深度学习引论

章 毅，张 蕾，郭 泉

四川大学·计算机学院·人工智能系

机器智能实验室

http://www.machineilab.org/

深度学习引论

第六章

深度神经网络结构

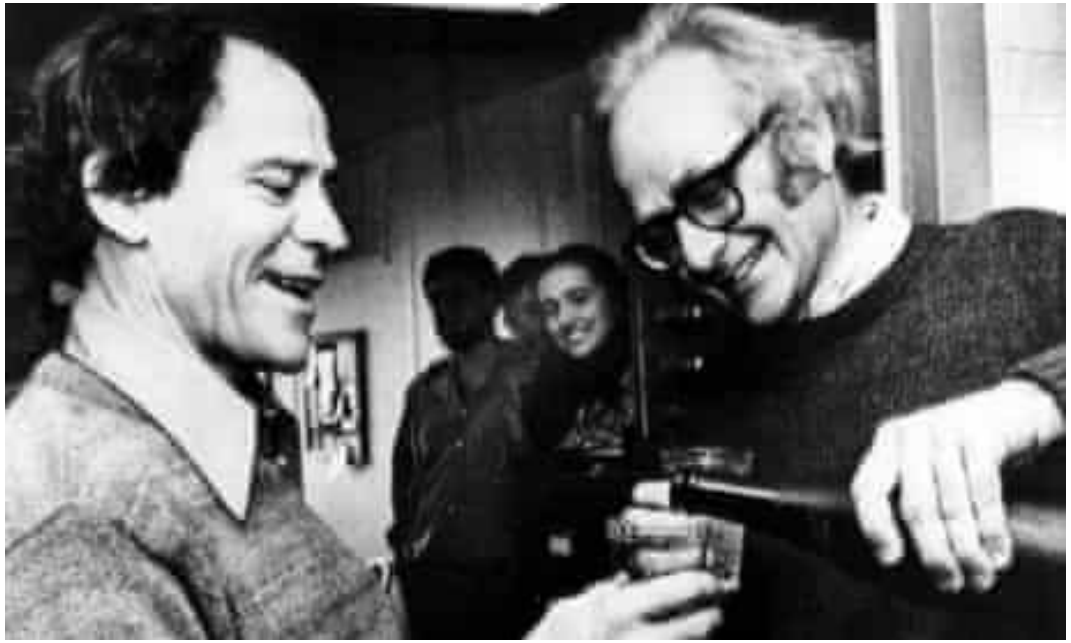
2022年 秋季

提纲

视觉：卷积神经网络

序列：多输出神经网络

视觉：卷积神经网络



Recipient of the Nobel Prize in Physiology or Medicine, 1981

Torsten Wiesel

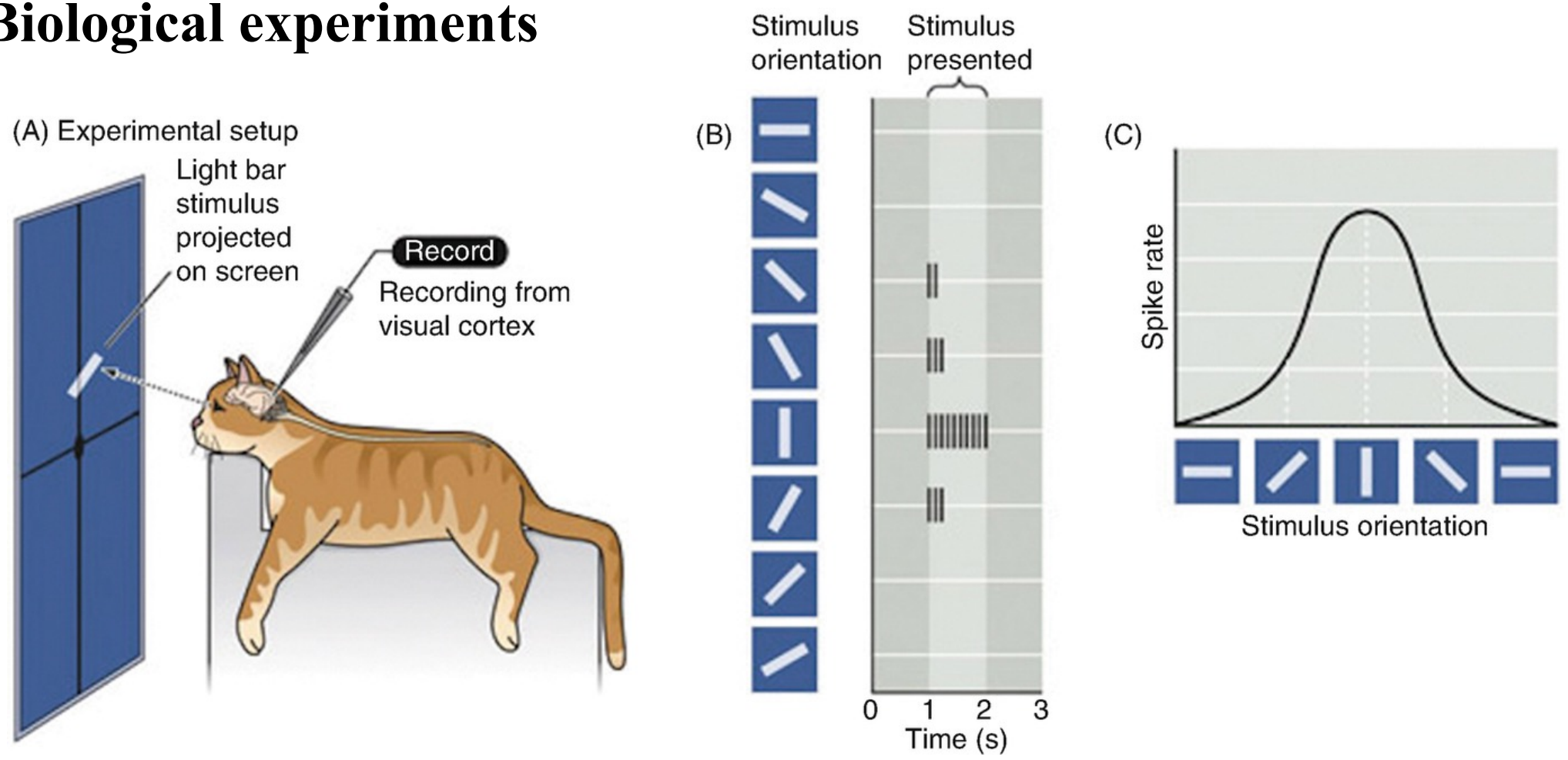
Professor of neurobiology
Harvard Medical School

David H. Hubel

Professor of Neurobiology
Johns Hopkins University and Harvard Medical School

视觉：卷积神经网络

Biological experiments

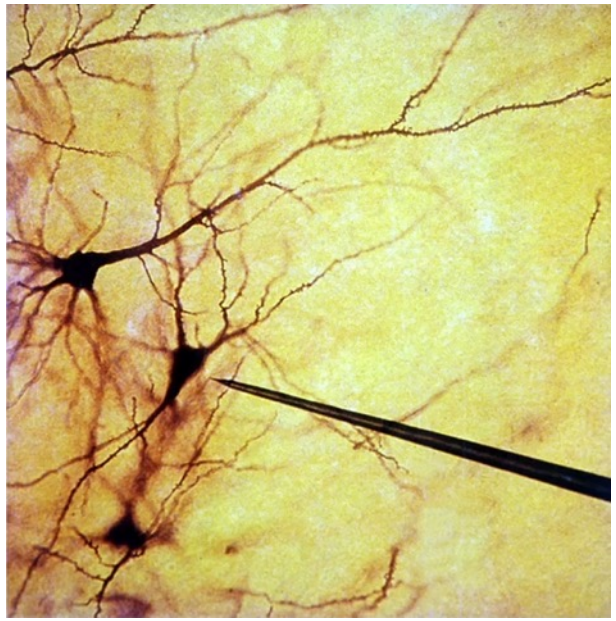


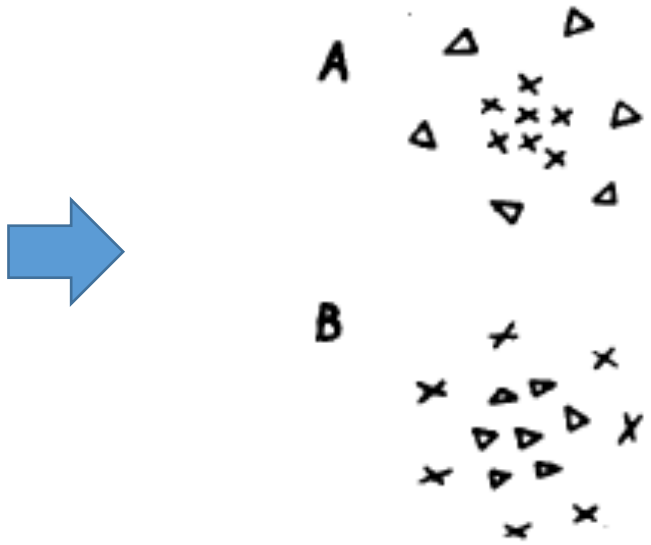
Hubel, David H., and Torsten N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." *The Journal of physiology*, **1962**

视觉：卷积神经网络

Receptive Fields (Light-sensitive area) of a cell

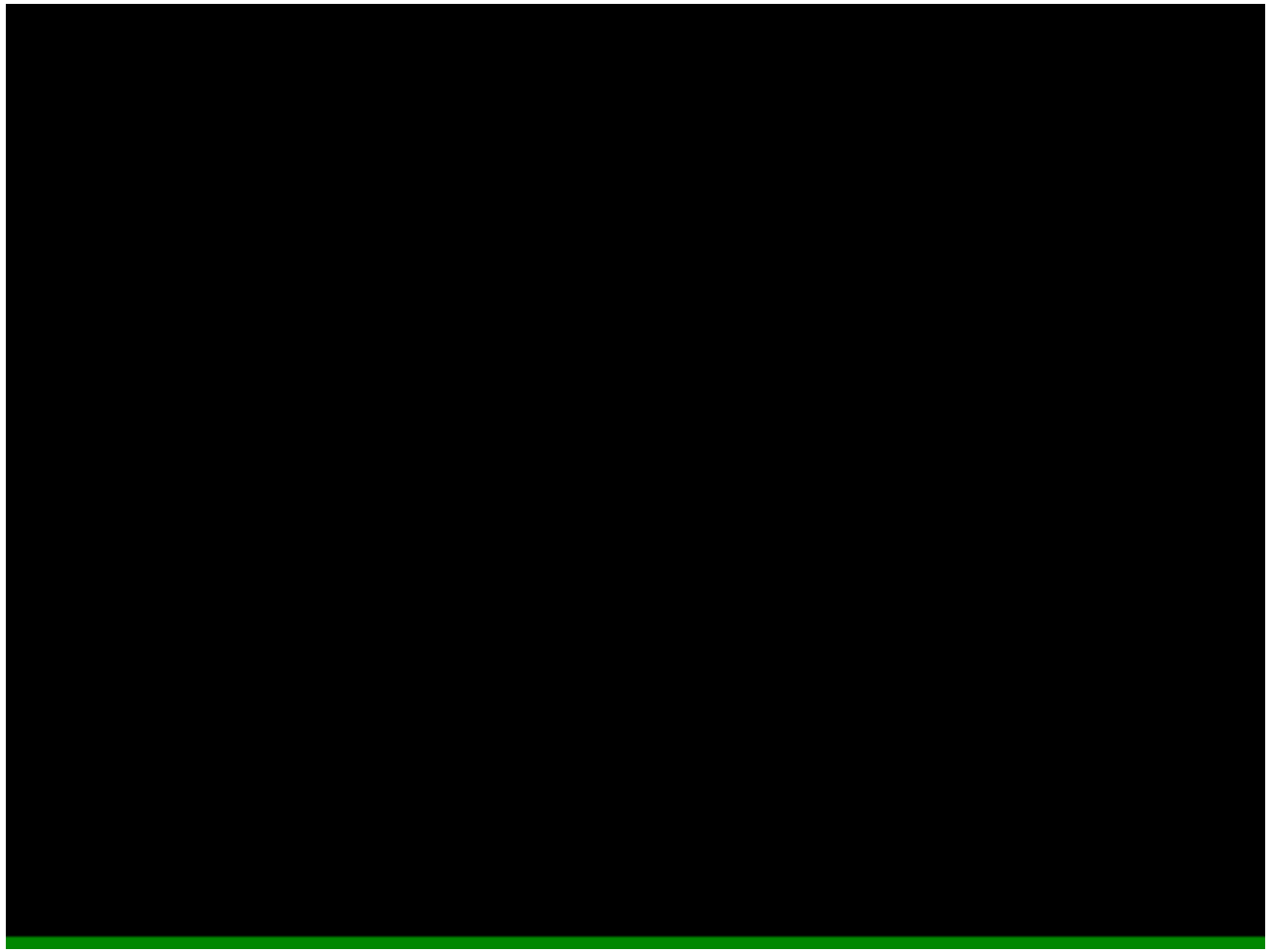
1. Area of retina that affects firing rate of a given neuron in the circuit
2. Receptive fields are determined by monitoring single cell responses
3. Stimulus is presented to retina and response of cell is measured by an electrode





视觉：卷积神经网络

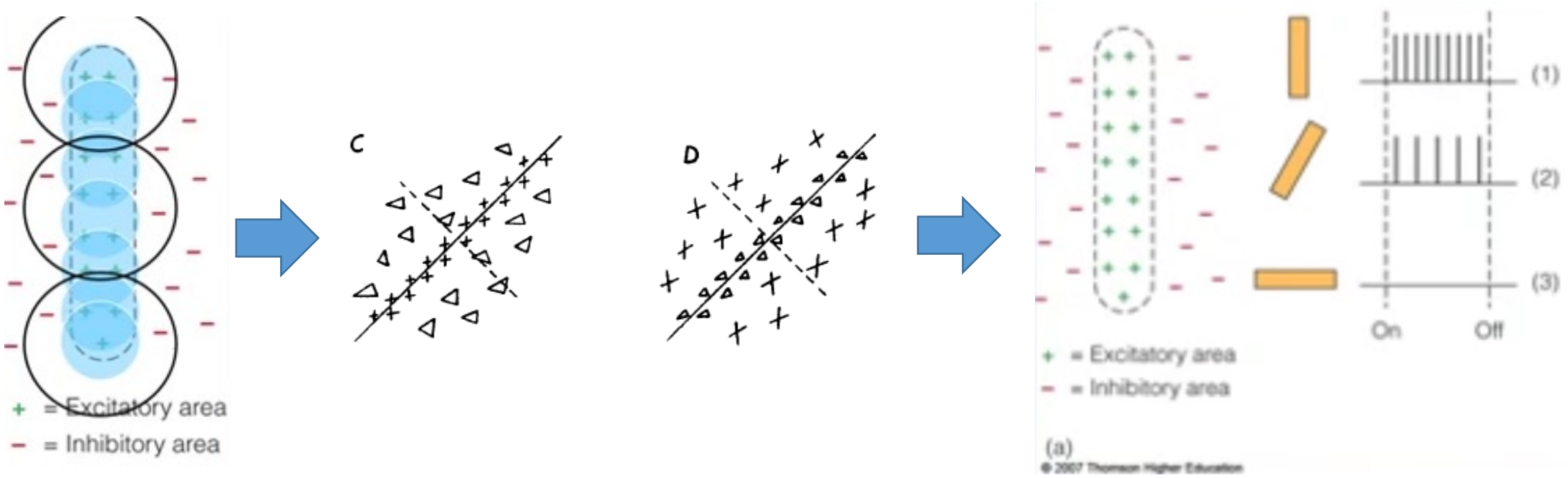
Biological experiments: simple cell



视觉：卷积神经网络

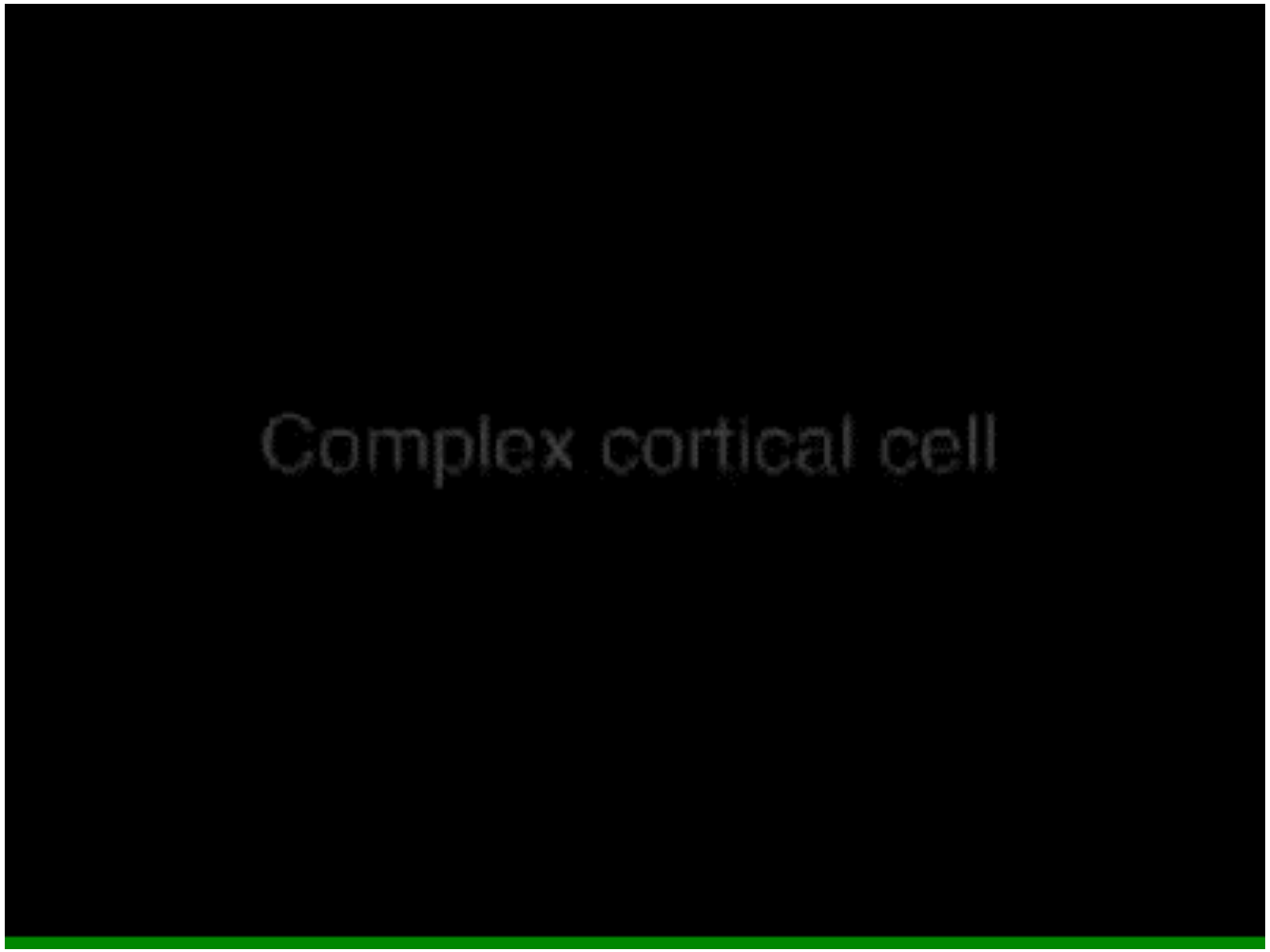
Simple Cells

1. Side-by-side excitatory and inhibitory receptive fields
2. Result of center-surround antagonism
3. Respond best (or least) to a bar of light in a specific orientation



视觉：卷积神经网络

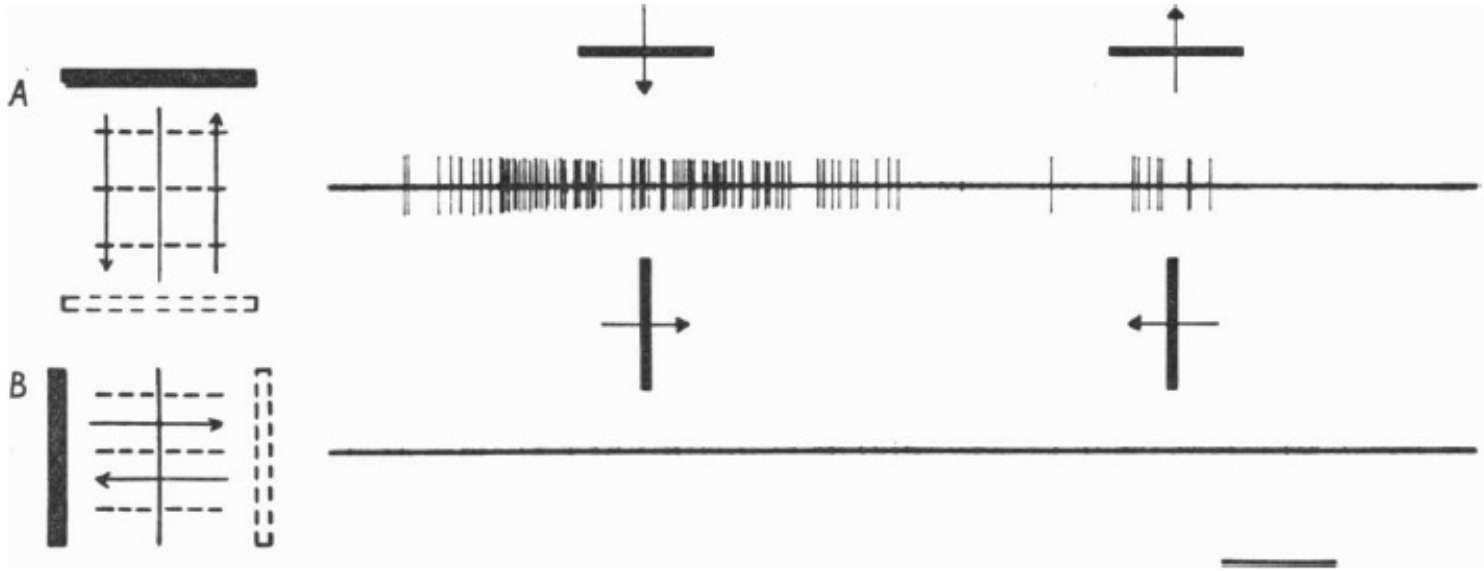
Biological experiments: complex cell



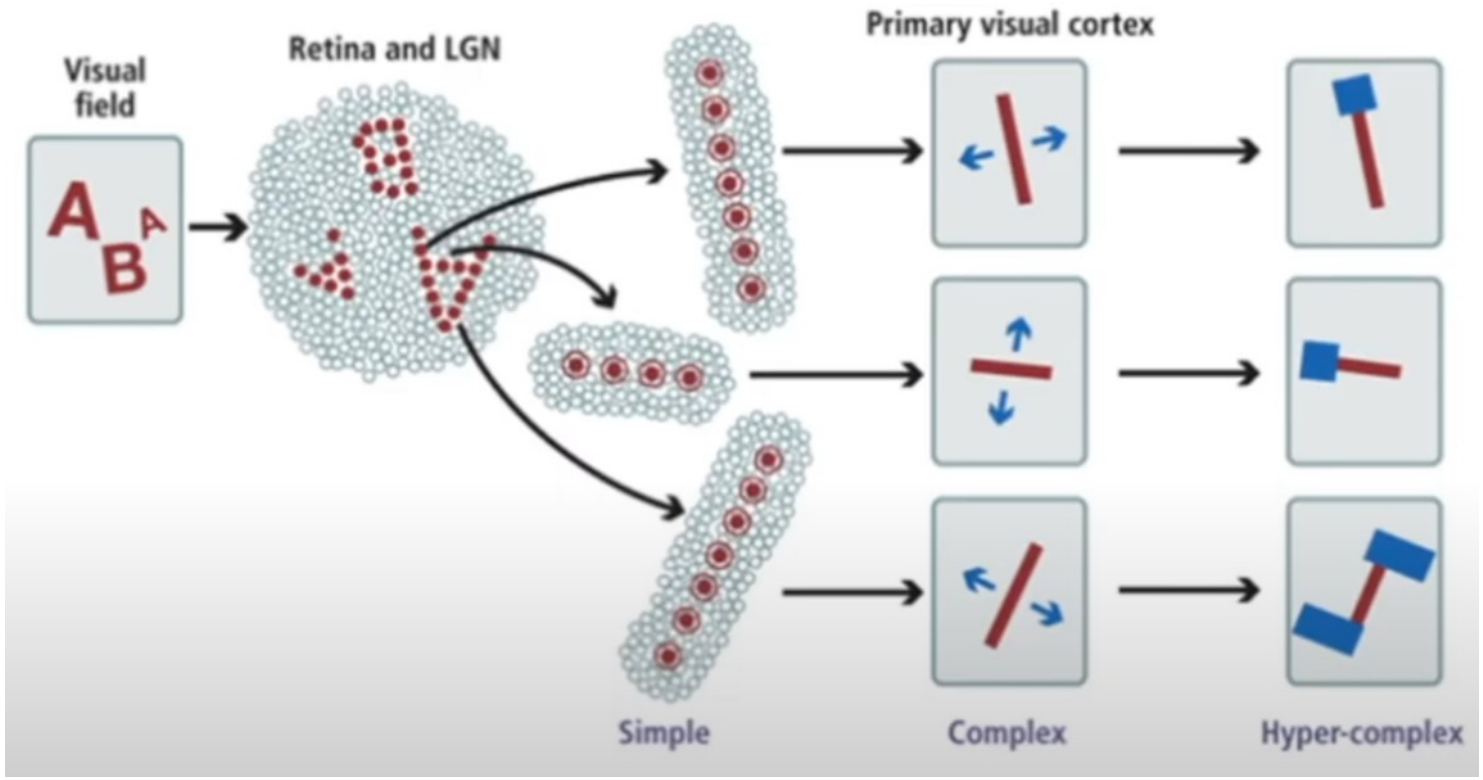
视觉：卷积神经网络

Complex Cells

1. Respond best to correctly oriented bars of light
2. Respond most when such a stimulus is moving across the cell's receptive field
3. Many respond best to movement in a particular direction



视觉：卷积神经网络


Visual field
Retina and LGN
Primary visual cortex
Simple
Complex
Hyper-complex

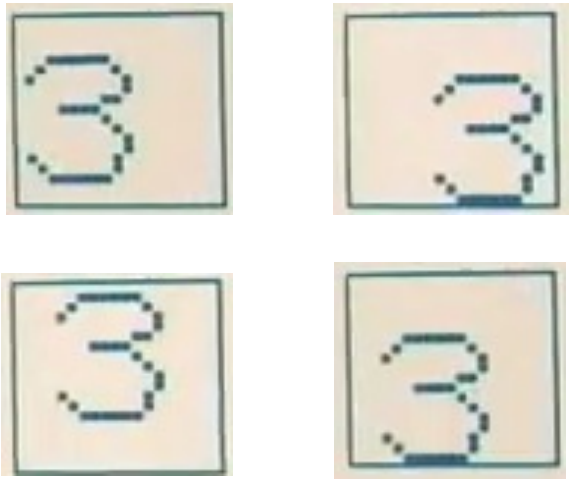

Edges
Simple shapes
Complex shapes
Faces and objects

视觉：卷积神经网络

福島 邦彦
（1936-）

Neural Networks Pioneer Award from IEEE in 2003
2021 Bower Award

Neocognitron



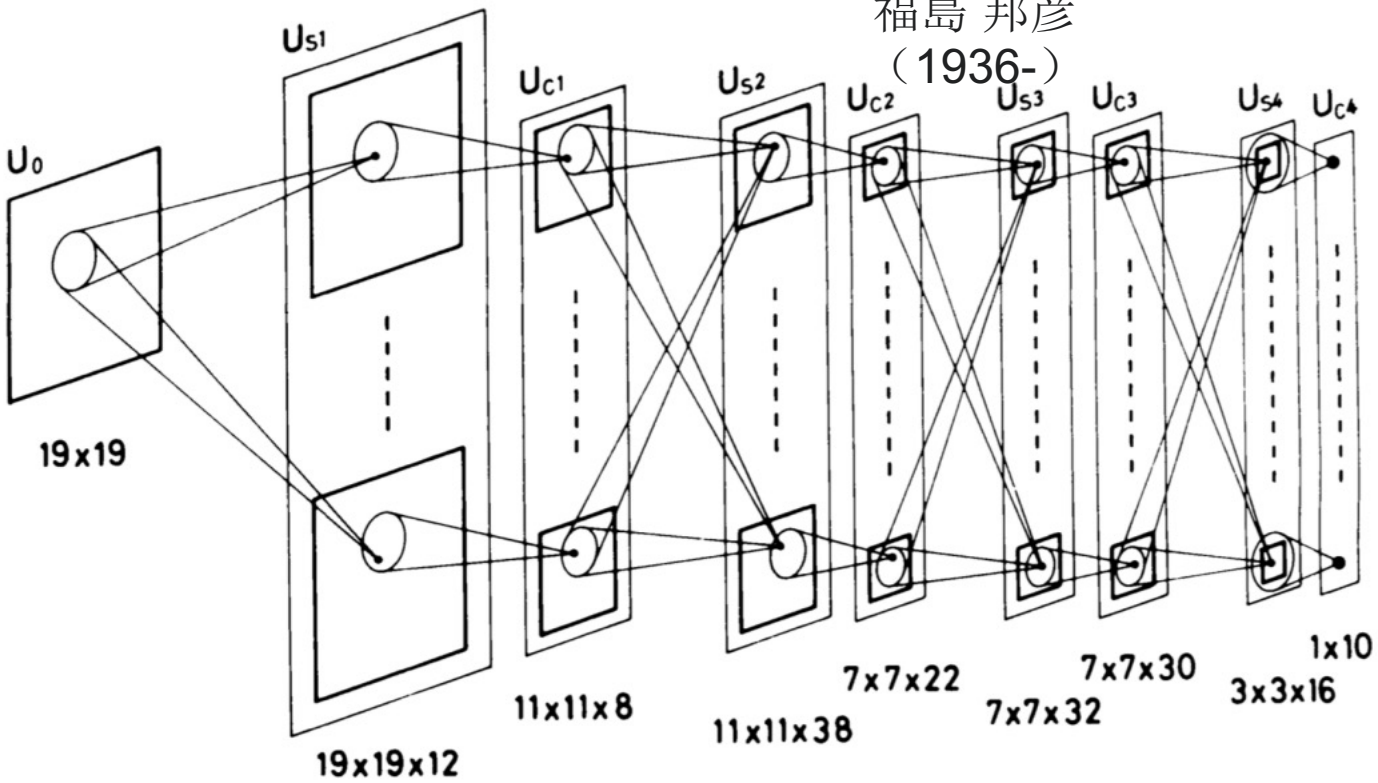


Fig. 2. Schematic diagram illustrating synaptic connections between layers in neocognitron.

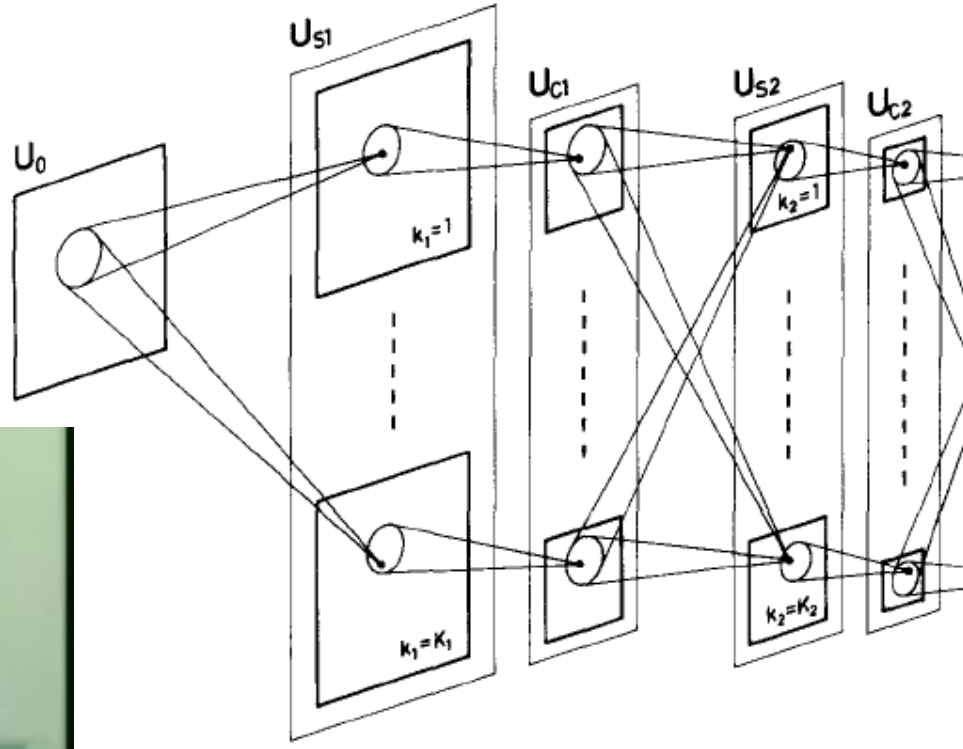
3

[1] **Fukushima, Kunihiko.** "Cognitron: A self-organizing multilayered neural network." *Biological cybernetics*, **1975**

[2] **Fukushima, Kunihiko**. "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position". *Biological cybernetics* **1980**

视觉：卷积神经网络

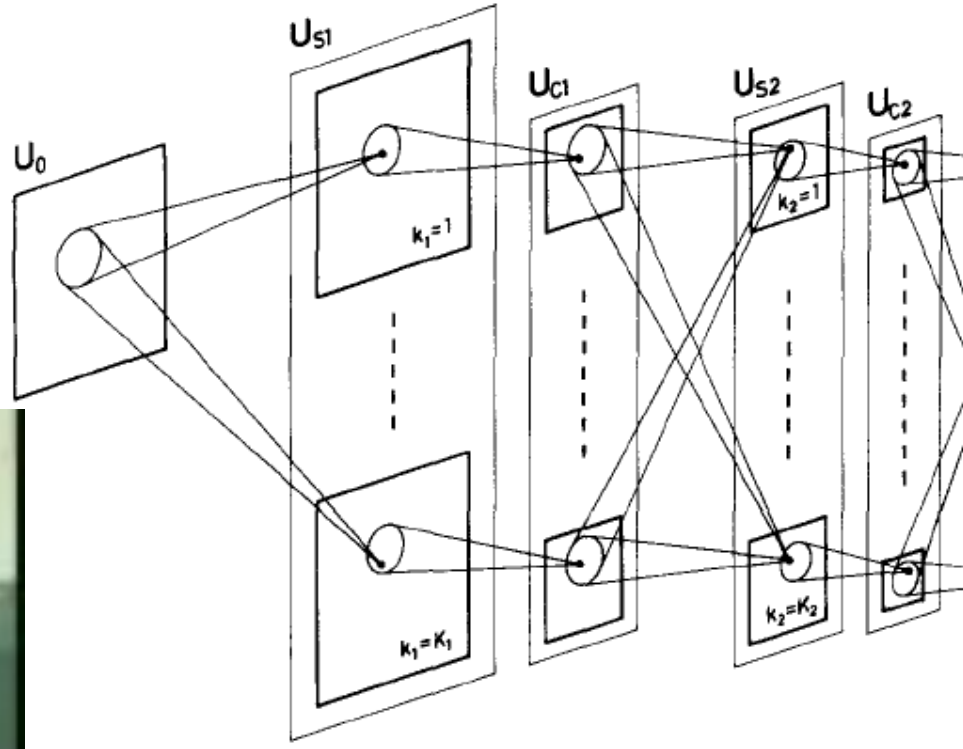
Application: pattern recognition (1980)



Neocognitron

视觉：卷积神经网络

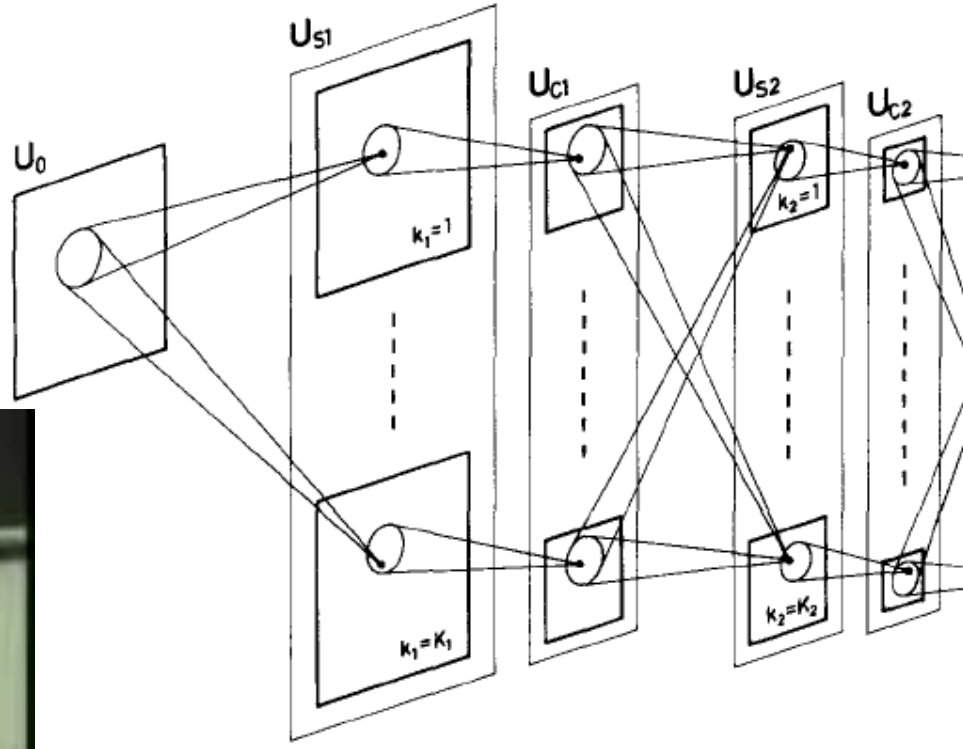
Application: position-shifted cases



Neocognitron

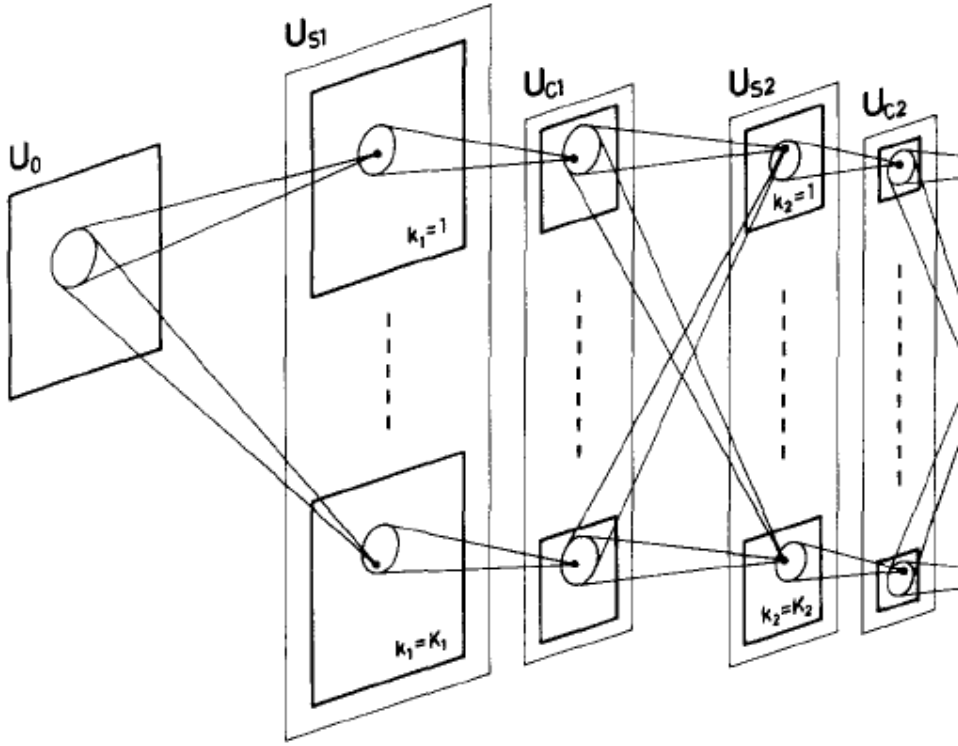
视觉：卷积神经网络

Application: deformed cases



Neocognitron

视觉：卷积神经网络



Self-organization Learning

$$\Delta a_l(k_{l-1}, \mathbf{v}, \hat{k}_l) = q_l \cdot c_{l-1}(\mathbf{v}) \cdot u_{Cl-1}(k_{l-1}, \hat{\mathbf{n}} + \mathbf{v}),$$

$$\Delta b_l(\hat{k}_l) = (q_l/2) \cdot v_{Cl-1}(\hat{\mathbf{n}}),$$

视觉：卷积神经网络

IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. 37, NO. 3, MARCH 1989

Phoneme Recognition Using Time-Delay Neural Networks

ALEXANDER WAIBEL, MEMBER, IEEE, TOSHIYUKI HANAZAWA, GEOFFREY HINTON, KIYOHIRO SHIKANO, MEMBER, IEEE, AND KEVIN J. LANG

Geoffrey Hinton

2018 ACM A.M. Turing Award

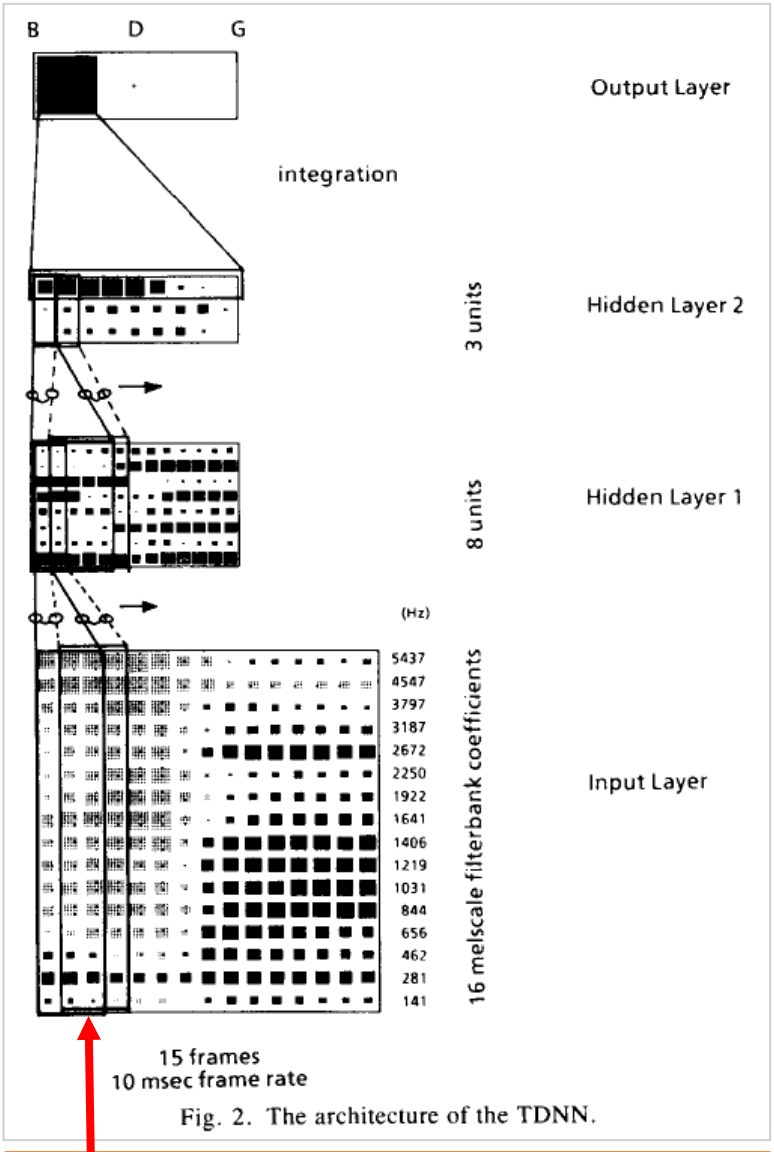


Fig. 2. The architecture of the TDNN.

The **first** convolutional neural network (CNN) trained by gradient descent, using the **backpropagation** algorithm.

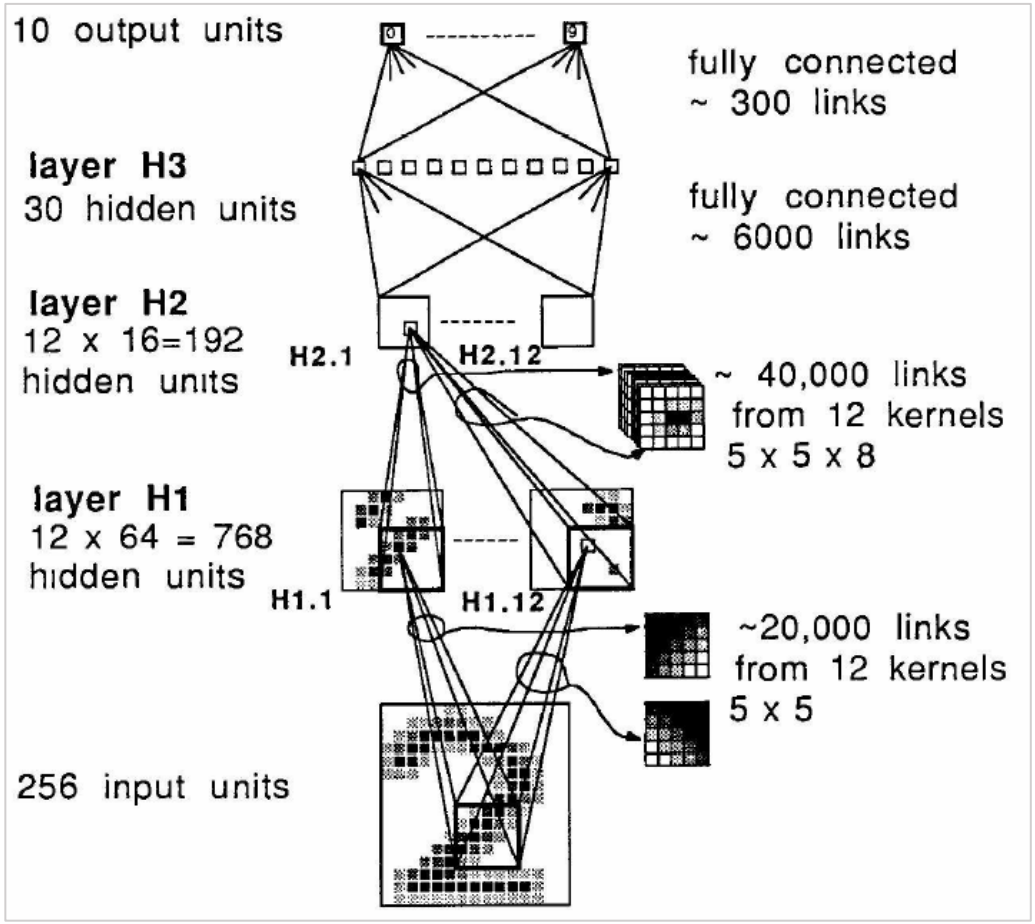
Convolution is performed along the time axis (1D CNN)

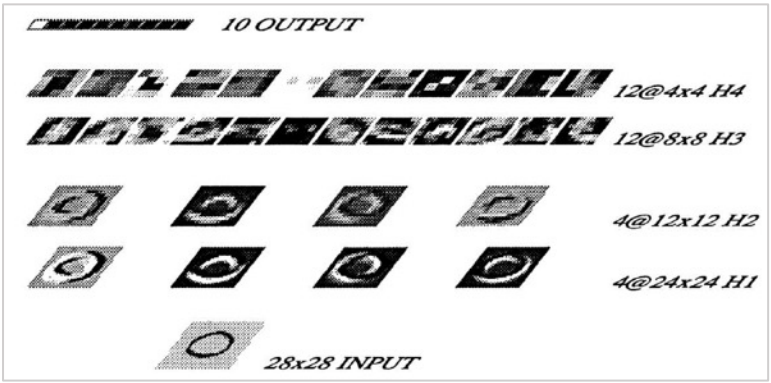
Waibel, Alex, et al. "Phoneme recognition using time-delay neural networks." *IEEE Transactions on acoustics, speech, and signal processing*,1989

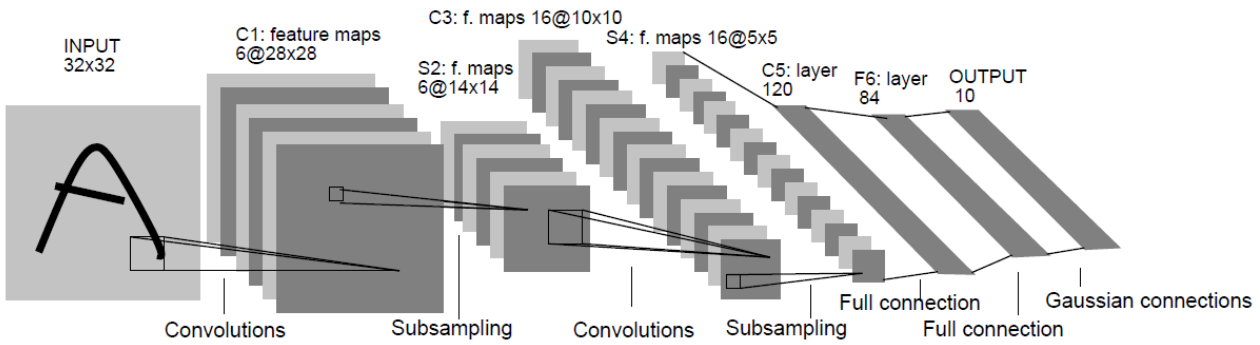
视觉：卷积神经网络

Yann LeCun

2018 ACM A.M. Turing Award







LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition." *Neural computation,* 1989

LeCun, Yann, et al. "Handwritten digit recognition with a back-propagation network." NIPS, 1989

LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, 1998

视觉：卷积神经网络

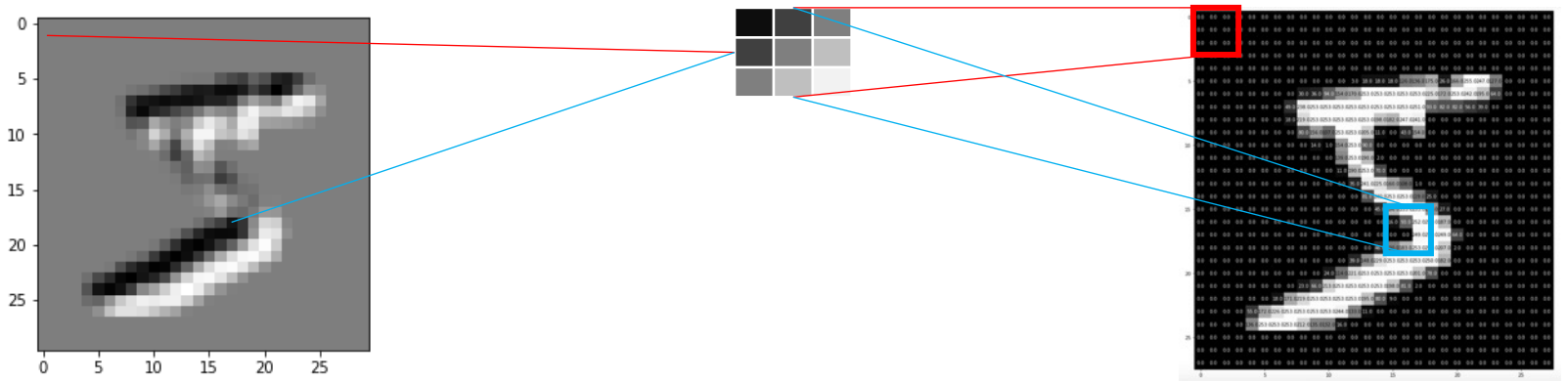
- Convolutional Network Demo from 1993

视觉：卷积神经网络

卷积层
Convolution

简单神经元
- 计算本质： 卷积
- 功能特点： 特征提取

I' K I



$$I'_{i,j} = \sum_{x=-a}^{a} \sum_{y=-b}^{b} I_{i+y,j+x} K_{y,x}$$

与数学卷积方向相反

$$I' = I * K$$

注意区分符号，使用时最好定义

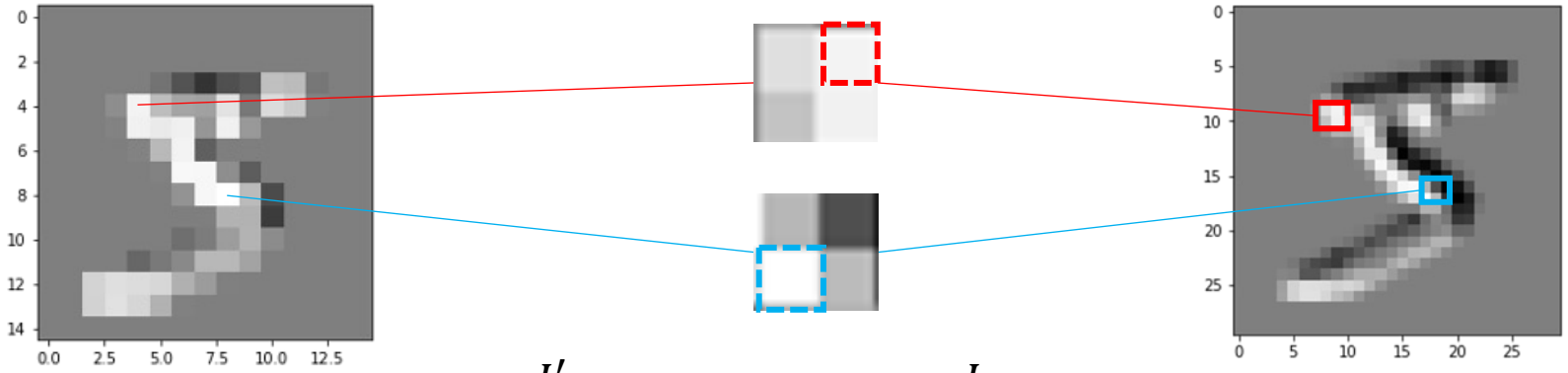
$$F'_{i,j} = \sum_{x=-a}^{a} \sum_{y=-b}^{b} F_{i-y,j-x} G_{y,x}$$

视觉：卷积神经网络

池化层
Pooling

复杂神经元
- 计算本质： 最大值（或者其他采样/聚合函数）
- 功能特点： 平移不变性

I' I

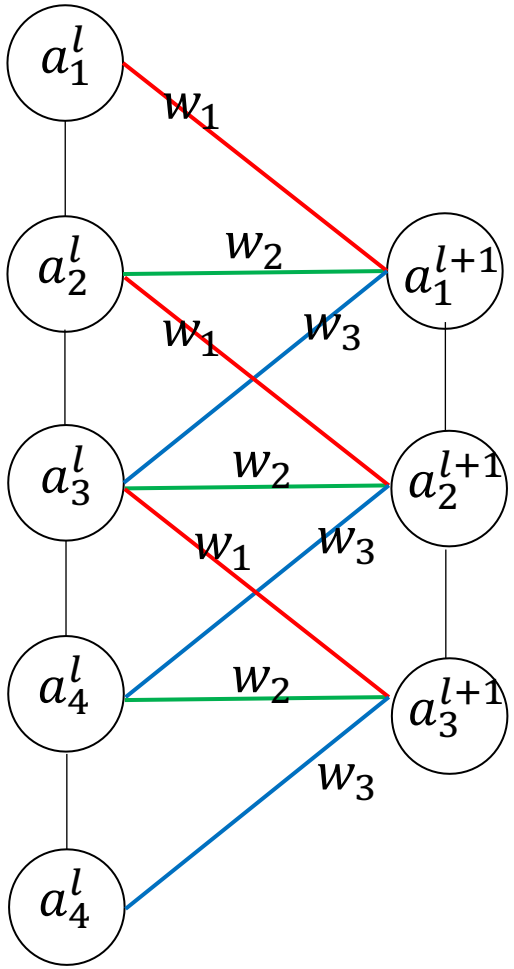


$$I'_{i,j} = \max_{x=[-a,a],y=[-b,b]} I_{i+y,j+x}$$

$$I'_{i,j} = \sum_{x=-a}^{a} \sum_{y=-b}^{b} I_{i+y,j+x} F_{y,x}(\cdot)$$

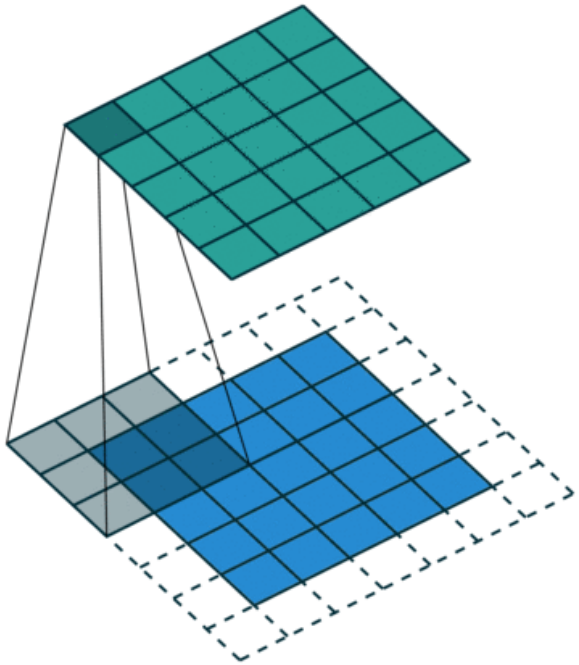
$$F_{y,x}(I_{i+y,j+x}) = \begin{cases} 1, & \text{最大值} \\ 0, & \text{其他} \end{cases} \qquad F_{y,x}(I_{i+y,j+x}) = 1/4 \qquad F_{y,x}(I_{i+y,j+x}) = \cdots$$

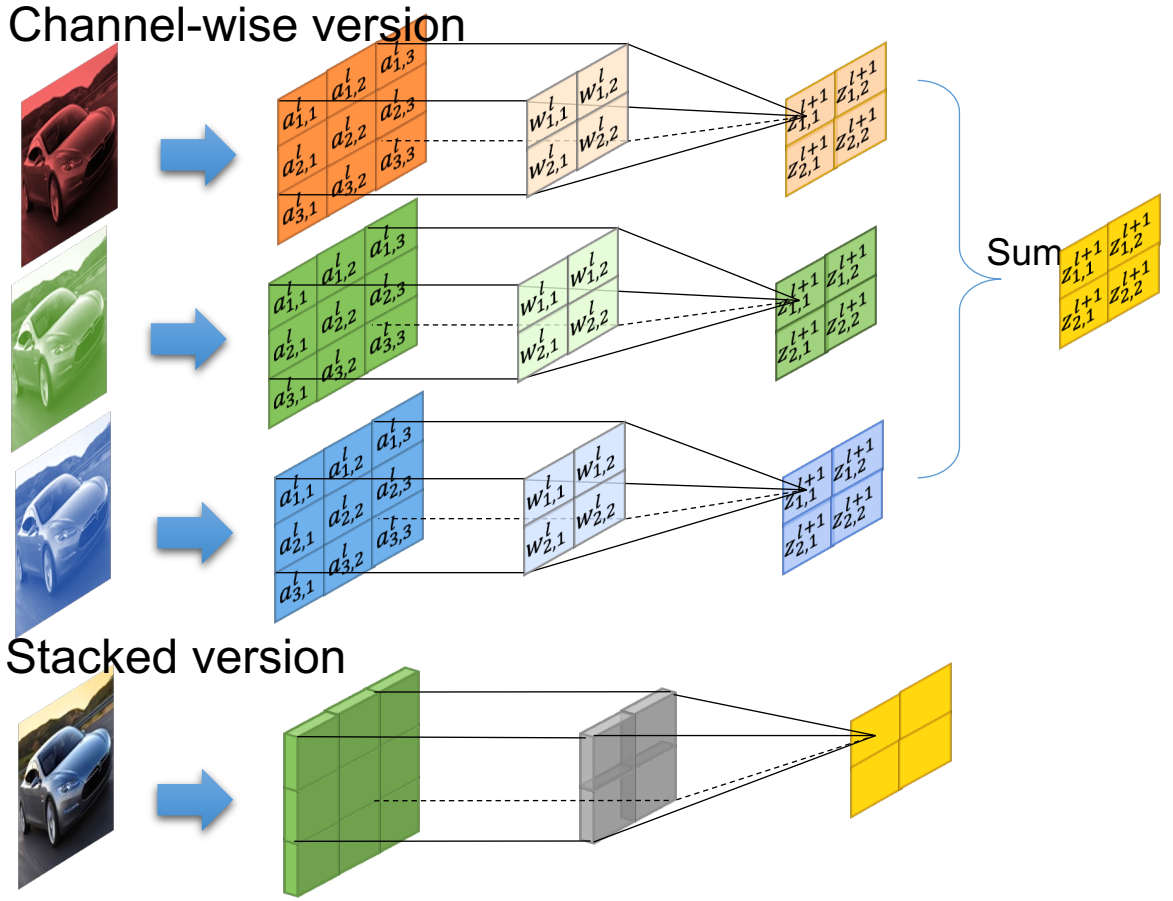
视觉：卷积神经网络



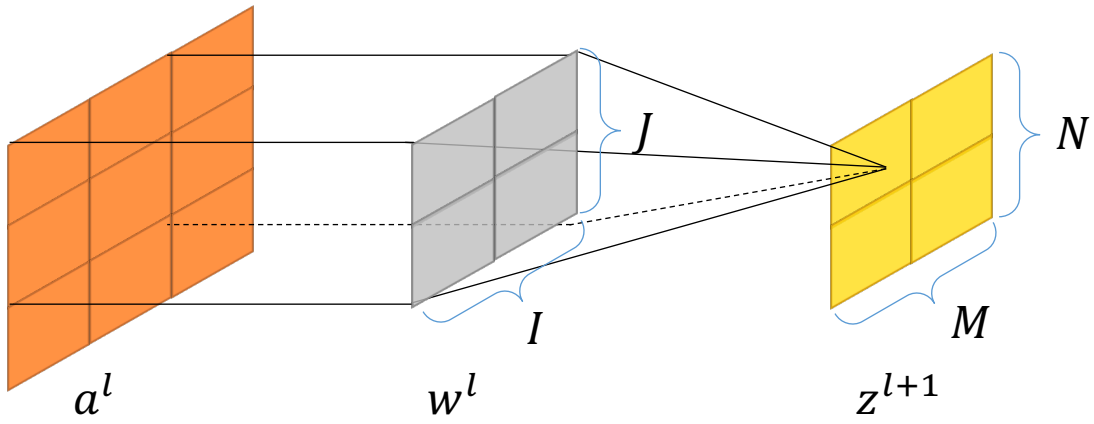
卷积神经网络

- 在两层之间连接权有共享，即相邻两层之间的连接权有相同的





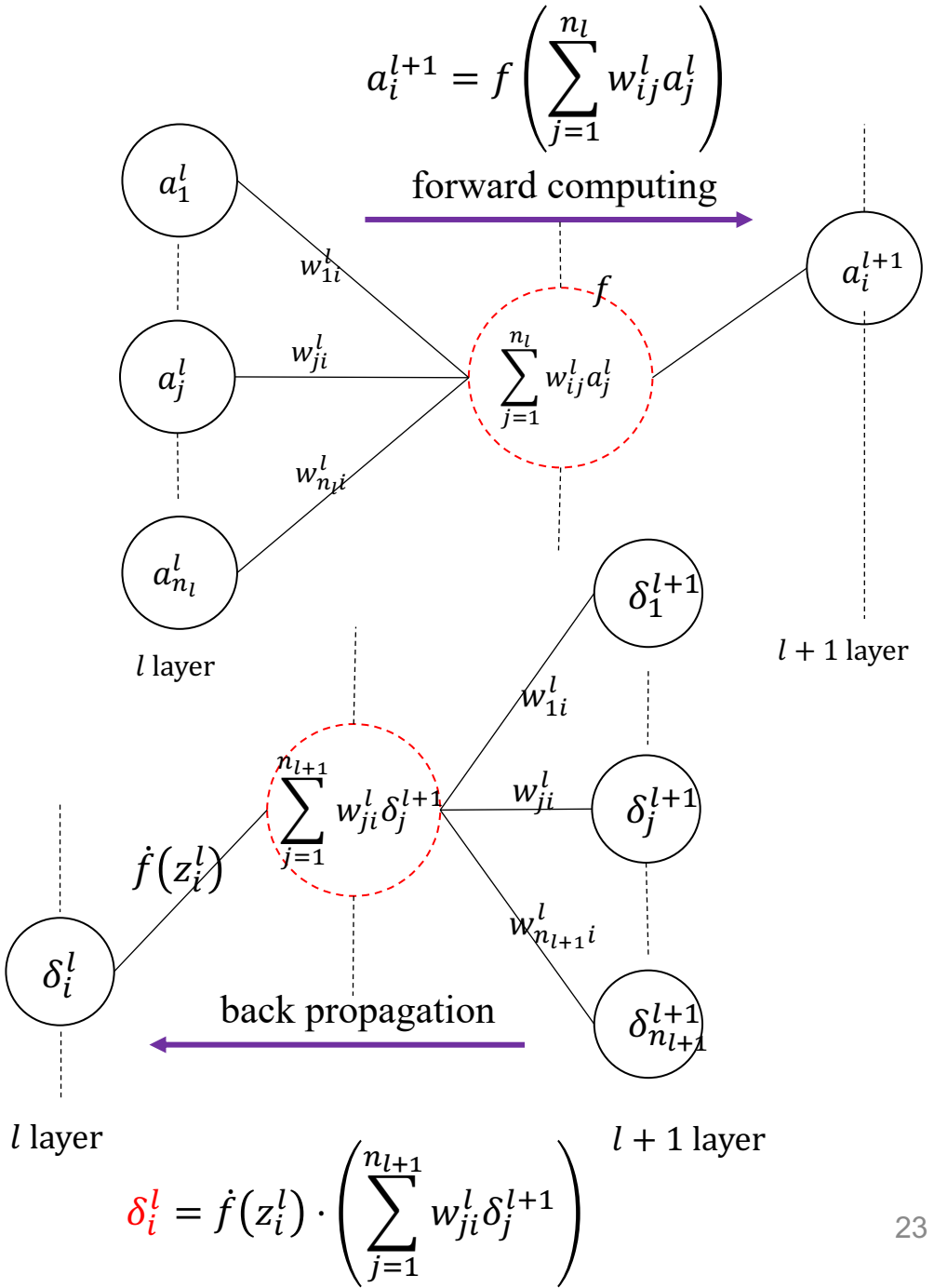
视觉：卷积神经网络



$$z_{n,m}^{l+1} = \sum_{i=1}^{I}\sum_{j=1}^{J} a_{n+j-1,m+i-1} \cdot w_{j,i}^{l}$$

- How to compute $\frac{\partial J}{\partial w_{j,i}^{l}}$ of the convolutional layer ?

The same with the fully-connected layer!

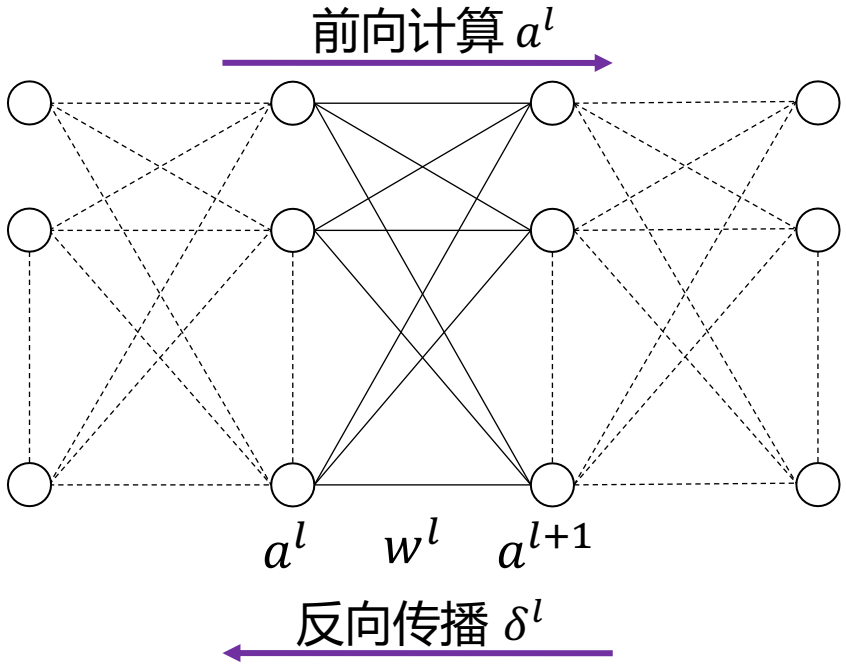


仅用1页ppt理解BP

性能函数: $J(w^1, \cdots, w^{L-1})$

更新规则: $w_{ji}^l \leftarrow w_{ji}^l - \alpha \cdot \frac{\partial J}{\partial w_{ji}^l}$

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



第 l 层的第 i 个神经元

$$a_i^l = f(z_i^l)$$

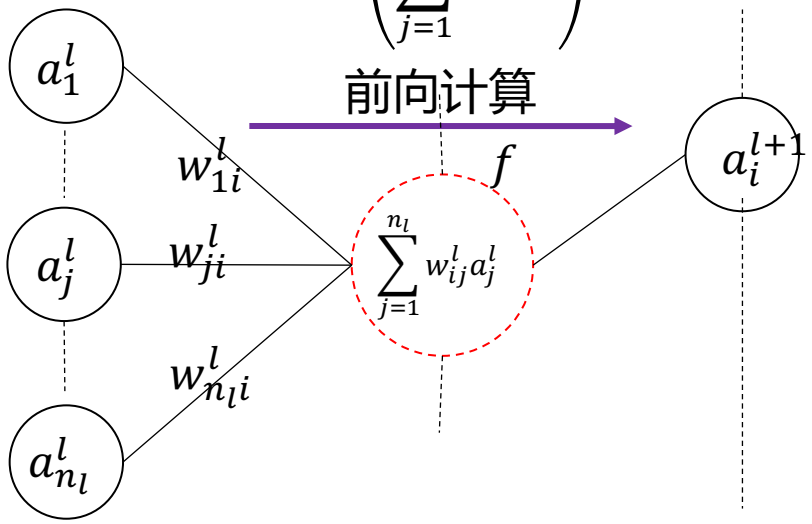
$$\delta_i^l = \frac{\partial J}{\partial z_i^l}$$

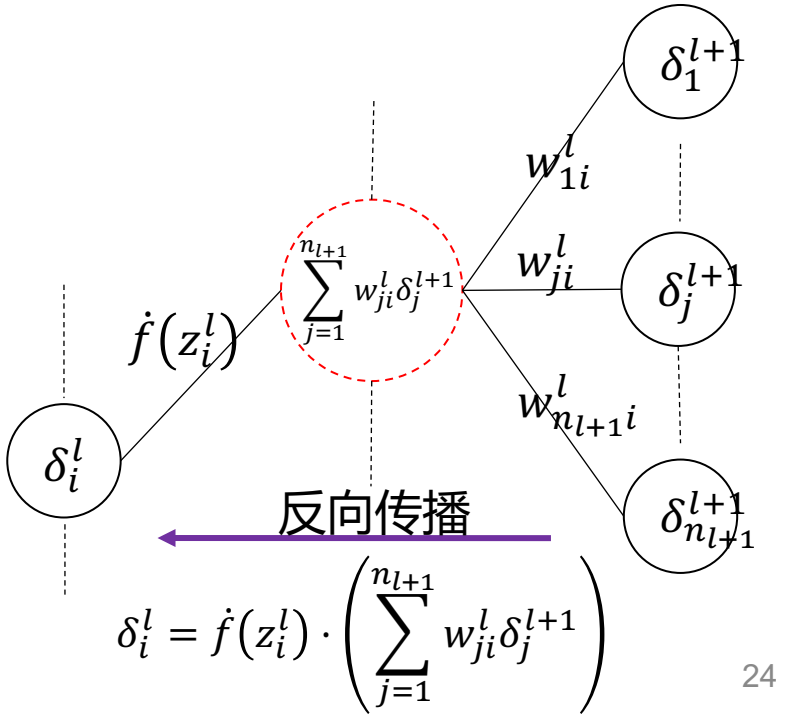
全局 J 局部 f

δ_i^l z_i^l a_i^l

桥梁

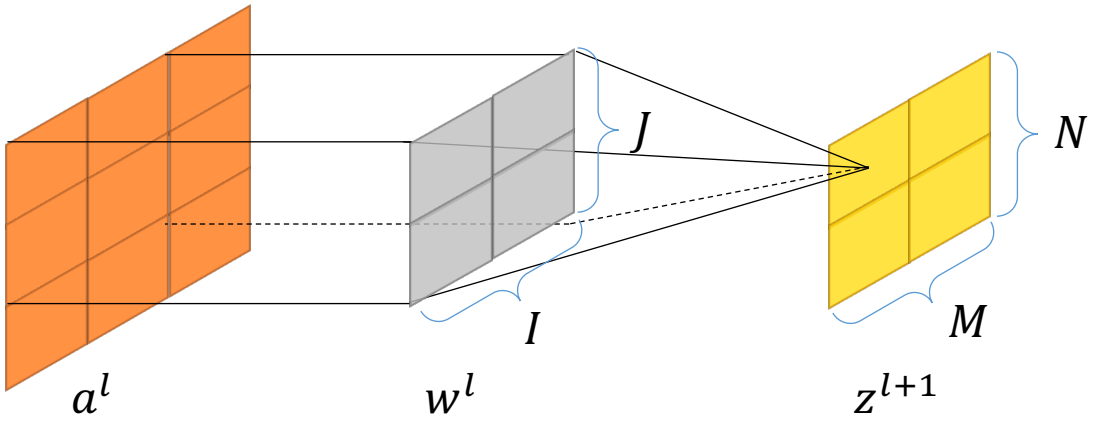
$$a_i^{l+1} = f\left(\sum_{j=1}^{n_l} w_{ij}^l a_j^l\right)$$





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

视觉：卷积神经网络



$$z_{n,m}^{l+1} = \sum_{i=1}^{I}\sum_{j=1}^{J} a_{n+j-1,m+i-1}^{l} \cdot w_{j,i}^{l}$$

Relationship: $$\frac{\partial J}{\partial w_{j',i'}^{l}} = \sum_{m=1}^{M}\sum_{n=1}^{N} \frac{\partial J}{\partial z_{n,m}^{l+1}} \cdot \frac{\partial z_{n,m}^{l+1}}{\partial w_{j',i'}^{l}} = \sum_{m=1}^{M}\sum_{n=1}^{N} \delta_{n,m}^{l+1} \cdot \frac{\partial z_{n,m}^{l+1}}{\partial w_{j',i'}^{l}}$$

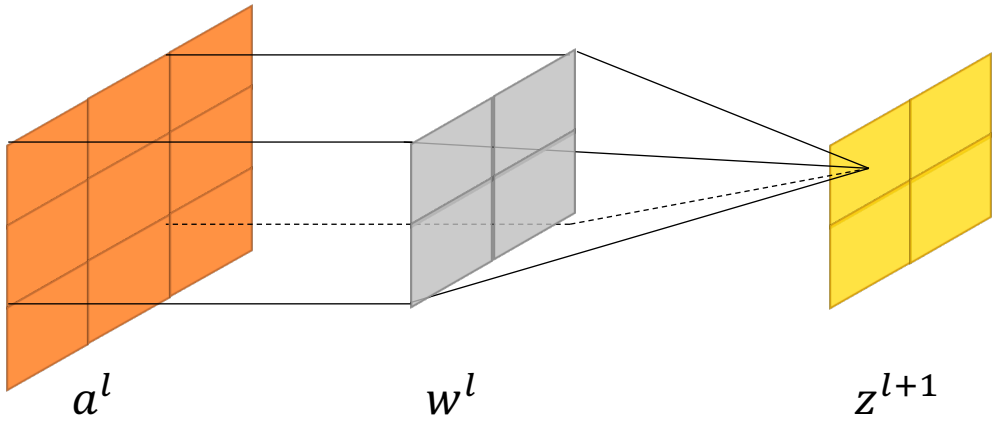
$$\frac{\partial z_{n,m}^{l+1}}{\partial w_{j',i'}^{l}} = \frac{\partial \sum_{i=1}^{I}\sum_{j=1}^{J} a_{n+j-1,m+i-1}^{l} \cdot w_{j,i}^{l}}{\partial w_{j',i'}^{l}} = \begin{cases} a_{n+j'-1,m+i'-1}^{l}, if\ i = i'\ and\ j = j' \\ 0, otherwise \end{cases}$$

$$\frac{\partial J}{\partial w_{j',i'}^{l}} = \sum_{m=1}^{M}\sum_{n=1}^{N} a_{n+j'-1,m+i'-1}^{l} \cdot \delta_{n,m}^{l+1}$$ Convolution!

视觉：卷积神经网络

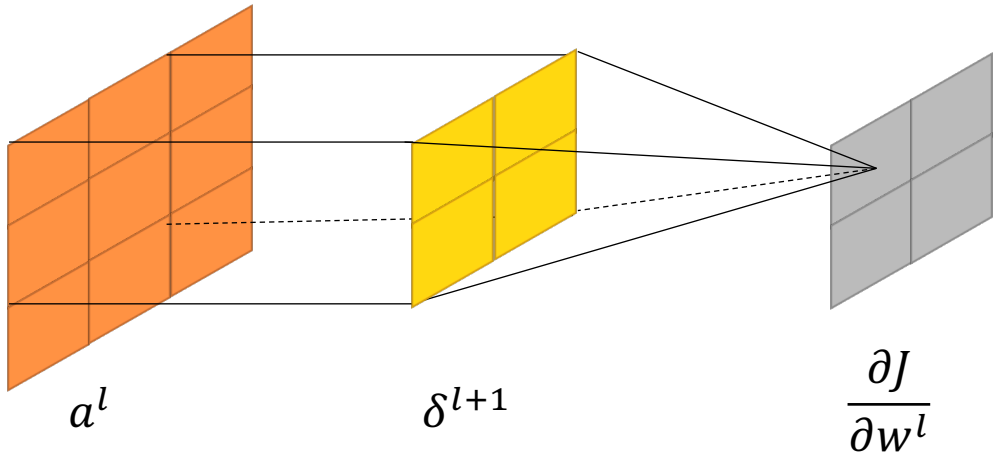
Forward computing

$$z_{n,m}^{l+1} = \sum_{i=1}^{I}\sum_{j=1}^{J} a_{n+j-1,m+i-1}^{l} \cdot w_{j,i}^{l}$$



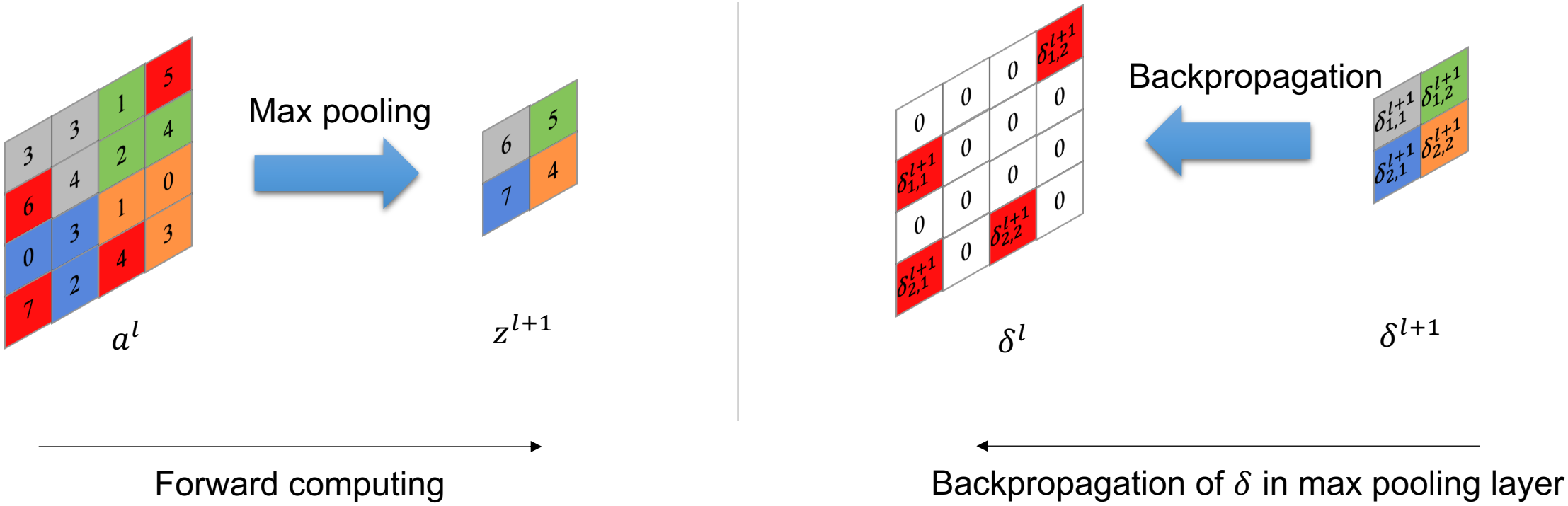
Relationship

$$\frac{\partial J}{\partial w_{j,i}^{l}} = \sum_{m=1}^{M}\sum_{n=1}^{N} a_{n+j-1,m+i-1}^{l} \cdot \delta_{n,m}^{l+1}$$

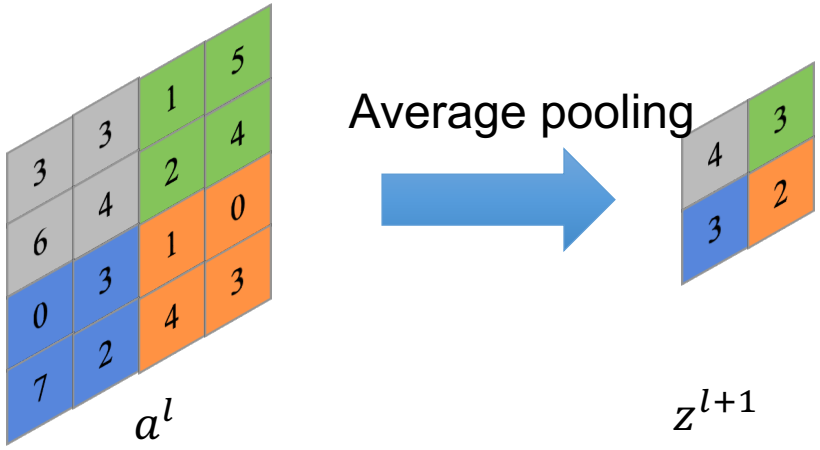


How to backpropagate δ in pooling and convolutional layers?

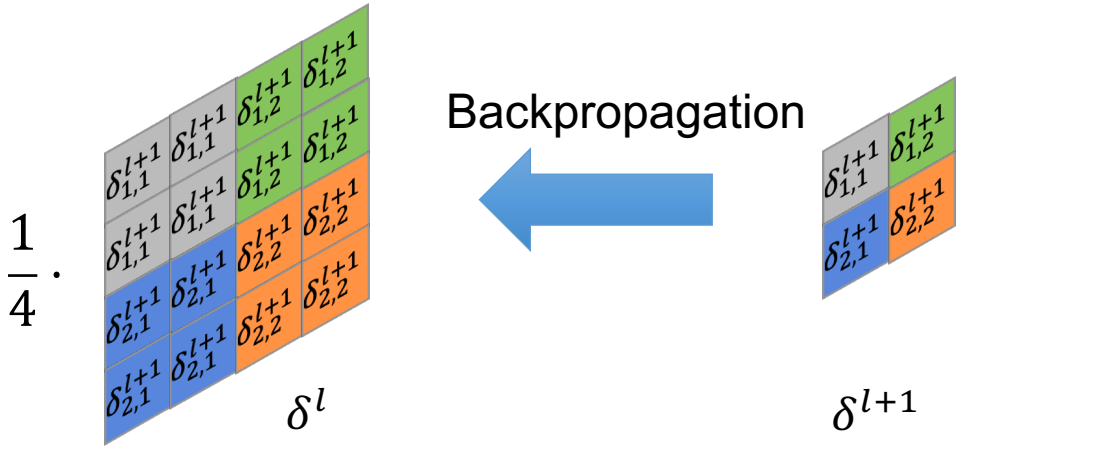
视觉：卷积神经网络


Max pooling
a^l
z^{l+1}
Backpropagation
δ^l
δ^{l+1}
Forward computing
Backpropagation of δ in max pooling layer

视觉：卷积神经网络



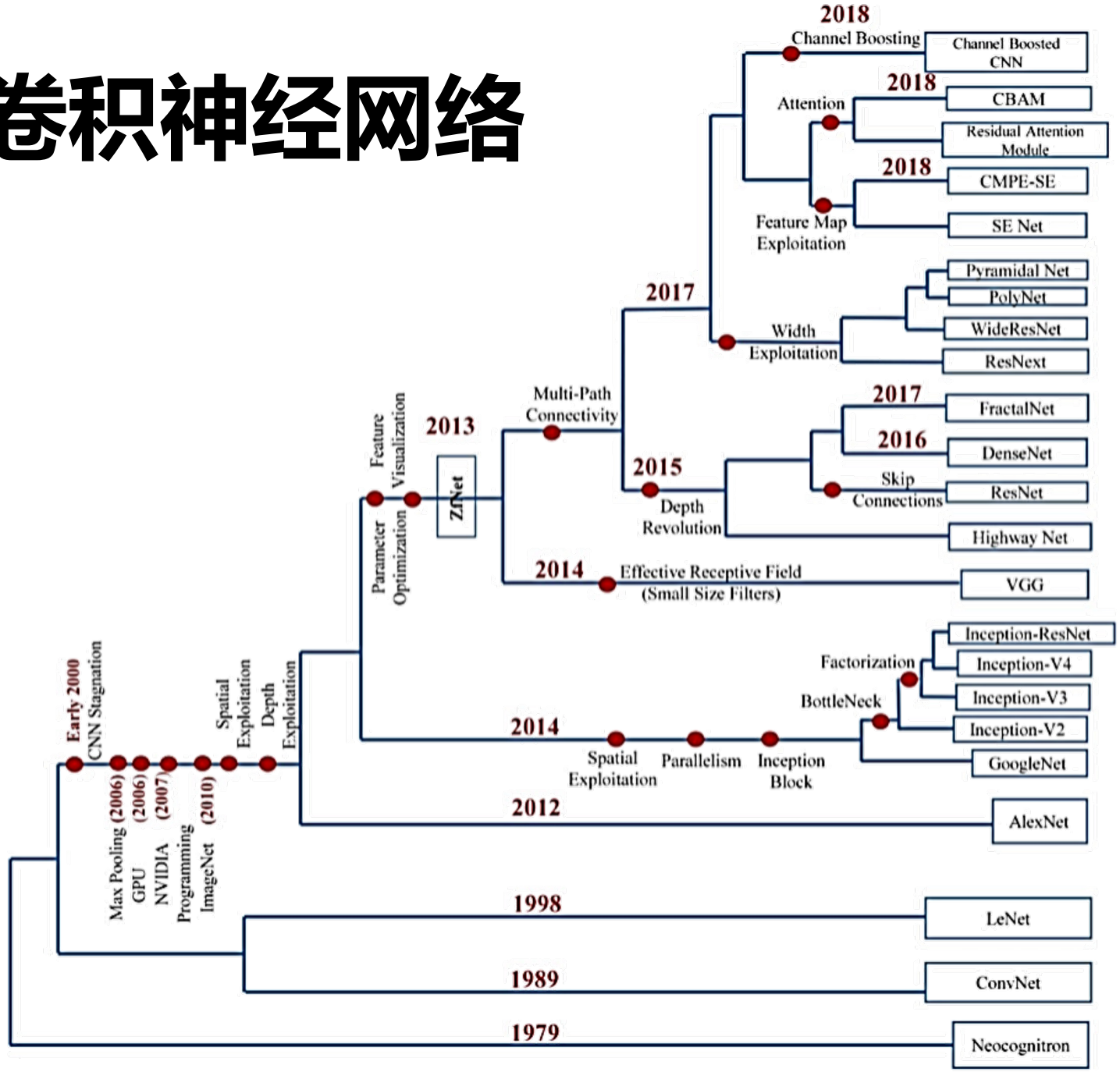
Forward computing



Backpropagation of δ in average pooling layer

How to backpropagate δ in convolutional layer?

视觉：卷积神经网络


2018
Channel Boosting
Channel Boosted CNN
2018
Attention
CBAM
Residual Attention Module
2018
CMPE-SE
SE Net
Feature Map Exploitation
2017
Pyramidal Net
PolyNet
WideResNet
ResNext
Width Exploitation
Multi-Path Connectivity
2017
FractalNet
2016
DenseNet
2015
Skip Connections
ResNet
Depth Revolution
Highway Net
Feature Visualization
2013
ZFNet
Parameter Optimization
2014
Effective Receptive Field (Small Size Filters)
VGG
Inception-ResNet
Factorization
Inception-V4
BottleNeck
Inception-V3
Inception-V2
2014
Spatial Exploitation
Parallelism
Inception Block
GoogleNet
Early 2000
CNN Stagnation
Spatial Exploitation
Depth Exploitation
2012
AlexNet
Max Pooling (2006)
GPU (2006)
NVIDIA (2007)
Programming (2010)
ImageNet
1998
LeNet
1989
ConvNet
1979
Neocognitron

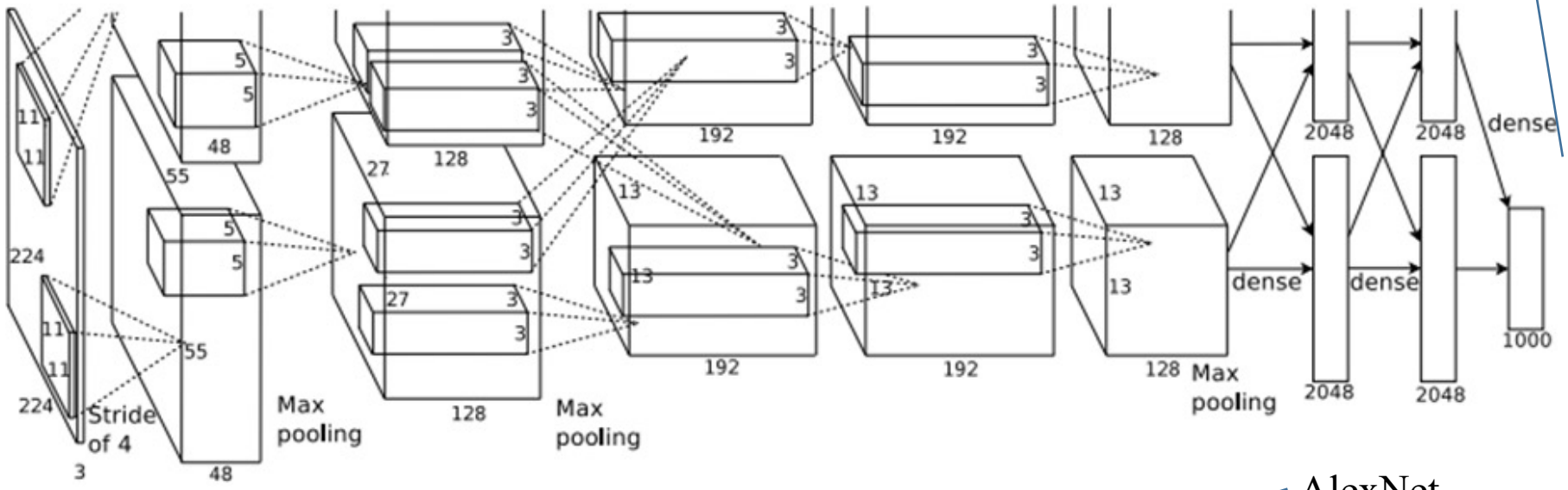
视觉：卷积神经网络

IMAGENET

ImageNet Large Scale Visual Recognition Challenge 2012

计算十分复杂，引入GPU加速计算

1000分类！
120万图像训练集；
5万验证图像；15万测试图像。



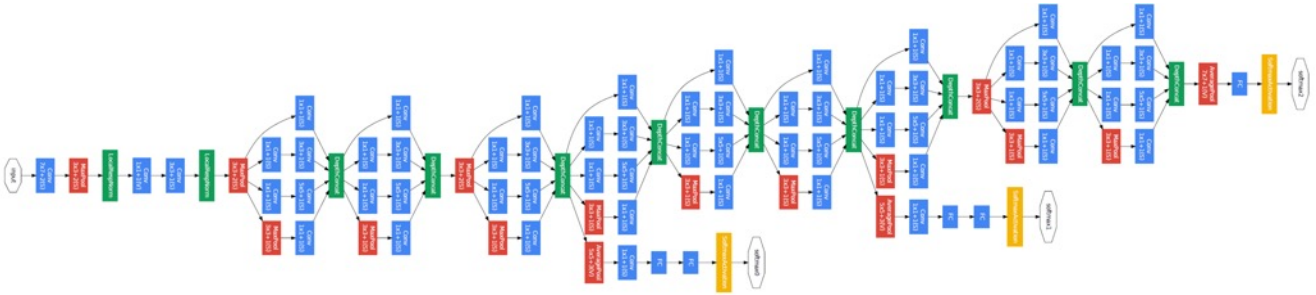
AlexNet

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012.

Top-5 正确率
85%

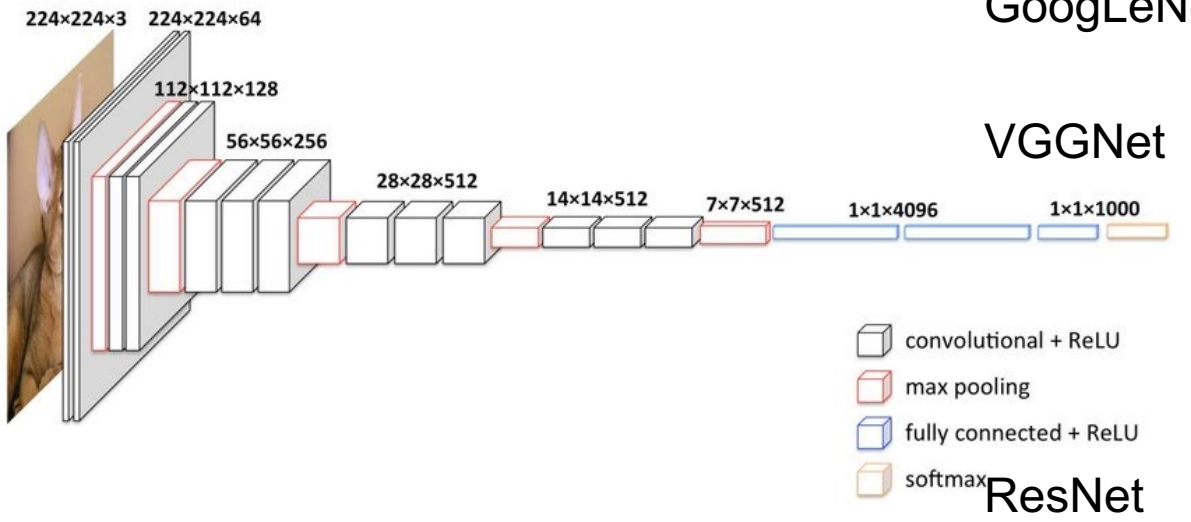
视觉：卷积神经网络

C. Szegedy *et al*., Going deeper with convolutions, *CVPR* 2015.



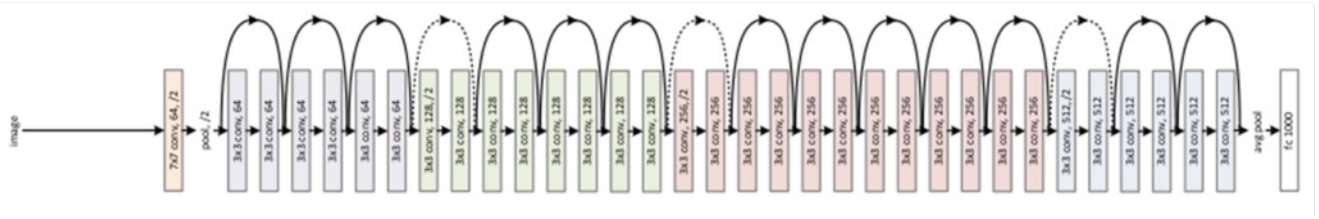
GoogLeNet

Simonyan, Karen and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ICLR 2015*.



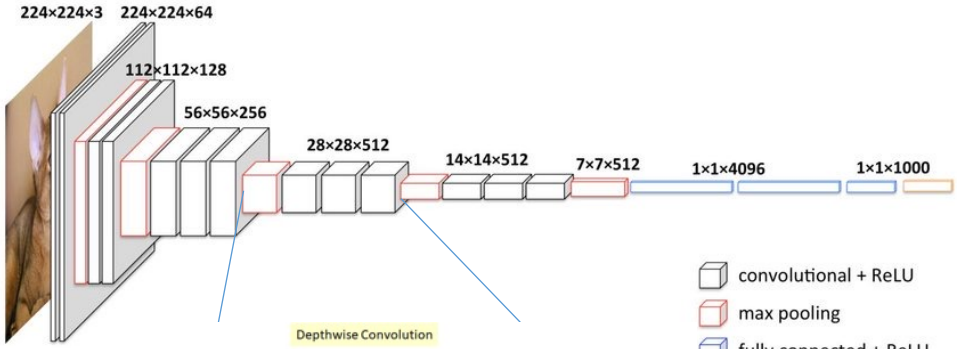
VGGNet

He, Kaiming, X. Zhang, Shaoqing Ren and Jian Sun. “Deep Residual Learning for Image Recognition.” *CVPR* 2016: 770-778.

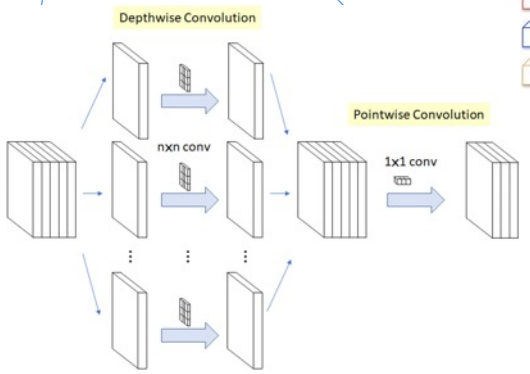


ResNet

视觉：卷积神经网络



Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

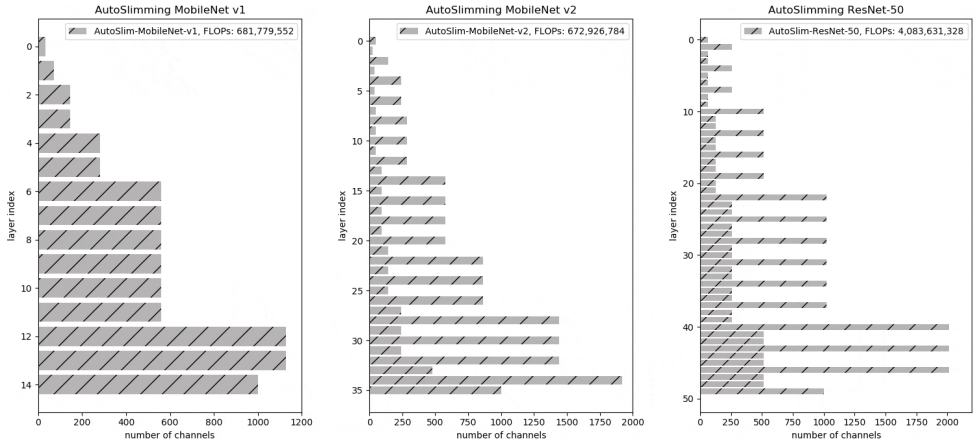


MobileNet

Howard, Andrew G., Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le and Hartwig Adam. Searching for MobileNetV3. *ICCV* 2019: 1314-1324.

MobileNet V3

Jiahui Yu; Thomas Huang, Universally Slimmable Networks and Improved Training Techniques, ICCV 2019.



NAS

提纲

视觉：卷积神经网络

序列：多输出神经网络

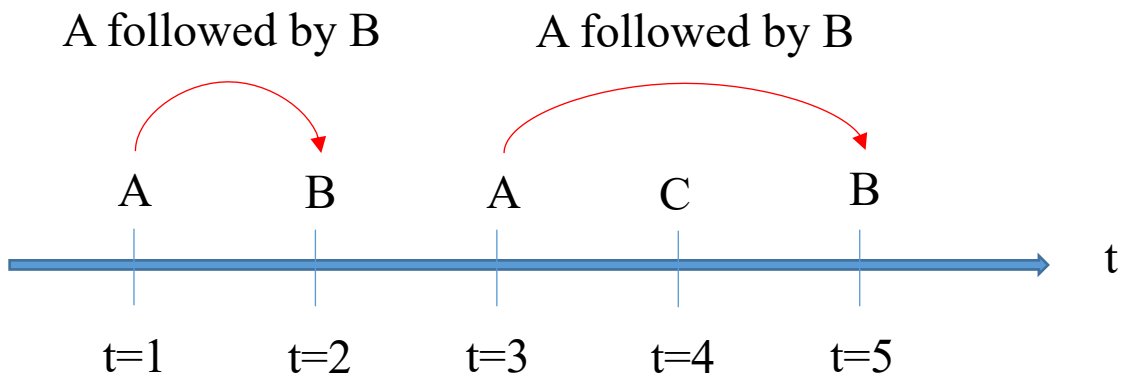
一个序列识别的例子

生成序列：

1. ABACB
2. CCBBA
3. CACCB
4. ACCCB
5. CACBC
6. AAACB
7. BAACB
8. CCBAB
9. BCCAB
10. CABAC

……..

任务: Recognize A followed by B.



一个序列识别的例子

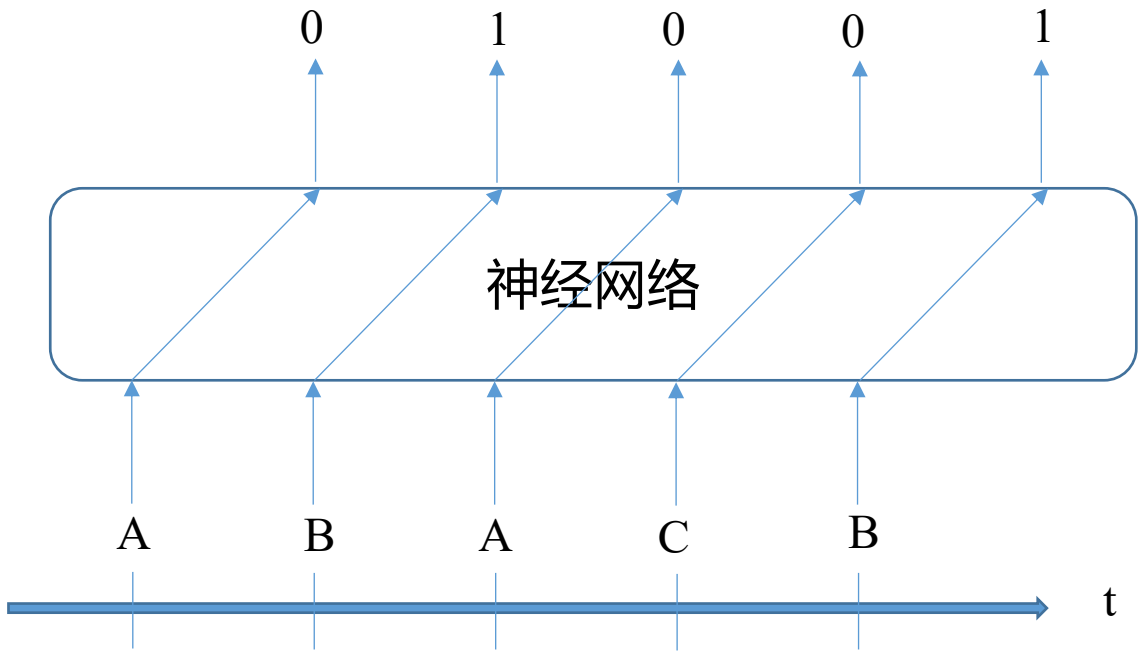
任务: Recognize A followed by B.

生成的序列：

1. ABCAB
2. CCBBA
3. CACCB
4. ACCCB
5. CACBC
6. AAACB
7. BAACB
8. CCBAB
9. BCCAB
10. CABAC

……

问题: 可否用神经网络解决这个?



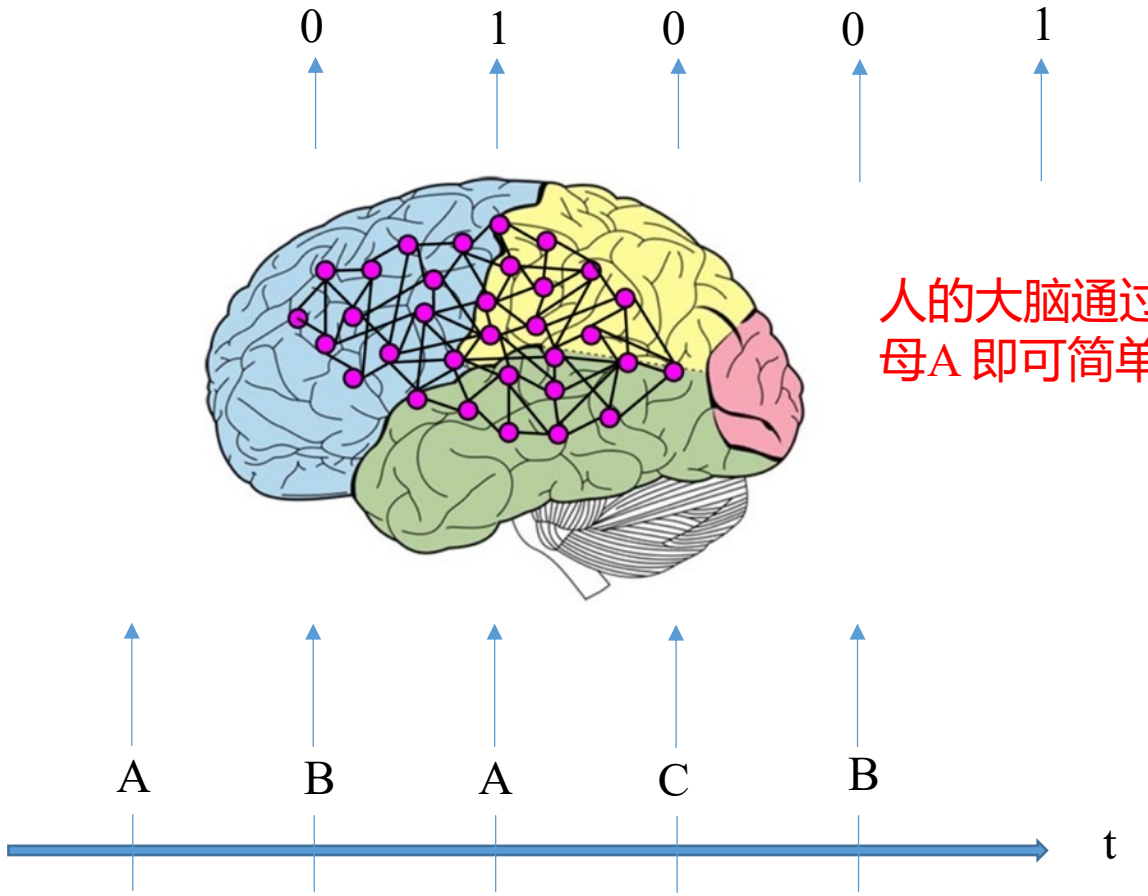
一个序列识别的例子

任务: Recognize A followed by B.

生成的序列：

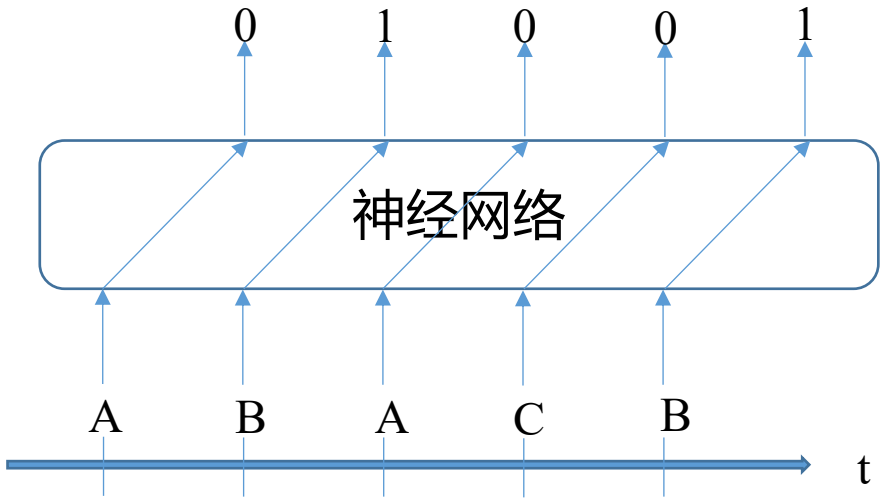
1. ABCAB
2. CCBBA
3. CACCB
4. ACCCB
5. CACBC
6. AAACB
7. BAACB
8. CCBAB
9. BCCAB
10. CABAC

……



人的大脑通过记住前面出现的字母A 即可简单地解决这个问题。

一个序列识别的例子

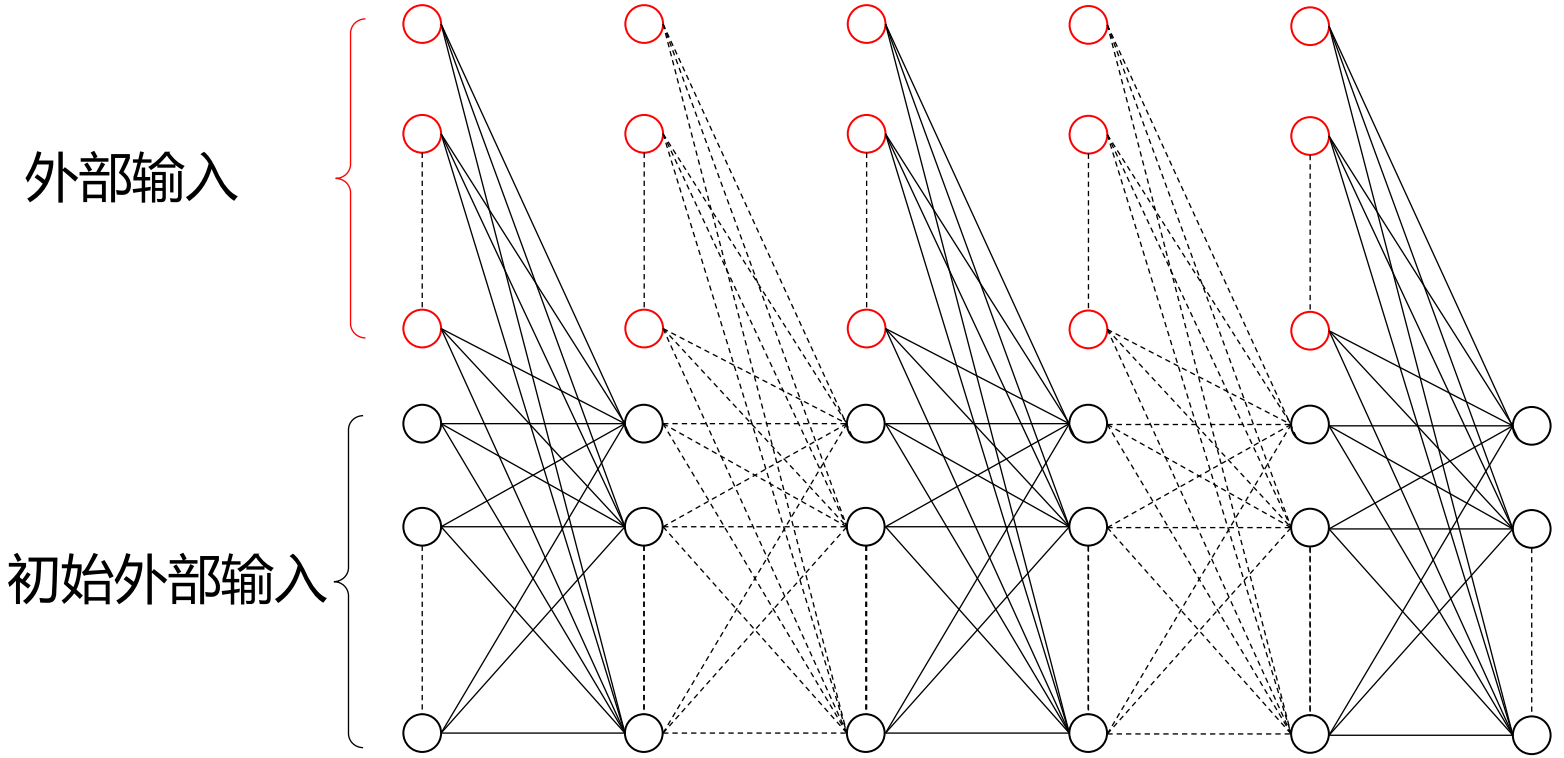


任务: Recognize A followed by B.

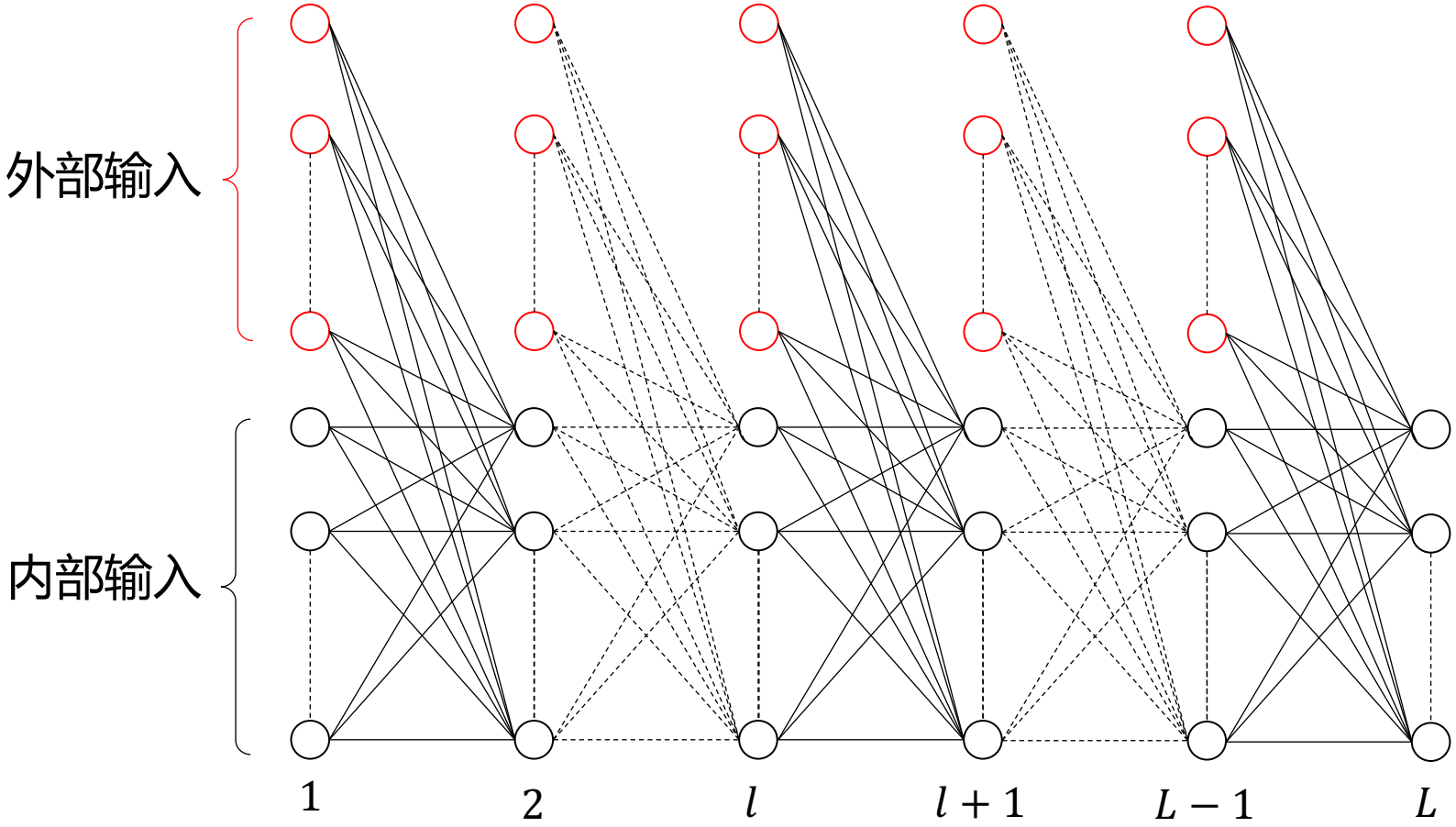
生成的序列：

1. ABCAB
2. CCBBA
3. CACCB
4. ACCCB
5. CACBC
6. AAACB
7. BAACB
8. CCBAB
9. BCCAB
10. CABAC

……



一个序列识别的例子



单目标输出

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

仅在最后一层输出

单目标输出网络不能解决序列识别问题。

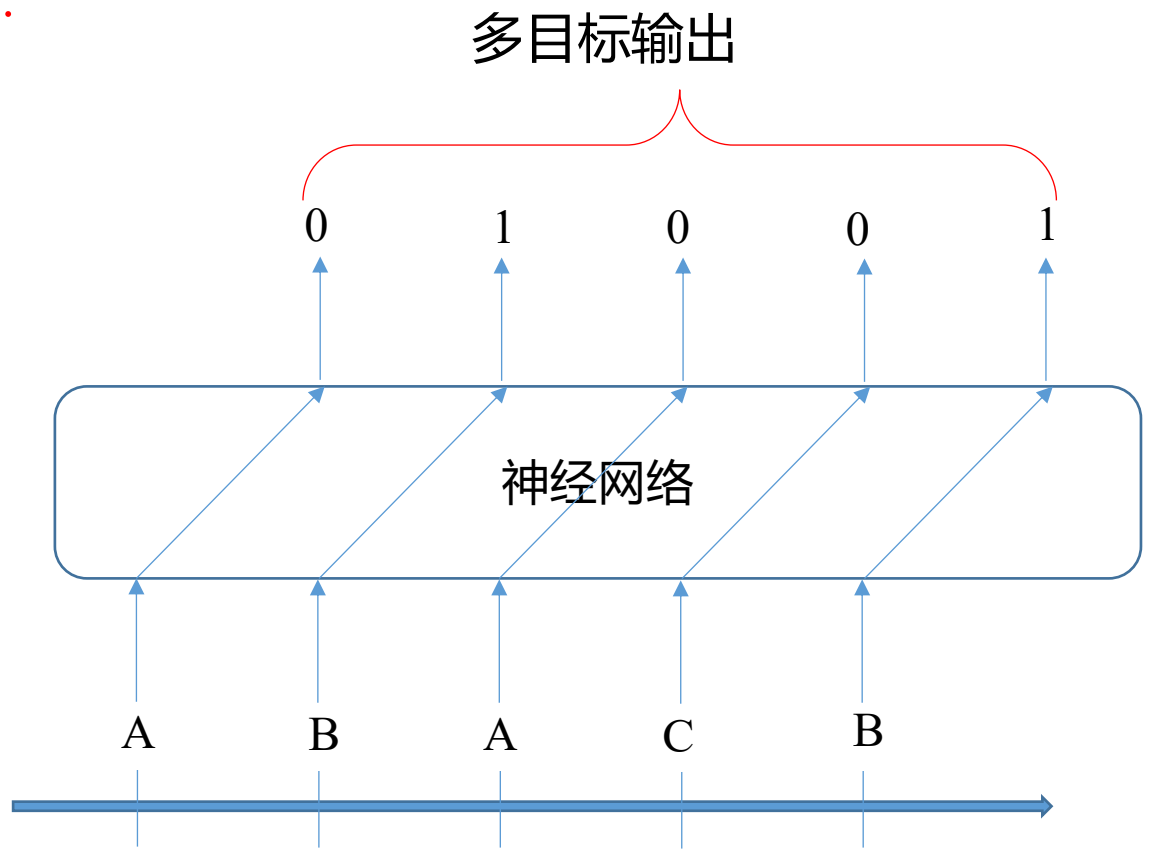
一个序列识别的例子

任务: Recognize A followed by B.

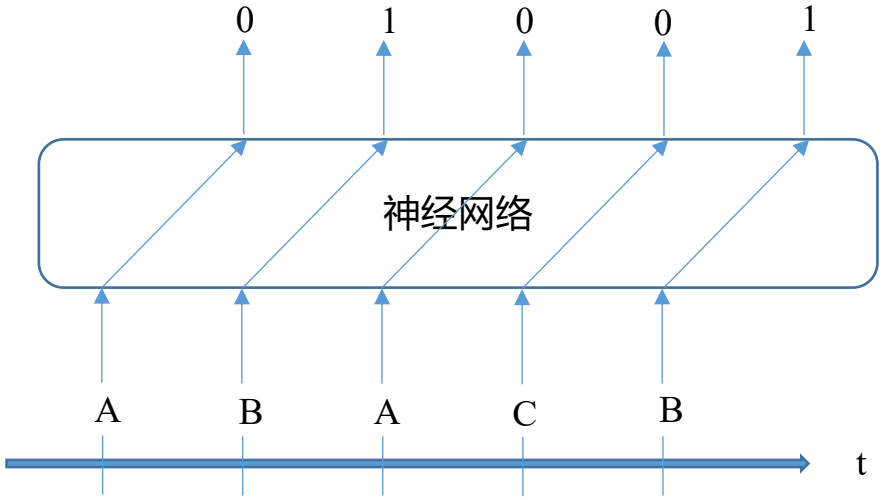
生成的序列：

1. ABCAB
2. CCBBA
3. CACCB
4. ACCCB
5. CACBC
6. AAACB
7. BAACB
8. CCBAB
9. BCCAB
10. CABAC

……

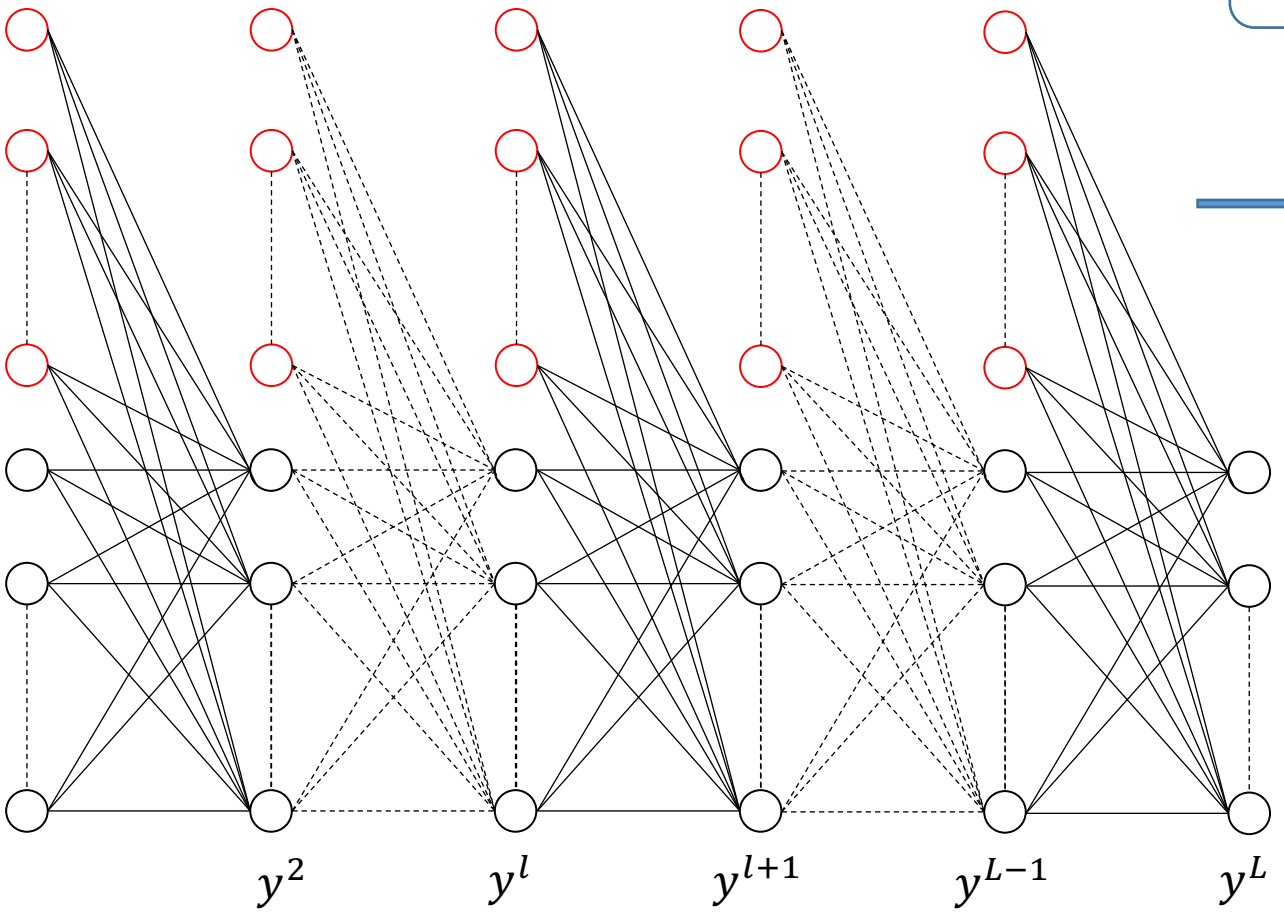


一个序列识别的例子



外部输入

初始外部输入



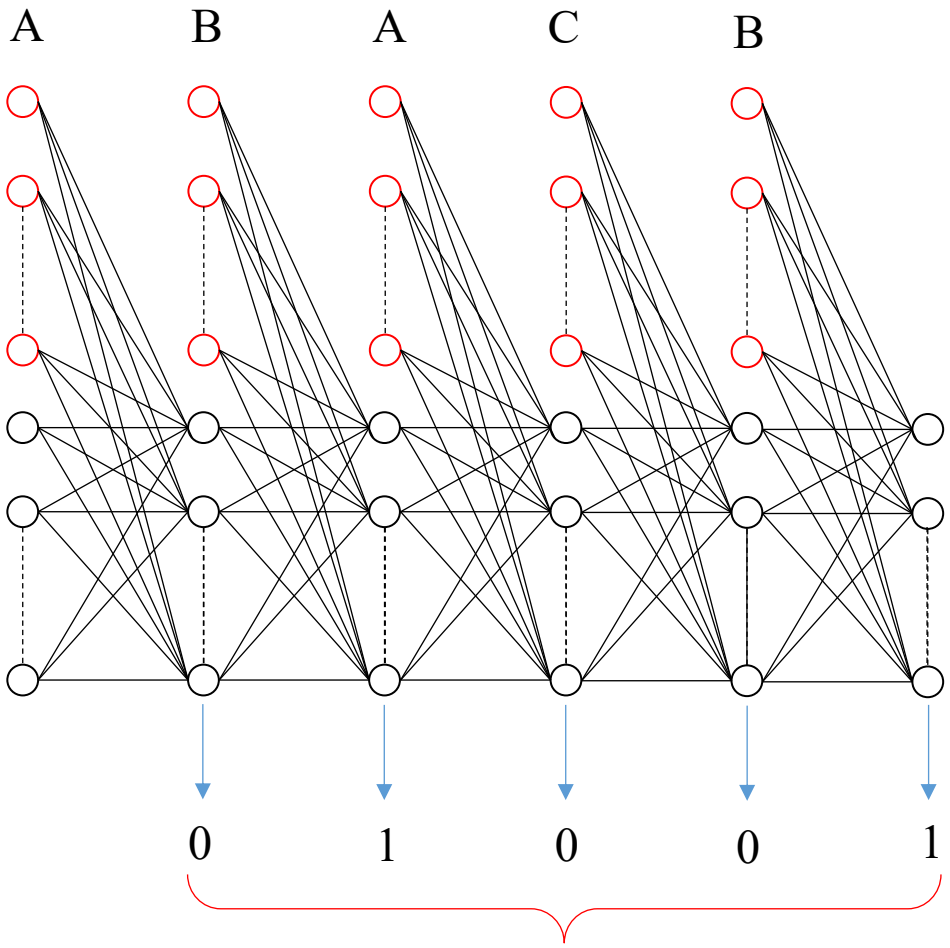
多目标输出

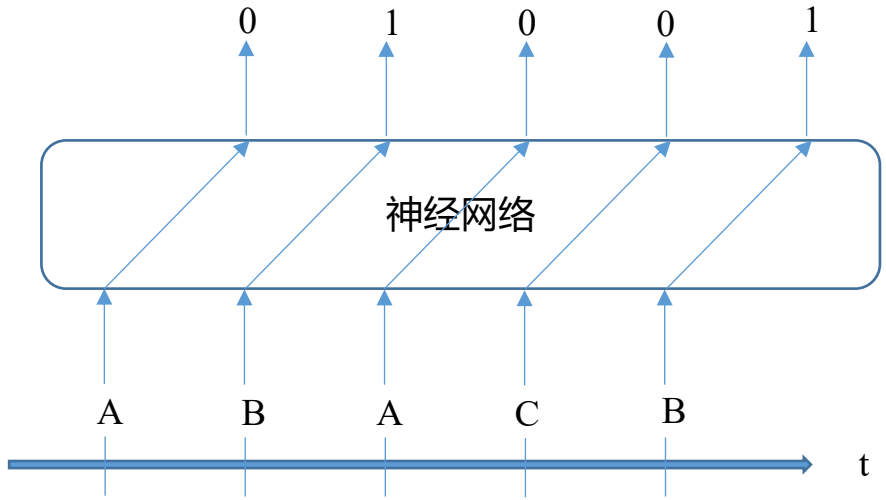
多目标输出

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

$$(l = 2, \cdots, L)$$

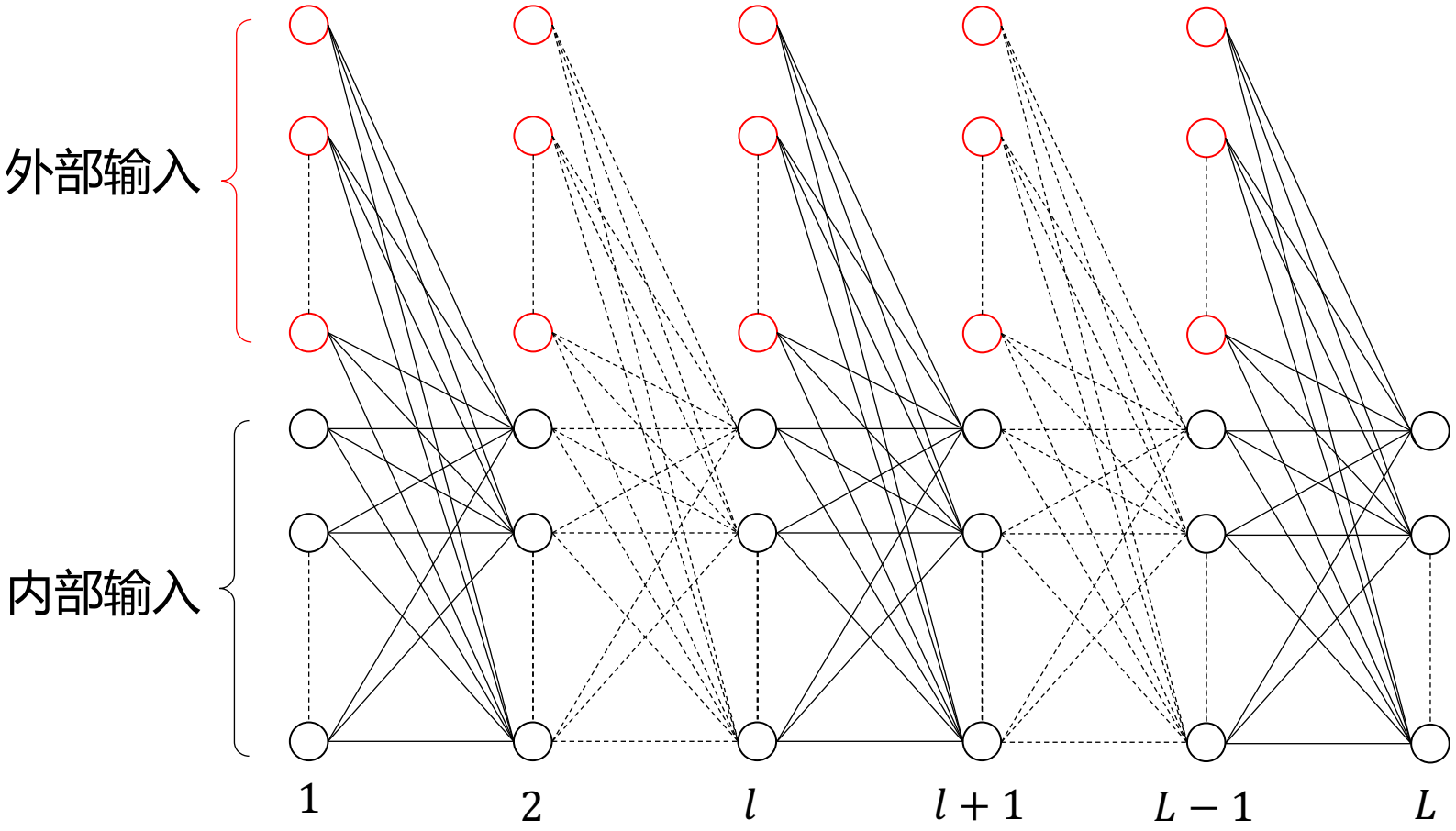
一个序列识别的例子





问题:
如何构造一个算法来训练这个网络?

一个序列识别的例子



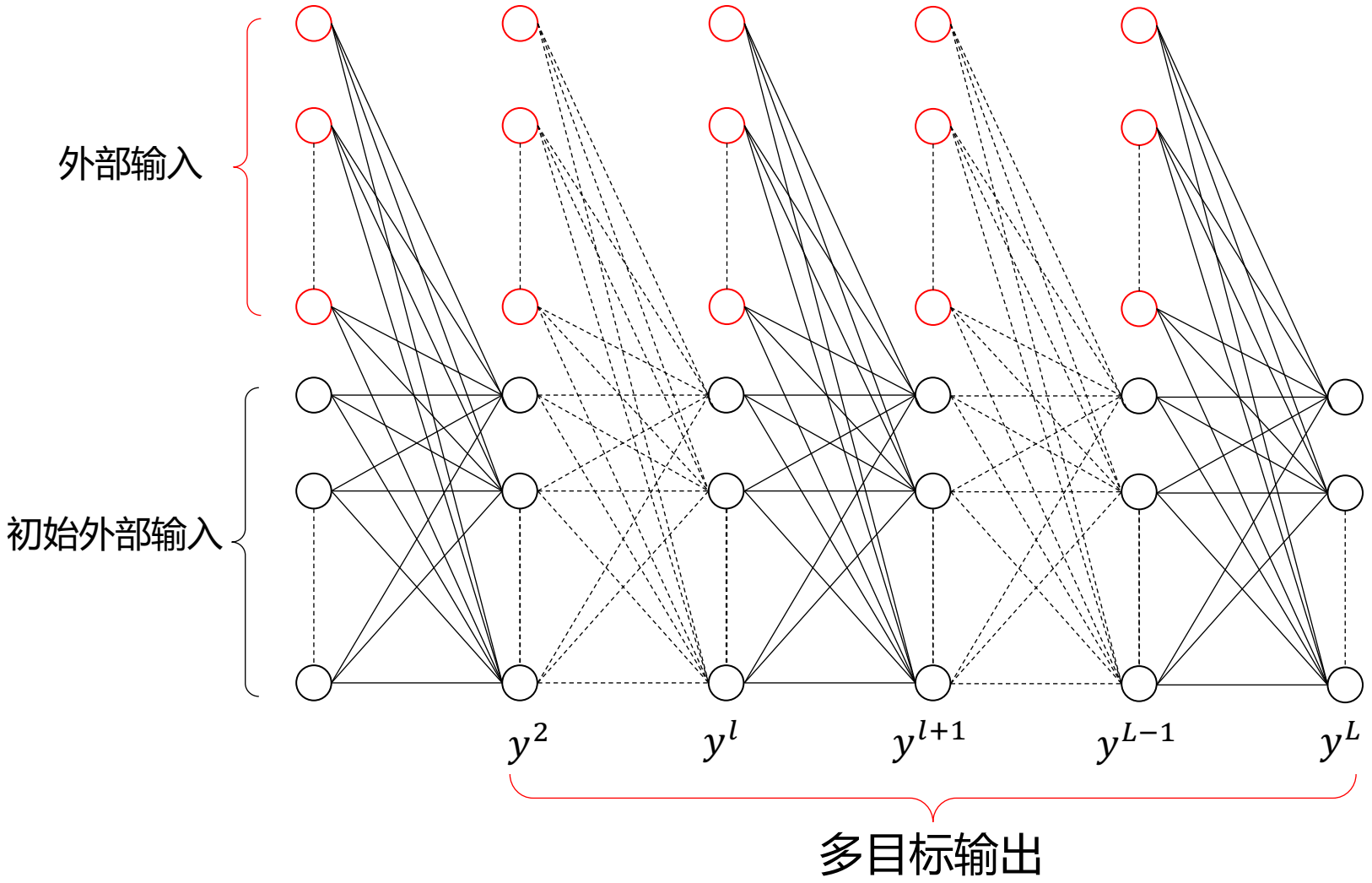
单目标输出

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

仅在最后一层输出

在单目标输出中已经有BP算法可以训练网络。

一个序列识别的例子



问题:

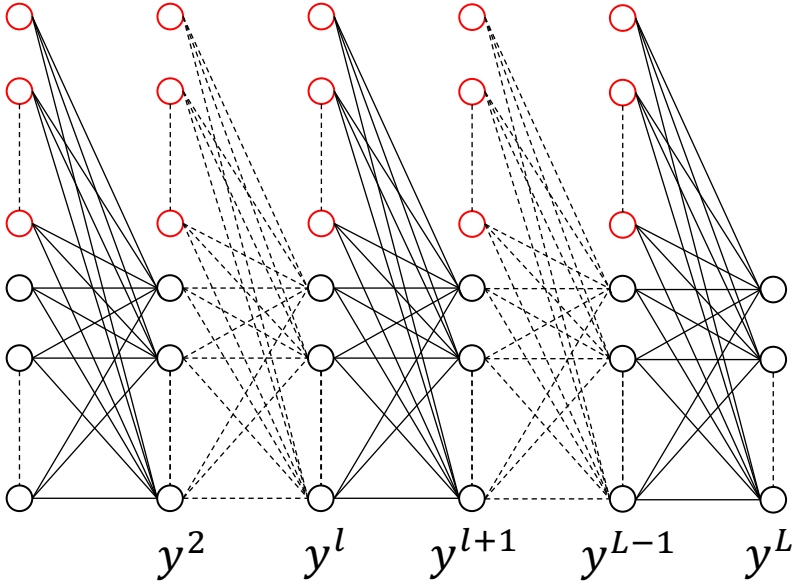
可否构造一个类似BP算法的多目标输出学习算法?

多目标输出

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

$$(l = 2, \cdots, L)$$

一个序列识别的例子

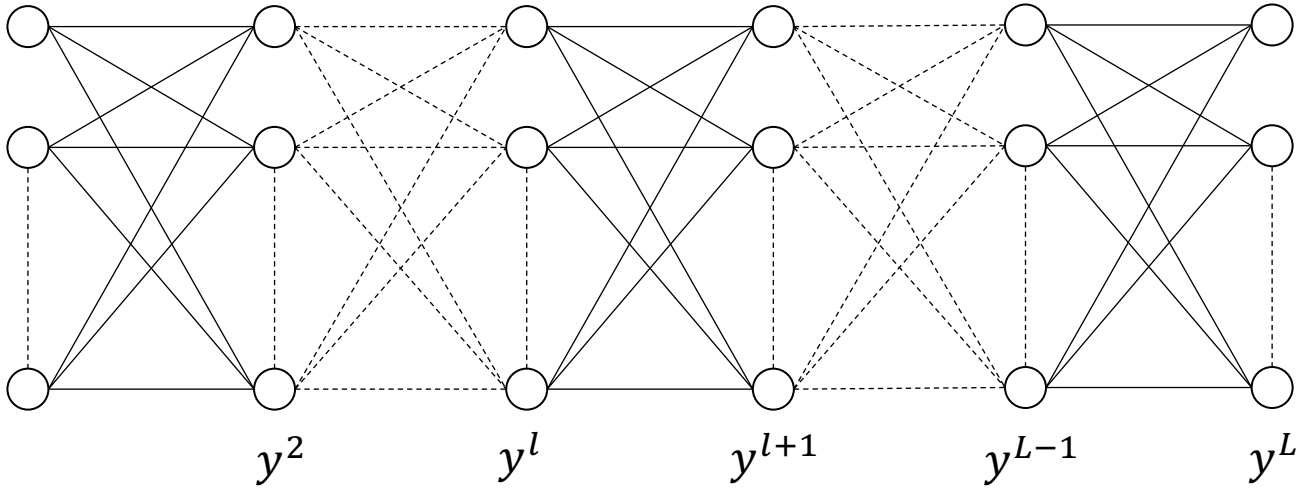


等价

多目标输出

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

$(l = 2, \cdots, L)$



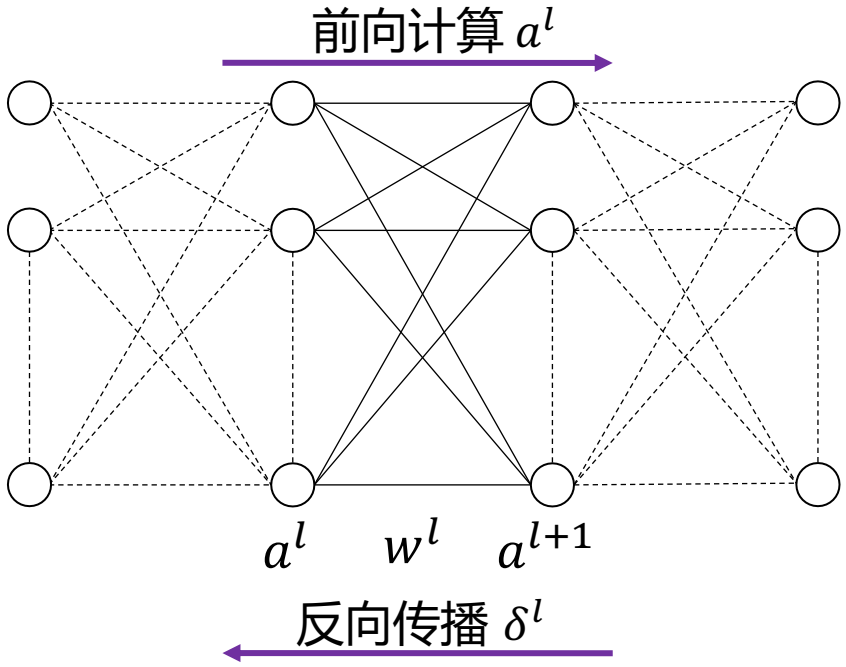
问题:
是否可以构造一个类似BP的学习算法?

仅用1页ppt理解BP

性能函数: $J(w^1, \cdots, w^{L-1})$

更新规则: $w_{ji}^l \leftarrow w_{ji}^l - \alpha \cdot \frac{\partial J}{\partial w_{ji}^l}$

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



第 l 层的第 i 个神经元

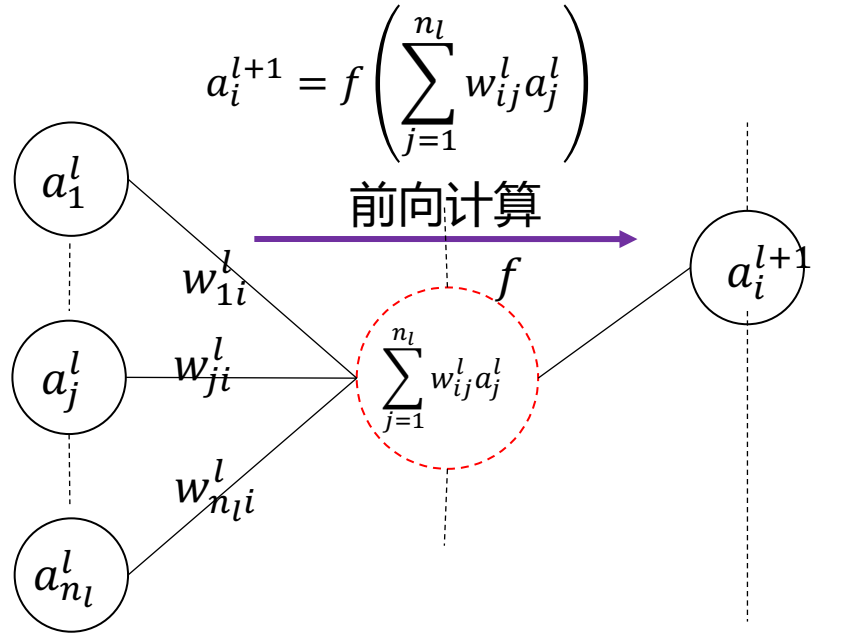
$$a_i^l = f(z_i^l)$$

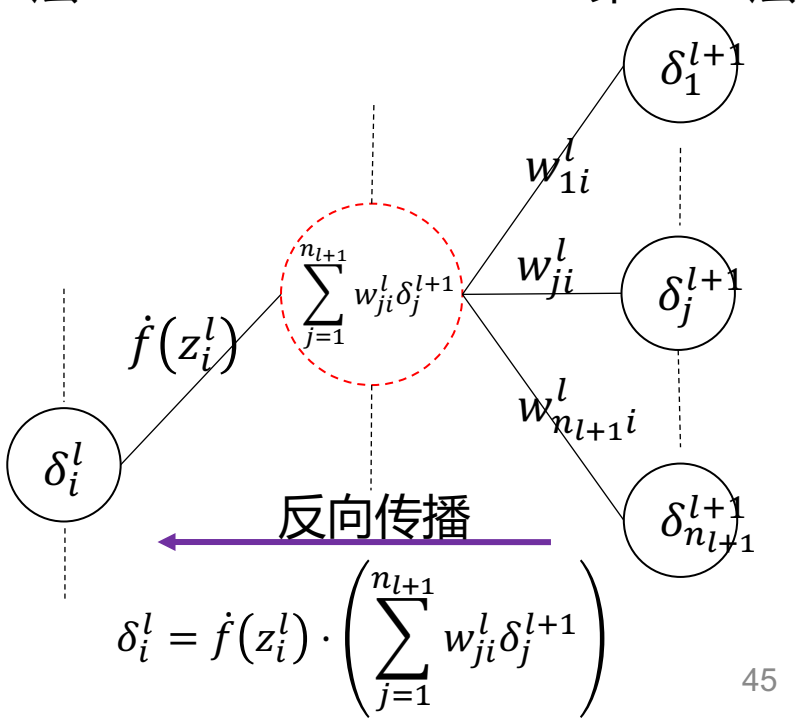
$$\delta_i^l = \frac{\partial J}{\partial z_i^l}$$

全局 J　　局部 f

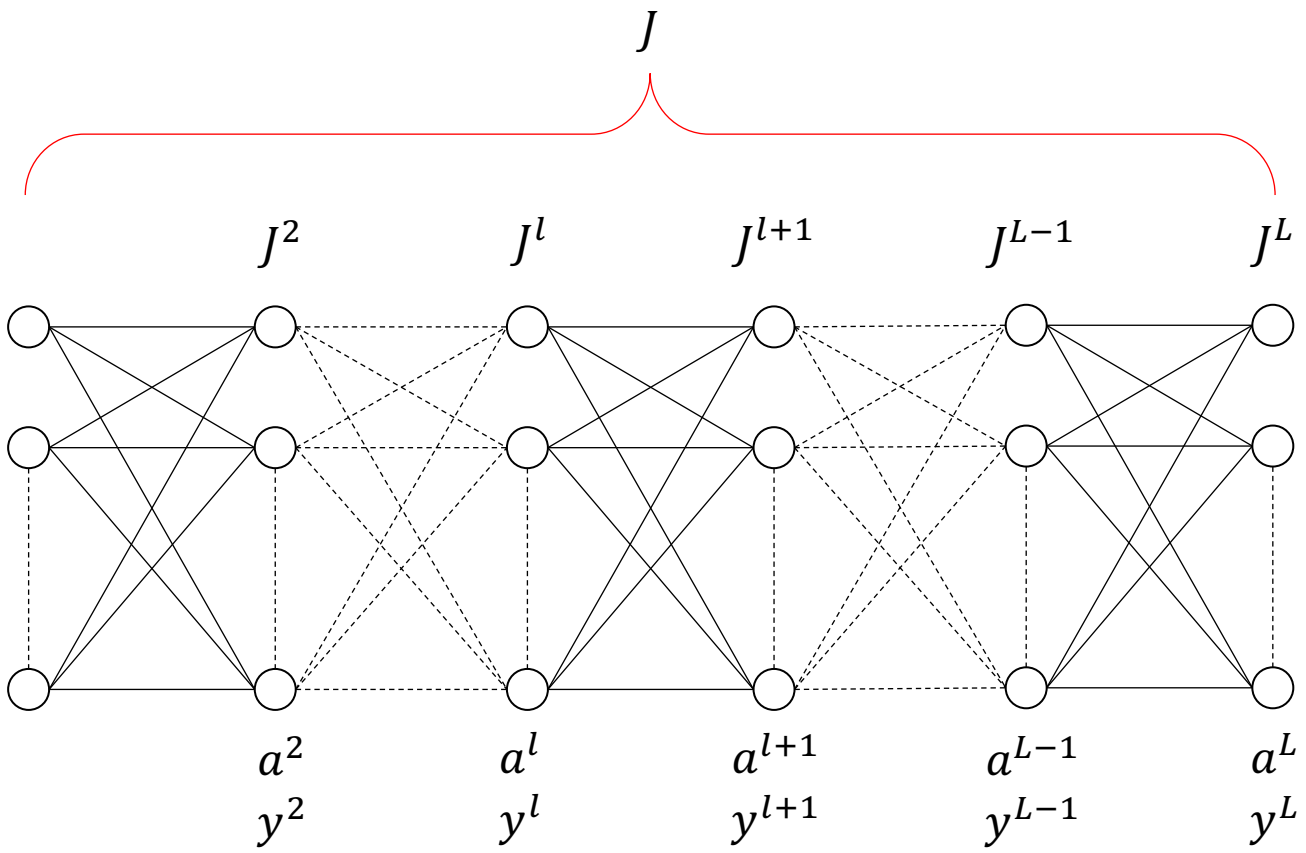
δ_i^l ← z_i^l → a_i^l

桥梁





多目标输出神经网络的BP原理



性能函数

$$J^l = \frac{1}{2}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2, (l = 2, \cdots, L)$$

$$J = \sum_{l=2}^{L} J^l = \frac{1}{2}\sum_{l=2}^{L}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2$$

网络预测

$$a^l = \begin{bmatrix} a_1^l \\ \vdots \\ a_{n_l}^l \end{bmatrix}$$

$$(l = 2, \cdots, L)$$

多目标输出

$$y^l = \begin{bmatrix} y_1^l \\ \vdots \\ y_{n_l}^l \end{bmatrix}$$

$$(l = 2, \cdots, L)$$

多目标输出神经网络的BP原理

最速下降法

$$J^l = \frac{1}{2}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2, (l = 2, \cdots, L)$$

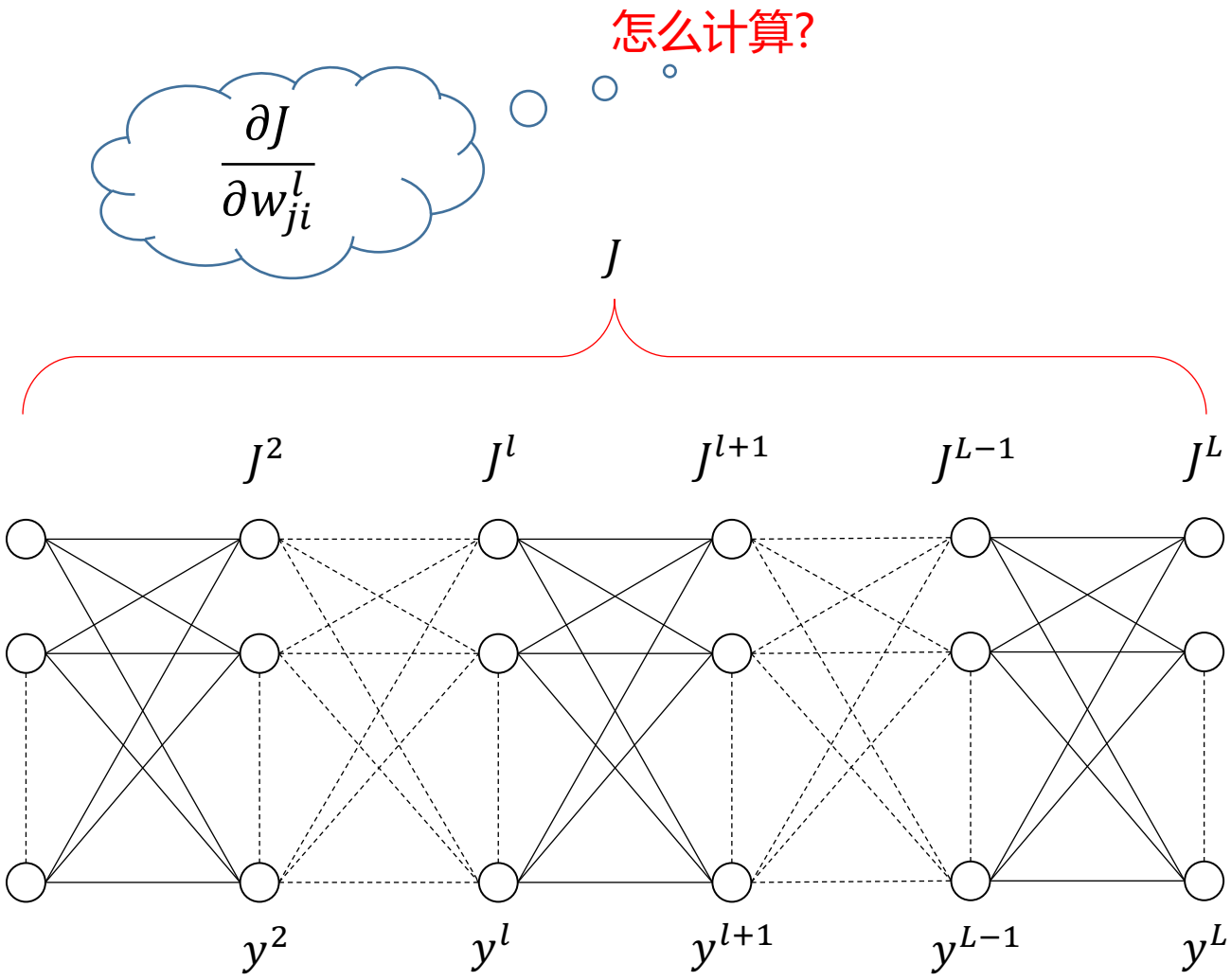
$$J = \sum_{l=2}^{L} J^l = \frac{1}{2}\sum_{l=2}^{L}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2$$

1. 计算

$$\frac{\partial J}{\partial w_{ji}^l}$$

2. 迭代更新

$$w_{ji}^l \leftarrow w_{ji}^l - \alpha \cdot \frac{\partial J}{\partial w_{ji}^l}$$



l layer

$$a_i^l = f(z_i^l)$$

define $\delta_i^l = \frac{\partial J}{\partial z_i^l}$

δ_i^l 和 $\frac{\partial J}{\partial w_{ji}^l}$ 的关系

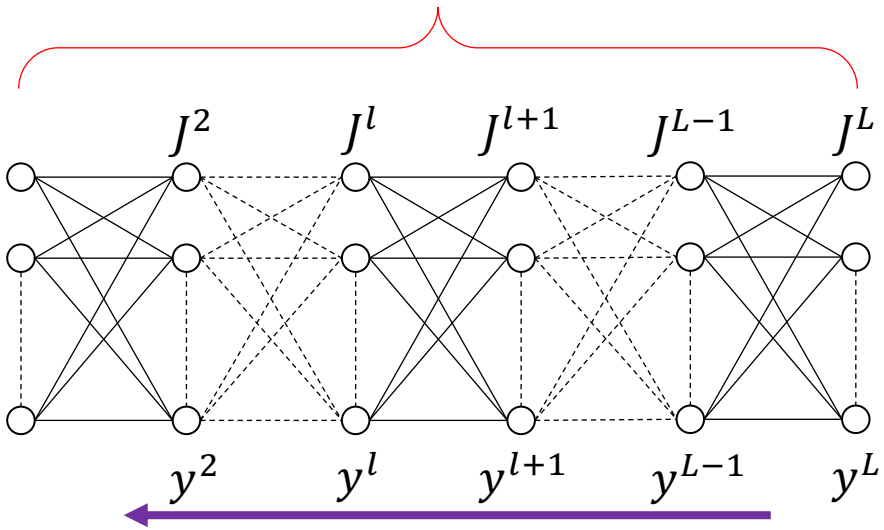
$$\frac{\partial J}{\partial w_{ji}^l} = \delta_j^{l+1} \cdot a_i^l$$

为什么?

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

$$J^l = \frac{1}{2}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2, (l = 2, \cdots, L)$$

$$J = \sum_{l=2}^{L} J^l = \frac{1}{2}\sum_{l=2}^{L}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2$$



问题：我们可以反向计算 δ^l 吗?

第l层

$$a_i^l = f(z_i^l)$$

$$\delta_i^l = \frac{\partial J}{\partial z_i^l}$$

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

w_{ji}^l

第 $l+1$ 层

$$\delta_j^{l+1} = \frac{\partial J}{\partial z_j^{l+1}}$$

$$z_j^{l+1} = \sum_{i=1}^{n_l} w_{ji}^l a_i^l$$

Step 1: 计算最后一层的 δ^L

第L 层

$$a_i^L = f(z_i^L)$$

$$\delta_i^L = \frac{\partial J}{\partial z_i^L}$$

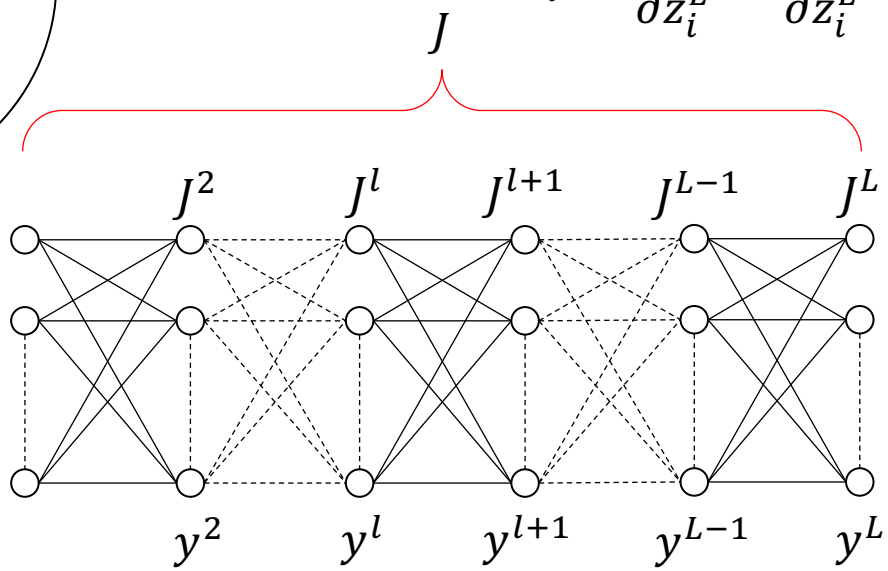
$$J^l = \frac{1}{2}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2, (l = 2, \cdots, L)$$

$$J = \sum_{l=2}^{L} J^l = \frac{1}{2}\sum_{l=2}^{L}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2$$

可推出,

$$\delta_i^L = \frac{\partial J}{\partial z_i^L} = \frac{\partial J^L}{\partial z_i^L} = \frac{1}{2}\cdot\frac{\partial\left(a_i^l - y_i^l\right)^2}{\partial z_i^L} = \left(a_i^L - y_i^L\right)\cdot\frac{\partial a_i^L}{\partial z_i^L} = \left(a_i^L - y_i^L\right)\cdot\dot{f}(z_i^L)$$

$a_i^L = f(z_i^L)$

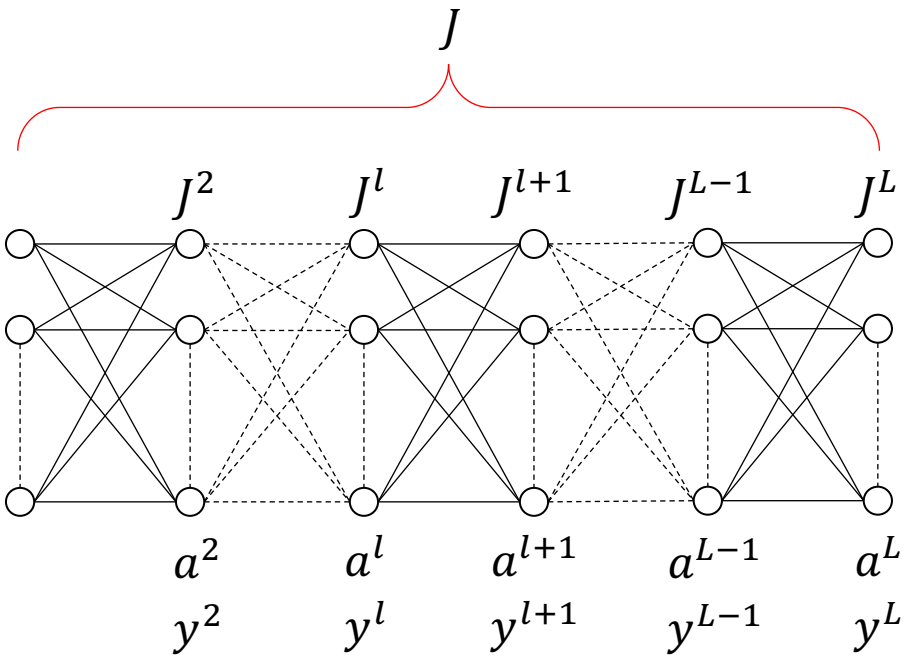


Step 2: δ^l 和 δ^{l+1} 的关系

$$J^l = \frac{1}{2}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2 = \frac{1}{2}\sum_{i=1}^{n_l}\left(f_i^l\left(z_i^l\right) - y_i^l\right)^2, (l = 2, \cdots, L)$$

$$J = \sum_{l=2}^{L} J^l = \sum_{r=2}^{l-1} J^r + J^l + \sum_{r=l+1}^{L} J^r = \frac{1}{2}\sum_{l=2}^{L}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2$$

J 不但对 z_i^l 有显式依赖关系，而且对 z_i^l 还有隐式依赖关系。



Step 2: δ^l 和 δ^{l+1} 的关系

$$J^l = \frac{1}{2}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2 = \frac{1}{2}\sum_{i=1}^{n_l}\left(f_i^l\left(z_i^l\right) - y_i^l\right)^2, (l = 2, \cdots, L)$$

$$J = \sum_{l=2}^{L} J^l = \sum_{r=2}^{l-1} J^r + J^l + \sum_{r=l+1}^{L} J^r = \frac{1}{2}\sum_{l=2}^{L}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2$$

J 不但对 z_i^l 有显式依赖关系，而且对 z_i^l 还有隐式依赖关系。为了避免解释偏导数时产生歧义，定义：$z_i^l(*) = z_i^l$

于是，有

$$\delta_i^l = \frac{\partial J}{\partial z_i^l} = \frac{\partial J}{\partial z_i^l(*)} \cdot \frac{\partial z_i^l(*)}{\partial z_i^l} + \sum_{j=1}^{n_{l+1}} \frac{\partial J}{\partial z_j^{l+1}} \cdot \frac{\partial z_j^{l+1}}{\partial z_i^l}$$

一个说明性的例子

$$J = x + y, y = \exp(x)$$

$$\frac{\partial J}{\partial x} = \frac{\partial J}{\partial x} + \frac{\partial J}{\partial y} \cdot \frac{\partial y}{\partial x}$$

$$x^* = x$$

$$\frac{\partial J}{\partial x} = \frac{\partial J}{\partial x^*} \cdot \frac{\partial x^*}{\partial x} + \frac{\partial J}{\partial y} \cdot \frac{\partial y}{\partial x}$$

Step 2: δ^l 和 δ^{l+1} 的关系

$$\delta_i^l = \frac{\partial J}{\partial z_i^l} = \frac{\partial J}{\partial z_i^l(*)} \cdot \frac{\partial z_i^l(*)}{\partial z_i^l} + \sum_{j=1}^{n_{l+1}} \frac{\partial J}{\partial z_j^{l+1}} \cdot \frac{\partial z_j^{l+1}}{\partial z_i^l}$$

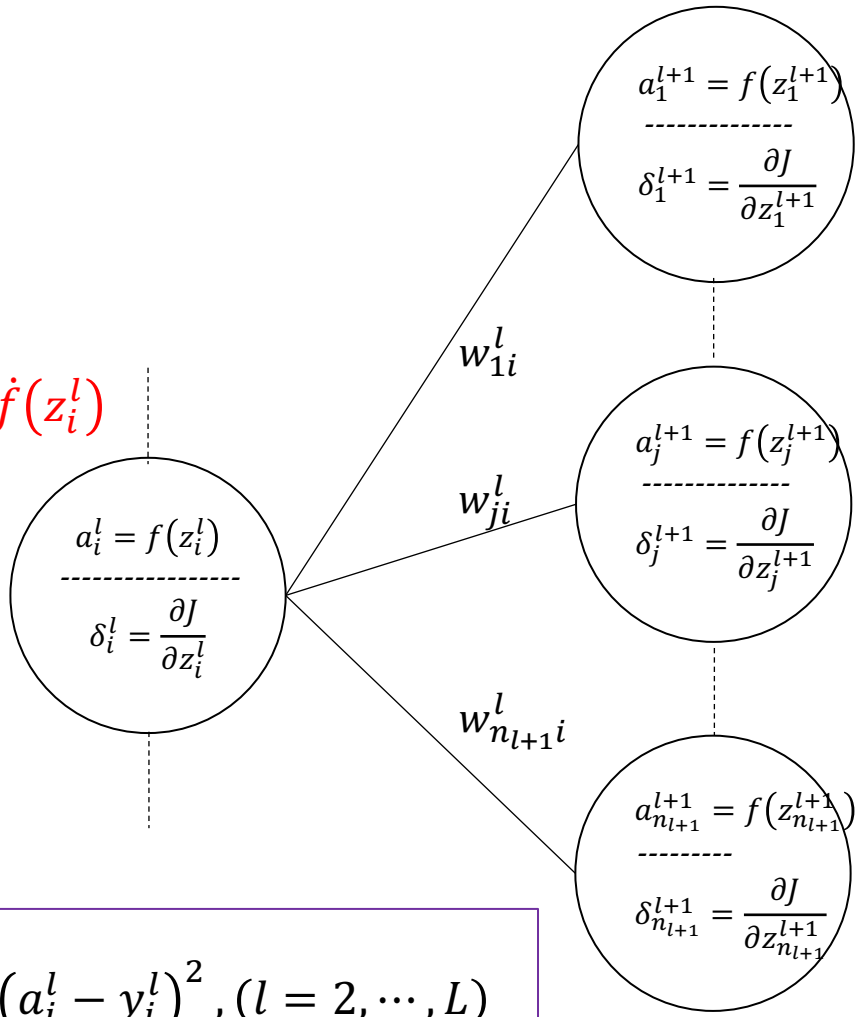
$$\frac{\partial J}{\partial z_i^l(*)} \cdot \frac{\partial z_i^l(*)}{\partial z_i^l} = \frac{\partial J^l}{\partial z_i^l} = \frac{1}{2} \cdot \frac{\partial \left(a_i^l - y_i^l\right)^2}{\partial z_i^l} = \left(a_i^l - y_i^l\right) \cdot \frac{\partial a_i^l}{\partial z_i^l} = \left(a_i^l - y_i^l\right) \cdot \dot{f}\left(z_i^l\right)$$

$$\sum_{j=1}^{n_{l+1}} \frac{\partial J}{\partial z_j^{l+1}} \cdot \frac{\partial z_j^{l+1}}{\partial z_i^l} = \sum_{j=1}^{n_{l+1}} \delta_j^{l+1} \cdot \frac{\partial z_j^{l+1}}{\partial z_i^l} = \dot{f}\left(z_i^l\right) \cdot \left(\sum_{j=1}^{n_{l+1}} \delta_j^{l+1} \cdot w_{ji}^l \right)$$

$$\frac{\partial z_j^{l+1}}{\partial z_i^l} = w_{ji}^l \cdot \frac{\partial a_j^l}{\partial z_i^l} = w_{ji}^l \cdot \dot{f}\left(z_i^l\right)$$

$$z_j^{l+1} = \sum_{i=1}^{n_l} w_{ji}^l a_j^l$$
$$a_i^{l+1} = f\left(z_i^{l+1}\right)$$

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



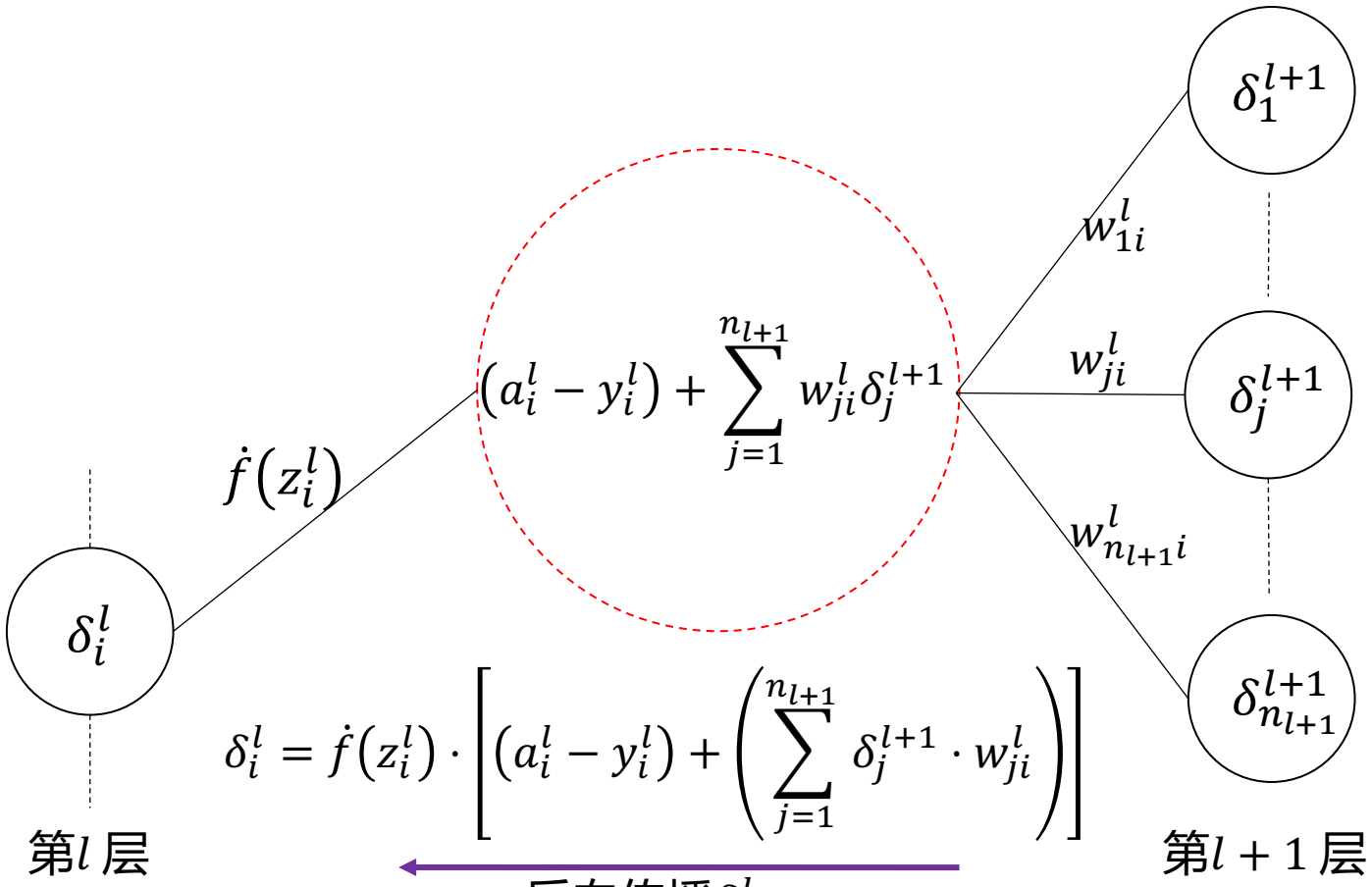
$$J^l = \frac{1}{2} \sum_{i=1}^{n_l} \left(a_i^l - y_i^l\right)^2, (l = 2, \cdots, L)$$

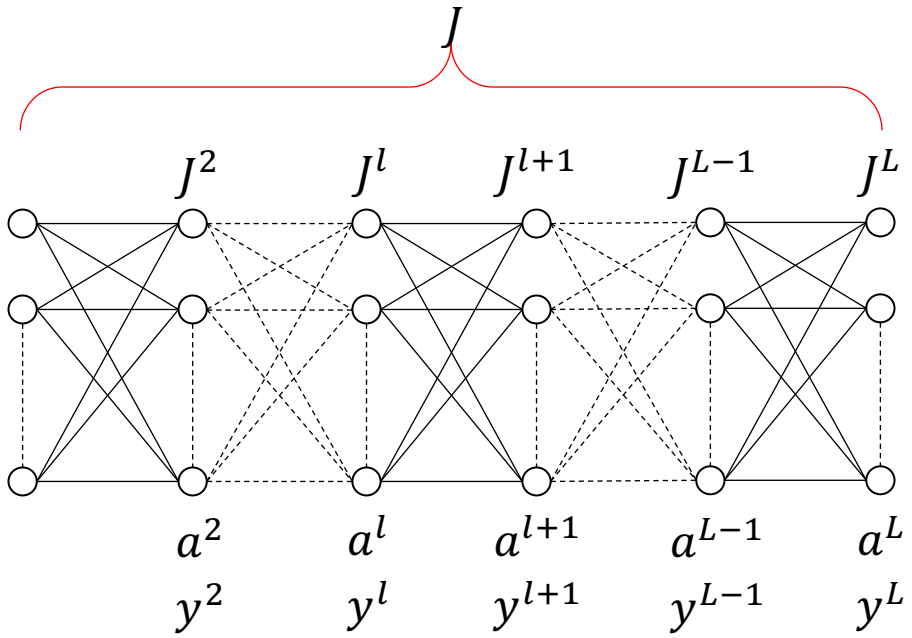
$$J = \sum_{l=2}^{L} J^l = \frac{1}{2} \sum_{l=2}^{L} \sum_{i=1}^{n_l} \left(a_i^l - y_i^l\right)^2$$

Step 3: 反向传播 δ^l

$$J^l = \frac{1}{2}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2, (l = 2, \cdots, L)$$

$$J = \sum_{l=2}^{L} J^l = \frac{1}{2}\sum_{l=2}^{L}\sum_{i=1}^{n_l}\left(a_i^l - y_i^l\right)^2$$





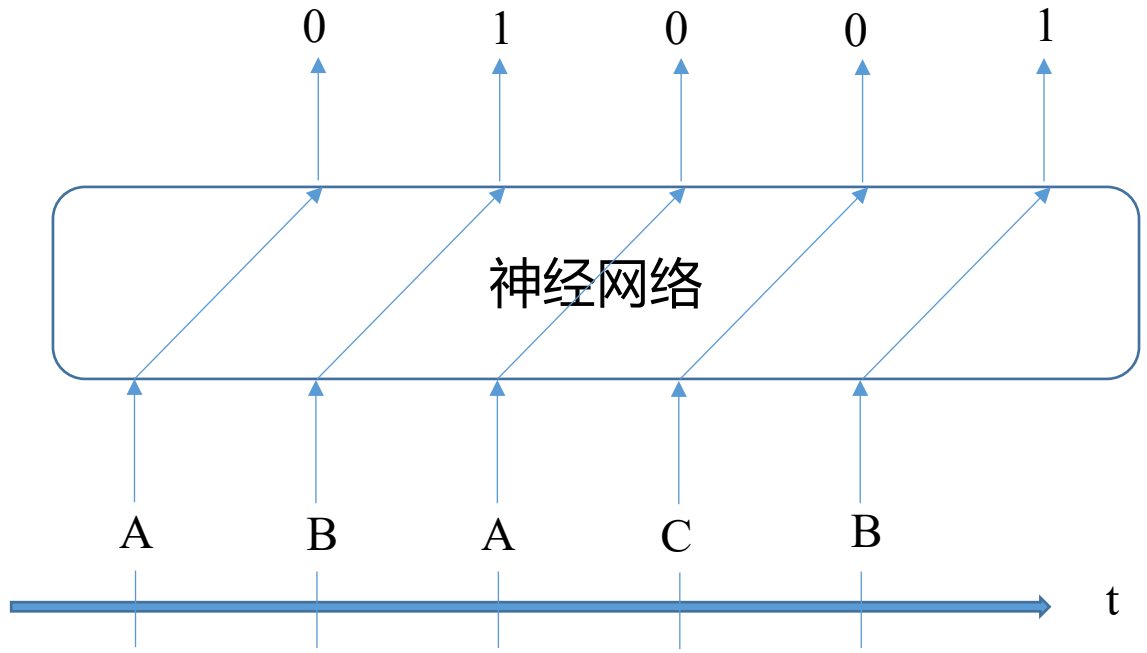
说明示例：序列识别

任务: Recognize A followed by B.

生成的序列

1. ABACB
2. CCBBA
3. CACCB
4. ACCCB
5. CACBC
6. AAACB
7. BAACB
8. CCBAB
9. BCCAB
10. CABAC

……



说明示例：序列识别

输入编码：

$$A=\begin{bmatrix}1\\0\\0\end{bmatrix}\quad B=\begin{bmatrix}0\\1\\0\end{bmatrix}\quad C=\begin{bmatrix}0\\0\\1\end{bmatrix}$$

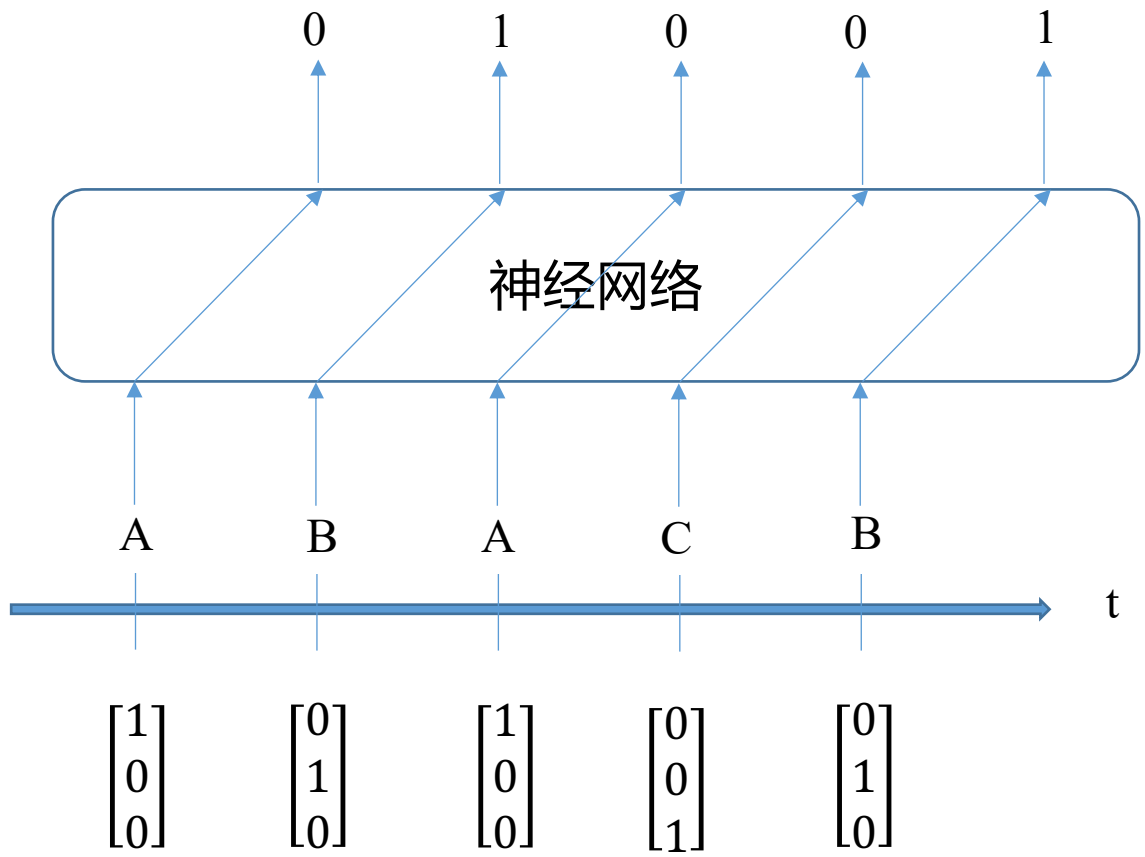
生成序列编码：

1. A B C A B

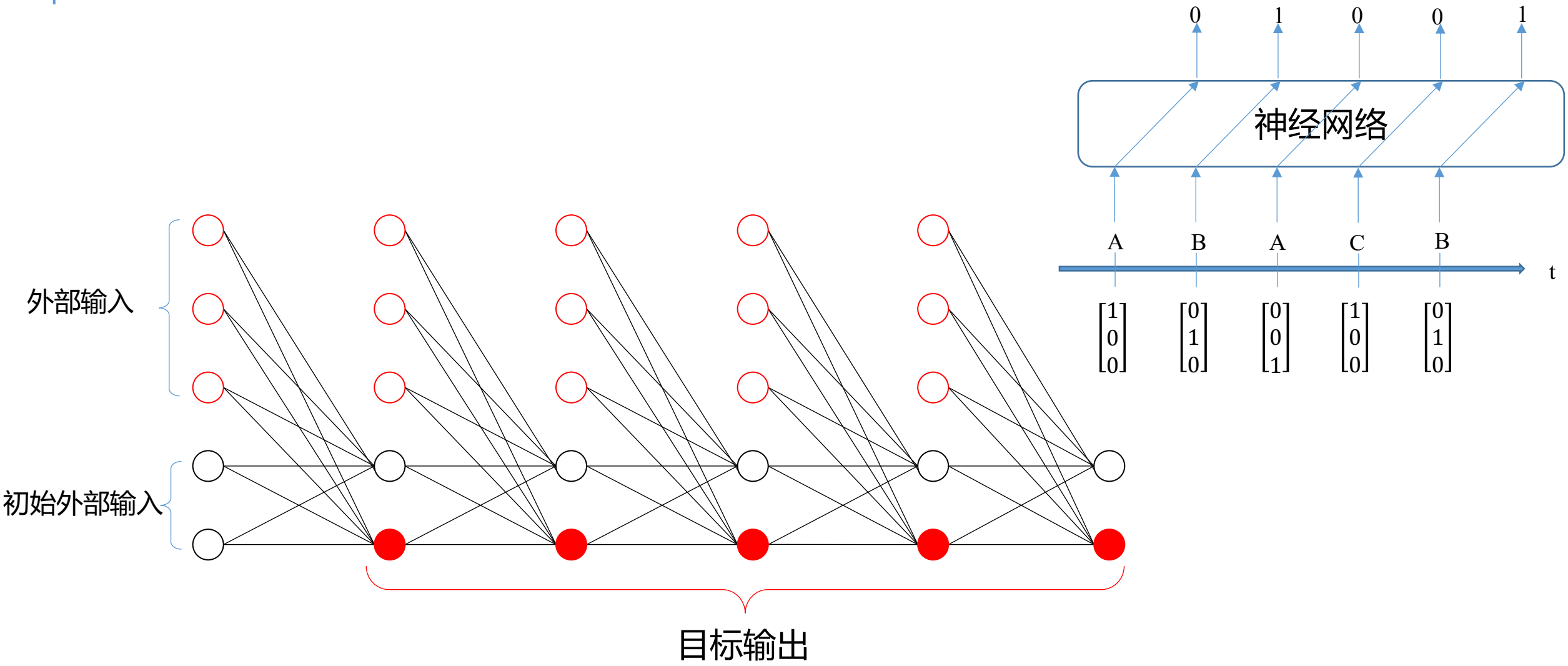
$$\begin{bmatrix}1\\0\\0\end{bmatrix}\ \begin{bmatrix}0\\1\\0\end{bmatrix}\ \begin{bmatrix}0\\0\\1\end{bmatrix}\ \begin{bmatrix}1\\0\\0\end{bmatrix}\ \begin{bmatrix}0\\1\\0\end{bmatrix}$$

2. C A C C B

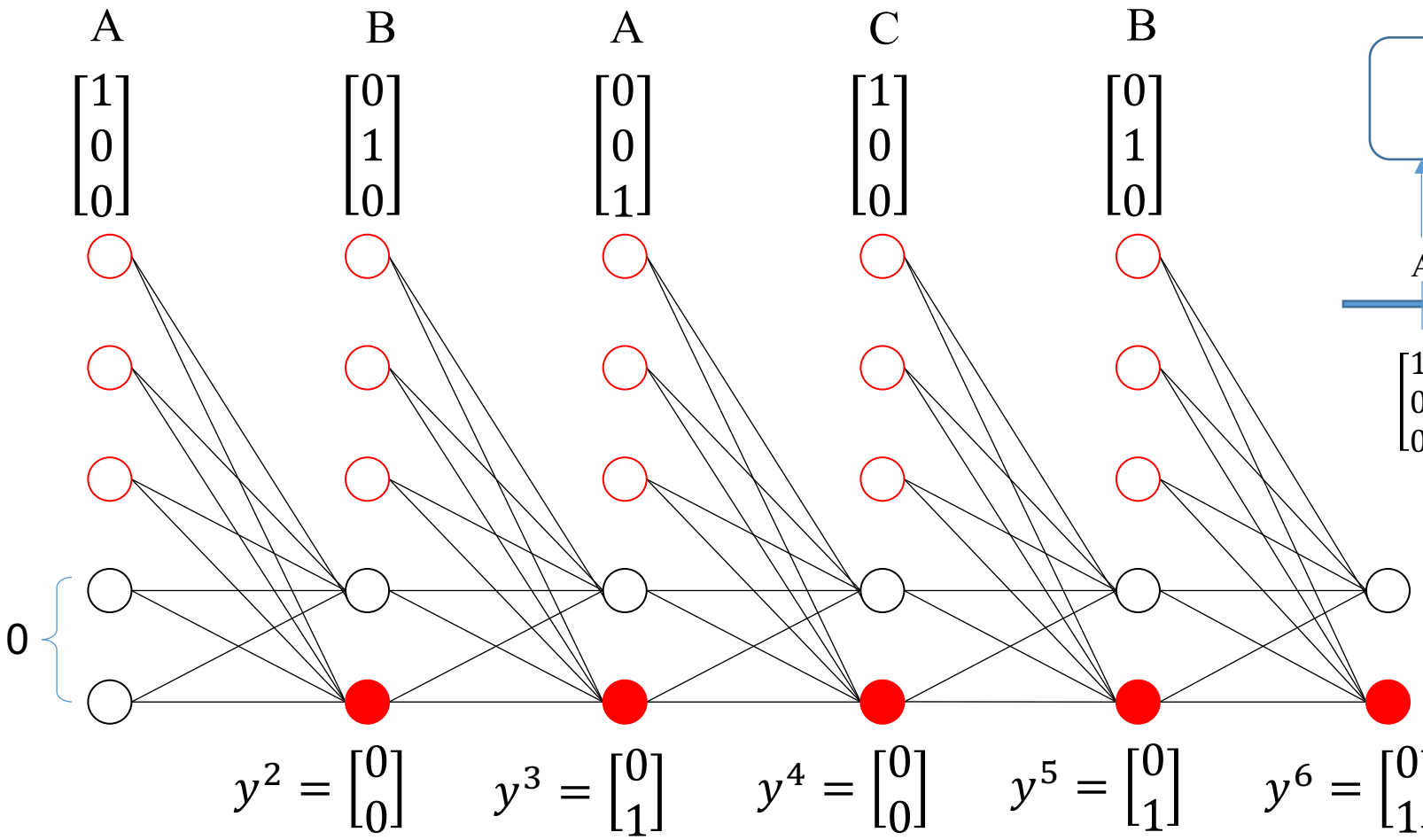
$$\begin{bmatrix}0\\0\\1\end{bmatrix}\ \begin{bmatrix}1\\0\\0\end{bmatrix}\ \begin{bmatrix}0\\0\\1\end{bmatrix}\ \begin{bmatrix}0\\0\\1\end{bmatrix}\ \begin{bmatrix}0\\1\\0\end{bmatrix}$$

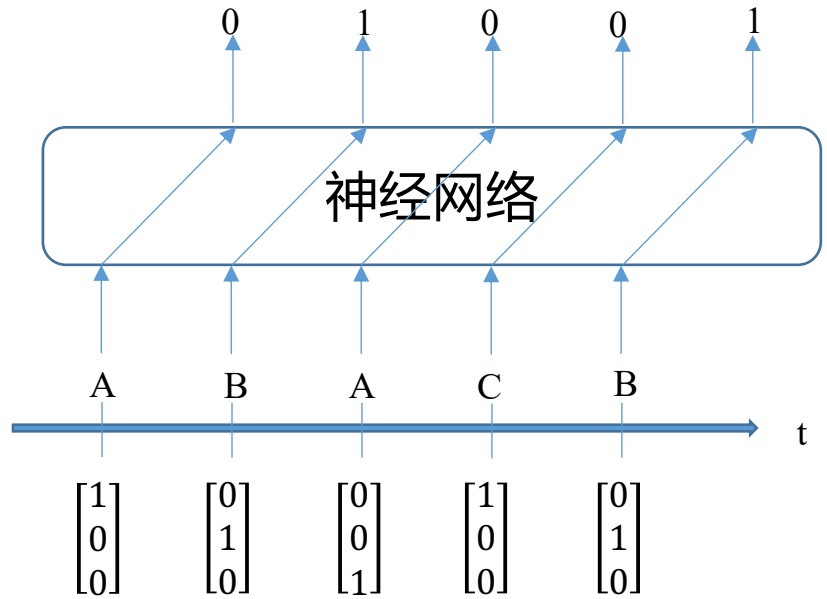


说明示例：序列识别



说明示例：序列识别


A
B
A
C
B
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\1\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
0
$y^2 = \begin{bmatrix}0\\0\end{bmatrix}$
$y^3 = \begin{bmatrix}0\\1\end{bmatrix}$
$y^4 = \begin{bmatrix}0\\0\end{bmatrix}$
$y^5 = \begin{bmatrix}0\\1\end{bmatrix}$
$y^6 = \begin{bmatrix}0\\1\end{bmatrix}$

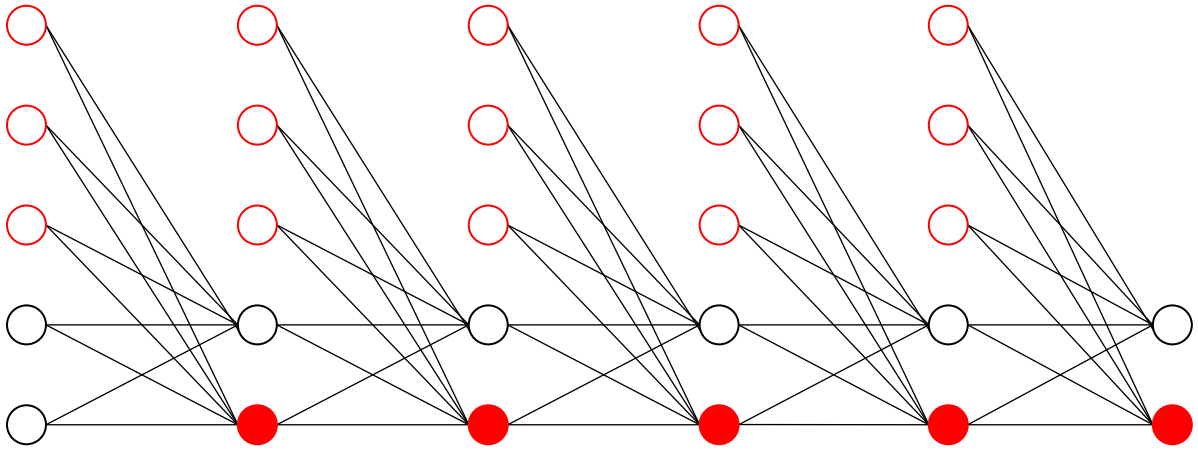

0
1
0
0
1
神经网络
A
B
A
C
B
t
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$
$\begin{bmatrix}0\\0\\1\end{bmatrix}$
$\begin{bmatrix}1\\0\\0\end{bmatrix}$
$\begin{bmatrix}0\\1\\0\end{bmatrix}$

说明示例：序列识别

生成训练序列

1. ABCAB
2. CCBBA
3. CACCB
4. ACCCB
5. CACBC
6. AAACB
7. BAACB
8. CCBAB
9. BCCAB
10. CABAC

……..



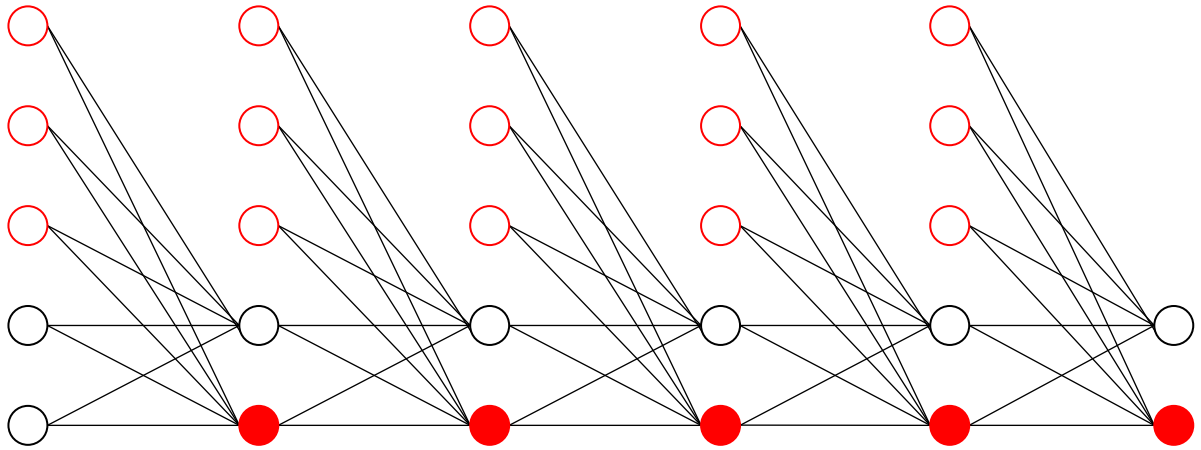
生成测试序列

1. CBCAC
2. ACBBA
3. BACCB
4. ACBCB
5. AACBC
6. BAACB
7. AAACB
8. CCBAB
9. BBCAB
10. AABAC

……..

说明示例：序列识别

$$A=\begin{bmatrix}1\\0\\0\end{bmatrix} \quad B=\begin{bmatrix}0\\1\\0\end{bmatrix} \quad C=\begin{bmatrix}0\\0\\1\end{bmatrix}$$



测试输出：

CAACB	→	0.0184	0.0001	0.0211	0.0801	0.9928
ABBCA	→	0.0179	0.9375	0.0267	0.0012	0.0000
AACBA	→	0.0179	0.0336	0.0286	0.8722	0.0000
CACBB	→	0.0184	0.0001	0.0170	0.8494	0.0013
BCAAA	→	0.0182	0.0001	0.0001	0.0622	0.0018

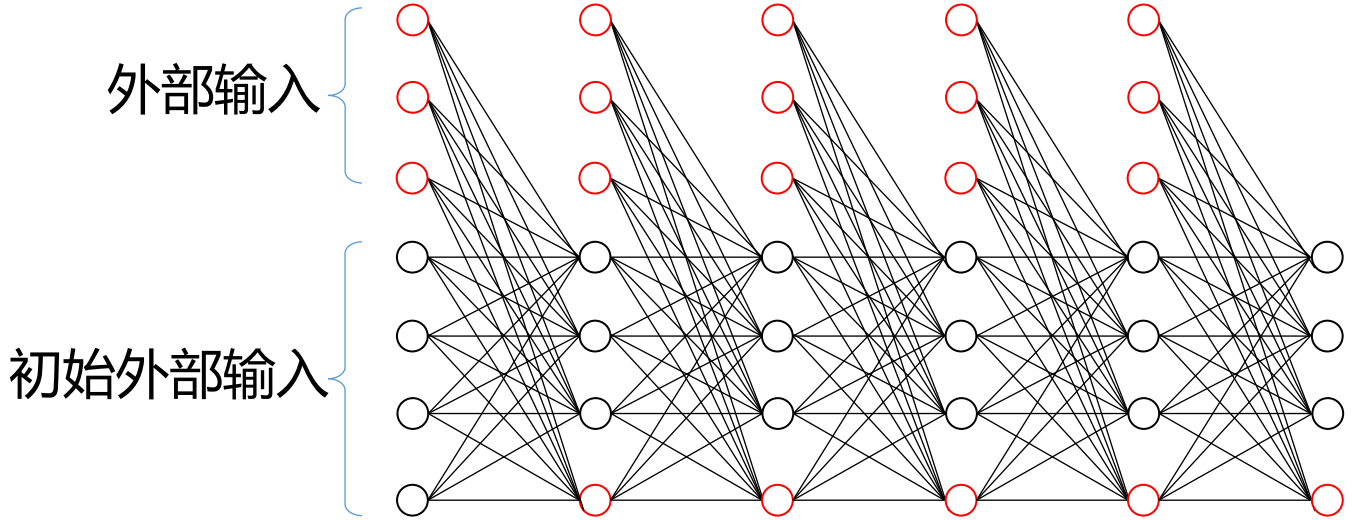
说明示例：序列识别

任务: Recognize A followed by B.

生成序列

1. ABCAB
2. CCBBA
3. CACCB
4. ACCCB
5. CACBC
6. AAACB
7. BAACB
8. CCBAB
9. BCCAB
10. CABAC

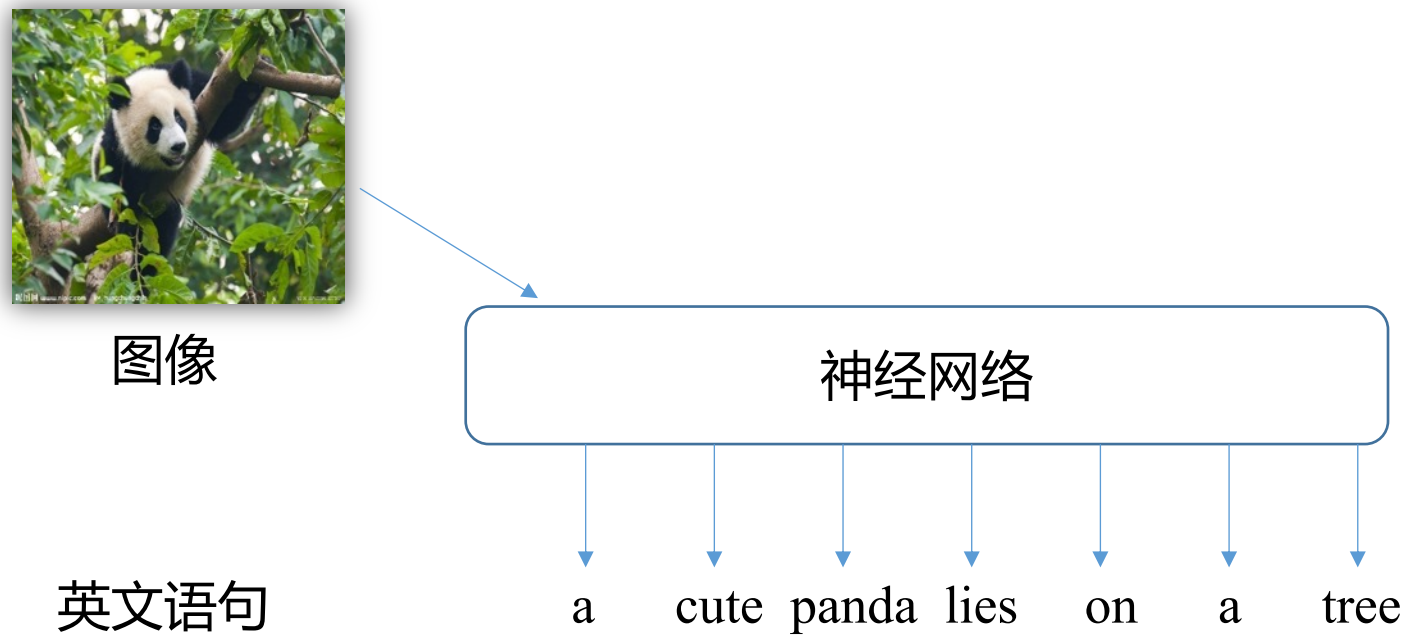
……



示例说明：Image Caption

Image Caption：

任务是使用格式正确的英语句子描述图像的内容

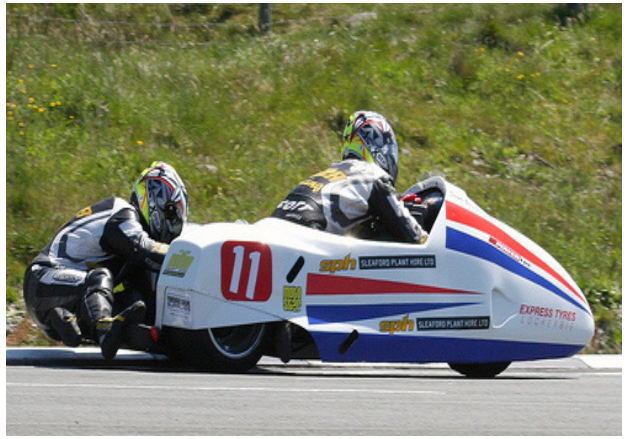


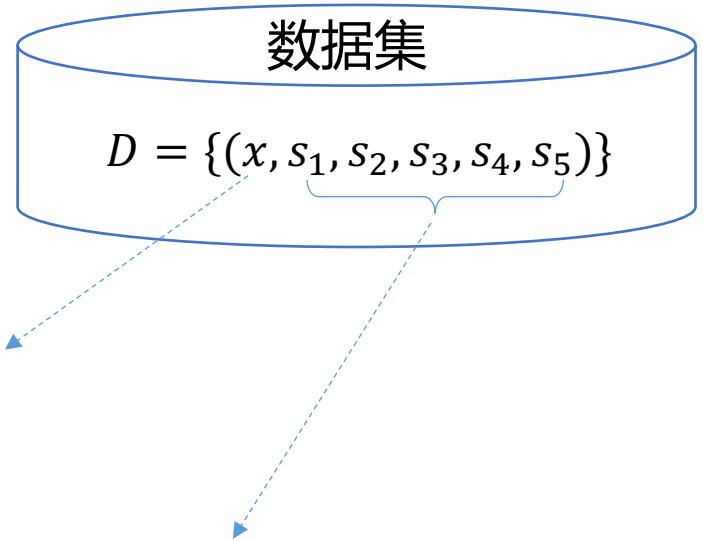
示例说明：Image Caption

数据集：COCO

COCO是微软赞助的新的图像识别、分割和字幕数据。
http://mscoco.org/dataset/#download
包含：

80,000 训练样本
40,000 验证样本
40,000 测试样本

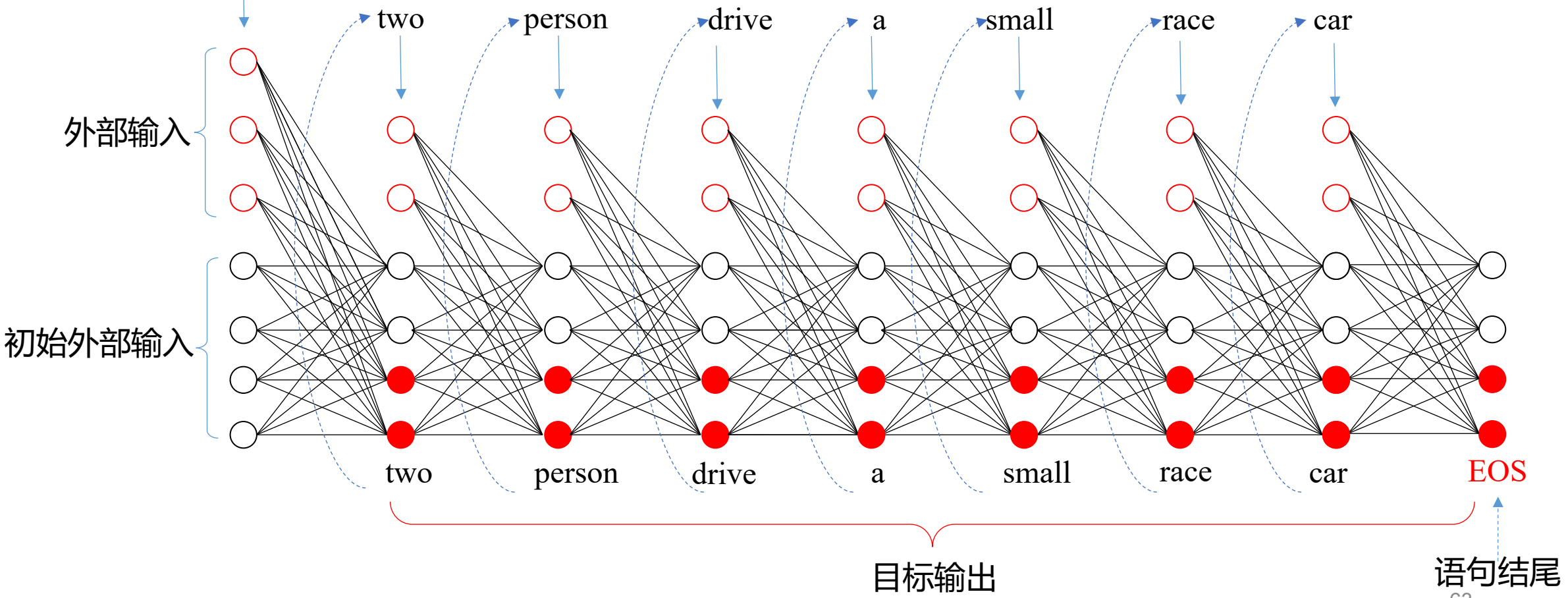




Microsoft

1.Two person drive a small race car .
2.Two racer drive a white bike down a road .
3.Two motorist be ride along on their vehicle that be oddly design and color .
4.Two person be in a small race car drive by a green hill .
5.Two person in race uniform in a street car .

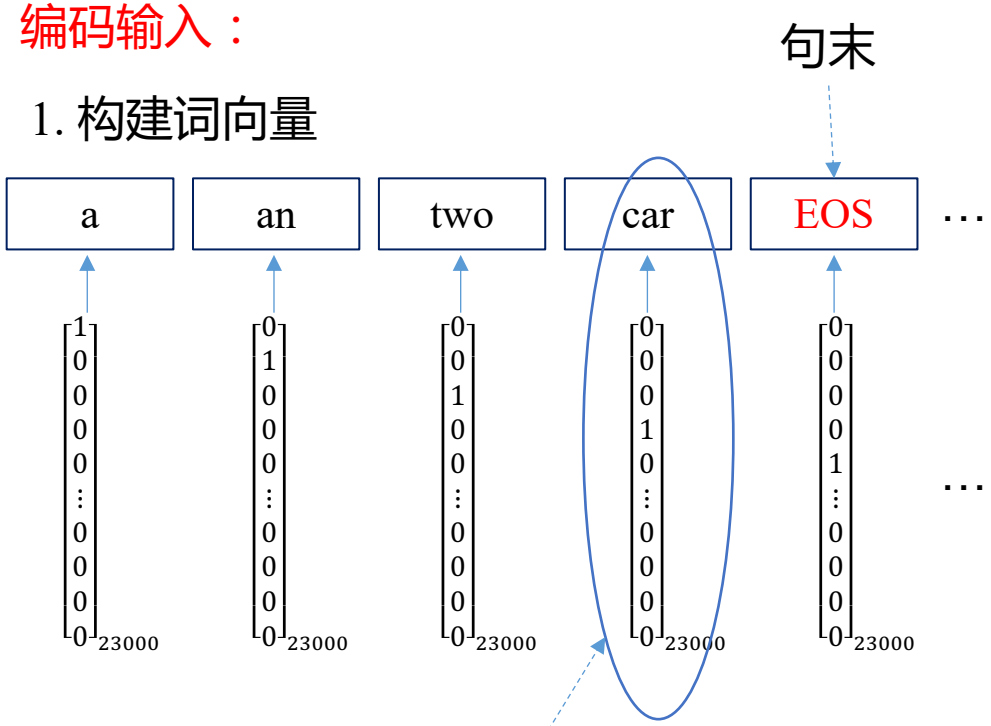
示例说明：Image Caption



示例说明：Image Caption

编码输入：

1. 构建词向量



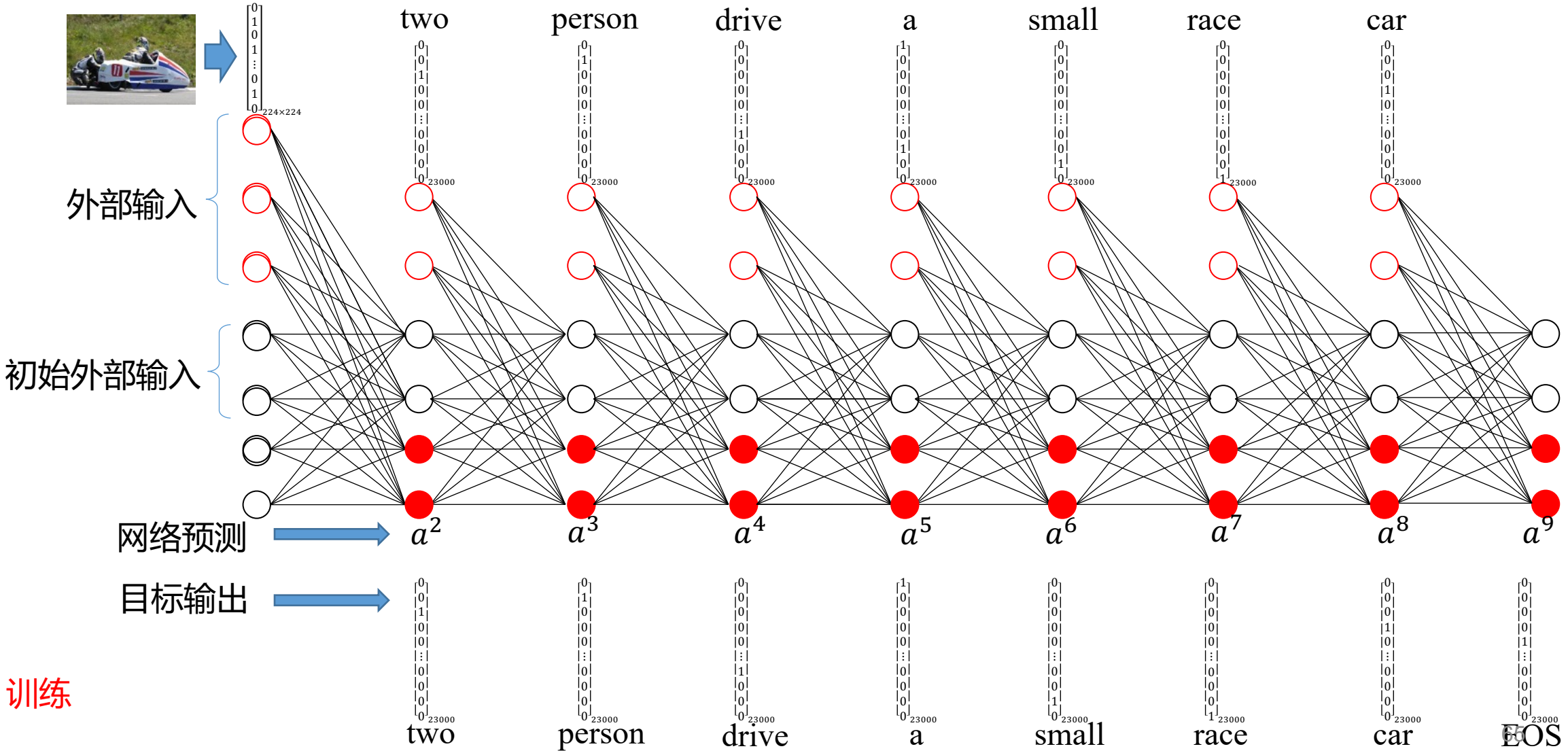
2. 对语句编码

s = two person drive a small race car EOS

3. 图片数字化

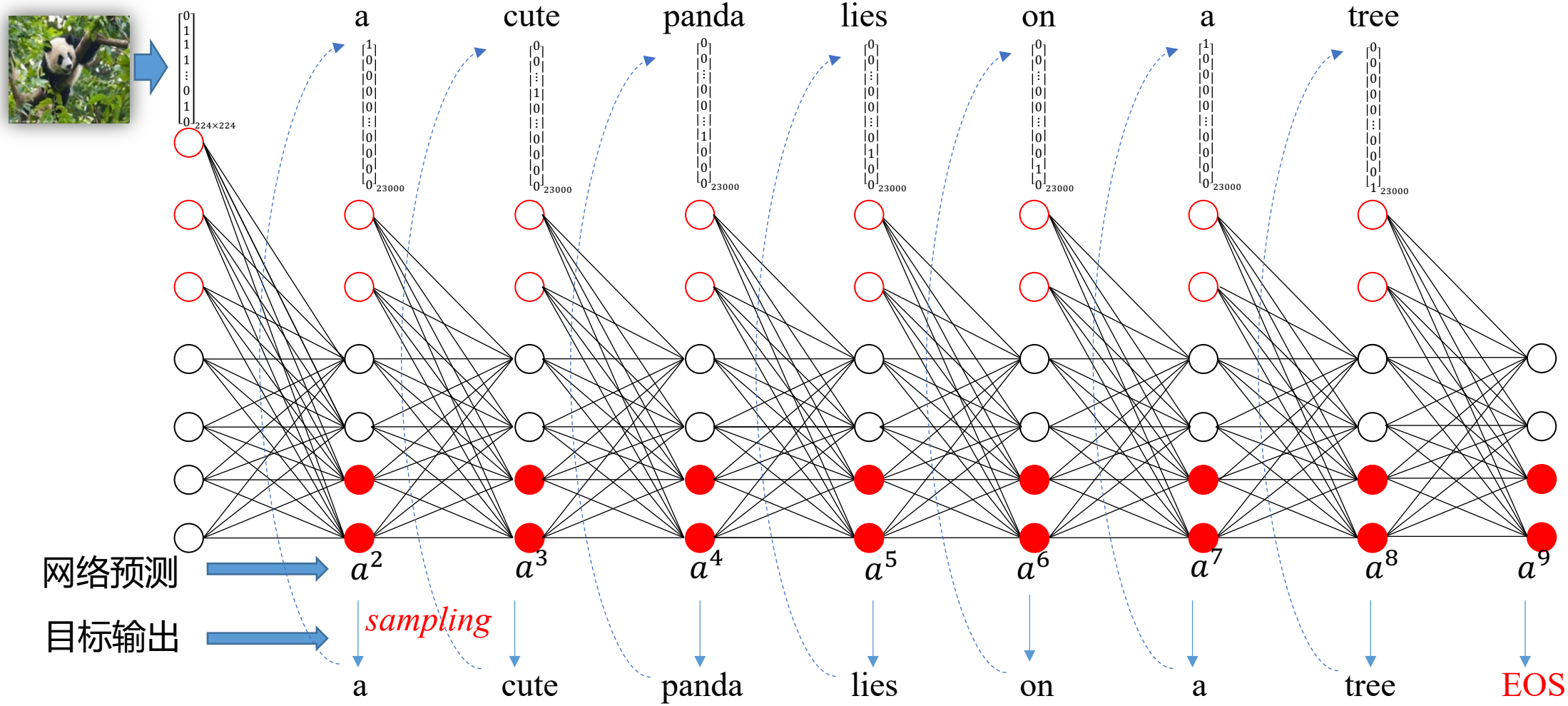
224×224

示例说明：Image Caption



训练

示例说明：Image Caption



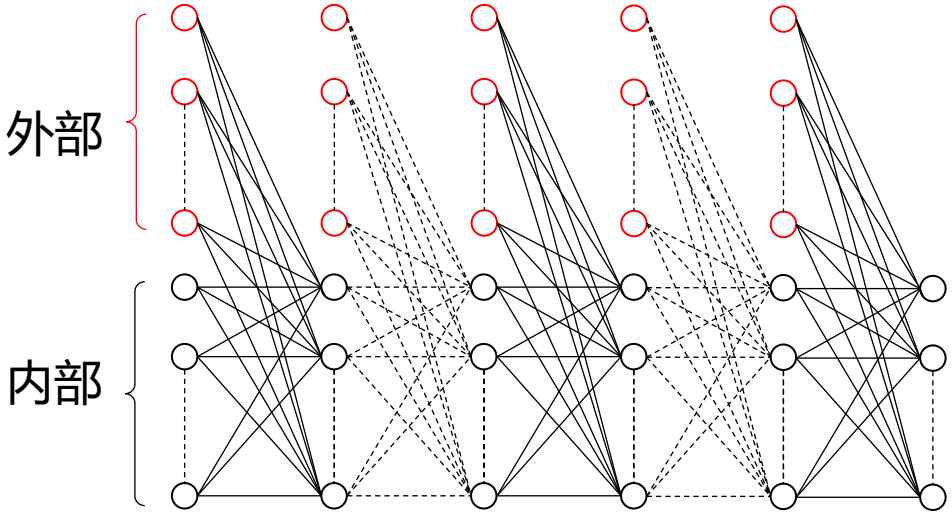
测试

一个接一个的生成单词，直到得到单词 'EOS'

示例说明：诗词创作

陆游
卜算子•咏梅

驿外断桥边，
寂寞开无主。
已是黄昏独自愁，
更著风和雨。
无意苦争春，
一任群芳妒。
零落成泥碾作尘，
只有香如故。

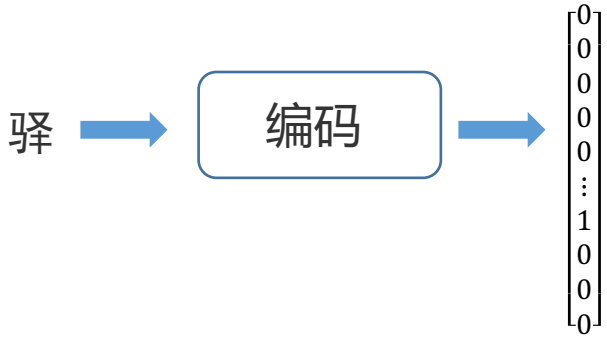


人工神经网络可以创作诗词吗？

毛泽东
卜算子•咏梅

风雨送春归，
飞雪迎春到。
已是悬崖百丈冰，
犹有花枝俏。
俏也不争春，
只把春来报。
待到山花烂漫时，
她在丛中笑。

示例说明：诗词创作

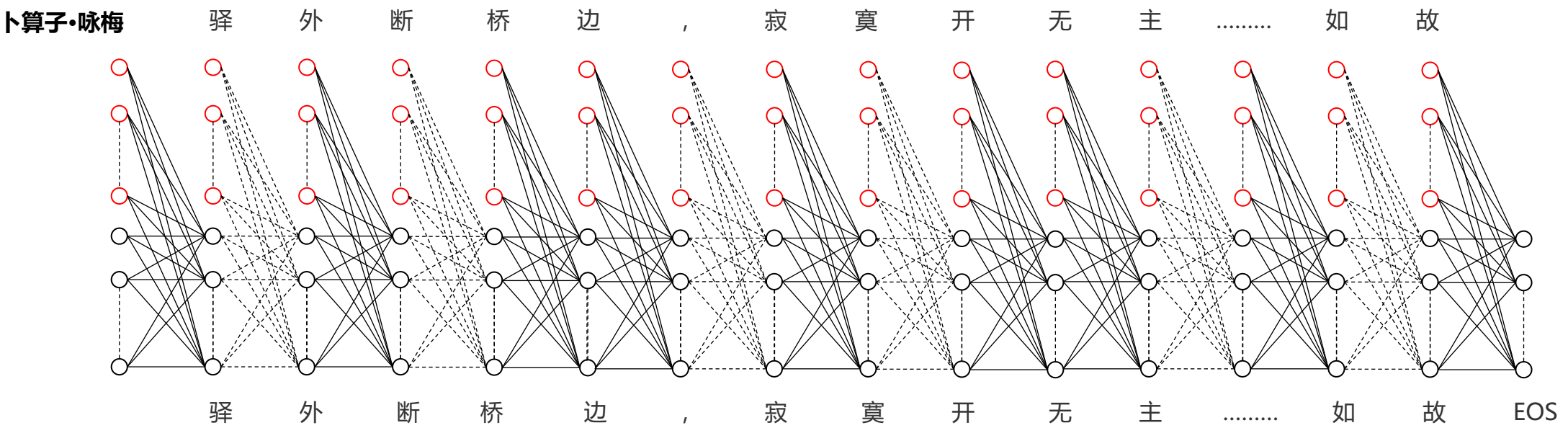


卜算子•咏梅

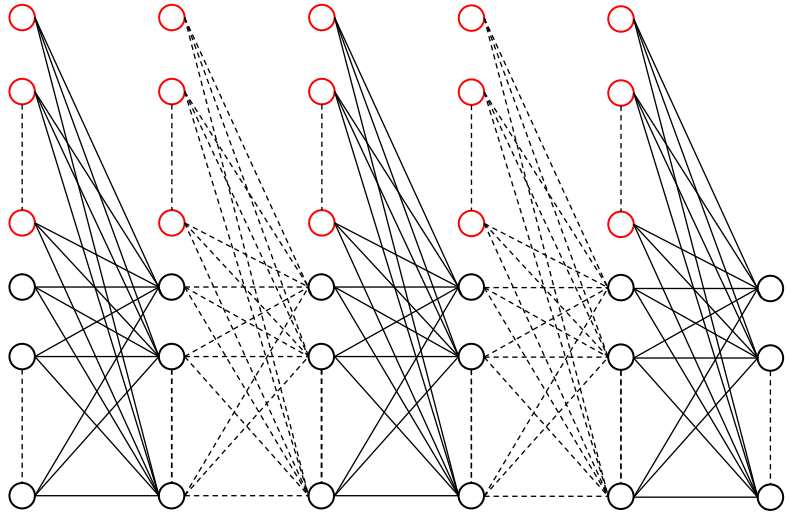
驿外断桥边，
寂寞开无主。
已是黄昏独自愁，
更著风和雨。
无意苦争春，
一任群芳妒。
零落成泥碾作尘，
只有香如故。

卜算子•咏梅

风雨送春归，
飞雪迎春到。
已是悬崖百丈冰，
犹有花枝俏。
俏也不争春，
只把春来报。
待到山花烂漫时，
她在丛中笑。



示例说明：诗词创作



卜算子•咏梅

花谢早疏篱，
几度陶潜里。
永日梅花昔底寒，
比向梅花妒。
荣悴幻非凡，
谓是娇芳伴。
后著金陵几日时，
中酒争先理。

卜算子•咏梅

朱阁见幽芳，
露叶梅花里。
玉屑琼台地屑琼，
瑞鹊惊飞尾。
便倚彩毫归，
枝上簪盐谱。
别作千秋一笑随，
好趁伊家笑。

卜算子•咏梅

小试买梅花，
并蒂栖香粉。
肯向红蕖似竹姿，
一叶清风许。
心思寺炉高，
深院松间曲。
万纸参差故与黎，
效我何知道。

作业

设计一个多目标输出神经网络来学习下面这个序列补全任务：序列的前两个字母唯一的确定了后面的四个数字。

下周上课前交（10/14）

Training Dataset

AA1212	AC1231	AD1221	AE1213
BA2312	BB2323	BC2331	BE2313
CB3123	CC3131	CD3121	CE3113
DA2112	DB2123	DC2131	DD2121
EA1312	EB1323	ED1321	EE1313

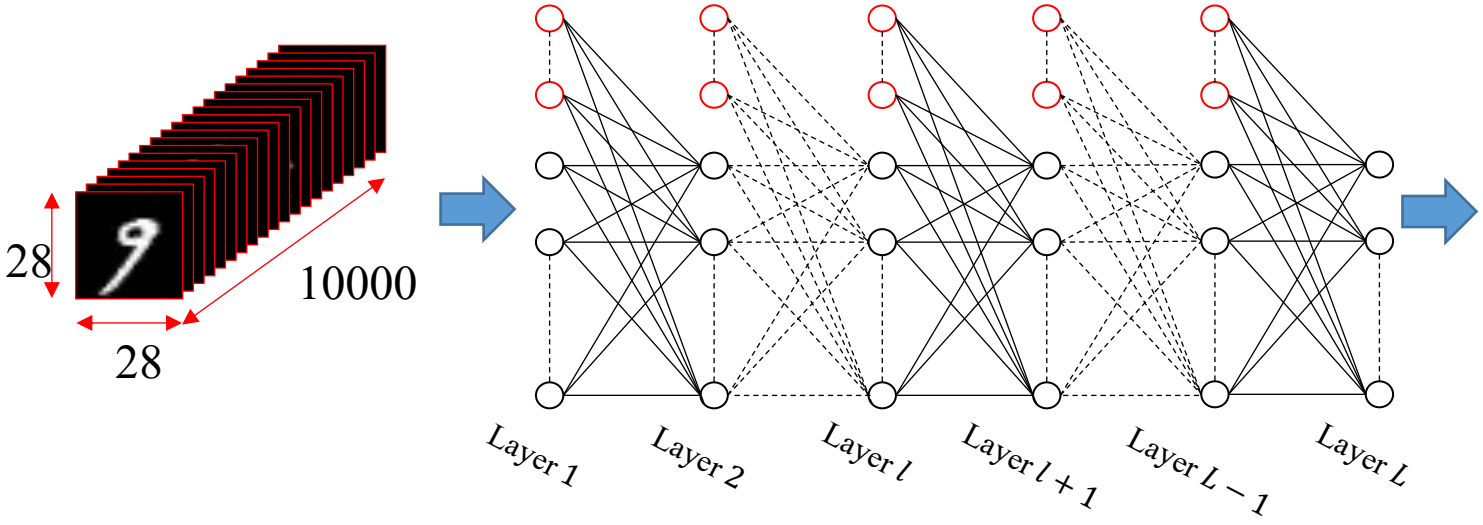
Testing Dataset

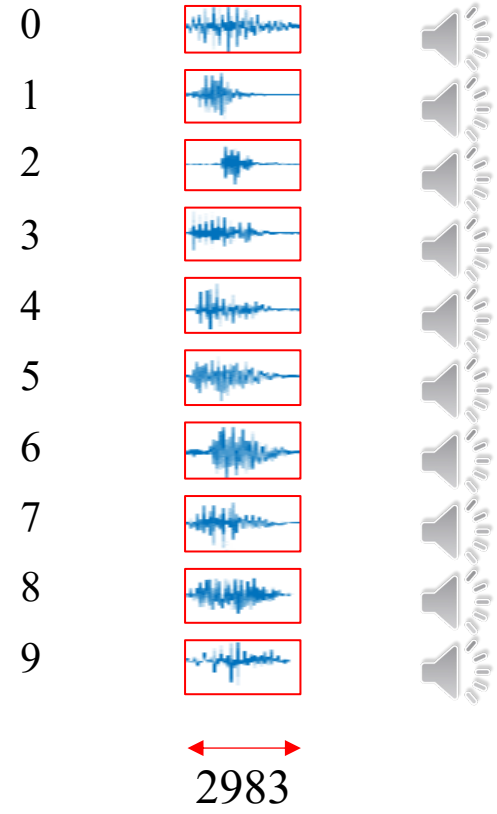
AB1223	BD2321	CA3112	DE2113	EC1331

作业（附加）

用程序实现一个手写数字语音转换器

非必需提交





课程信息

时间：2022年秋季学期 1-8周 周五 3-4节

线下：江安文科楼三区203

线上：

http://www.machineilab.org/

http://guoquan.net/

Thanks