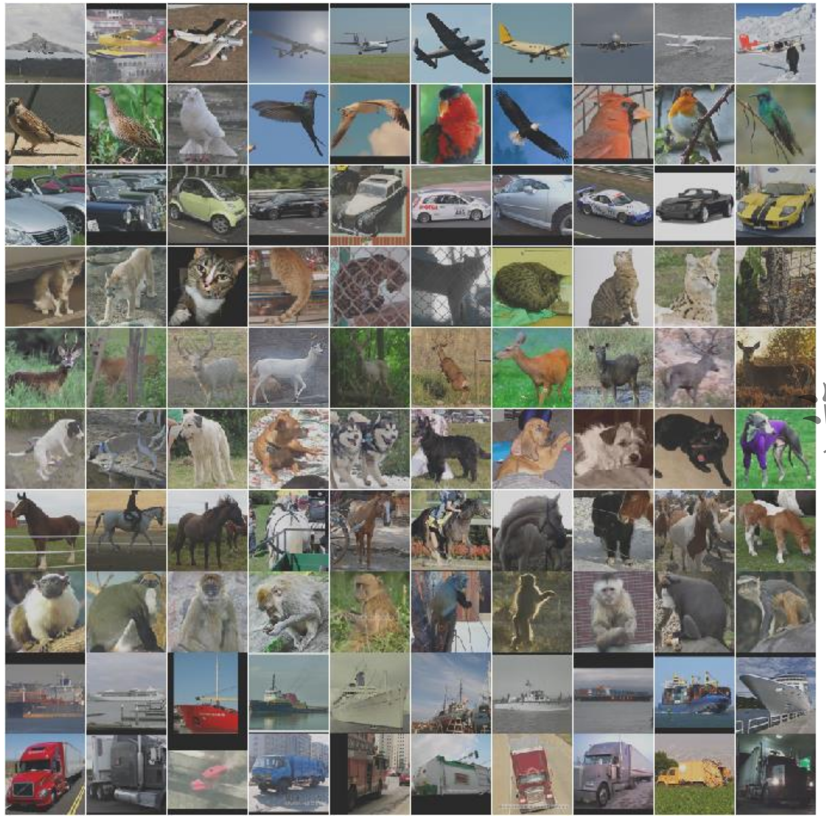
100

$$x_j = \frac{w_{ij}^{(1)}}{\sqrt{\sum_{j=1}^{n}\left(w_{ij}^{(1)}\right)^2}}$$

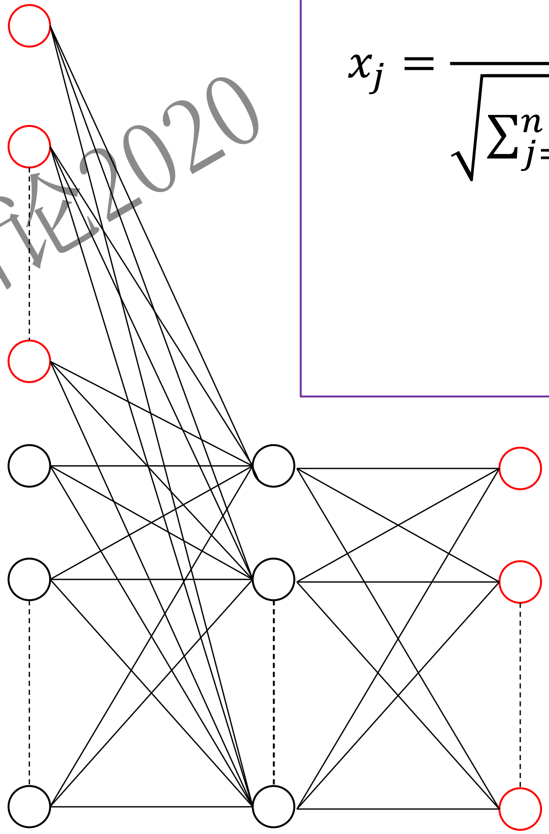


Autoencoder Neural Networks

Problem:
What is the feature that hidden neuron i is looking for?



Train the Net



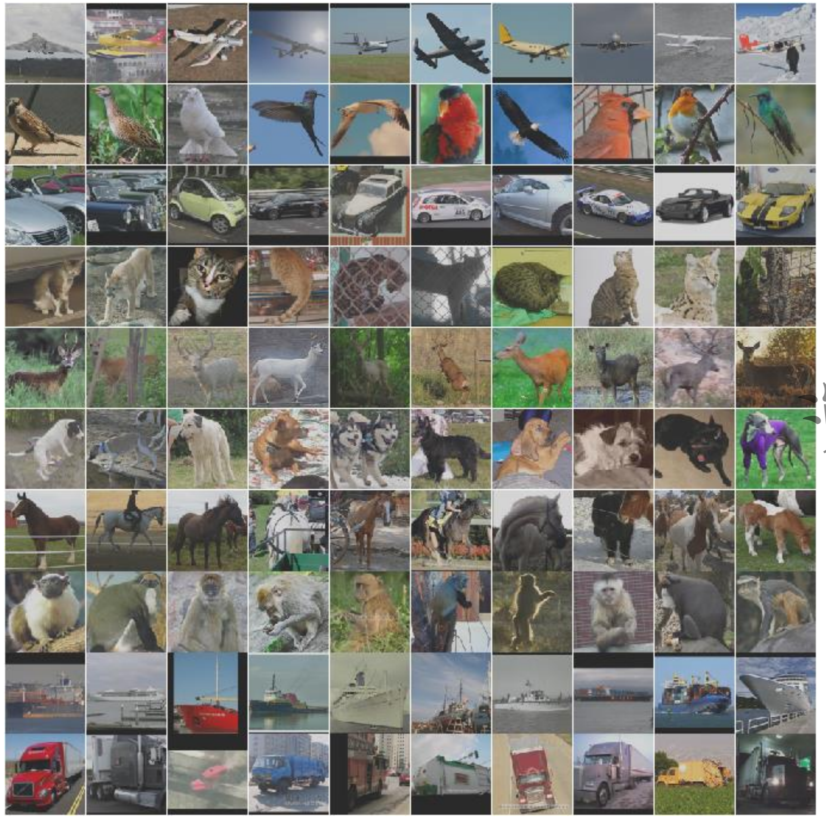
$$x_j = \frac{w_{ij}^{(1)}}{\sqrt{\sum_{j=1}^{n}\left(w_{ij}^{(1)}\right)^2}}, (j = 1, \cdots, n)$$

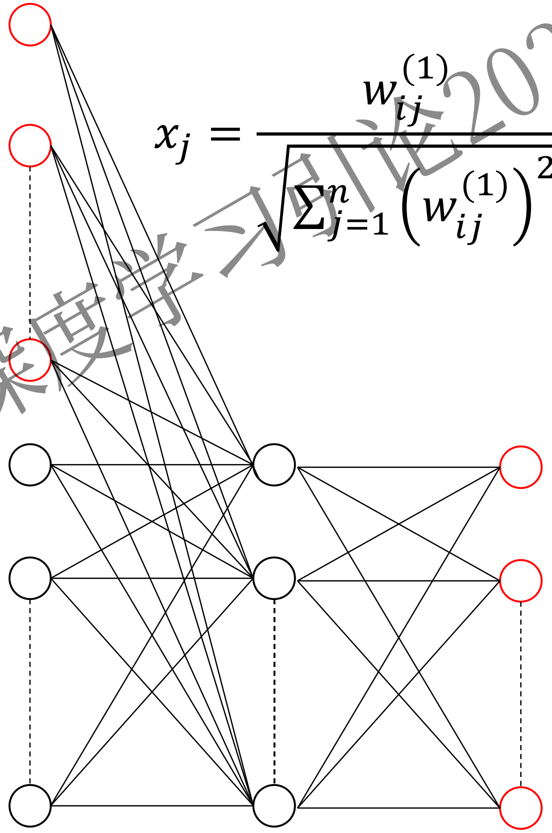
?

Autoencoder Neural Networks

Problem:

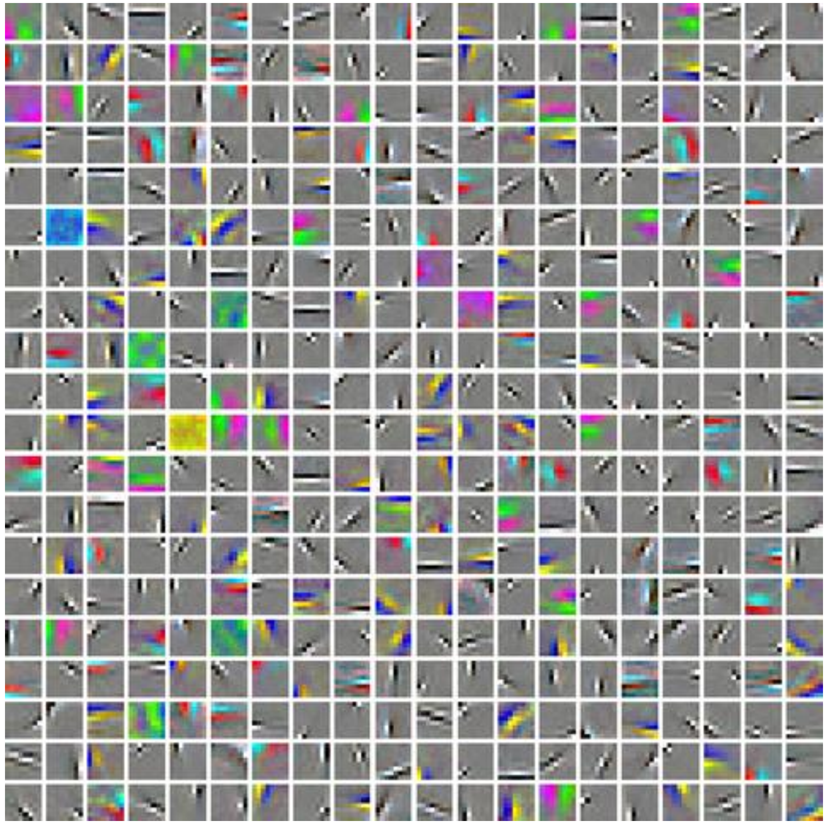
What is the feature that hidden neuron *i* is looking for?





$$x_j = \frac{w_{ij}^{(1)}}{\sqrt{\sum_{j=1}^{n}\left(w_{ij}^{(1)}\right)^2}}$$

Edges at different positions and orientations

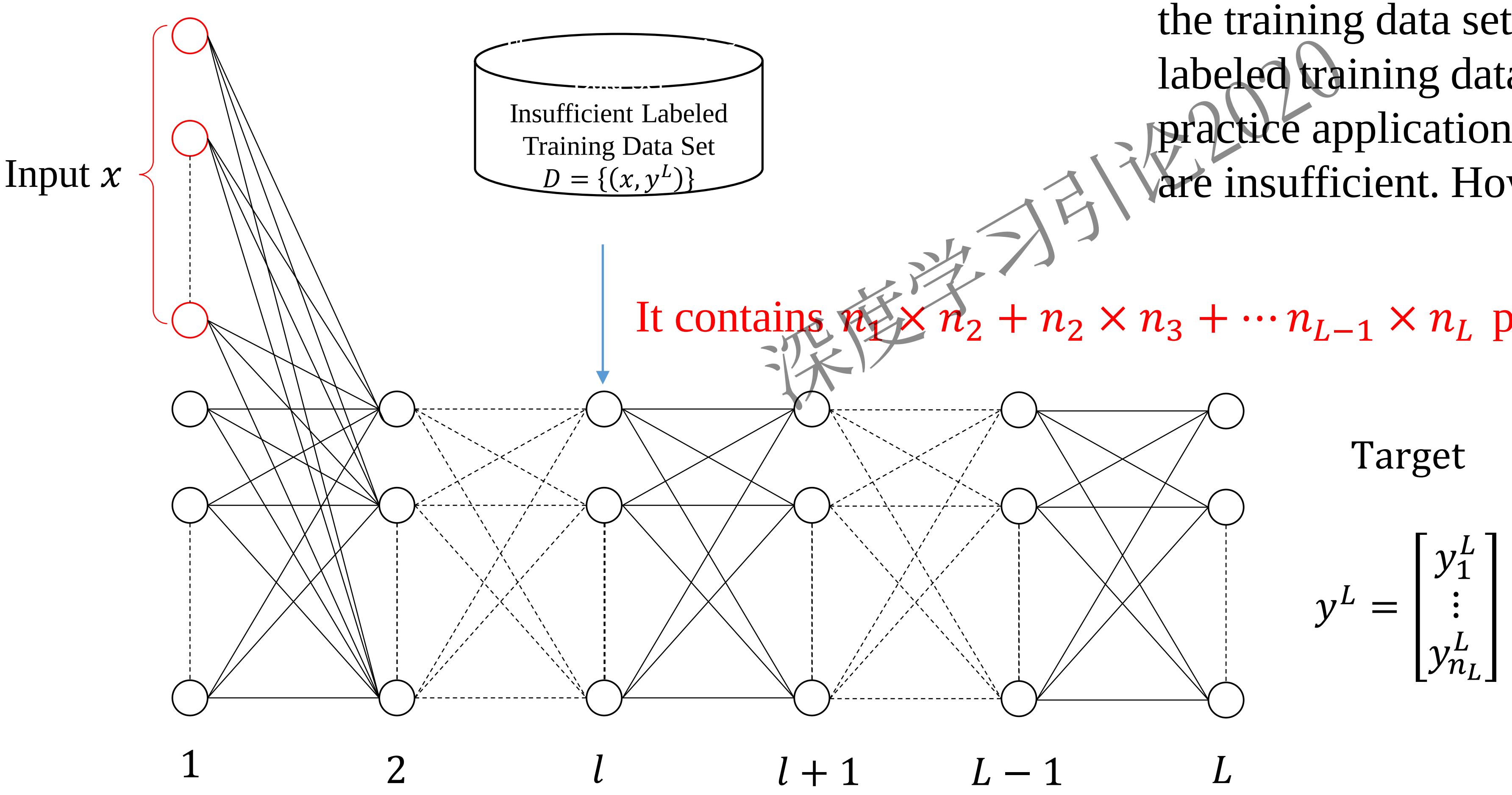


Outline

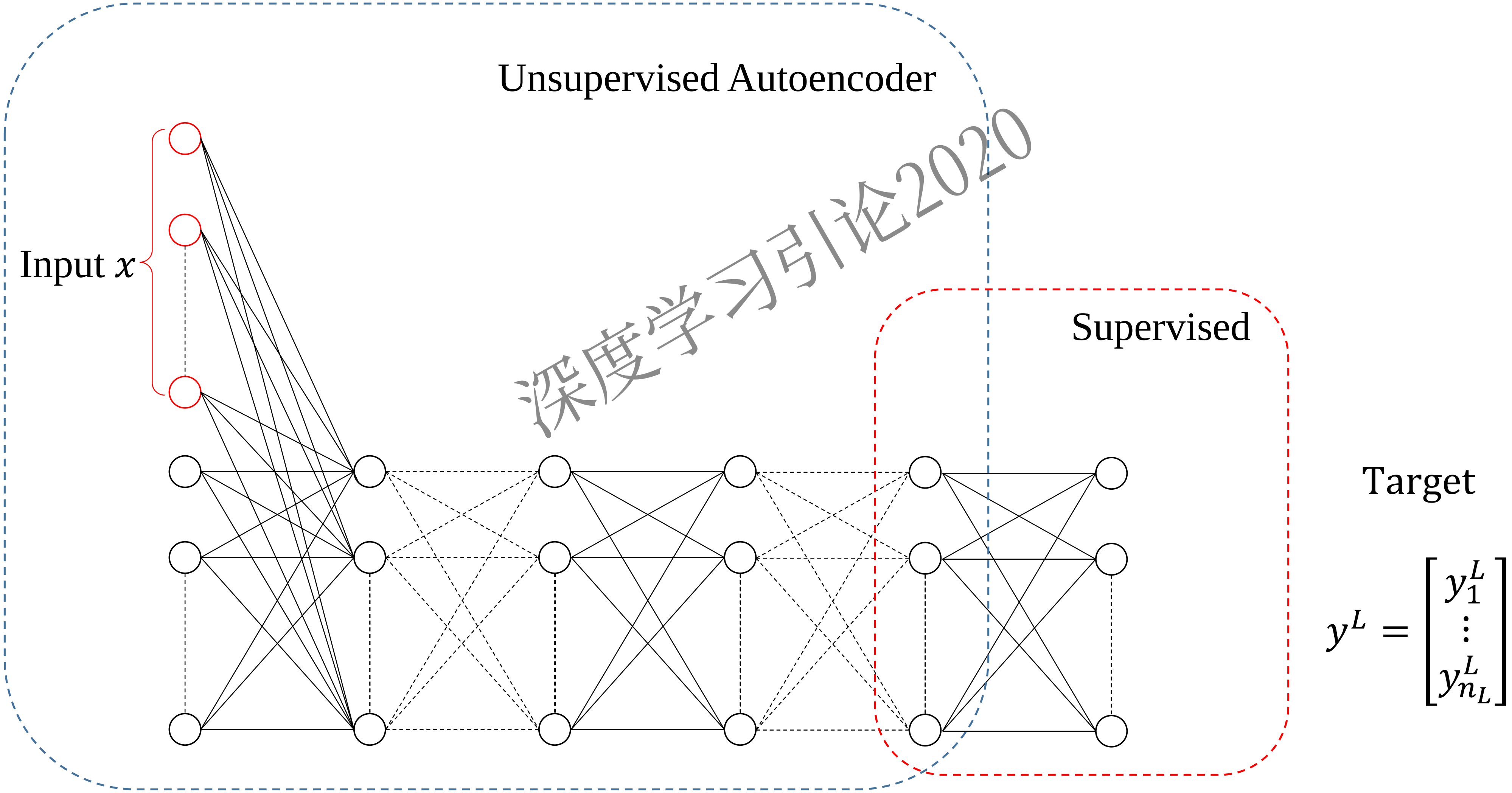
- Learning in Neural Networks
- Supervised Learning
- Unsupervised Learning
- Autoencoder Neural Networks
  - The Network Structure
  - Compressive Representation
  - Sparse Representation
  - Feature Learning
  - Application to Supervised Learning
- Generative Adversarial Networks
- Assignment

Application to Supervised Learning

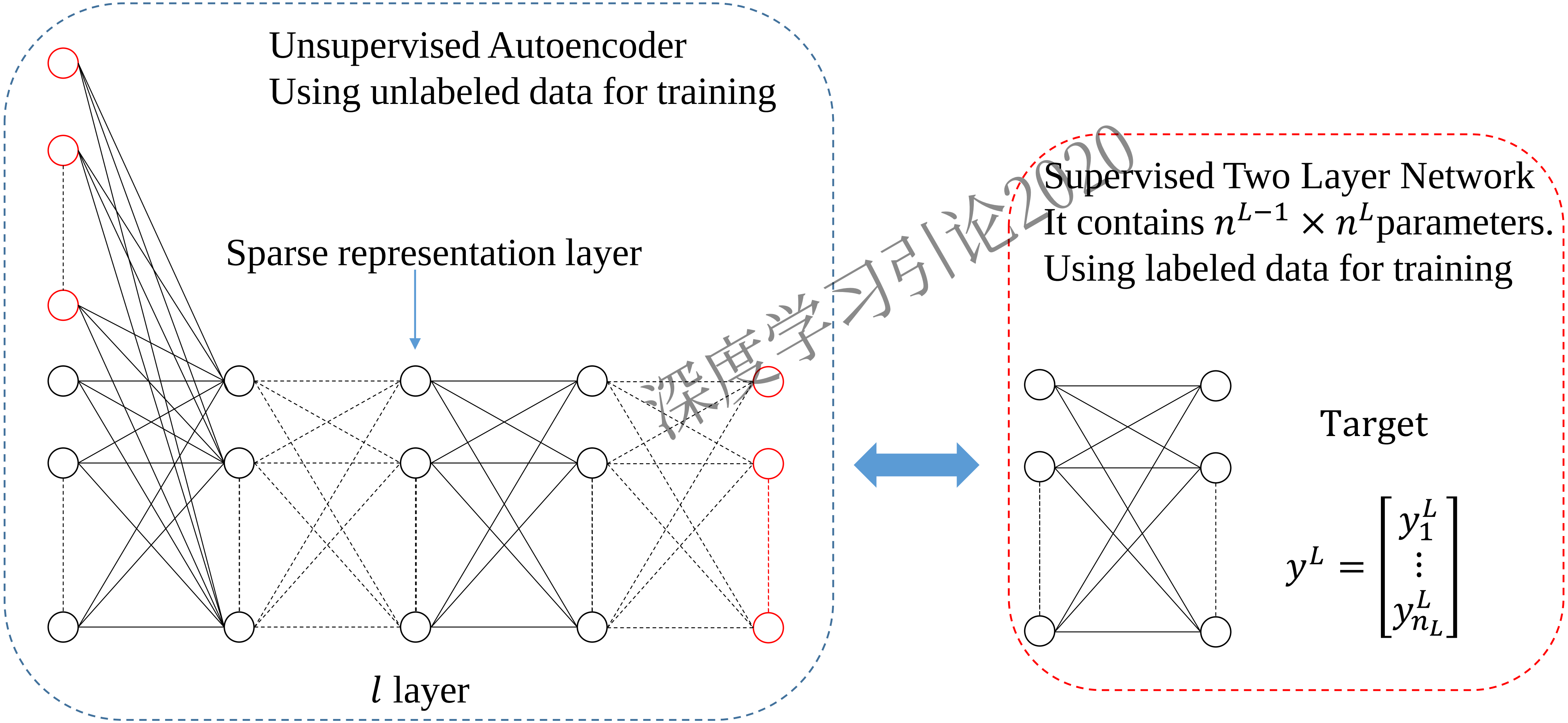
In supervised learning, to train the network, the training data set should contain sufficient labeled training data. However, in many practice applications, the labeled training data are insufficient. How to solve this problem?



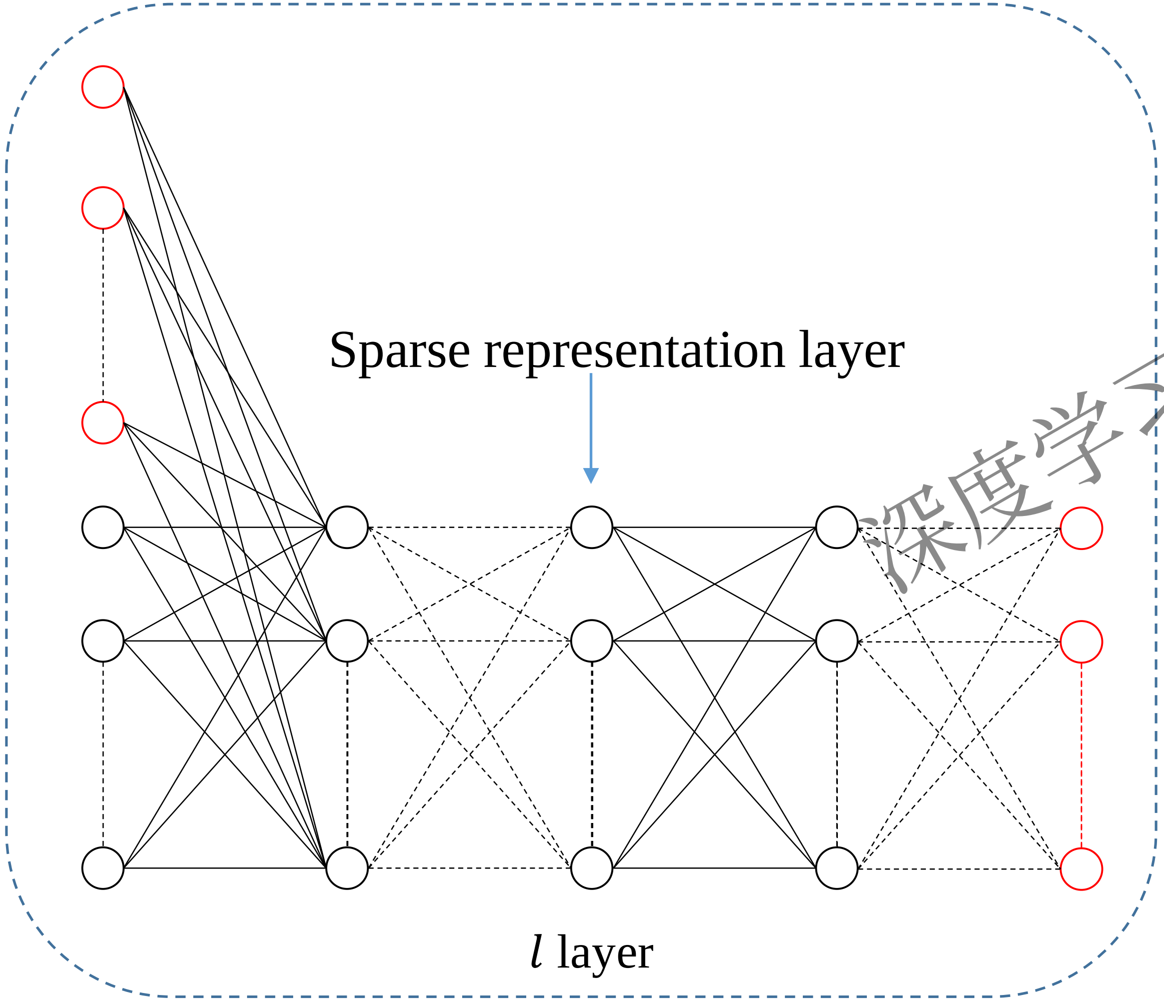
Application to Supervised Learning



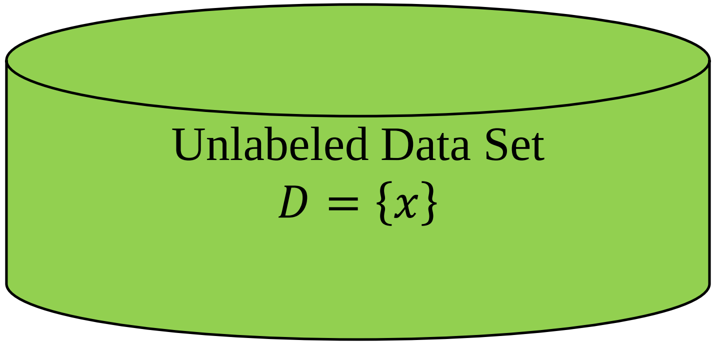
Application to Supervised Learning



Application to Supervised Learning

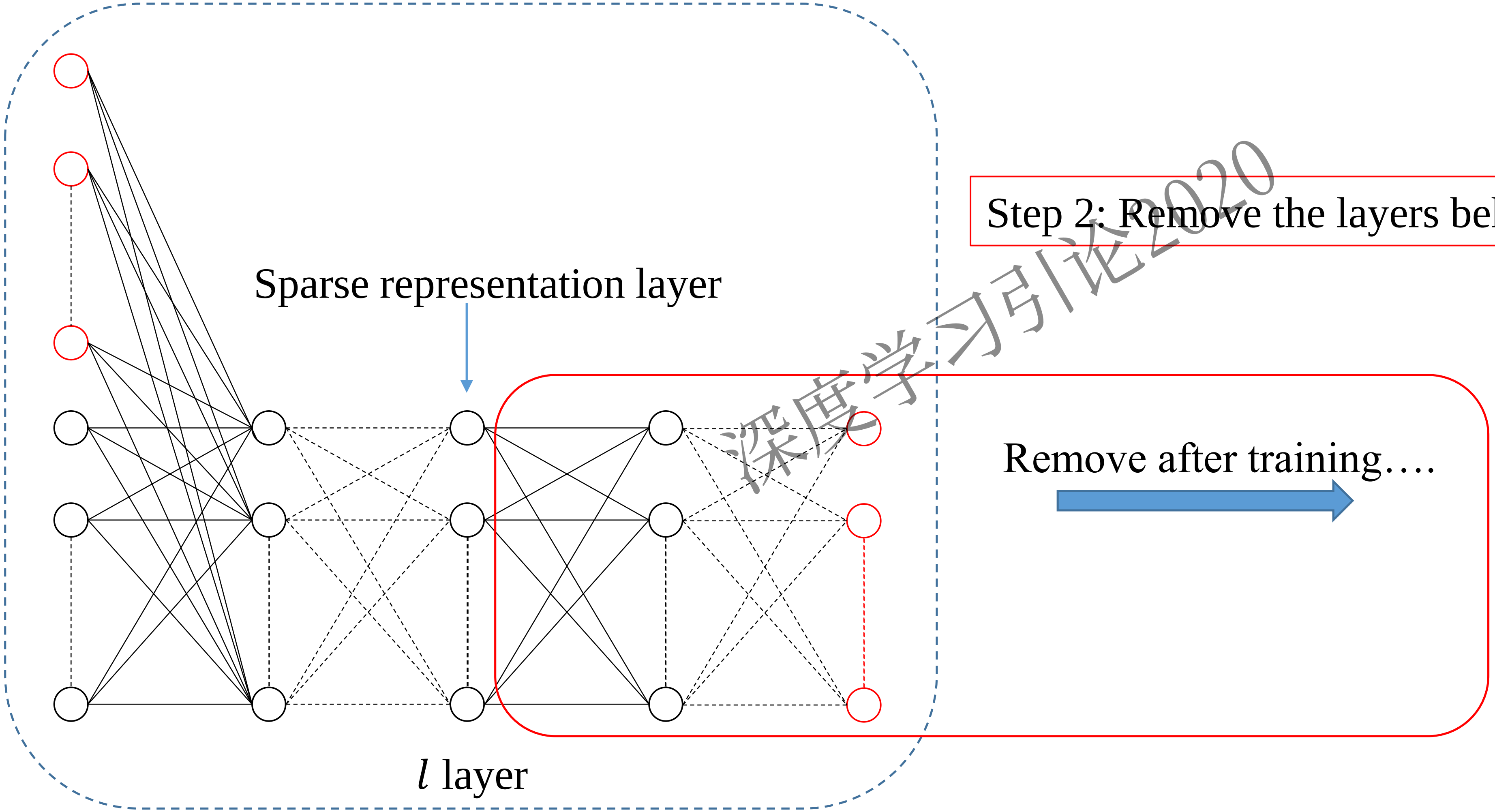


Step 1: Train the autoencoder by using unlabeled data.

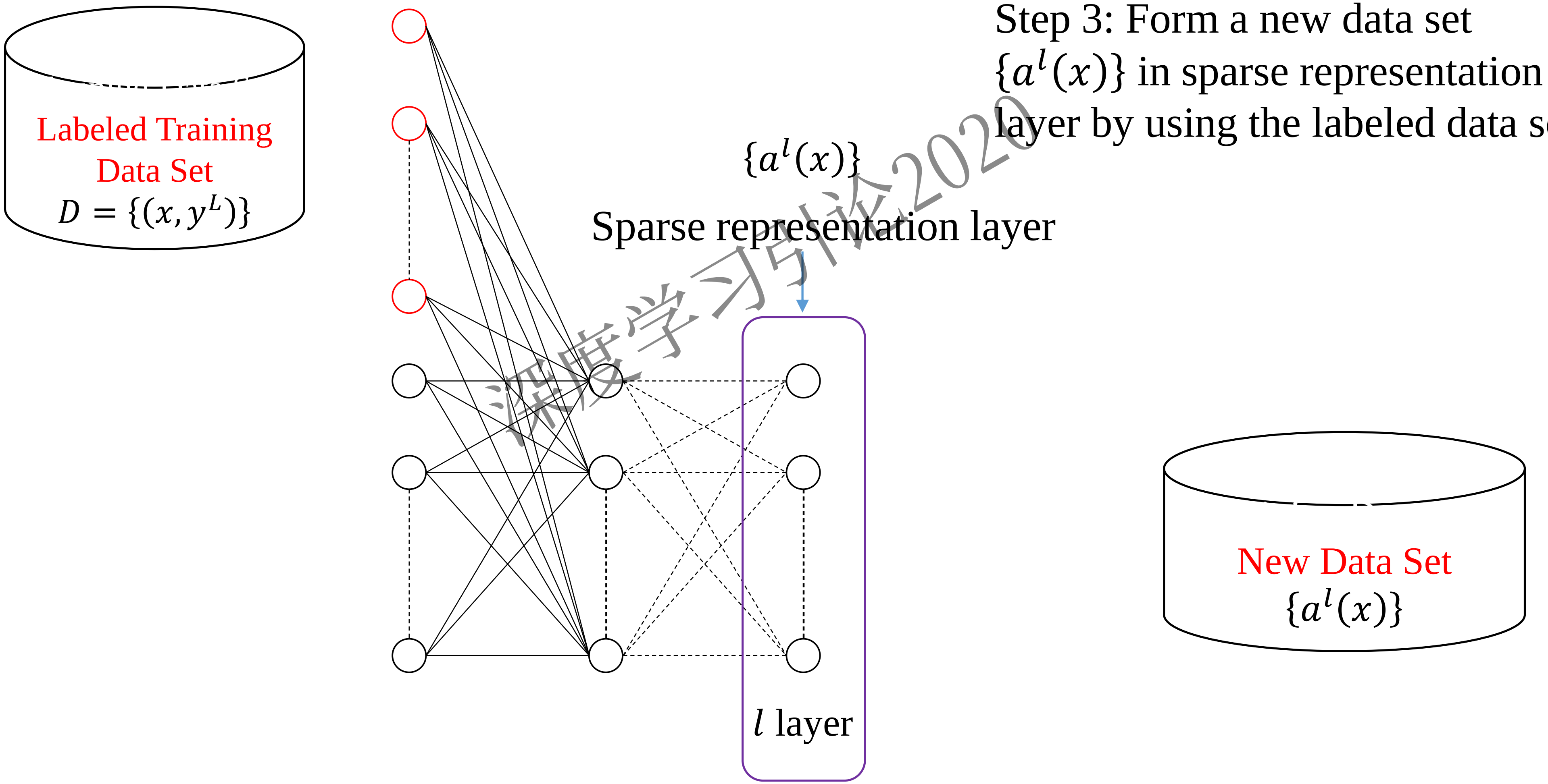


Application to Supervised Learning

Step 2: Remove the layers behind l^{th} layer after training.



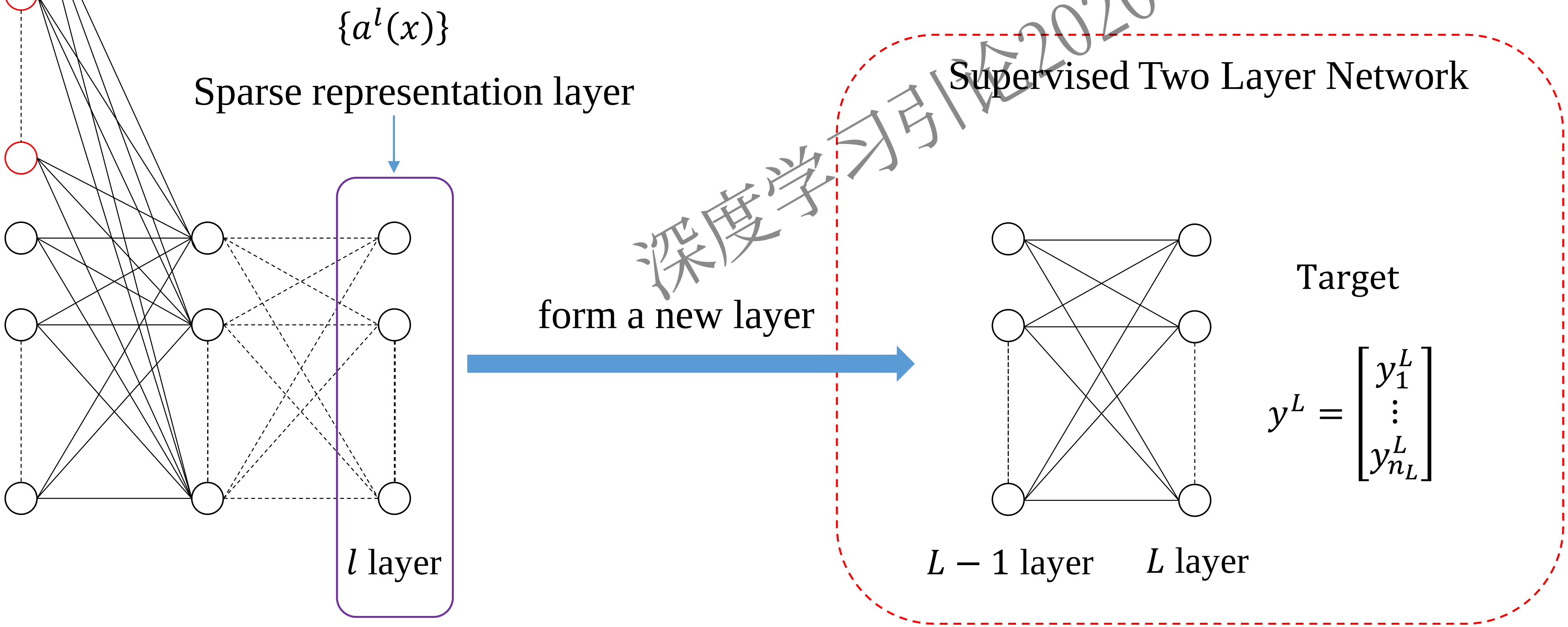
Application to Supervised Learning



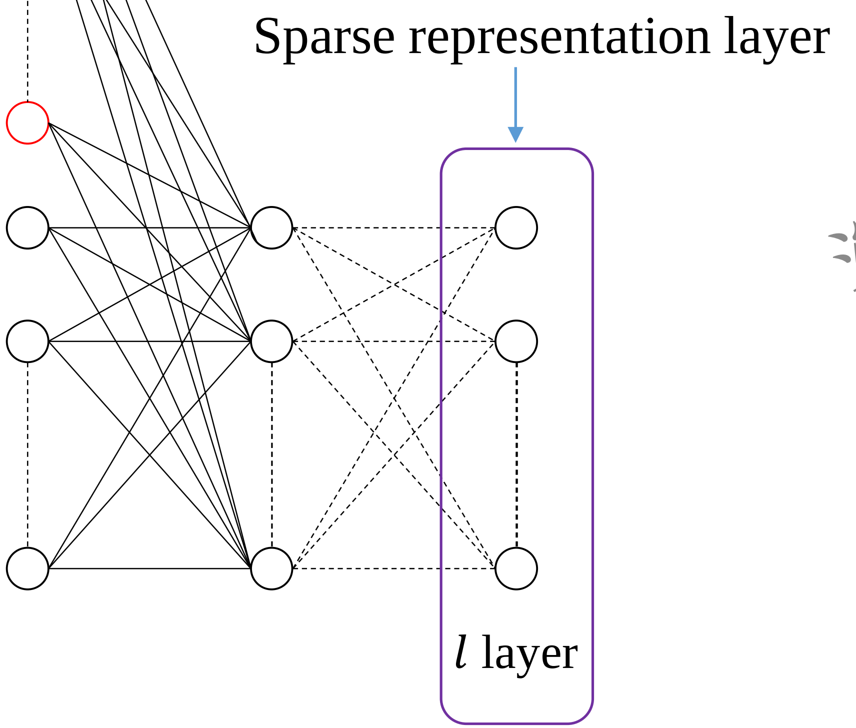
Step 3: Form a new data set $\{a^l(x)\}$ in sparse representation layer by using the labeled data set.

Application to Supervised Learning

Step 4: Form a new training data set $D_{new} = \{(a^l(x), y^L)\}$ for supervised network.



Application to Supervised Learning

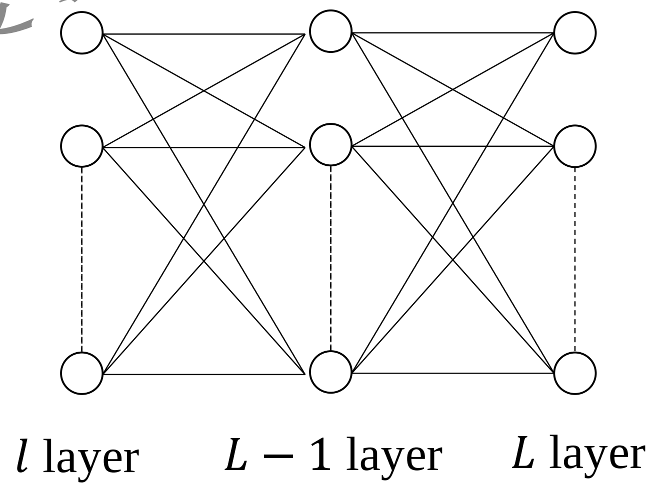


Step 5: Training the network by using the new data set

$$D_{new} = \{(a^l(x), y^L)\}$$

Supervised Three Layers Network

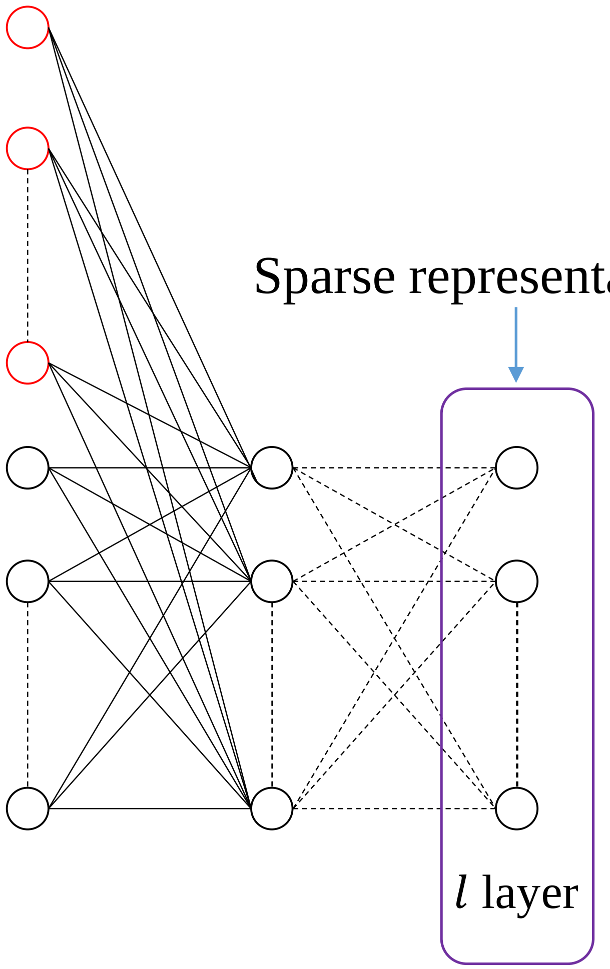
It contains $n_l \times n_{L-1} + n_{L-1} \times n_L$ parameters.



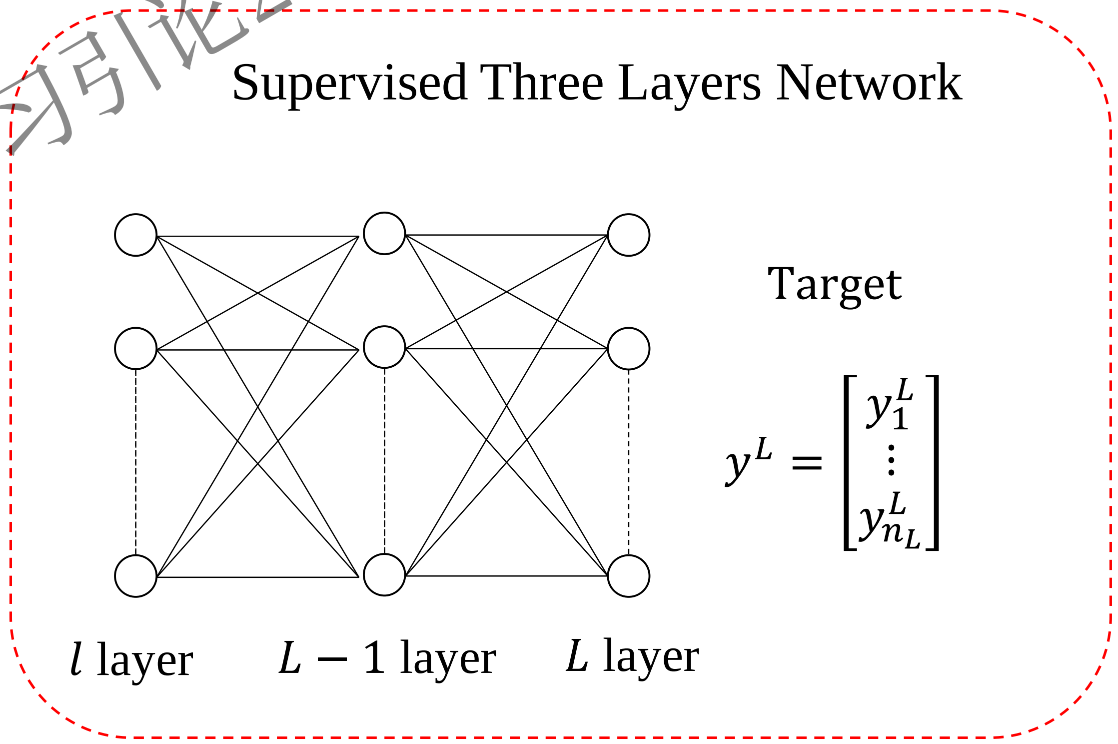
Target

$$y^L = \begin{bmatrix} y_1^L \\ \vdots \\ y_{n_L}^L \end{bmatrix}$$

Application to Supervised Learning

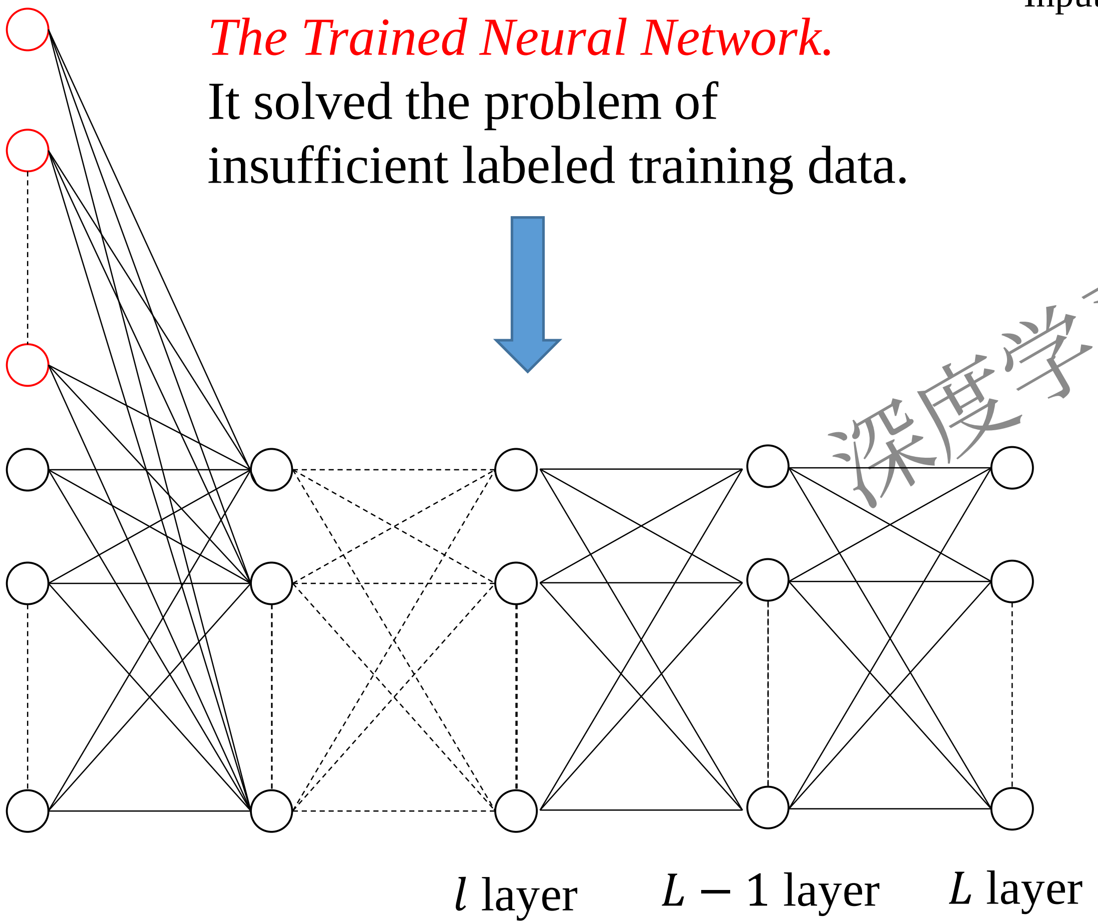


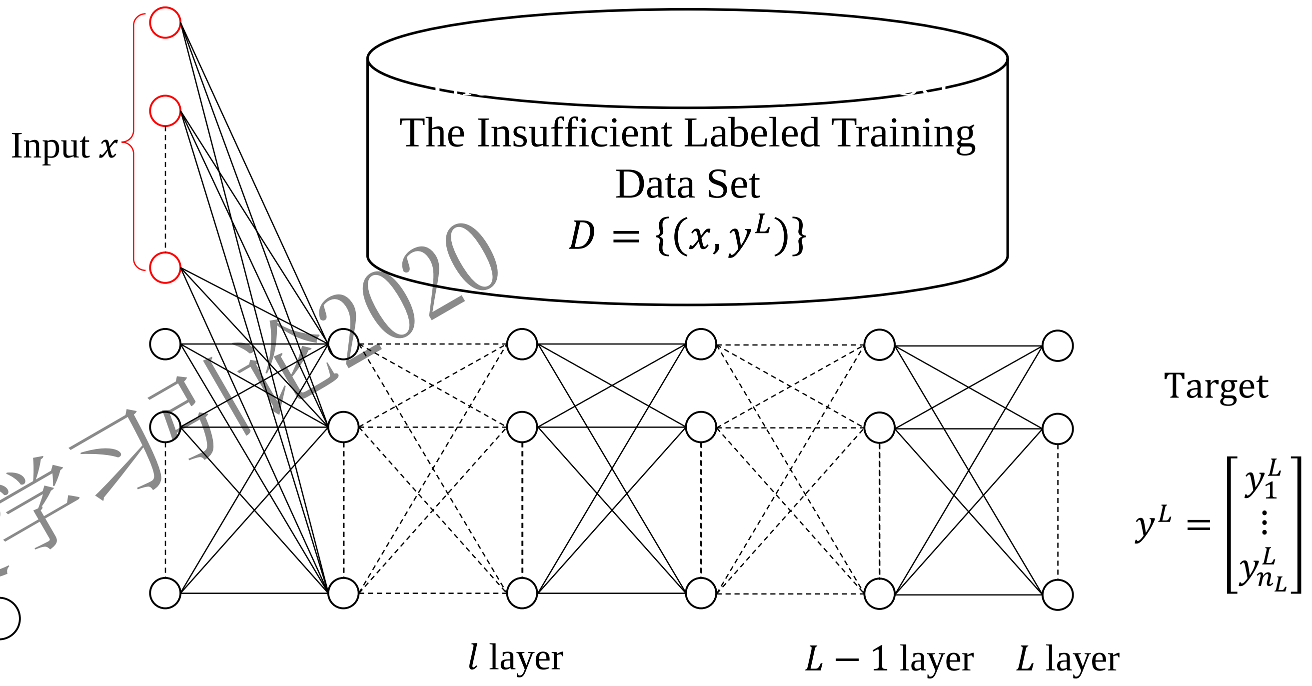
Step 6: combine the two networks.



Application to Supervised Learning

The Trained Neural Network.
It solved the problem of insufficient labeled training data.





The Original Neural Network

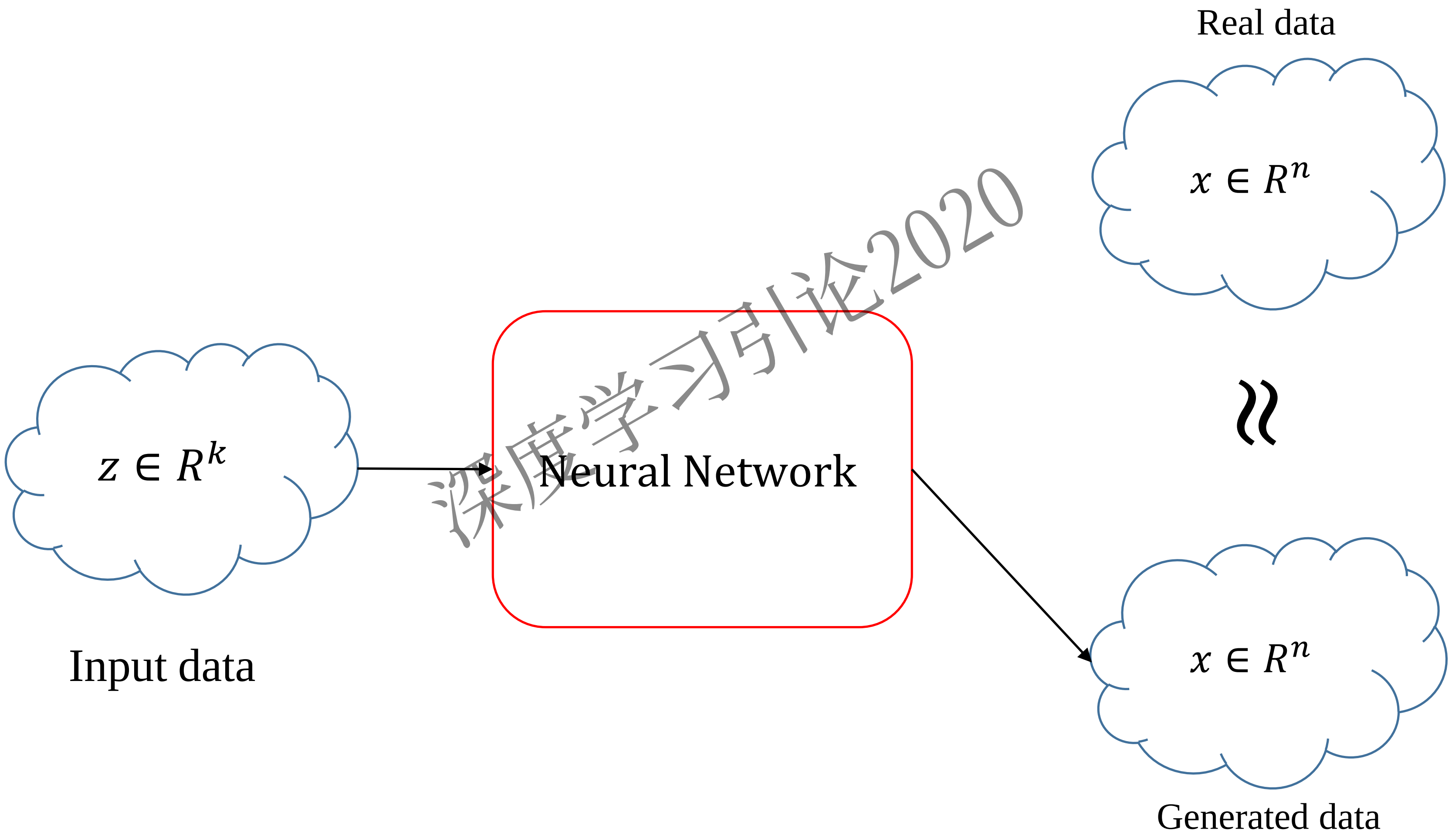
Outline

- Learning in Neural Networks
- Supervised Learning
- Unsupervised Learning
- *Autoencoder Neural Networks*
  - The Network Structure
  - Compressive Representation
  - Sparse Representation
  - Feature Learning
  - Application to Supervised Learning
- Generative Adversarial Networks
- Assignment

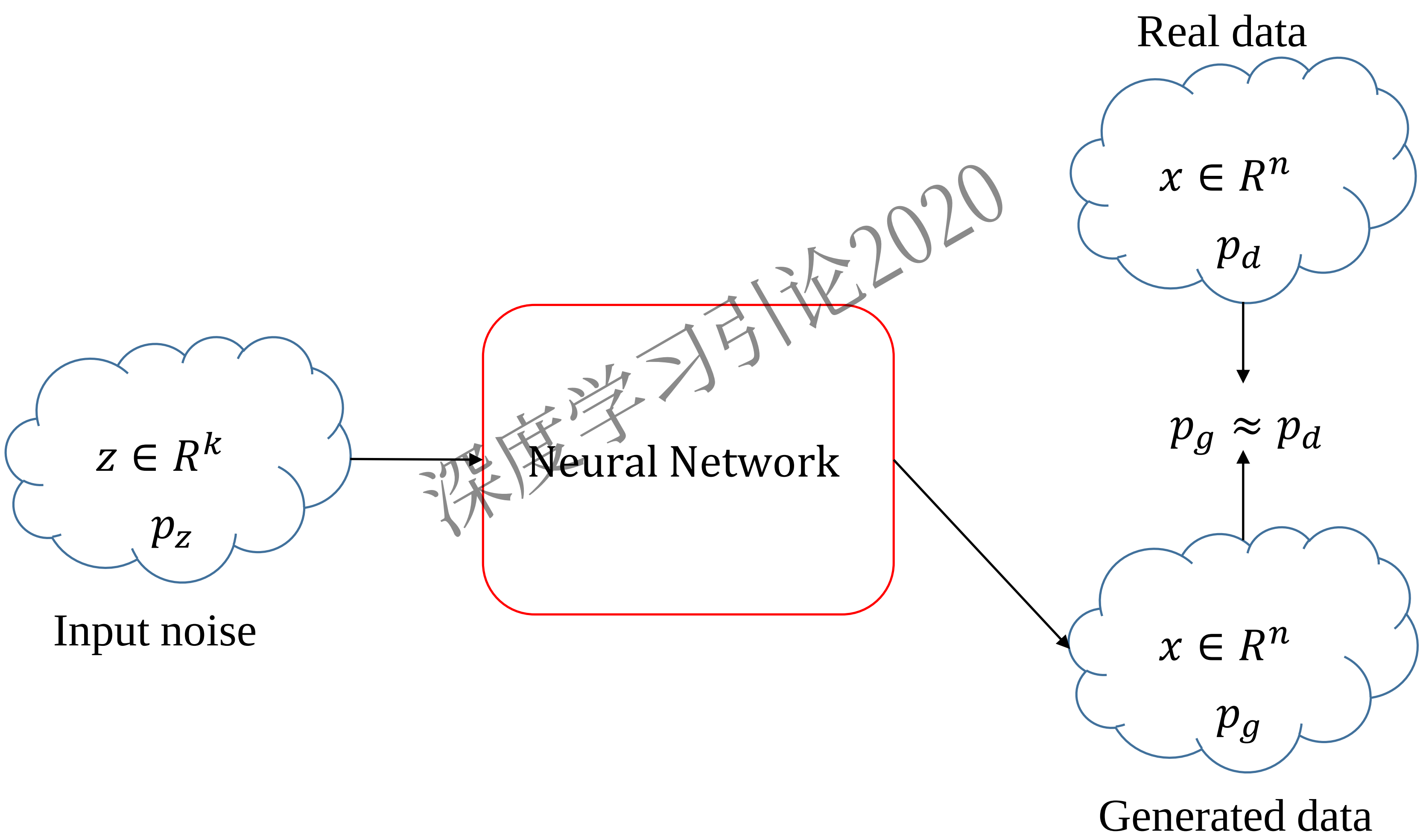
Generative Adversarial Networks

Can we generate similar images by using neural networks?

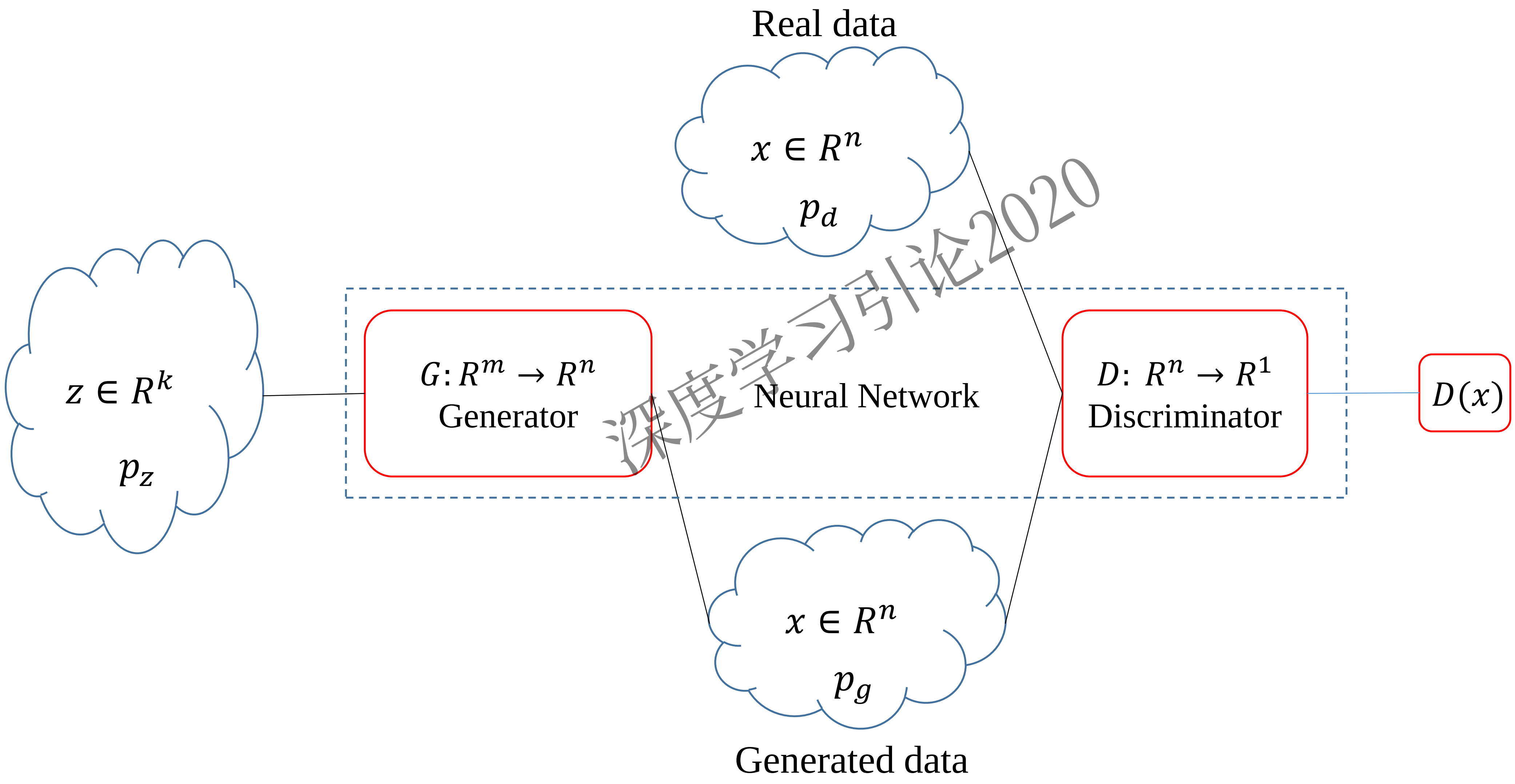
Generative Adversarial Network



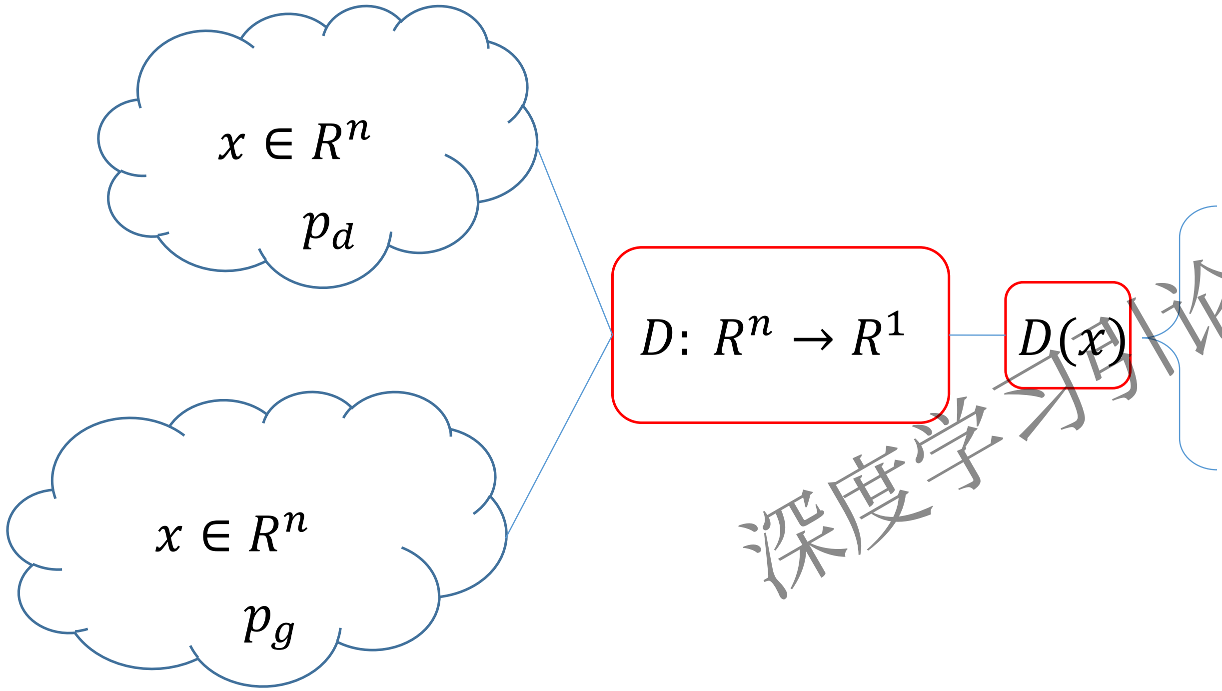
Generative Adversarial Network



Generative Adversarial Network



Discriminative Model



$$\max_D E_{x\sim p_d}[logD(x)]$$

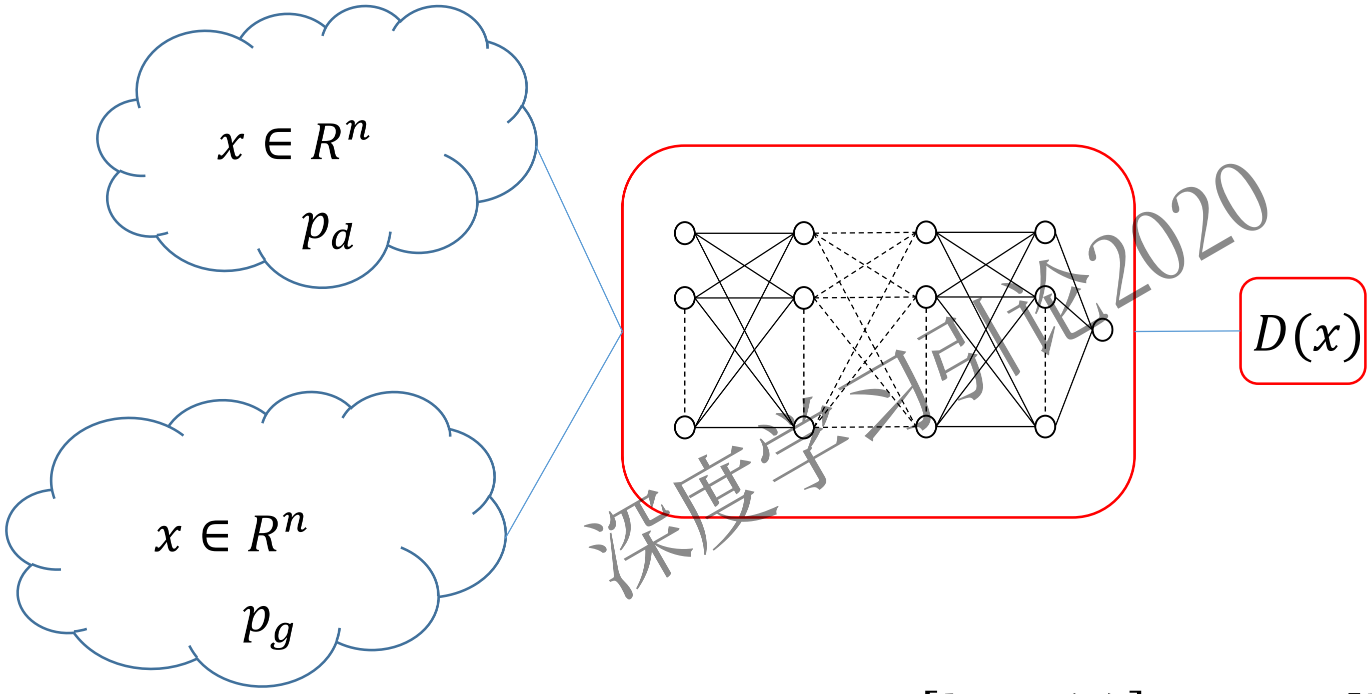
1 $\max_D D(x) \Leftrightarrow \max_D logD(x)$

0 $\min_D D(x) \Leftrightarrow \max_D log(1 - D(x))$

$$\max_D E_{x\sim p_g}[log\big(1 - D(x)\big)]$$

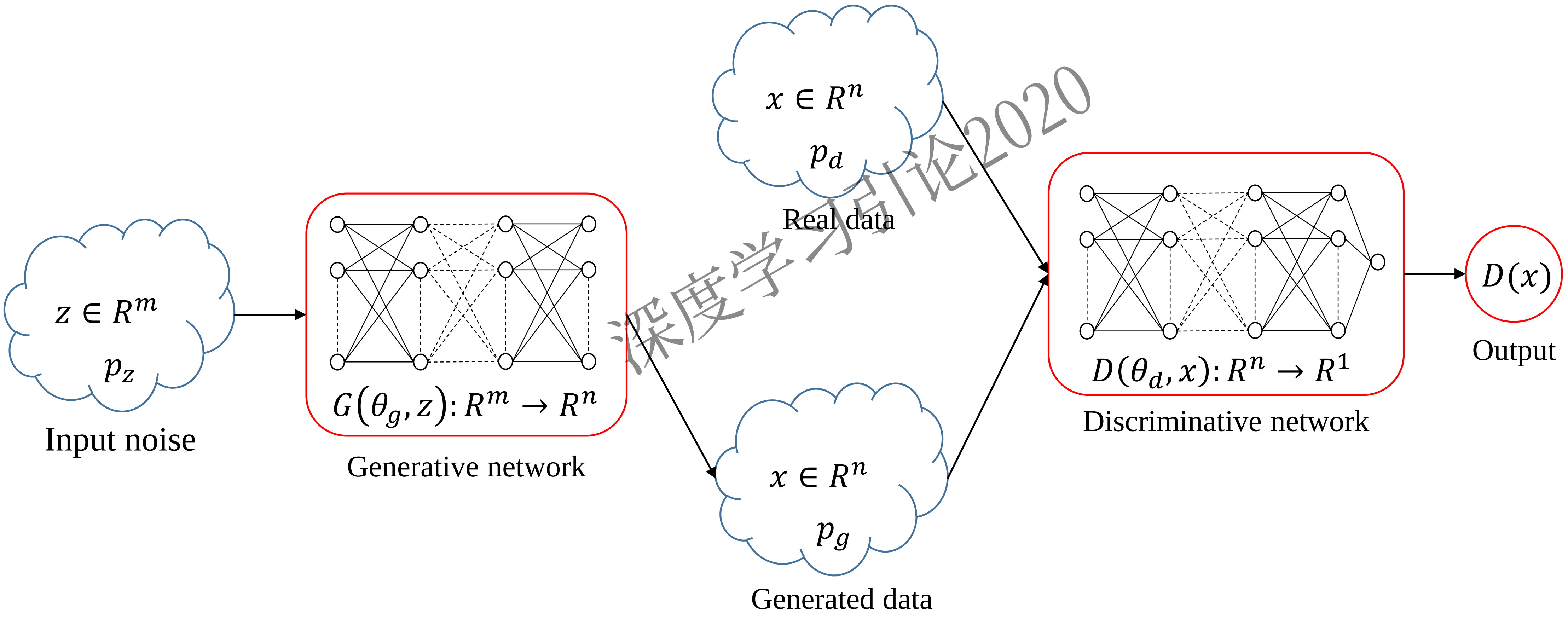
$$\max_D E_{x\sim p_d}[logD(x)] + E_{x\sim p_g}[log\big(1 - D(x)\big)]$$

Discriminative Model

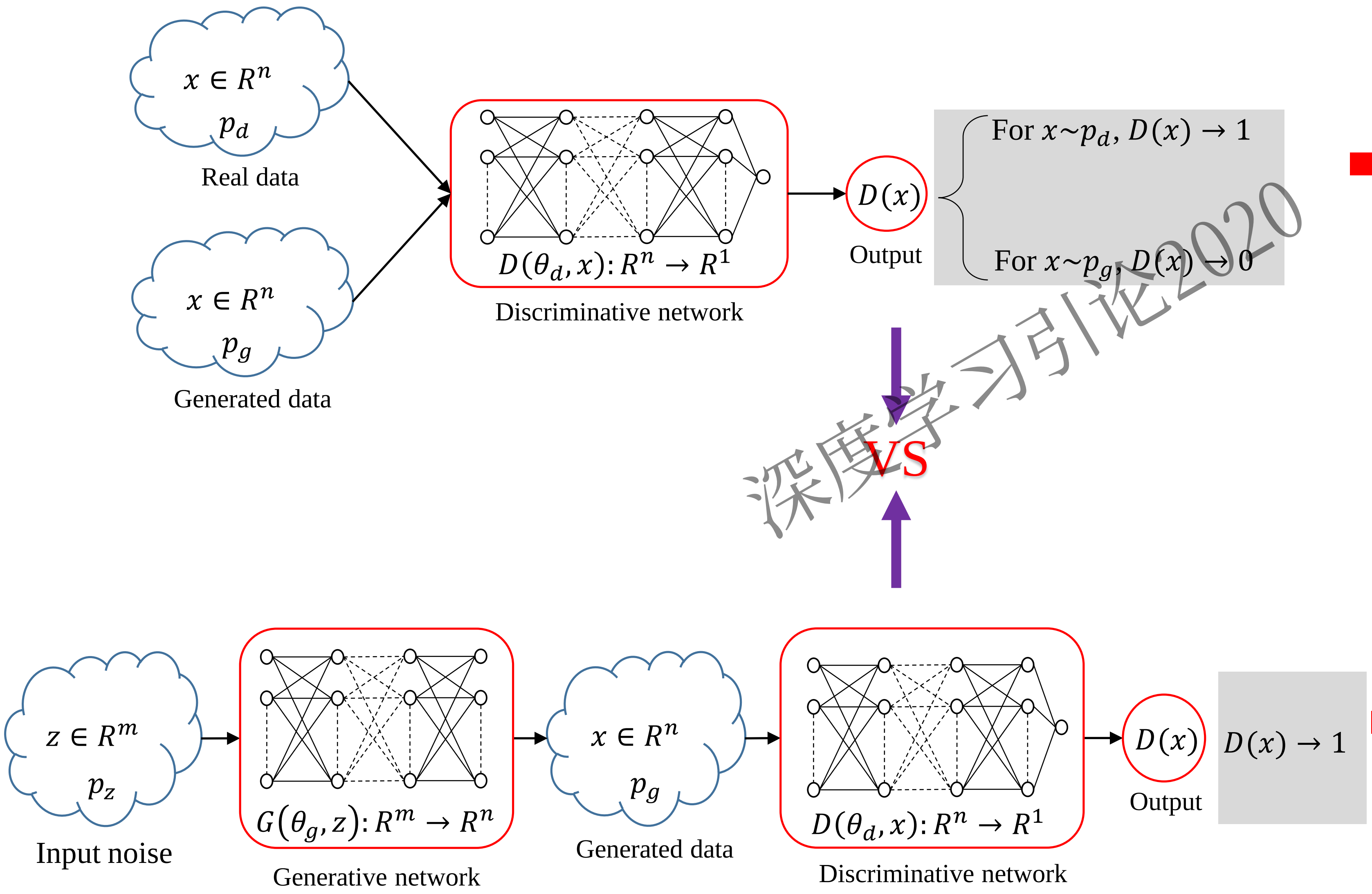


$$\max_D E_{x\sim p_d}[logD(x)] + E_{x\sim p_g}[log\big(1 - D(x)\big)]$$

Generative Adversarial Network



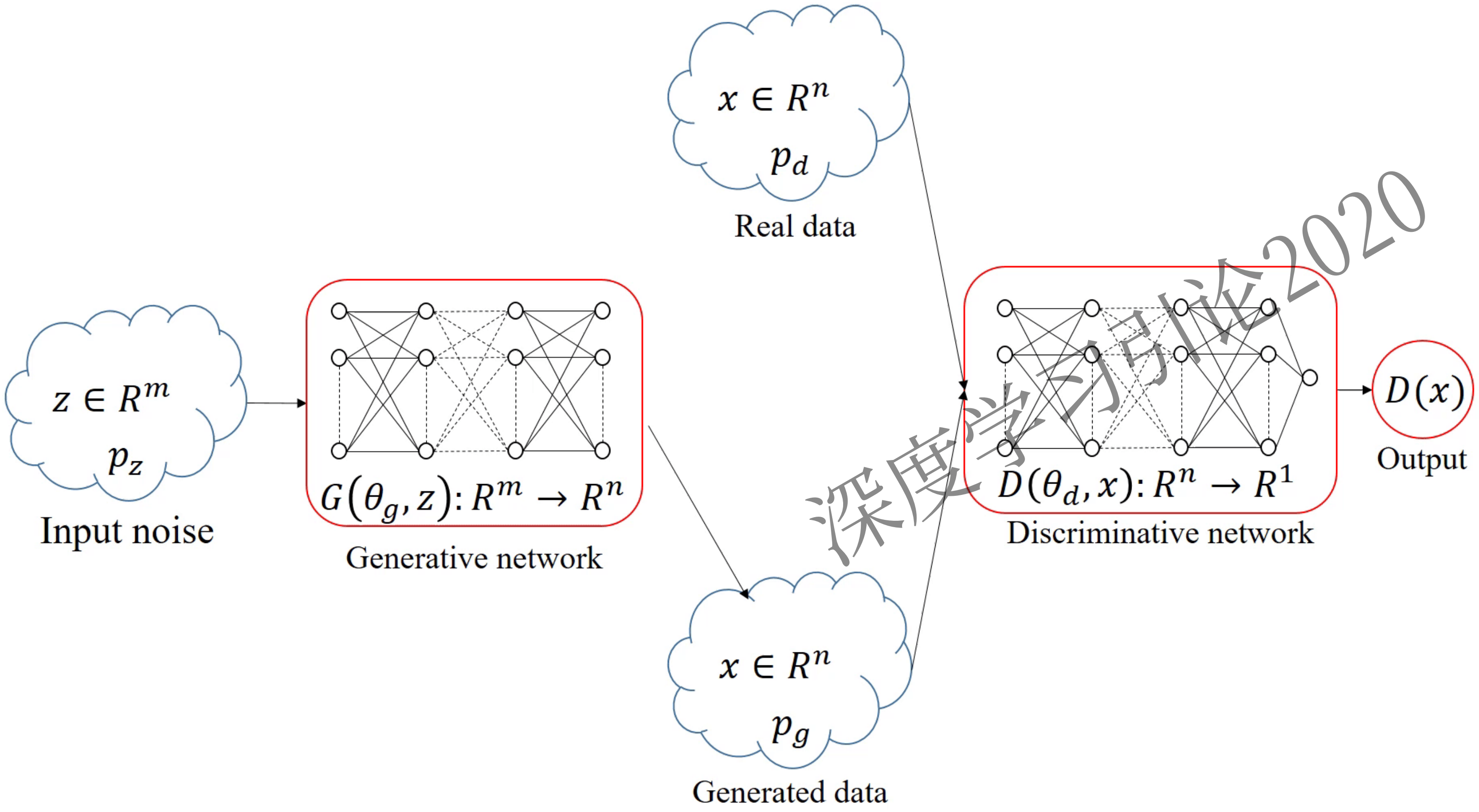
Generative Adversarial Network



- Discriminative network (D) aims to discriminate the real and generated data

- Generative network (G) aims to cheat the discriminative network

Generative Adversarial Networks



Algorithm

Step 1. Sample t noise samples $\{z^1, \dots, z^t\}$ from p_z

Step 2. Generate t samples $\{x^1, \dots, x^t\}$ from p_g by using $\{z^1, \dots, z^t\}$

Step 3. Update discriminative network

$$\nabla_{\theta_d} \frac{1}{t} \sum_{i=1}^{t} \left[log D(x^i) + log \left(1 - D\left(G(z^i)\right)\right)\right]$$

Step 4. Sample t noise samples $\{z^1, \dots, z^t\}$ from p_z

Step 5. Update generative network

$$\nabla_{\theta_g} \frac{1}{t} \sum_{i=1}^{t} log \left(1 - D\left(G(z^i)\right)\right)$$

Human Face Generation Examples

Original images

Generated images

Outline

- Learning in Neural Networks
- Supervised Learning
- Unsupervised Learning
- *Autoencoder Neural Networks*
  - The Network Structure
  - Compressive Representation
  - Sparse Representation
  - Feature Learning
  - Application to Supervised Learning
- Generative Adversarial Networks
- Assignment

Assignment 1

1. Given the cost function

$$J = \frac{1}{2}\sum_{i=1}^{m}\left(a_i^L - x_i\right)^2 + \beta \cdot \sum_{i=1}^{n_l} a_i^l$$

Prove that

$$\delta_i^l = \dot{f}\left(z_i^l\right) \cdot \left(\sum_{j=1}^{n_{l+1}} w_{ji}^l \delta_j^{l+1} + \beta\right)$$

2. Given the optimization problem

$$\begin{cases} \max \sum_{i=1}^{n} w_{ij}^{(1)} x_j \\ s.t. \sum_{i=1}^{n} x_i^2 \le 1 \end{cases}$$

Prove that

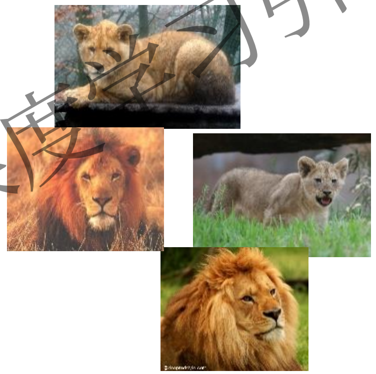
$$x_j = \frac{w_{ij}^{(1)}}{\sqrt{\sum_{j=1}^{n}\left(w_{ij}^{(1)}\right)^2}}, (j = 1, \cdots, n)$$

Assignment 2

Assignment:

In this example, the labeled training data are insufficient to train a classifier by using BP directly. However, a good classifier can be developed by using the autoencoder method. Please do it.

Unlabeled data



Label: Lion : 0

Label: Tiger : 1

Q & A

The End