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

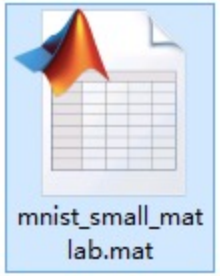
机器智能实验室

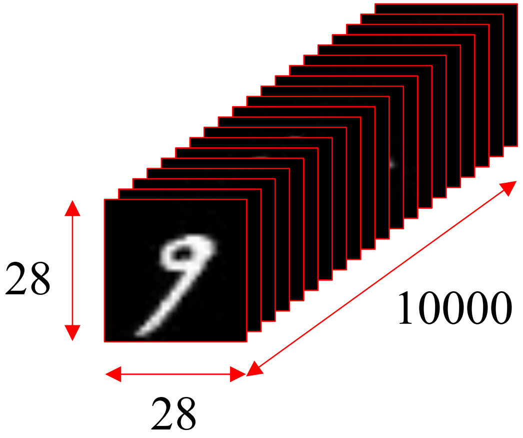
http://www.machineilab.org/

作业回顾

- 使用BP算法实现一个简单的手写数字识别。

  - 提供MATLAB模版和Python模版
  - 可以使用MATLAB或Python





回顾

Step 1:
数据准备

Step 2:
网络结构设计

Step 3:
网络参数初始化

Step 4:
定义网络性能函数

Step 5:
定义网络评价指标

Step 6:
训练网络

Step 7:
测试网络

Step 8:
存储网络

Step 9:
实际应用

第五章

深入理解BP算法

2022年 秋季

提纲

I

- 网络结构问题
- 学习算法问题
- 目标输出问题
- 网络输入问题

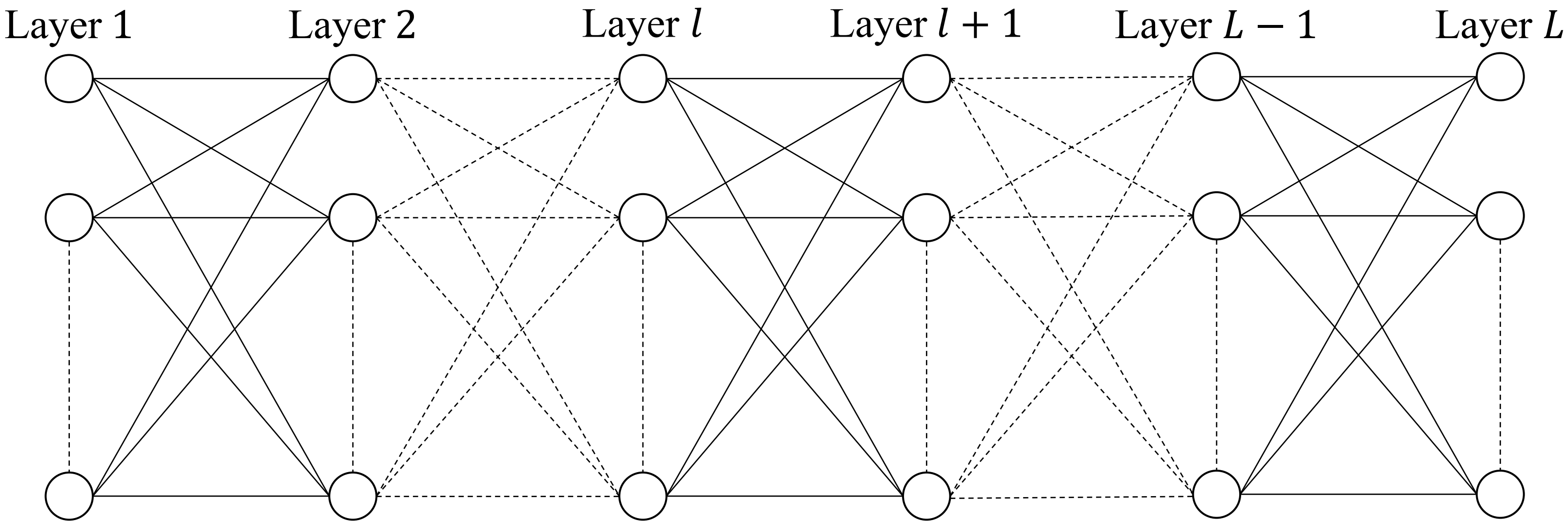
II

- 网络预测问题
- 性能函数问题
- 网络深度问题
- 训练数据问题

网络结构问题

两大重要特征：

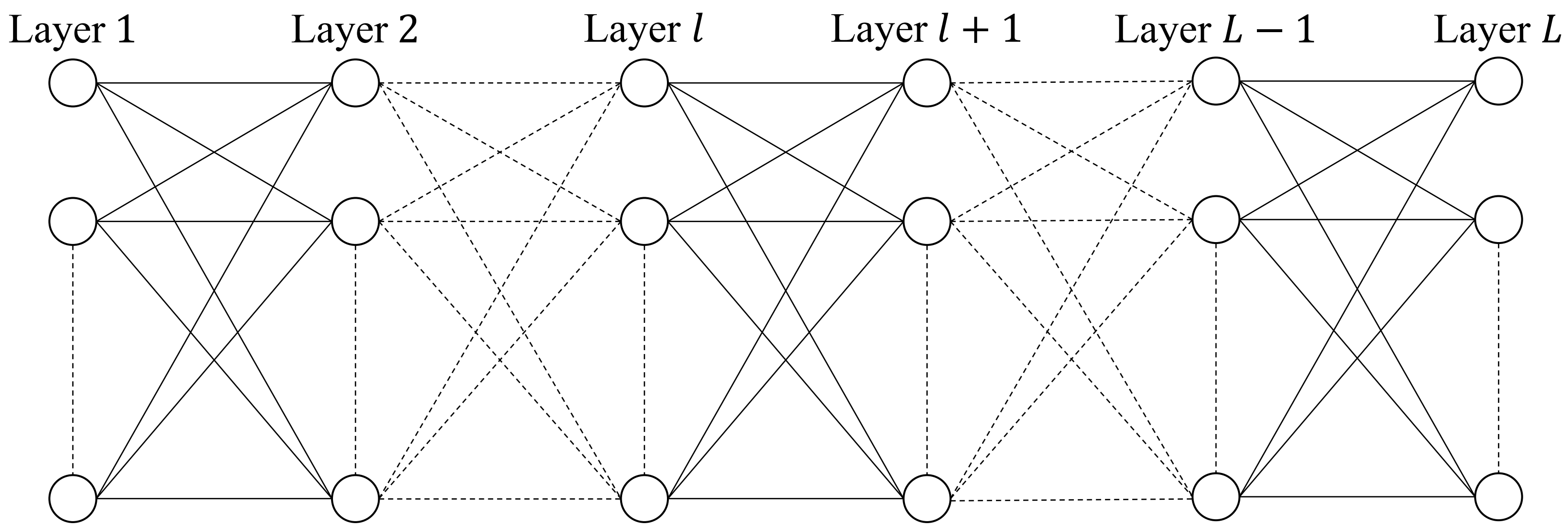
- 同层神经元间没有连接
- 跨层神经元间没有连接



神经元	n_1		n_2		n_l		n_{l+1}		n_{L-1}		n_L
连接权		W^1				W^l				W^{L-1}	
预测	a^1		a^2		a^l		a^{l+1}		a^{L-1}		a^L

网络结构问题

- 回复神经网络：
  - 每层神经元数量相同
  - 连接权矩阵共享

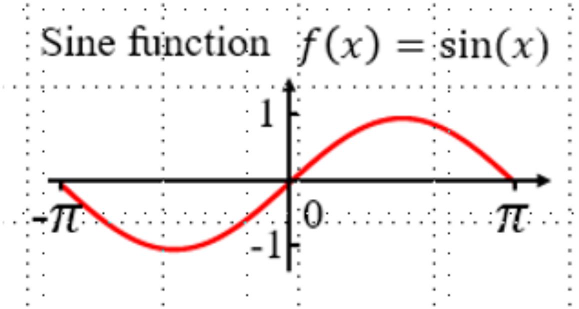


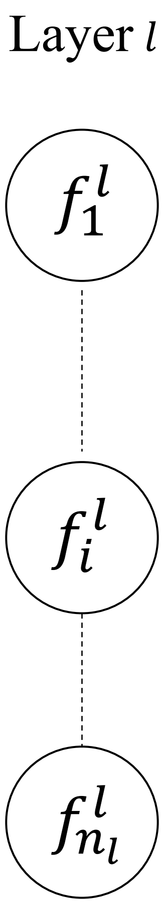
神经元	n		n		n		n		n		n
连接权		W				W				W	
预测	a^1		a^2		a^l		a^{l+1}		a^{L-1}		a^L

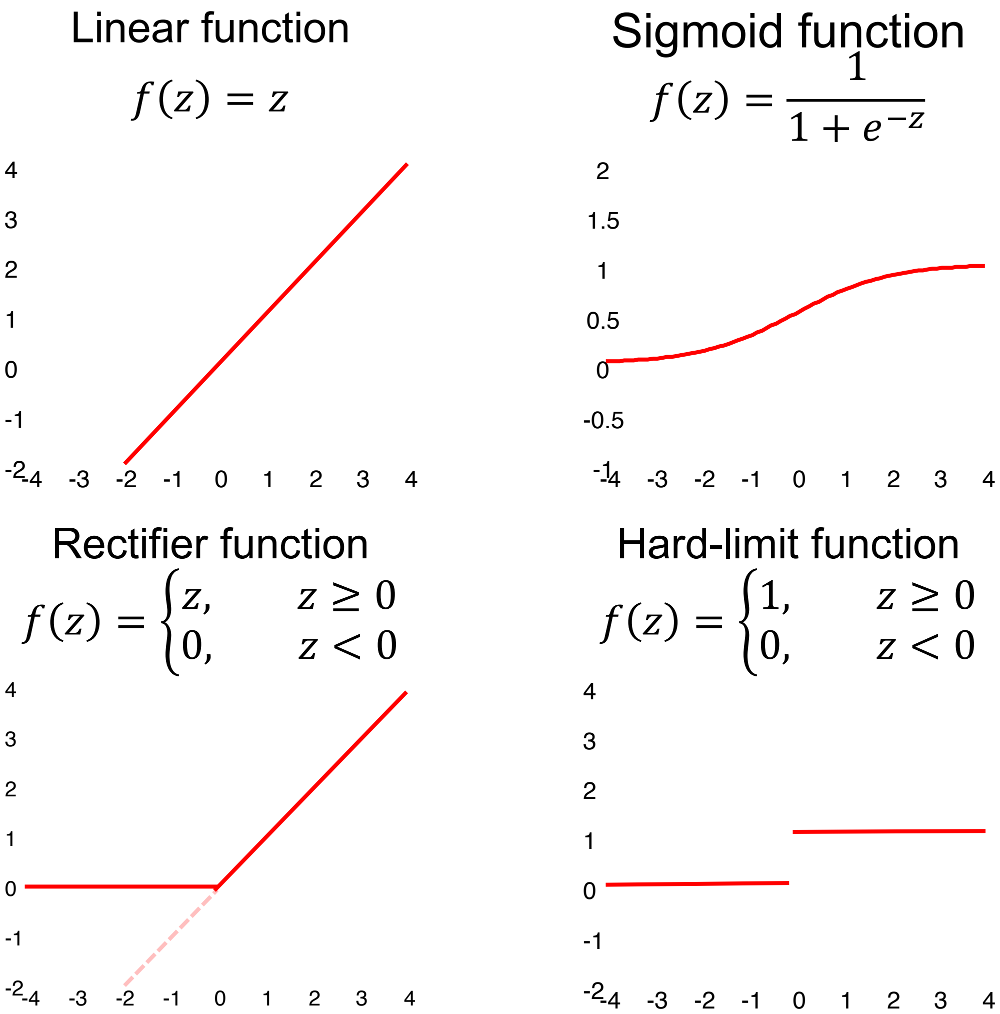
$$a^{l+1} = f\left(Wa^l\right)$$

网络结构问题

网络中每个神经元激活函数可以是不相同

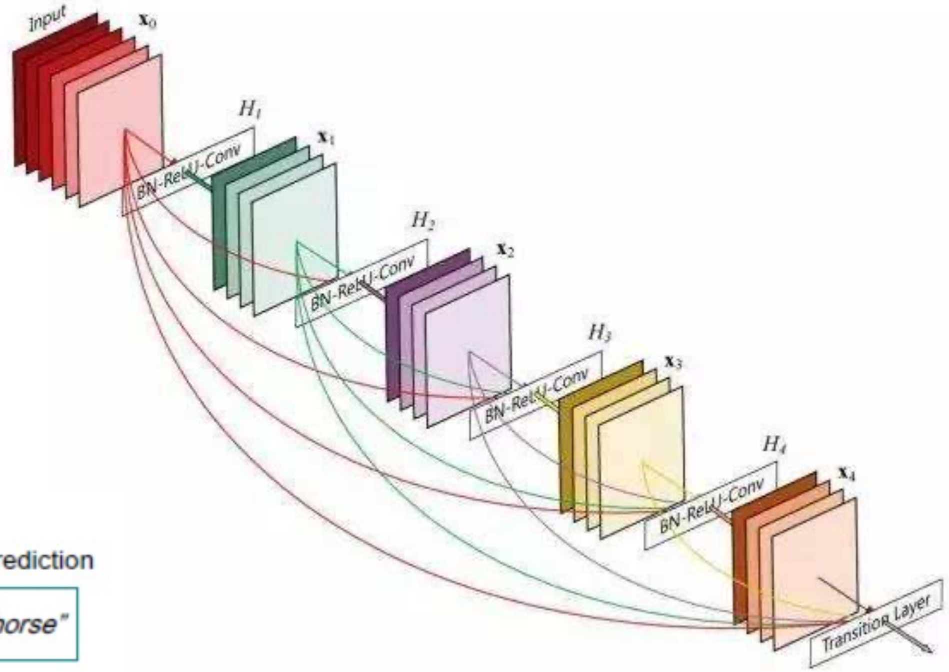


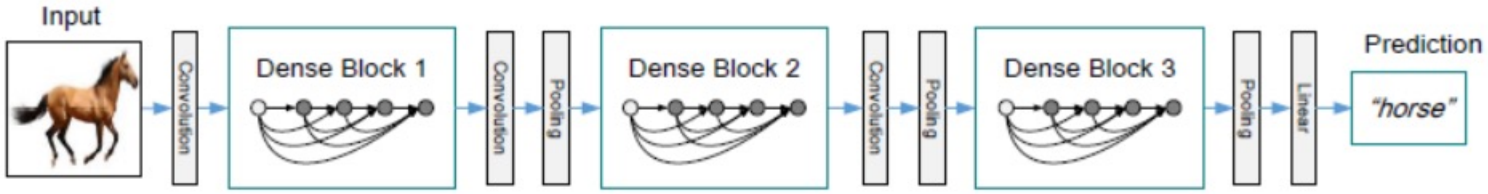


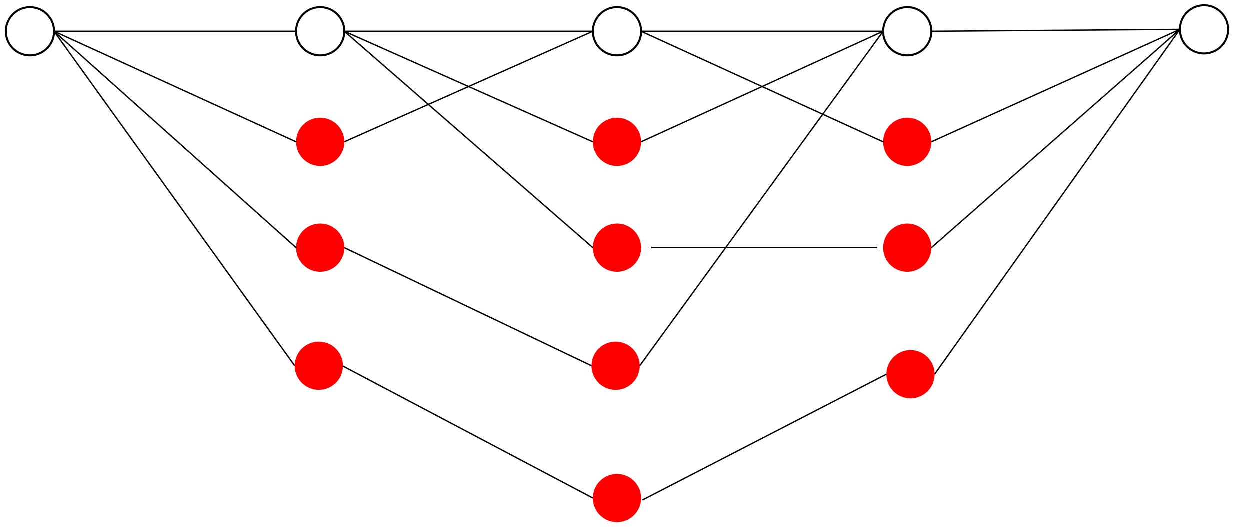


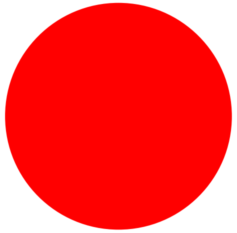
网络结构问题

DenseNets









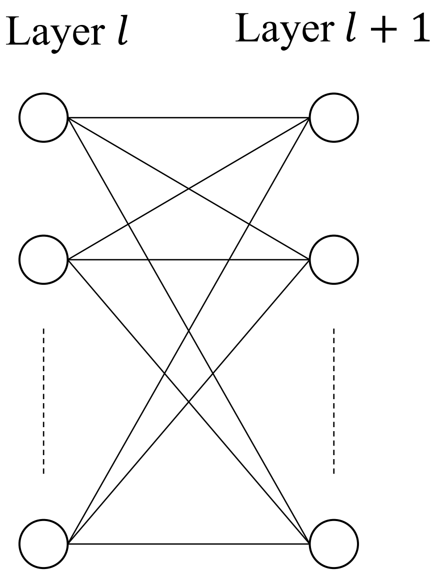
Linear neuron

$f(s) = s$

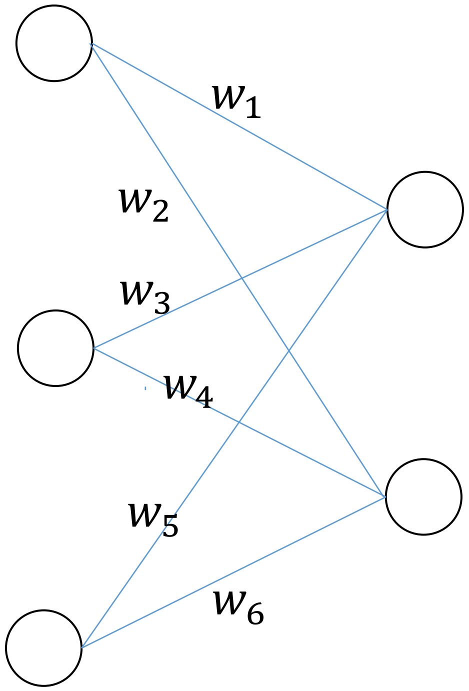
网络结构问题

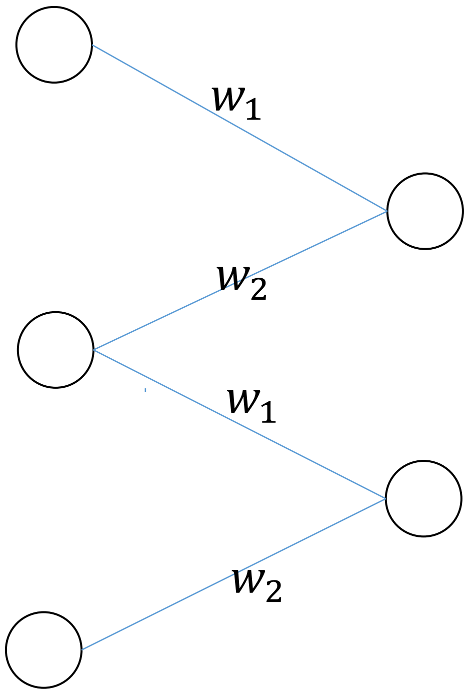
卷积神经网络

相邻层之间的神经元共享部分连接权



$$W^l = \left(w_{ij}^l\right)_{n_{l+1}\times n_l}$$

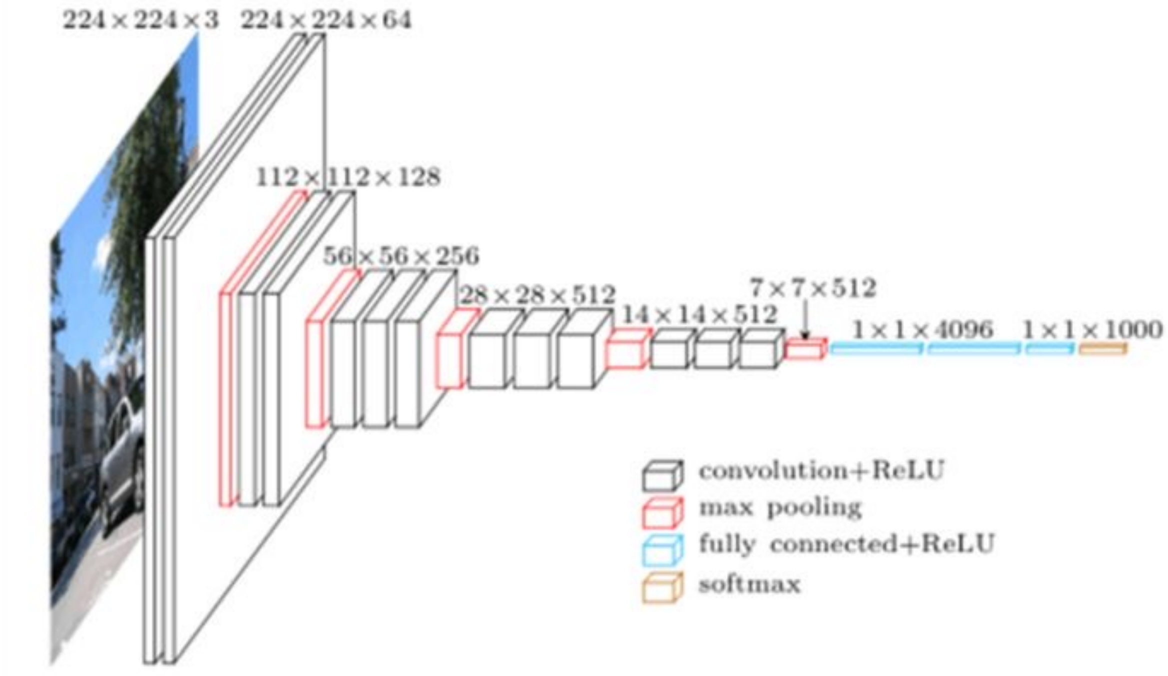


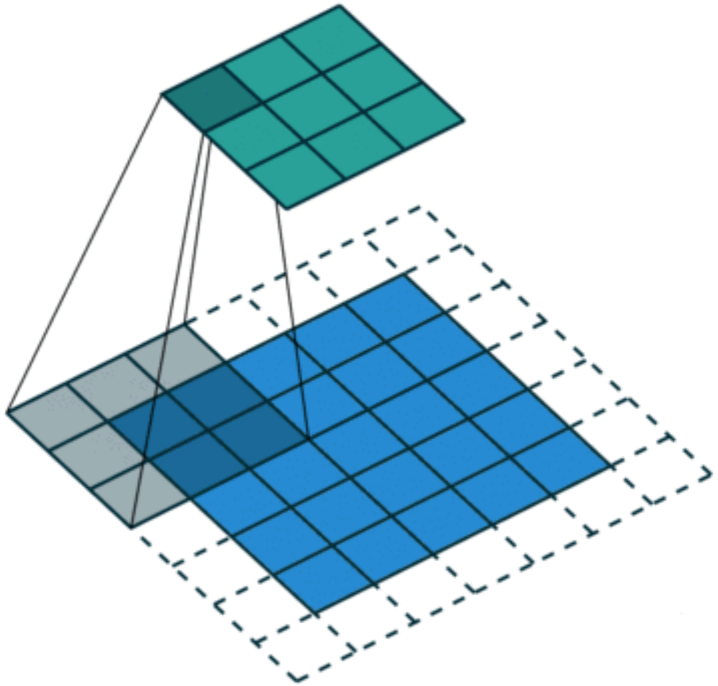


网络结构问题

卷积神经网络

相邻层之间的神经元共享部分连接权





提纲

I

- 网络结构问题
- 学习算法问题
- 目标输出问题
- 网络输入问题

II

- 网络预测问题
- 性能函数问题
- 网络深度问题
- 训练数据问题

学习算法问题

梯度下降算法

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

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

BP算法

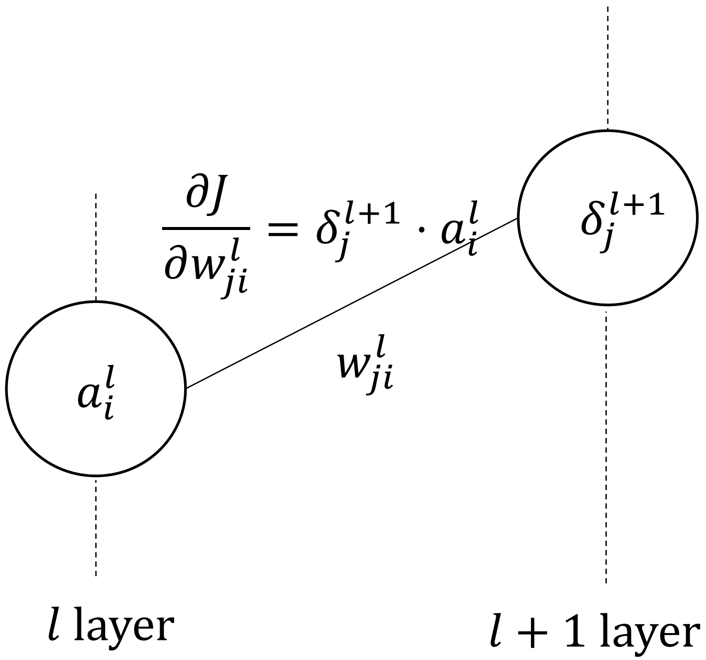
$$w_{ji}^{l} \leftarrow w_{ji}^{l} - \alpha \cdot \left(\delta_j^{l+1} \cdot a_i^l\right)$$

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

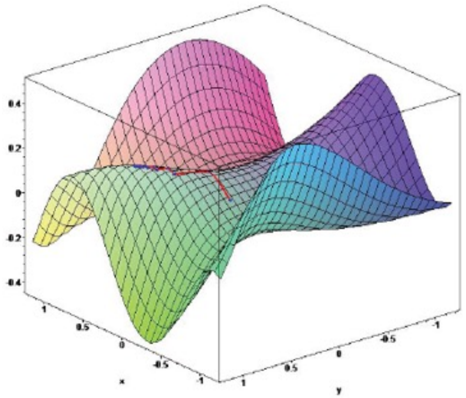
$$a_i^l = f\left(z_i^l\right)$$

BP算法

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



$$J = J(\cdots, w_{ji}^l, \cdots)$$



问题：

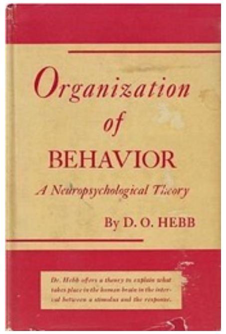
BP算法有没有神经科学依据？

学习算法问题

Hebb假说：

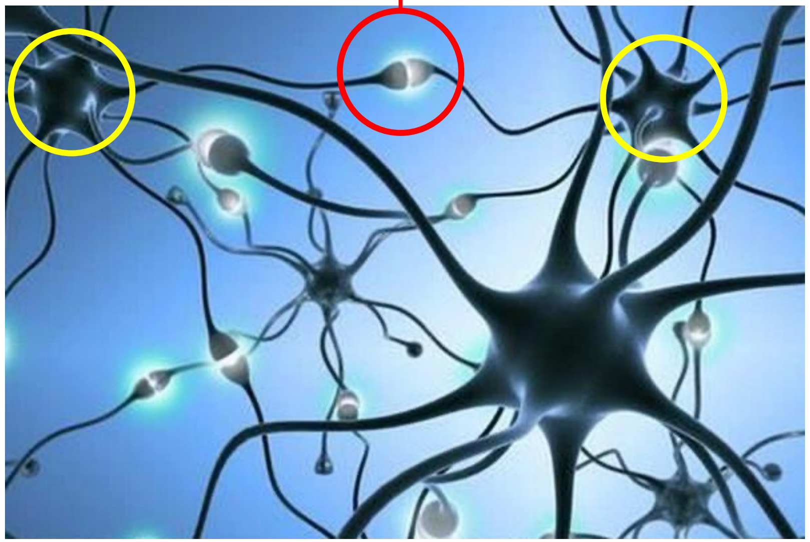
当神经元A的轴突足够接近到能够激发神经元B，且反复或持续地刺激神经元B，那么A或B中的一个或两个神经元将会产生某种增长过程或代谢变化，从而增强神经元A对神经元B的刺激效果。

——D.O Hebb，1949



D. O. Hebb
认知心理生物学之父
1904-1985

突触



学习算法问题

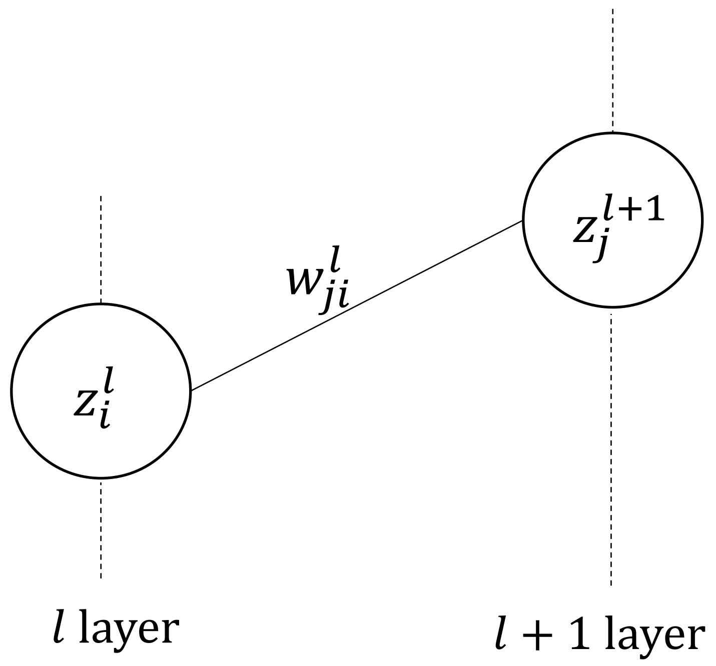
当神经元A的轴突足够接近到能够激发神经元B，且反复或持续地刺激神经元B，那么A或B中的一个或两个神经元将会产生某种增长过程或代谢变化，从而增强神经元A对神经元B的刺激效果。

如果连接突触两端的两个神经元同时被激活，这个突触的连接强度将会增强。

数学抽象

Hebbian 学习规则

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



学习算法问题

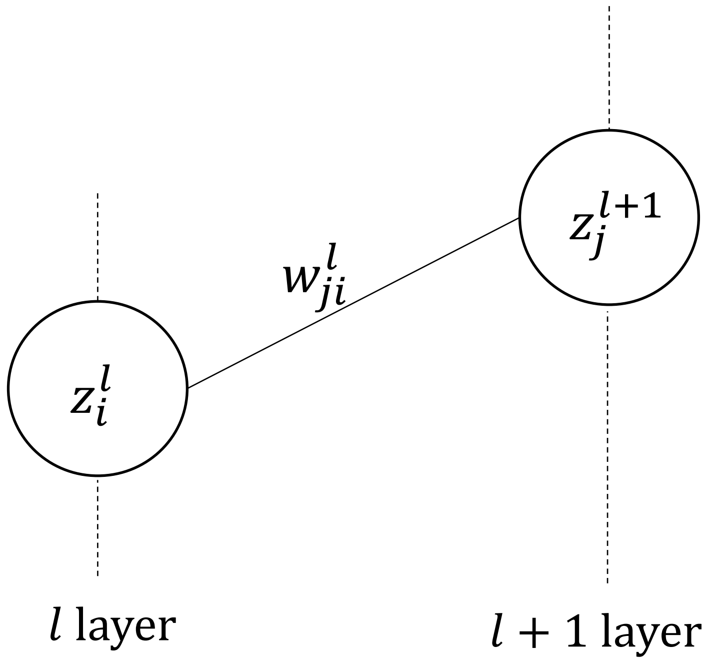
如果连接突触两端的两个神经元同时被激活，这个突触的连接强度将会增强。

Hebbian 学习规则

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

$$w_{ji}^{l} \leftarrow w_{ji}^{l} + \alpha \cdot f_j^{l+1}\left(z_j^{l+1}\right) \cdot f_i^{l}\left(z_i^{l}\right)$$

$$w_{ji}^{l} \leftarrow w_{ji}^{l} + \alpha \cdot a_j^{l+1} \cdot a_i^{l}$$



学习算法问题

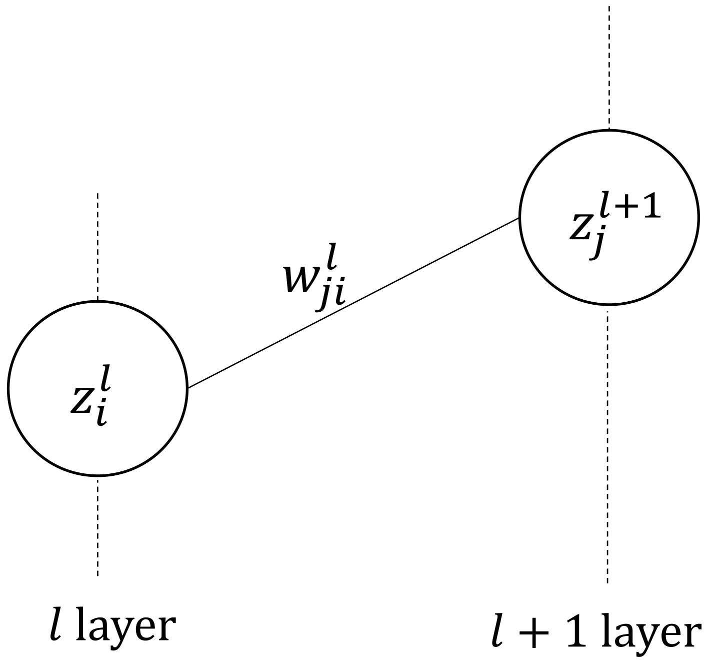
如果连接突触两端的两个神经元同时被激活，这个突触的连接强度将会增强。

Hebbian 学习算法

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

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

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



结论：BP算法是一种Hebb学习算法

提纲

I

- 网络结构问题
- 学习算法问题
- 目标输出问题
- 网络输入问题

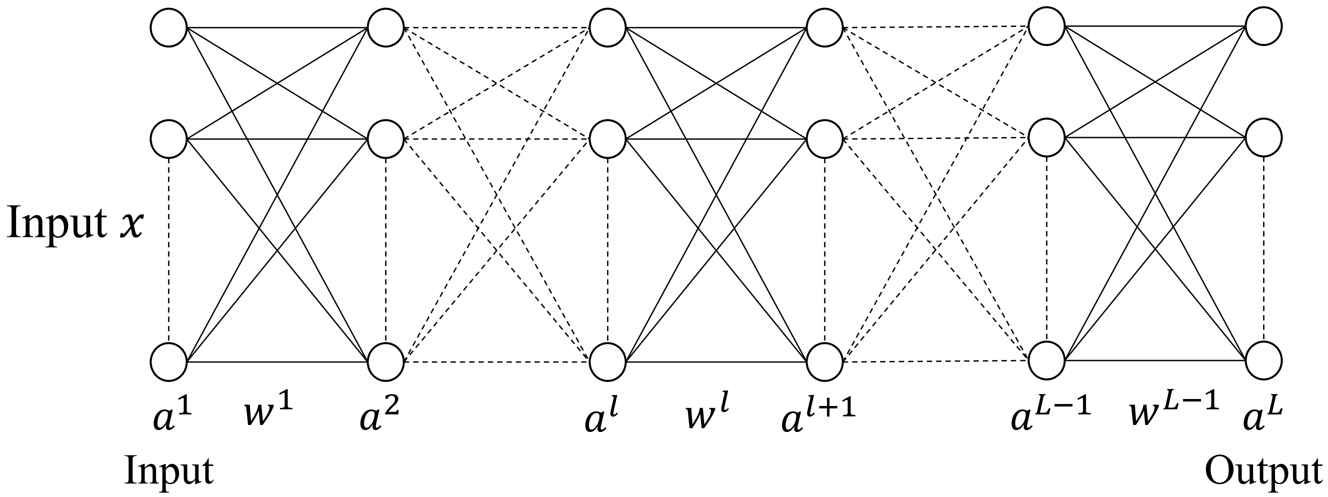
II

- 网络预测问题
- 性能函数问题
- 网络深度问题
- 训练数据问题

目标输出问题

问题：如何定义目标输出？

原则上，目标输出的定义必须符合具体任务的要求。因此目标输出是从具体任务 / 应用中产生的。此外，目标输出必须和输入相对应。



定义在最后一层的目标输出

$$\underbrace{y^L = \begin{bmatrix} y_1^L \\ \vdots \\ y_{n_L}^L \end{bmatrix} \quad \Longleftrightarrow \quad \text{Input } x}_{\text{一个训练样本 } (x, y^L)}$$

$$\dim(a^L) = \dim(y^L)$$

目标输出问题

分类问题：

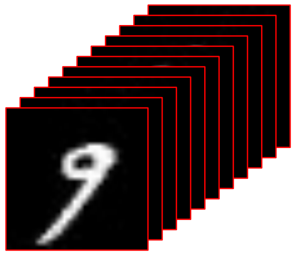
目标是将每个输入数据映射到它对应的分类标签，

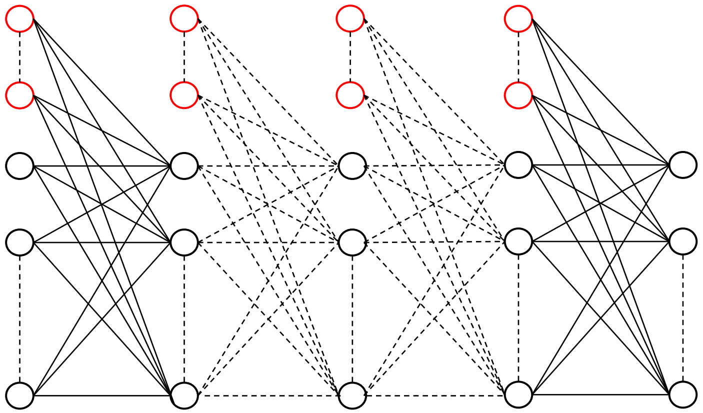
因此，目标输出定义为标签的表示。

与输入相关联的标签的具体表征

目标输出

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





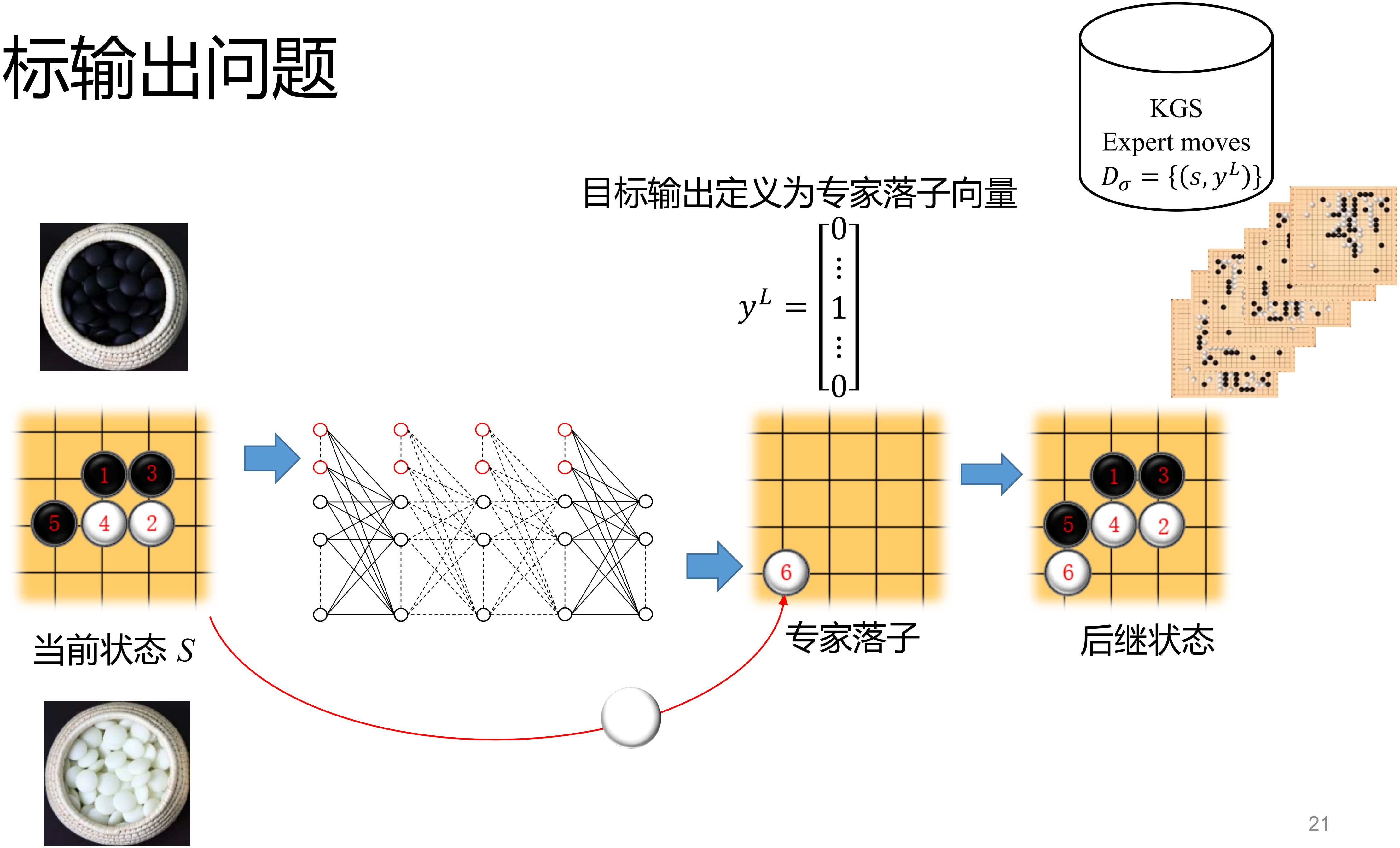
> Tip:
> 输出层神经元的数量等于类别数量。

分类标签

0	1	2	3	4	5	6	7	8	9
$\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\\0\\1\end{bmatrix}$	$\begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\1\\0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\1\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\1\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\\1\\0\end{bmatrix}$

表示

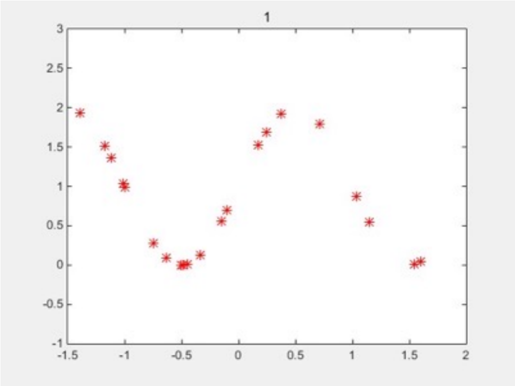
目标输出问题


KGS
Expert moves
$D_\sigma = \{(s, y^L)\}$
目标输出定义为专家落子向量
$y^L = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$
当前状态 s
专家落子
后继状态

目标输出问题

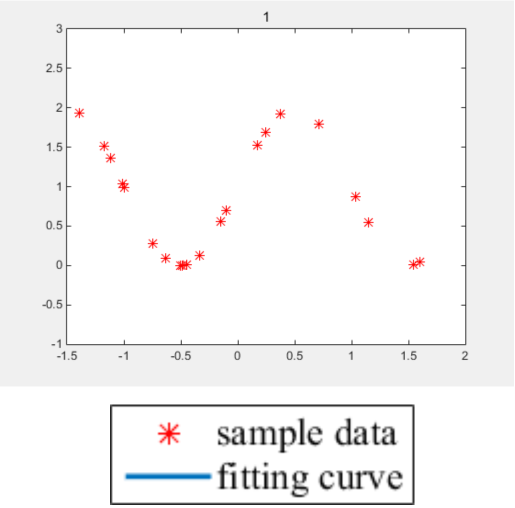
曲线拟合问题：

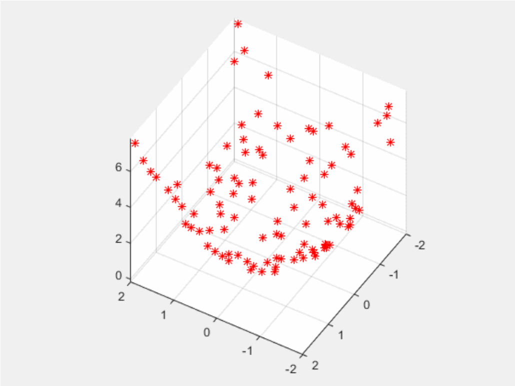
给定一组样本数据，来估计通过这些点的曲线。



采样数据

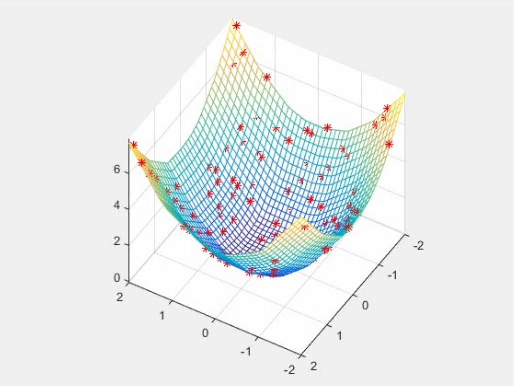
	1	2	3	4	5	6
x	-0.5000	0.1740	0.7100	-0.9980	-0.6340	1.0400
y	0	1.5198	1.7902	0.9937	0.0873	0.8747



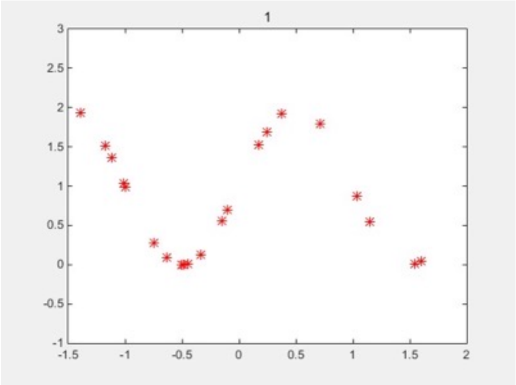


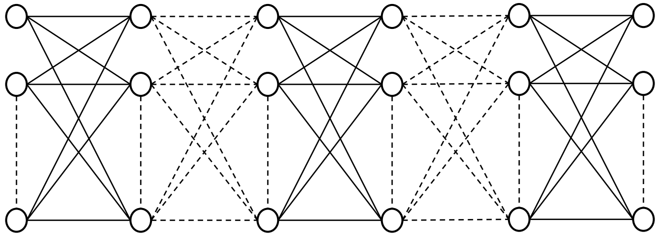
采样数据

	1	2	3	4	5	6
x	-0.2000	-1.9000	1.9000	0.4000	-1.9000	0.8000
	1.4000	-1.9000	-1.5000	-0.5000	0.3000	-0.1000
y	2.0000	7.2200	5.8600	0.4100	3.7000	0.6500

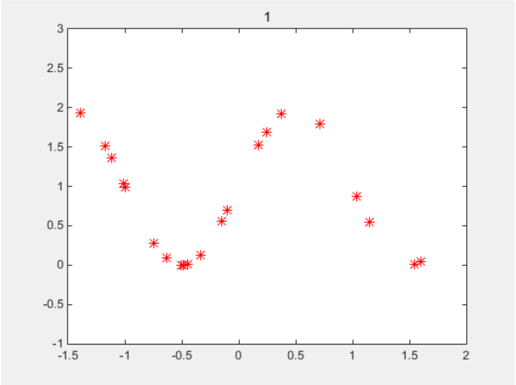


目标输出问题

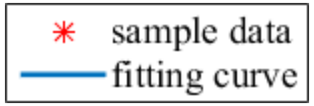


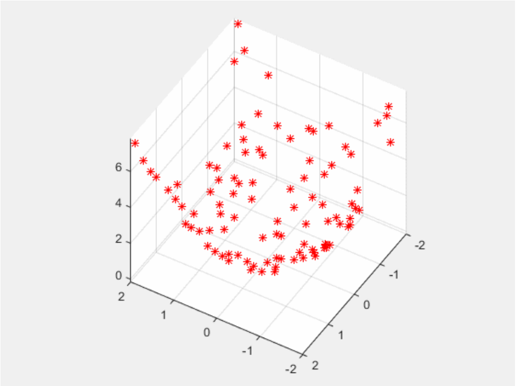


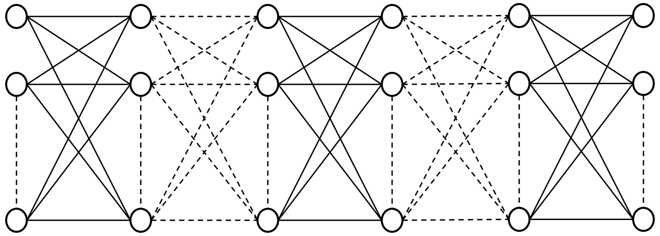
训练样本 (x, y)

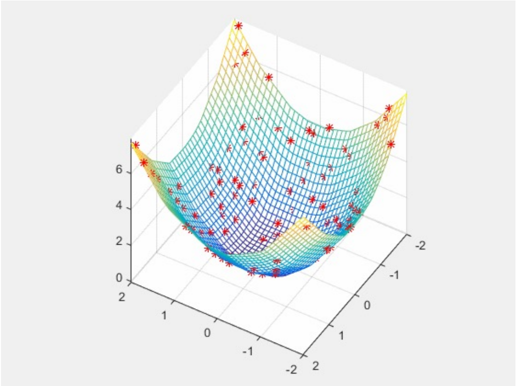


目标输出定义为每个样本x对应的y值，即，$y^L = y$





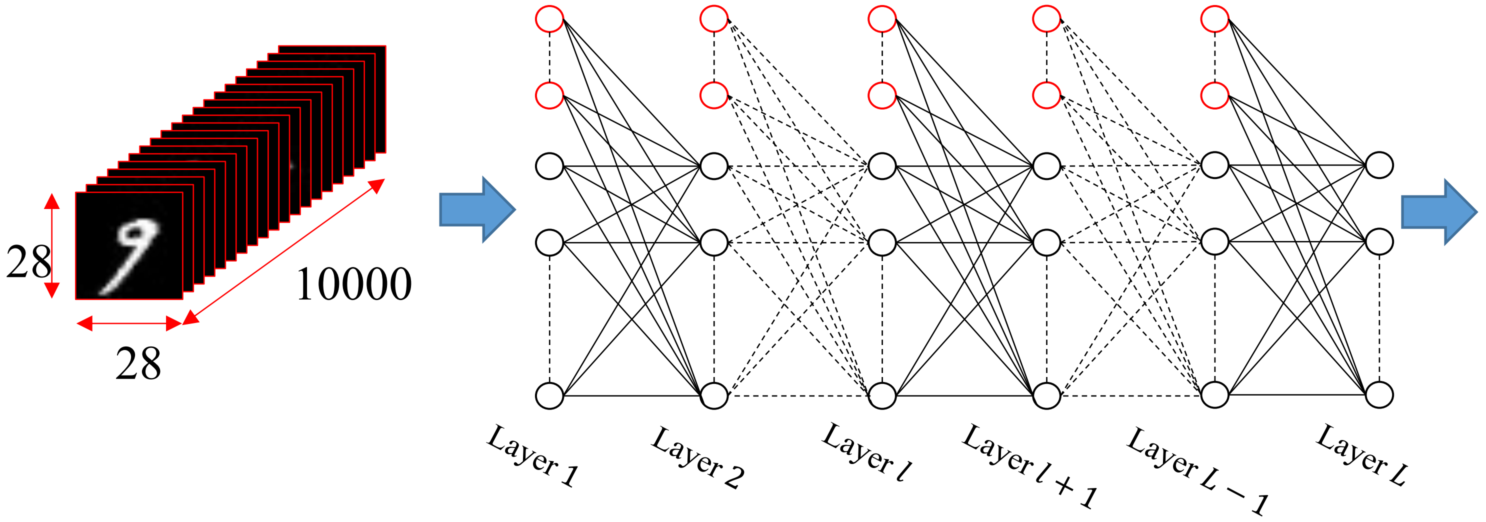


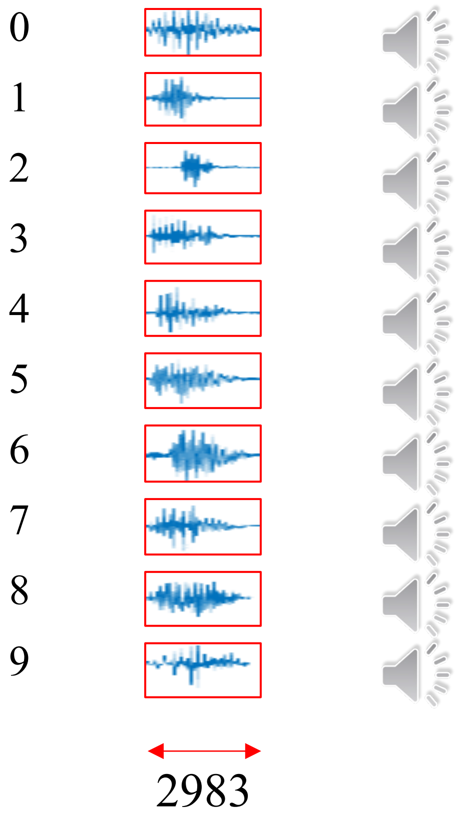


目标输出问题

目标输出定义为每个数字的语音向量

$$y^L = \begin{bmatrix} y_1^L \\ y_2^L \\ \vdots \\ y_{2983}^L \end{bmatrix}$$





提纲

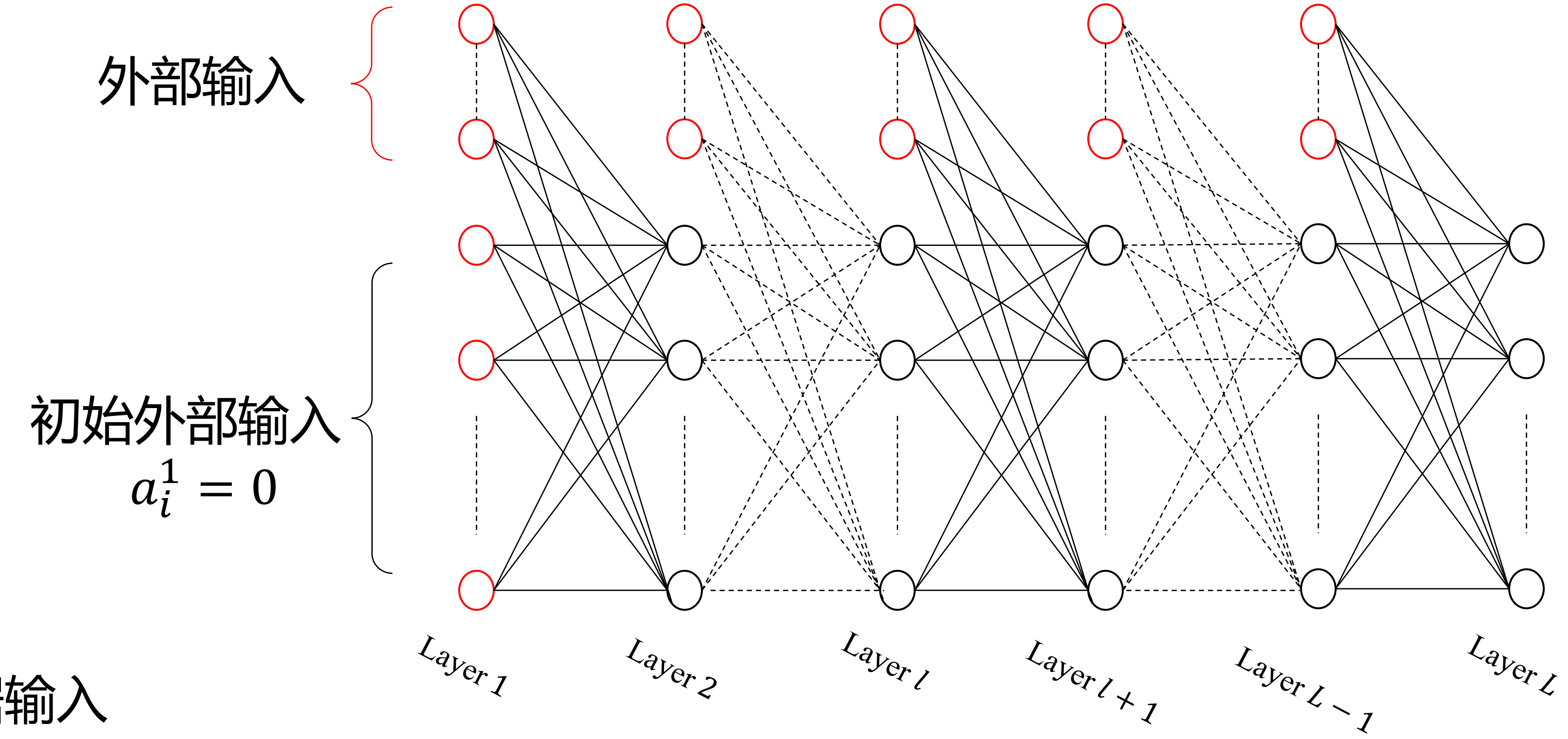
I

- 网络结构问题
- 学习算法问题
- 目标输出问题
- 网络输入问题

II

- 网络预测问题
- 性能函数问题
- 网络深度问题
- 训练数据问题

网络输入问题



外部输入

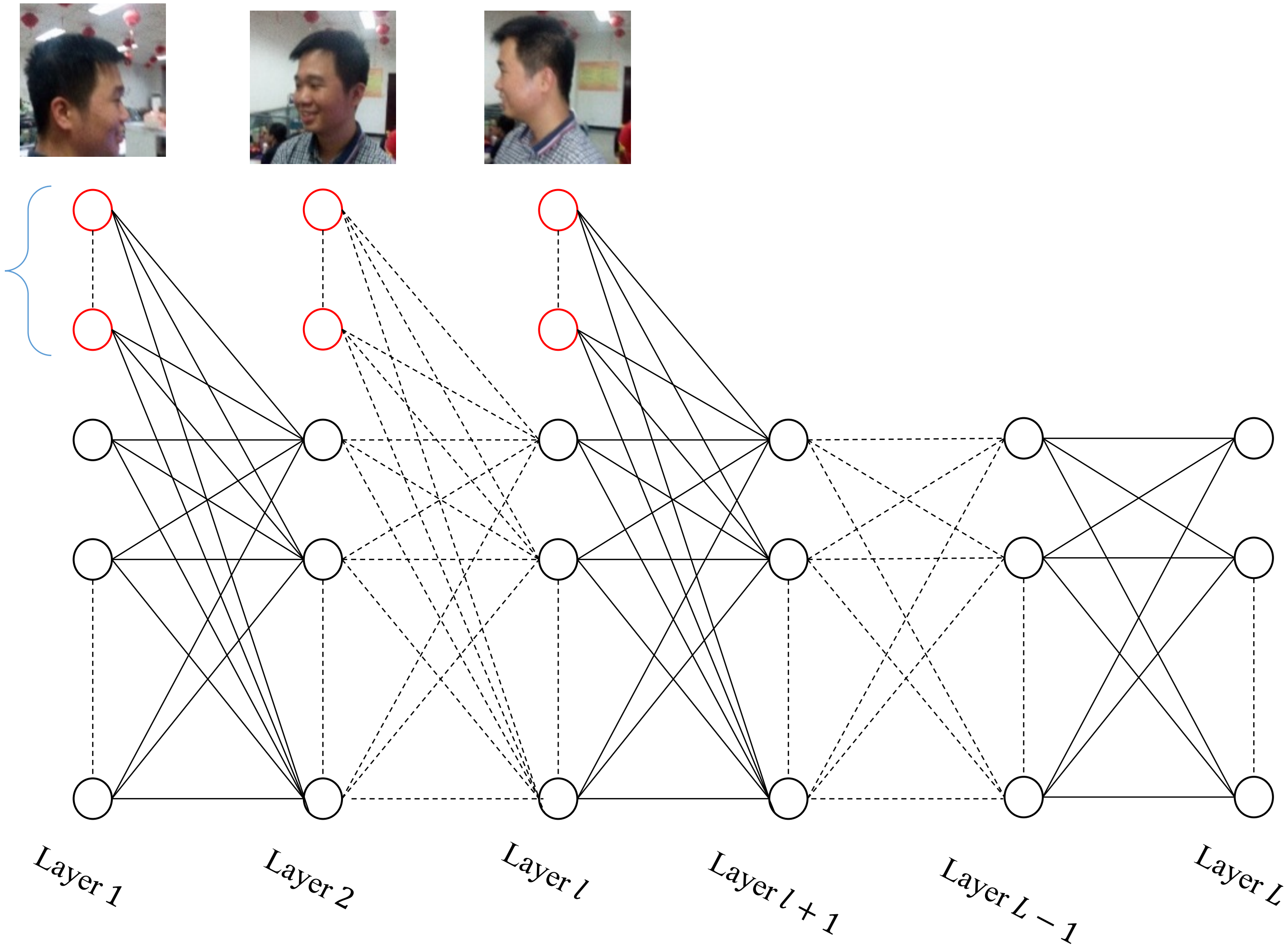
- 直接从数据输入

初始外部输入

- 根据应用情况设置
- 许多情况下可以直接取0

网络输入问题

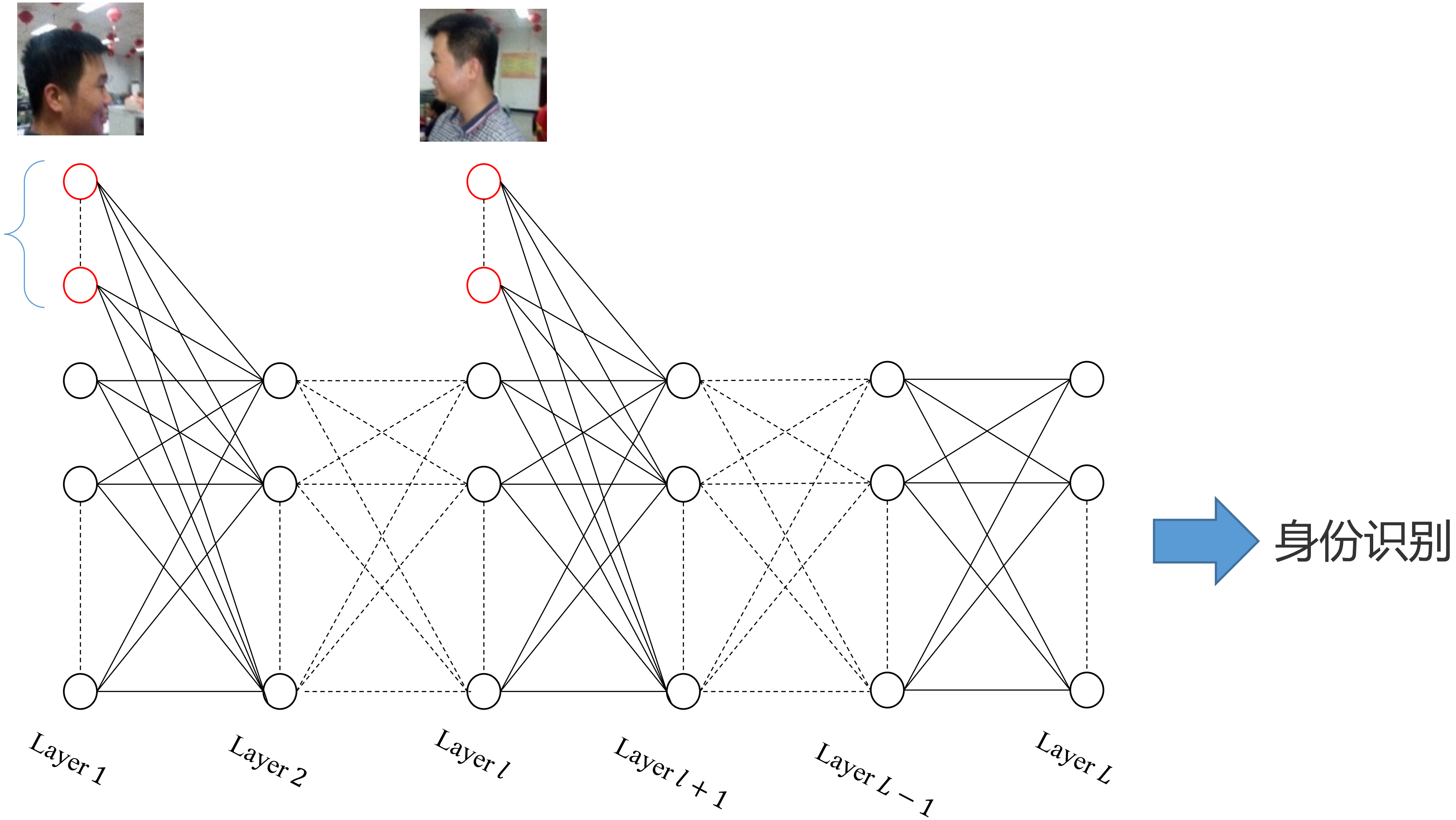
序列输入



身份识别

网络输入问题

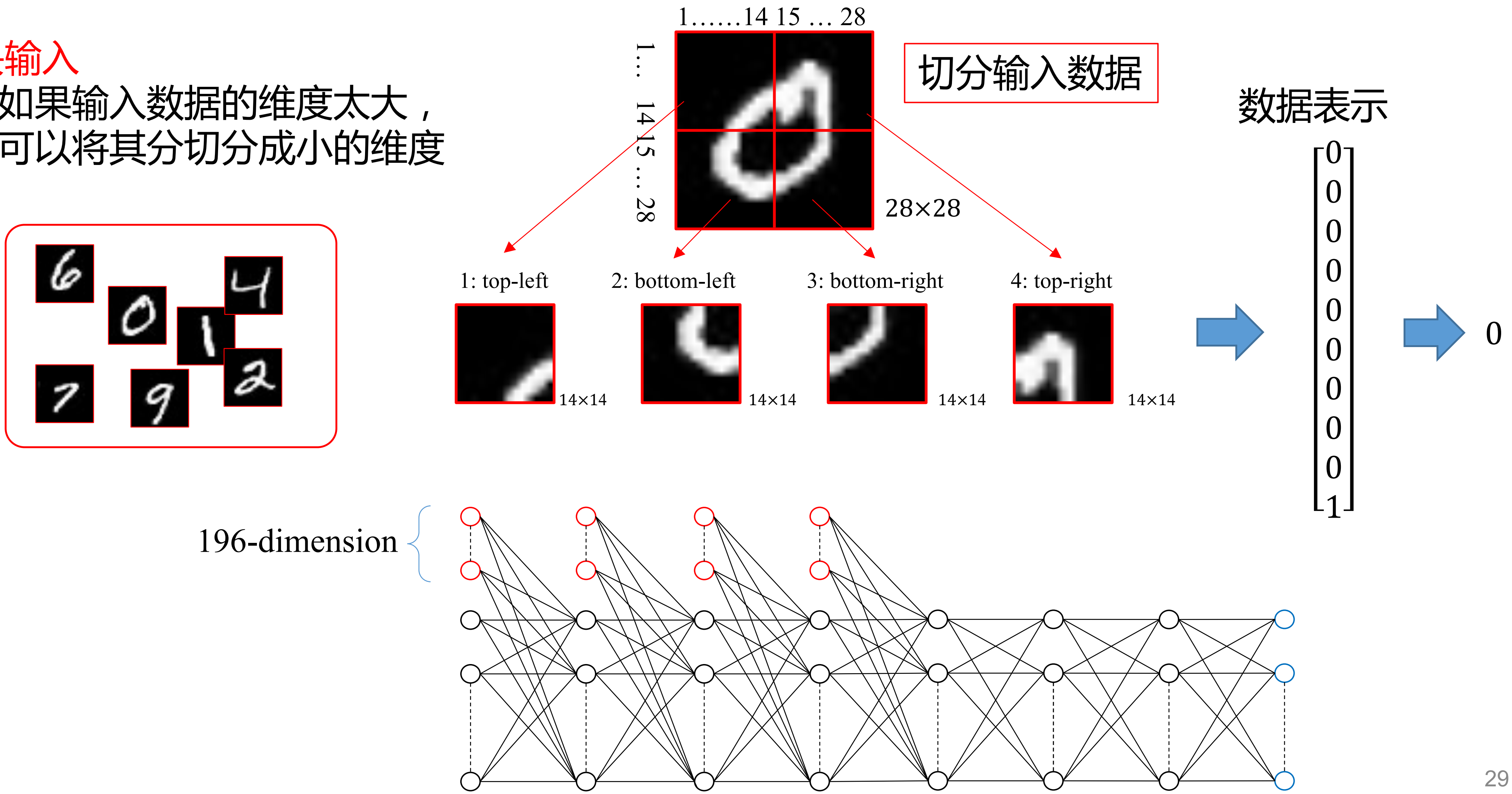
序列输入



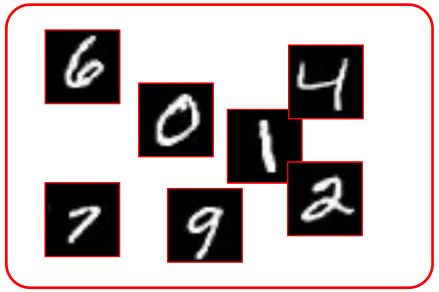
网络输入问题

分块输入

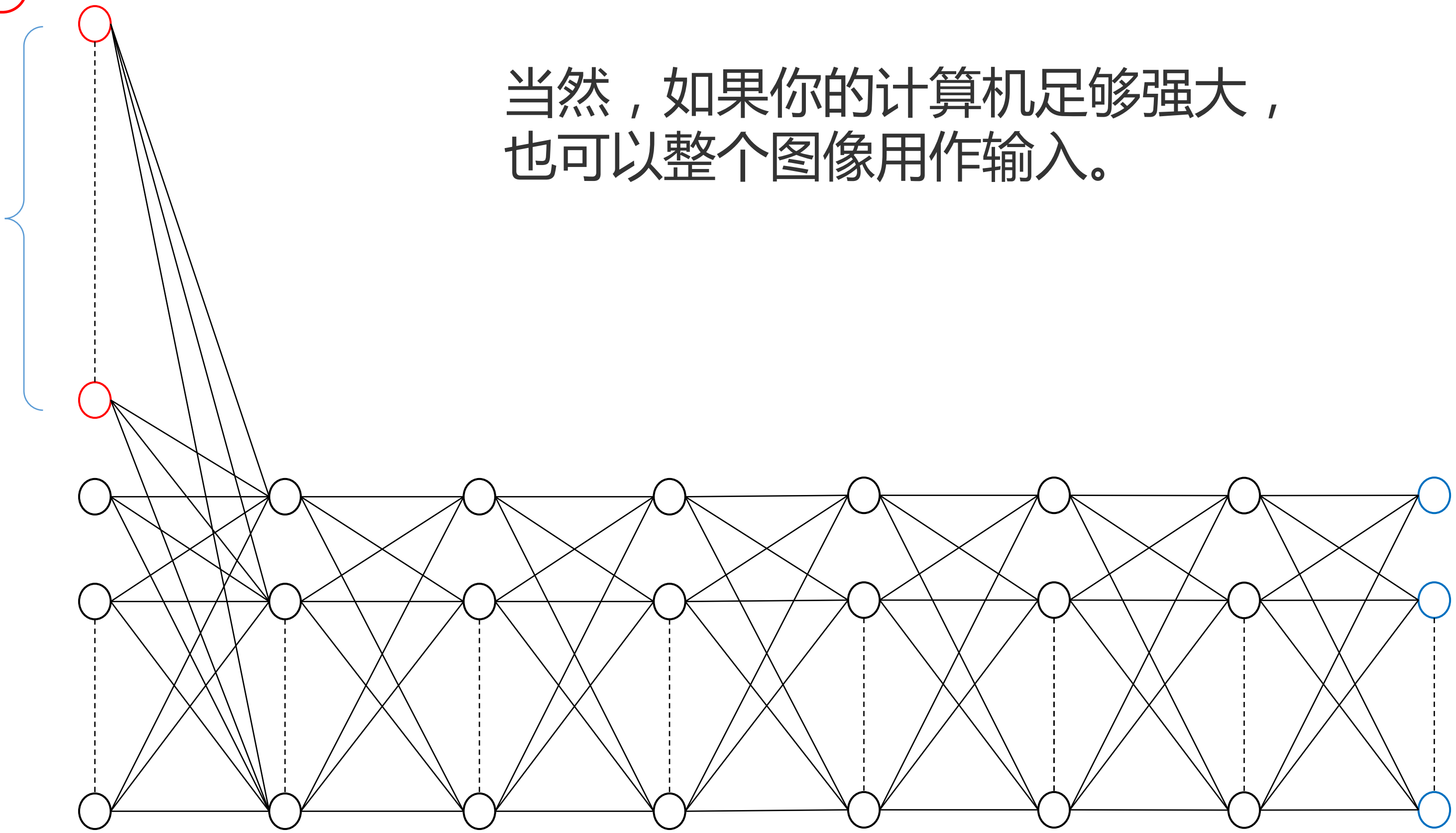
如果输入数据的维度太大，可以将其分切分成小的维度



网络输入问题



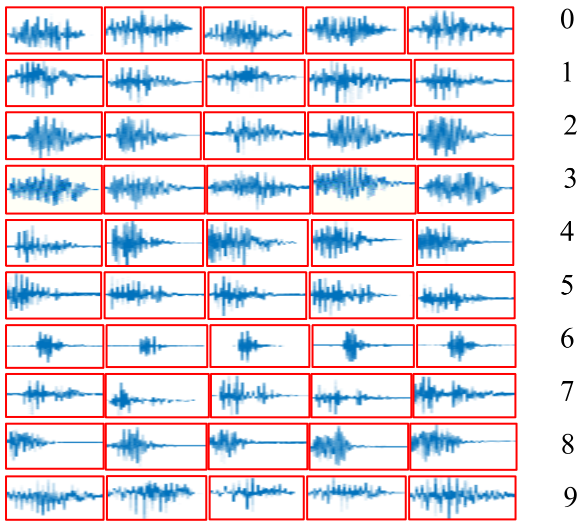
784-dimension

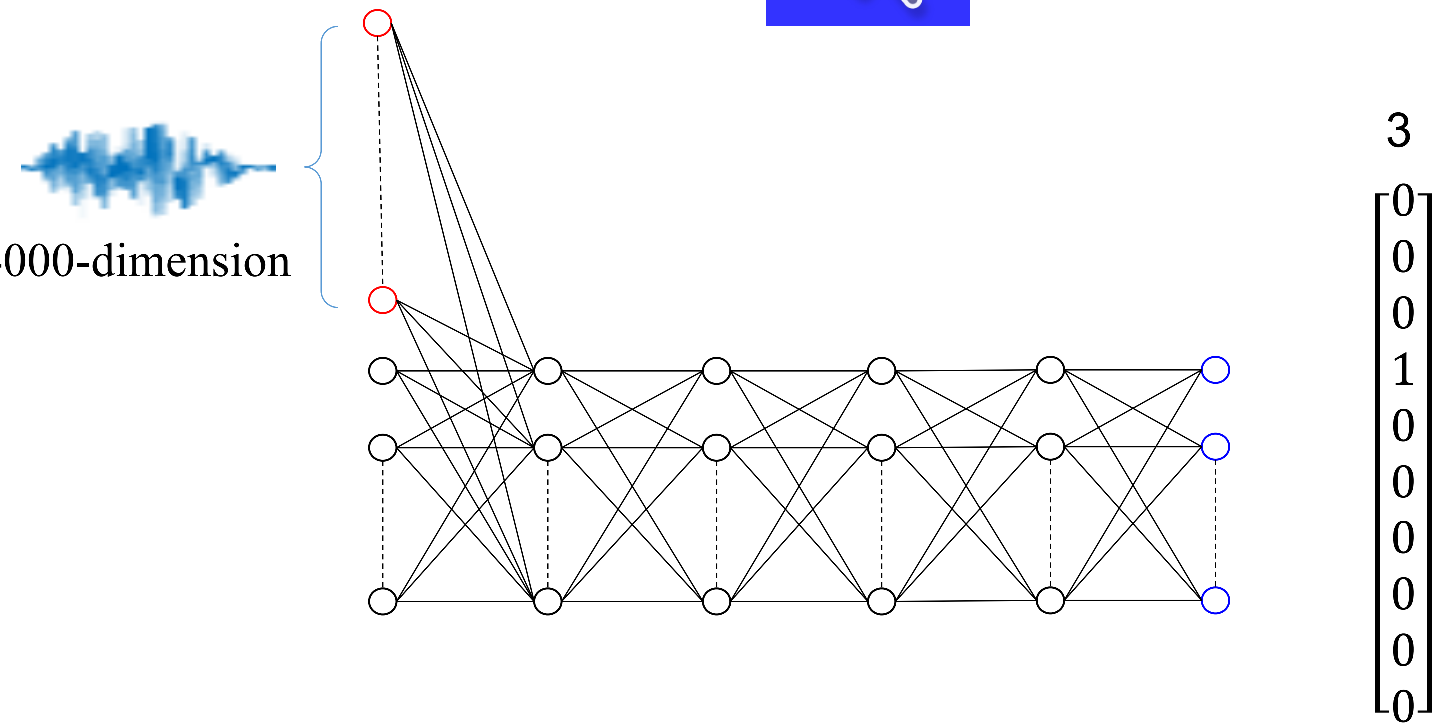


当然，如果你的计算机足够强大，
也可以整个图像用作输入。

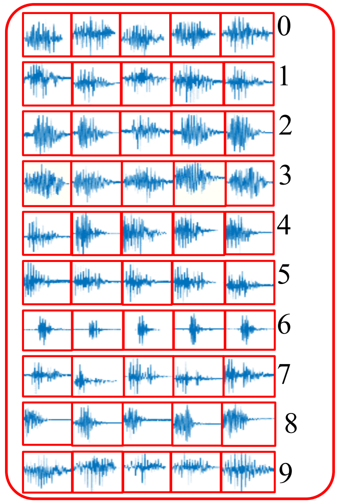
网络输入问题

语音数据输入

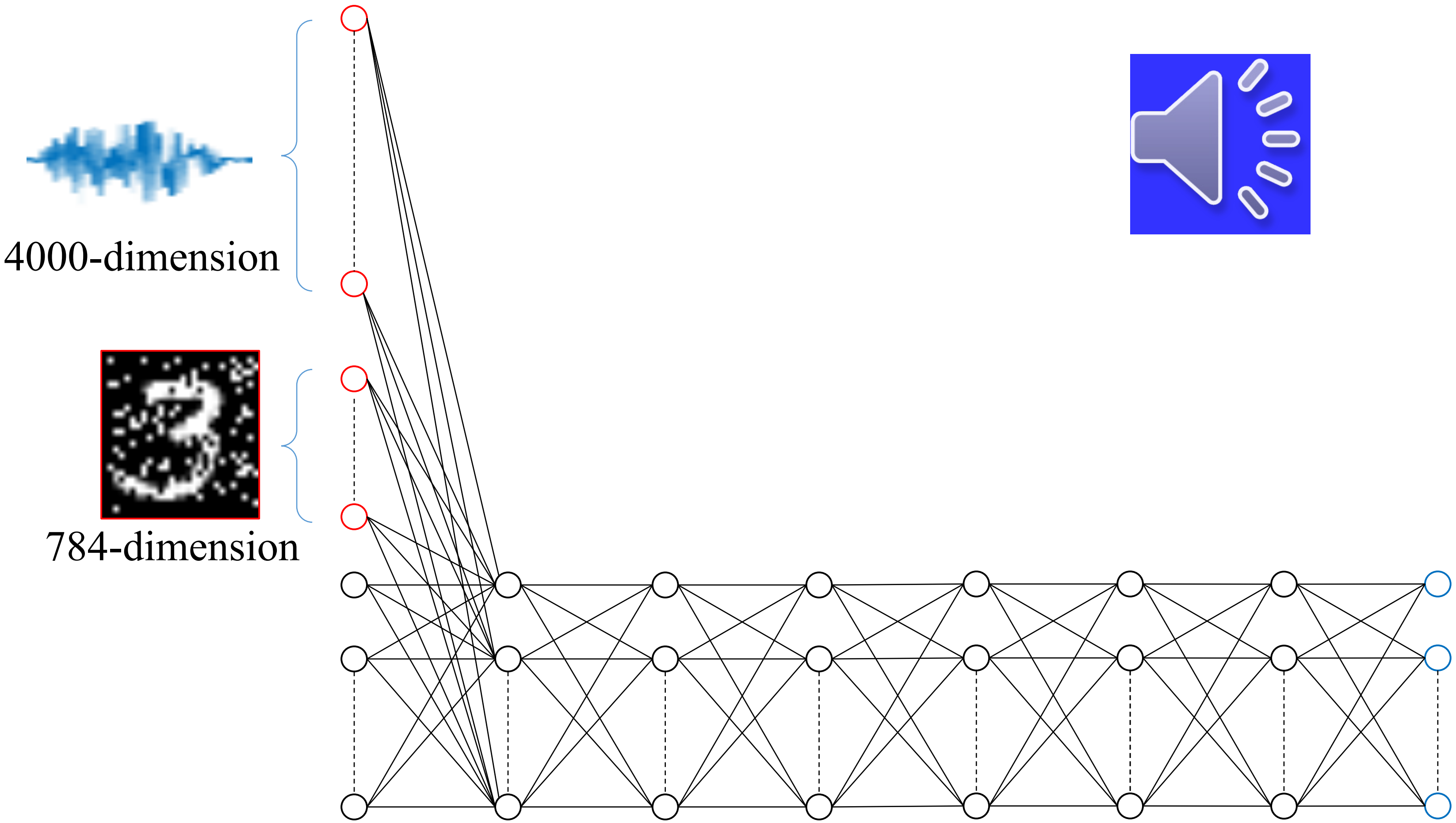




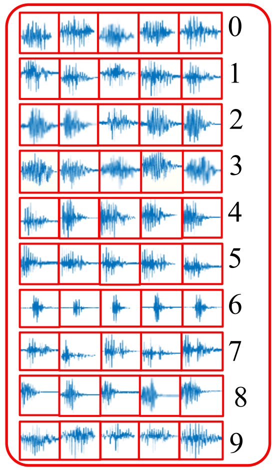
网络输入问题

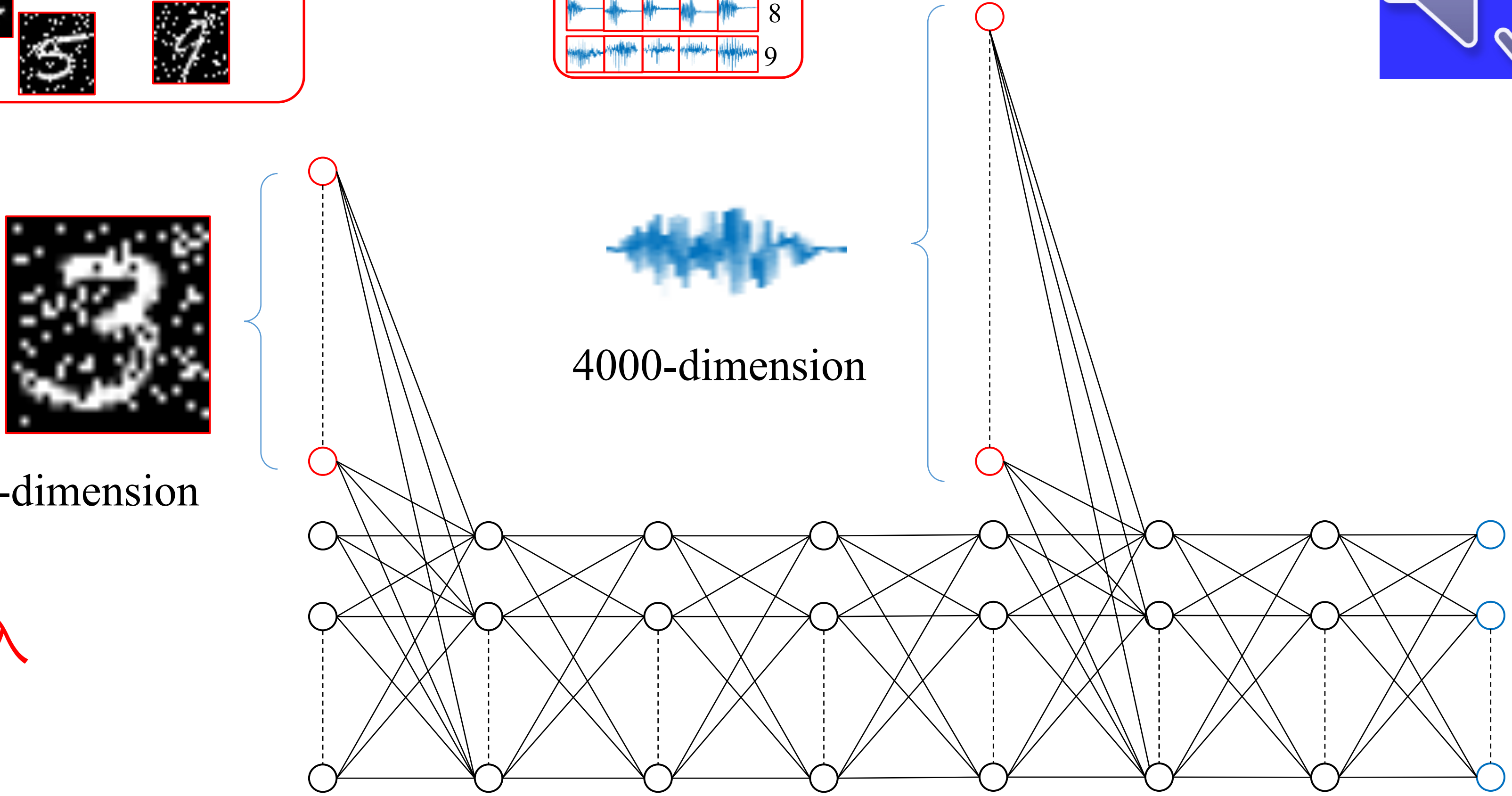


多模态数据输入



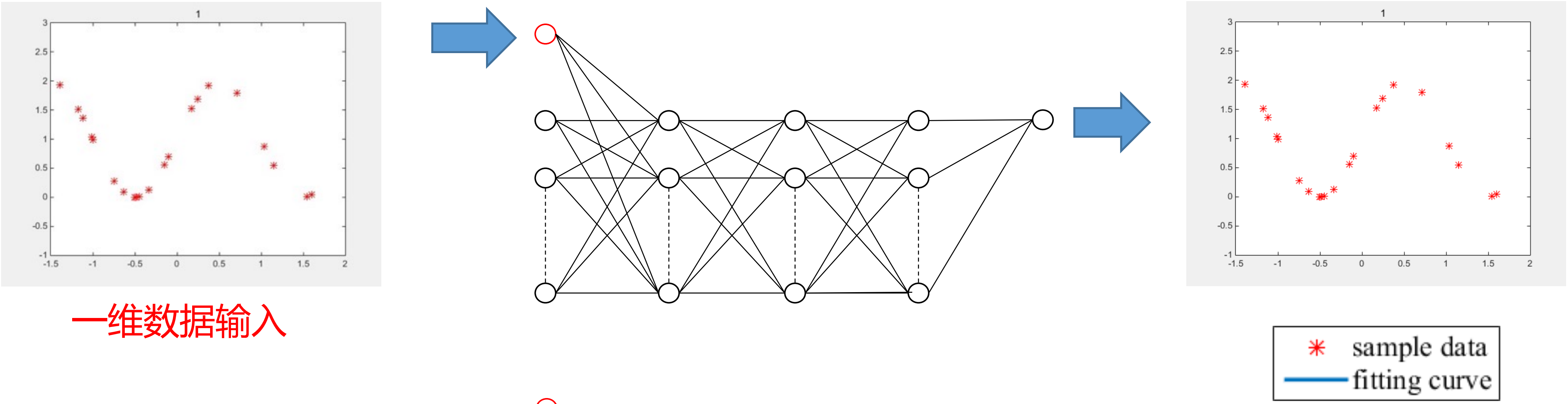
网络输入问题



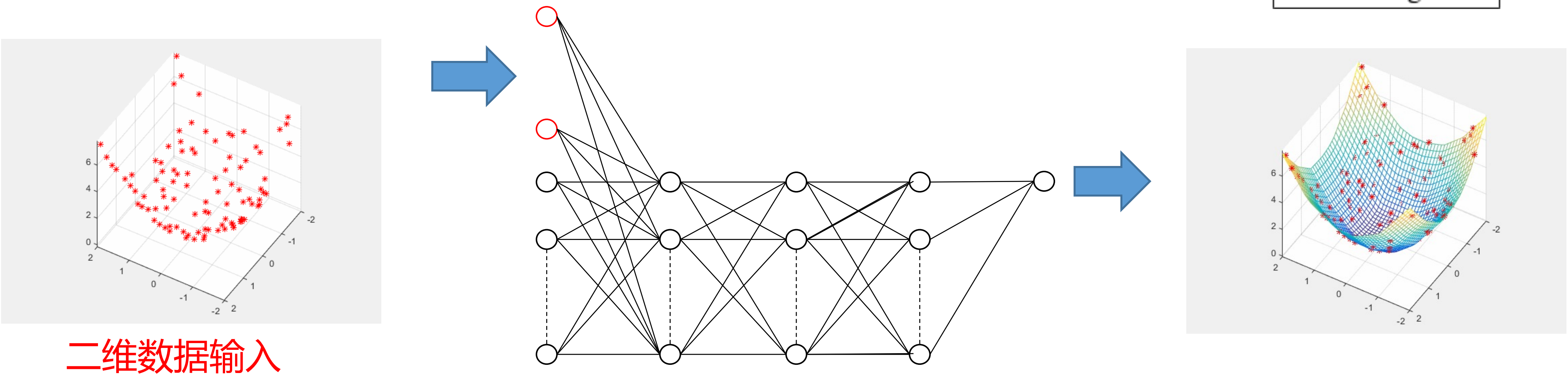


多模态数据输入

网络输入问题



一维数据输入

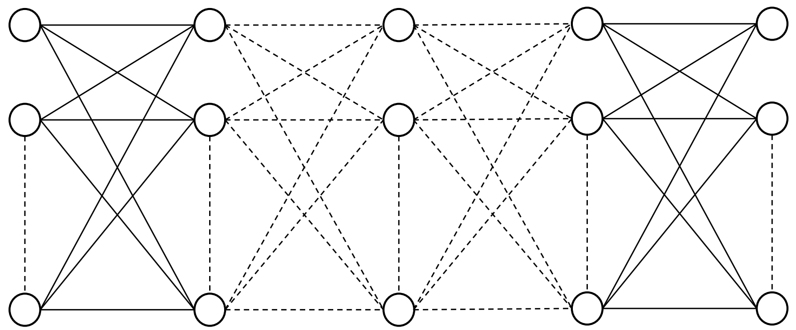


二维数据输入

提纲

I	II
网络结构问题	网络预测问题
学习算法问题	性能函数问题
目标输出问题	网络深度问题
网络输入问题	训练数据问题

网络预测问题



a^1 w^1 a^2 w^l a^l a^{L-1} w^{L-1} a^L

Input Output

网络预测

$$a^L = \begin{bmatrix} a_1^L \\ \vdots \\ a_{n_L}^L \end{bmatrix}$$

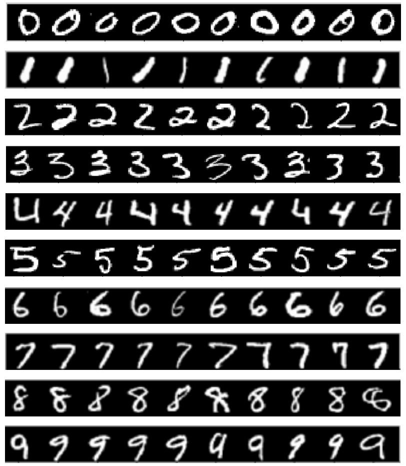
定义最后一层的激活函数 f^L 使得：

- 网络预测向量 a^L 和目标输出向量 y^L 的取值域相匹配
- 需要 f^L是可微的

目标输出

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

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

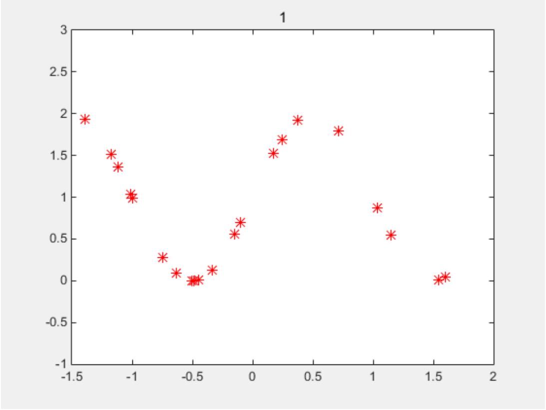


Sigmoid function

$$f(s) = \frac{1}{1+e^{-s}} \in (0,1)$$

$$\begin{bmatrix} a_1^L \\ \vdots \\ a_{n_L}^L \end{bmatrix} \xleftrightarrow{\text{Threshold } \theta} \begin{bmatrix} y_1^L \\ \vdots \\ y_{n_L}^L \end{bmatrix}$$

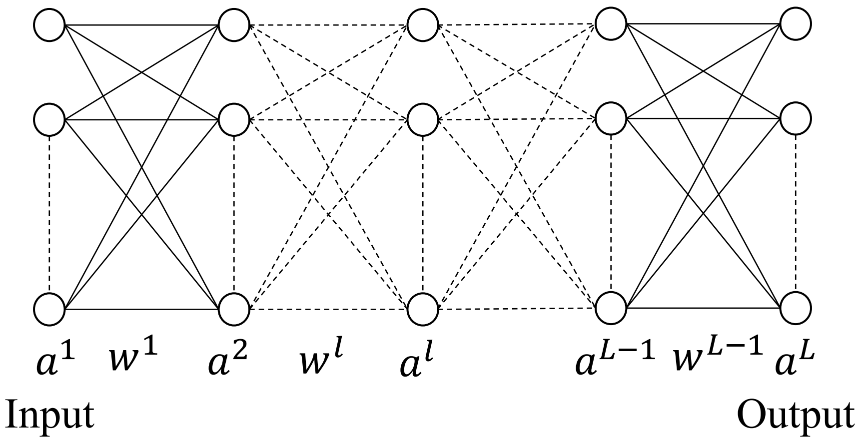
$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



Linear function

$$f(s) = s$$

网络预测问题



目标输出

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

$$0 \le y_i^L \le 1$$

$$\sum_{i=1}^{n_L} y_i^L = 1$$

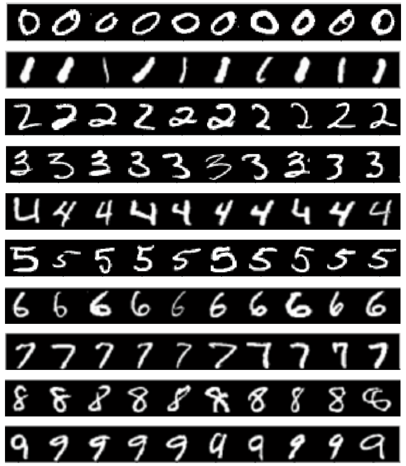
$$a_i^L = \frac{e^{z_i^L}}{e^{z_1^L} + \cdots + e^{z_{n_L}^L}}$$

Softmax function

网络预测

$$0 < a_i^L < 1$$

$$\sum_{i=1}^{n_L} a_i^L = 1$$



0	1	2	3	4	5	6	7	8	9
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0

$$y^L = \begin{bmatrix} y_1^L \\ \vdots \\ y_{n_L}^L \end{bmatrix} \qquad 0 \le y_i^L \le 1, \sum_{i=1}^{n_L} y_i^L = 1$$

提纲

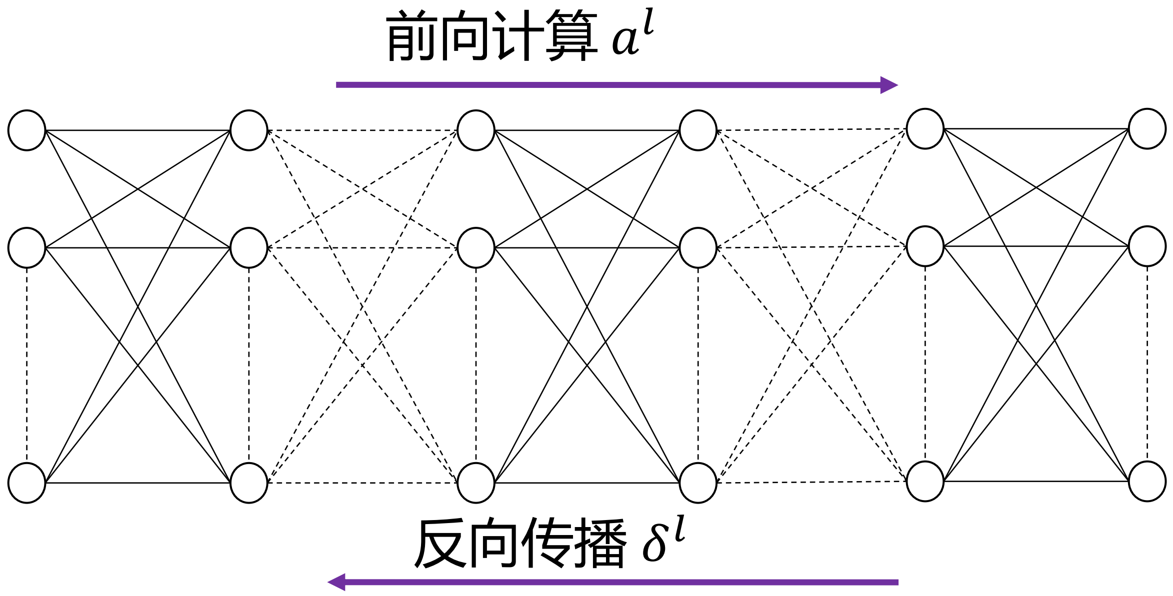
I

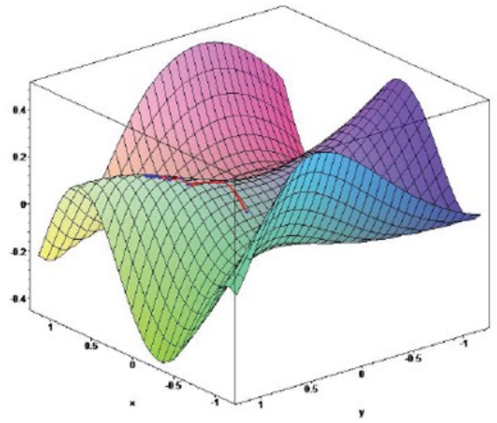
- 网络结构问题
- 学习算法问题
- 目标输出问题
- 网络输入问题

II

- 网络预测问题
- 性能函数问题
- 网络深度问题
- 训练数据问题

性能函数问题





网络输出

$$a^L = \begin{bmatrix} a_1^L \\ \vdots \\ a_{n_L}^L \end{bmatrix}$$

目标输出

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

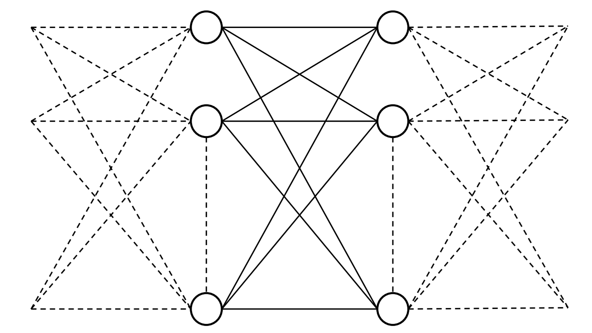
$$J(a^L, y^L)$$

性能函数 $J(a^L, y^L)$ 被用来刻画 a^L 和 y^L 的靠近程度，$J(a^L, y)$ 实际上是$(w^1, \cdots, w^{L-1})$的函数，例如：

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

性能函数问题

前向计算 a^l



反向传播 δ^l

欧式距离

$$\begin{cases} J = \dfrac{1}{2}\displaystyle\sum_{j=1}^{n^L}\left(a_j^L - y_j^L\right)^2 \\ \delta_i^L = \dfrac{\partial J}{\partial z_i^l} = \left(a_i^L - y_i^L\right)\cdot \dot{f}\left(z_i^L\right) \end{cases}$$

$$0 \le y_i^L \le 1\ (i = 1, \cdots, n_L)$$

$$a_i^L = f\left(z_i^L\right) = \frac{1}{1 + e^{-z_i^L}}$$

Sigmoid function

网络输出

$$a^L = \begin{bmatrix} a_1^L \\ \vdots \\ a_{n_L}^L \end{bmatrix}$$

目标输出

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

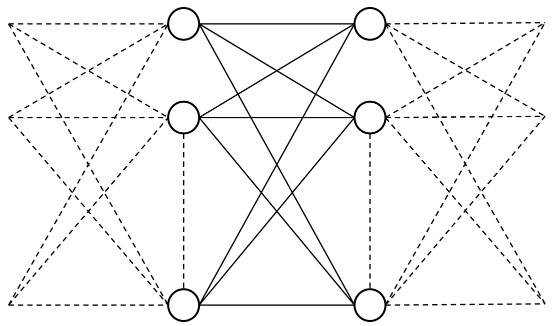
$$J(a^L, y^L)$$

性能函数 $J(a^L, y^L)$ 被用来刻画 a^L 和 y^L 的靠近程度，$J(a^L, y)$ 实际上是 $(w^1, \cdots, w^{L-1})$的函数，例如：

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

性能函数问题

前向计算 a^l



反向传播 δ^l

$$\sum_{j=1}^{n_L} y_i^L = 1$$

$$a_j^L = \frac{e^{z_j^L}}{e^{z_1^L} + \cdots + e^{z_{n_L}^L}}$$

Softmax function

网络输出

$$a^L = \begin{bmatrix} a_1^L \\ \vdots \\ a_{n_L}^L \end{bmatrix}$$

目标输出

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

$J(a^L, y^L)$

性能函数 $J(a^L, y^L)$ 被用来刻画 a^L 和 y^L 的靠近程度，$J(a^L, y)$ 实际上是$(w^1, \cdots, w^{L-1})$的函数，例如：

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

交叉熵

$$J = -\sum_{j=1}^{n_L} y_j^L \cdot \log(a_j^L) + \lambda \cdot \sum (w_{ij}^l)^2$$

$$a_j^L = \frac{e^{z_j^L}}{\sum_{i=1}^{n_L} e^{z_i^L}}$$

$$\delta_i^L = a_i^L - y_i^L$$

问题：为什么交叉熵可以刻画y^L和a^L的距离？

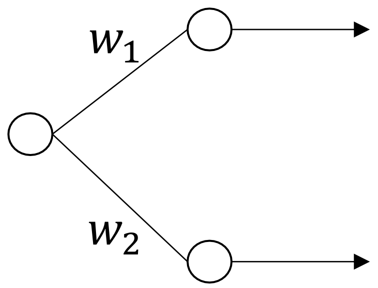
性能函数问题

一个例子

样本数据

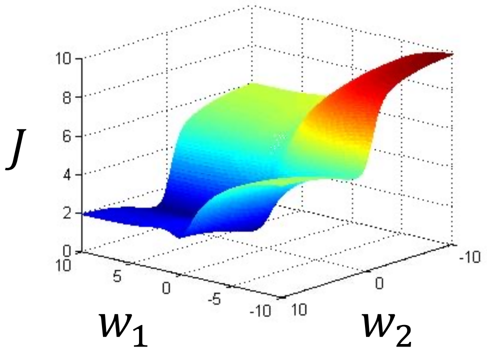
	1	2
x	0.8000	0.2000
y	0	1
	1	0

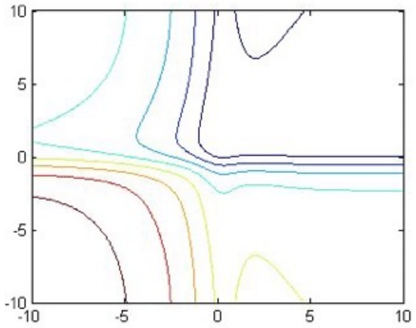
网络



欧式距离

$$\begin{cases} J = \frac{1}{2}\sum_{j=1}^{2}\left(a_j - y_j\right)^2 \\ a_j = \frac{1}{1+\exp(-z_j)} \\ z_j = w_j \cdot x \end{cases}$$

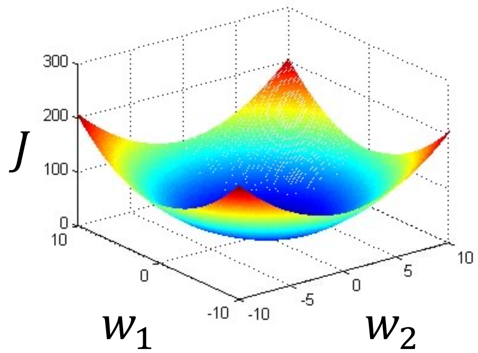


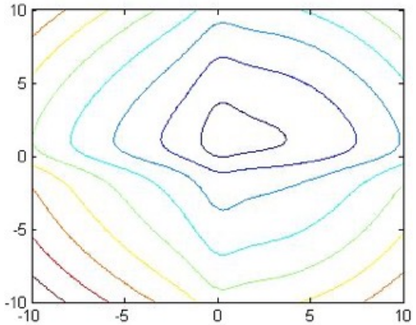


轮廓

交叉熵

$$\begin{cases} J = -\sum_{j=1}^{2} y_j \cdot \log\left(a_j\right) + \lambda\left(w_1^2 + w_2^2\right) \\ a_j = \frac{e^{z_j}}{\sum_{i=1}^{2} e^{z_i}} \\ z_j = w_j \cdot x \end{cases}$$





$\lambda = 0.05$

提纲

I

- 网络结构问题
- 学习算法问题
- 目标输出问题
- 网络输入问题

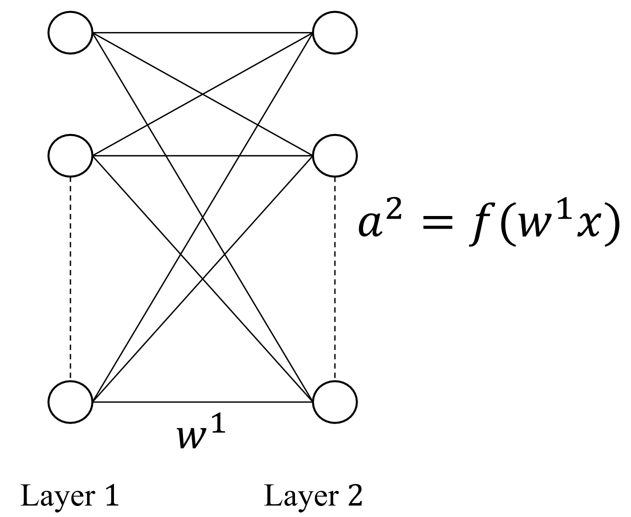
II

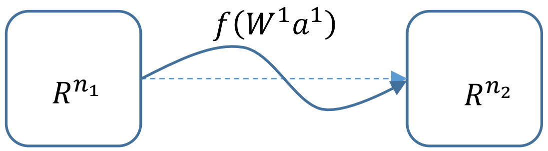
- 网络预测问题
- 性能函数问题
- 网络深度问题
- 训练数据问题

网络深度问题

浅层神经网络

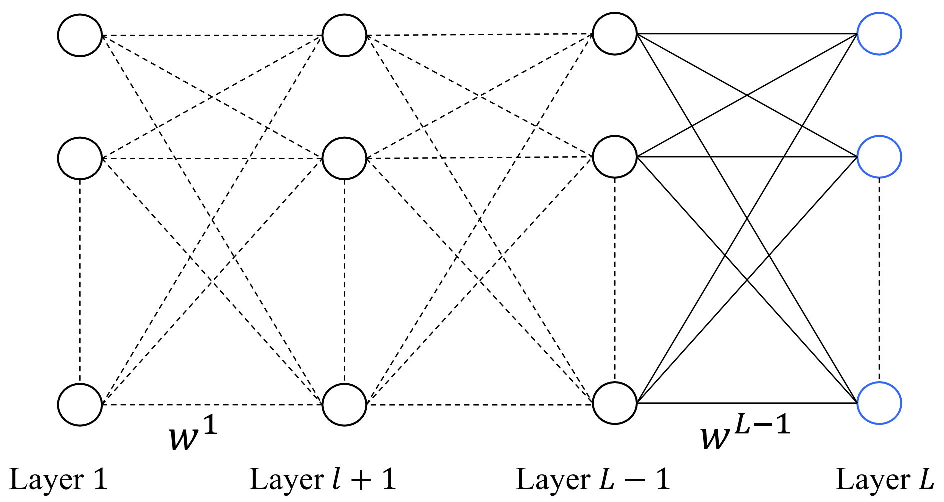
- $L = 2$
- 难以学习到复杂的非线性映射

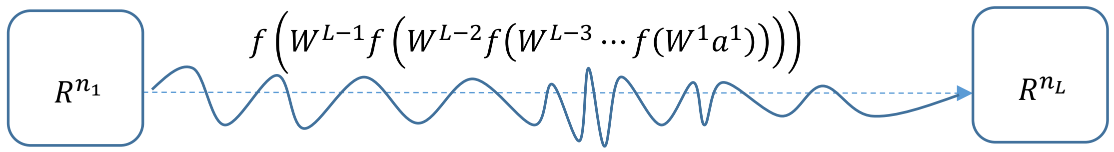




深层神经网络

- $L > 2$
- 只要网络中的神经元数目足够多，可以任意精度拟合任意非线性映射

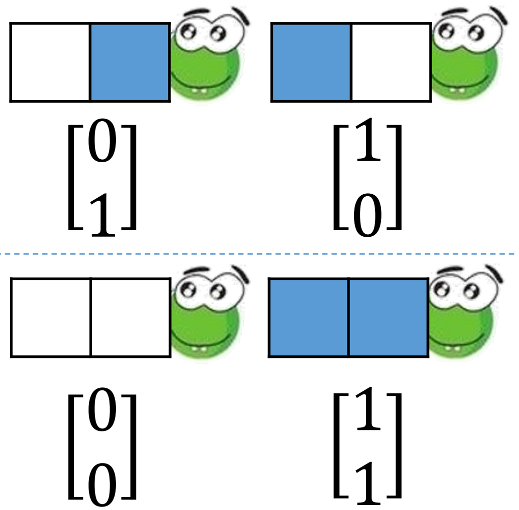


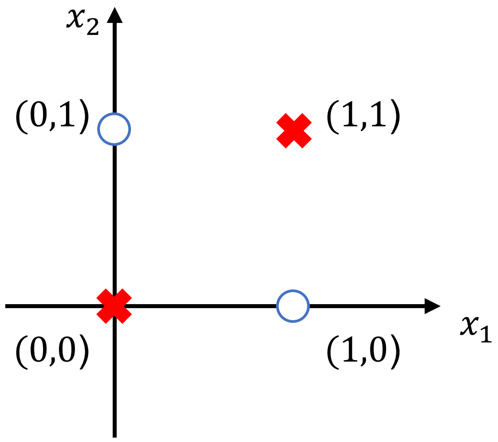


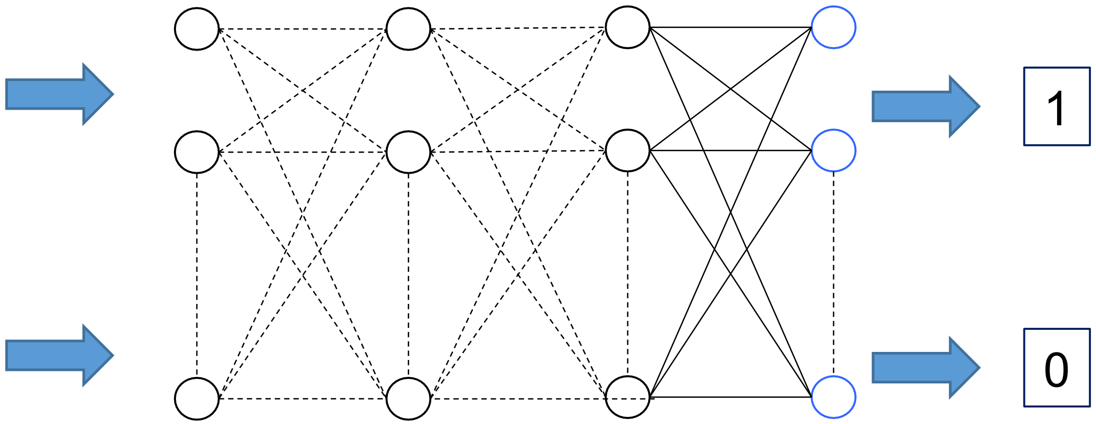
例子：异或问题

有斑点的毛毛虫

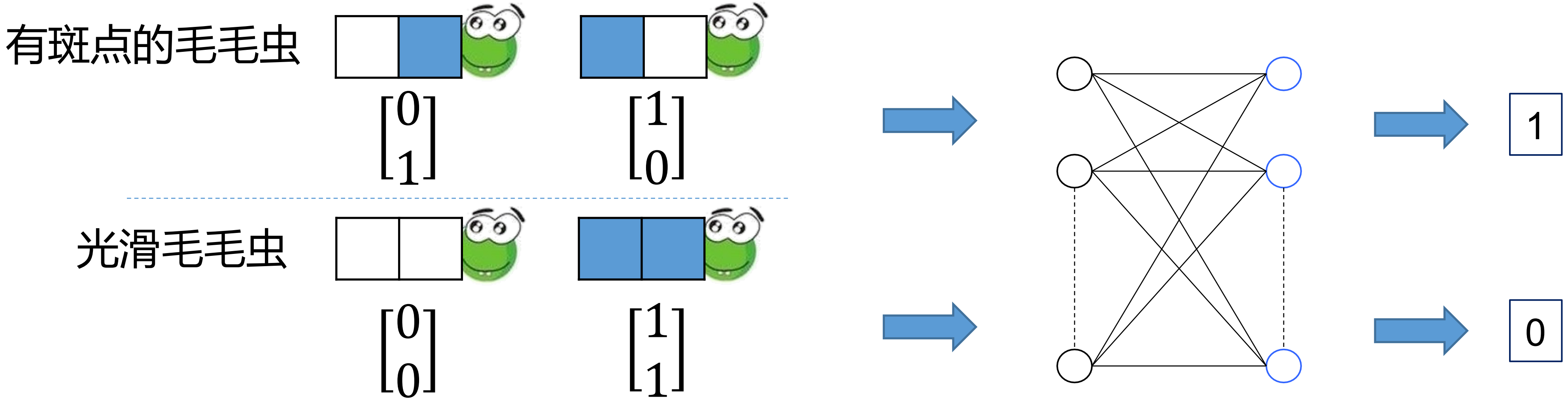
光滑毛毛虫

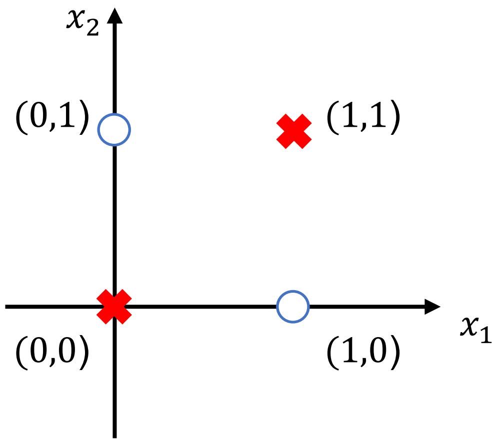






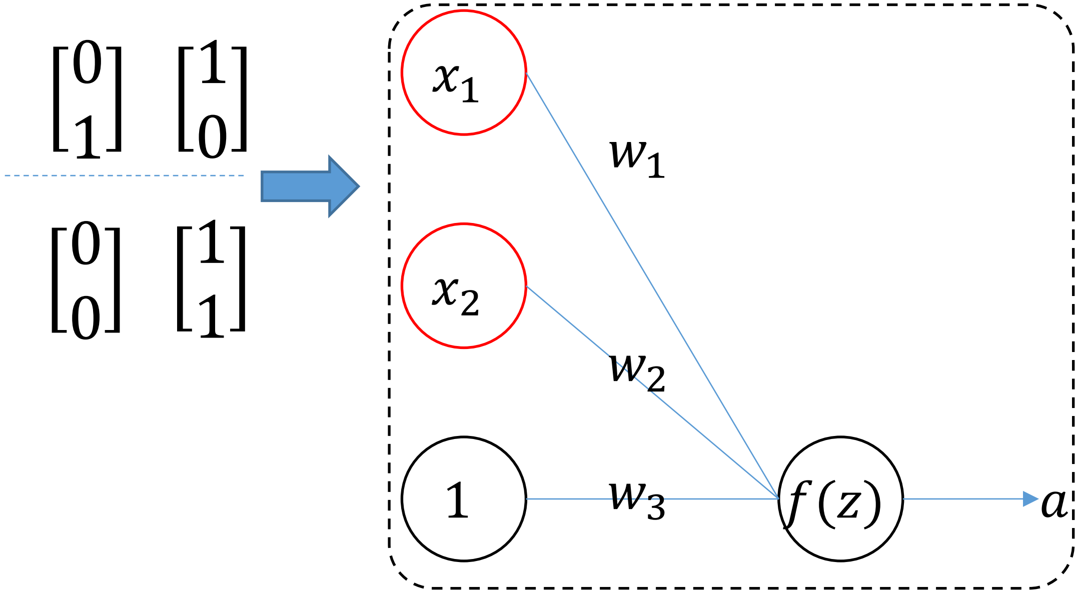
例子：异或问题





两层网络在激活函数是单调函数的情况下无法完成XOR分类任务

例子：异或问题



等价

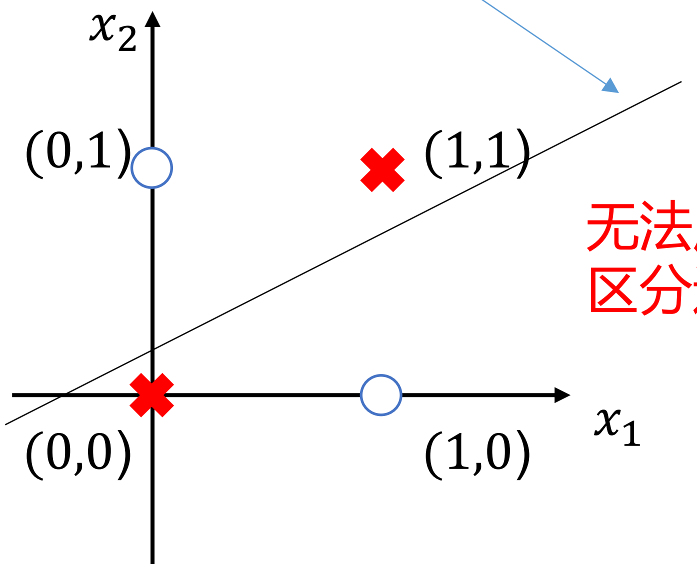
f是单调函数

$$a = f(z)$$

$$z = w_1x_1 + w_2x_2 + w_3$$

决策线

$$w_1x_1 + w_2x_2 + w_3 = 0$$

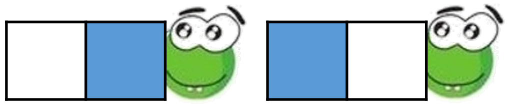


无法用一条直线区分这两个类别

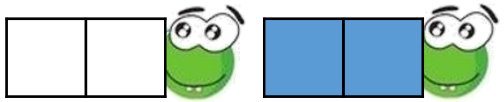
结论：两层网络在激活函数是单调函数的情况下无法完成XOR分类任务

问题：两层网络在激活函数为非单调函数的情况下能否完成XOR分类任务？

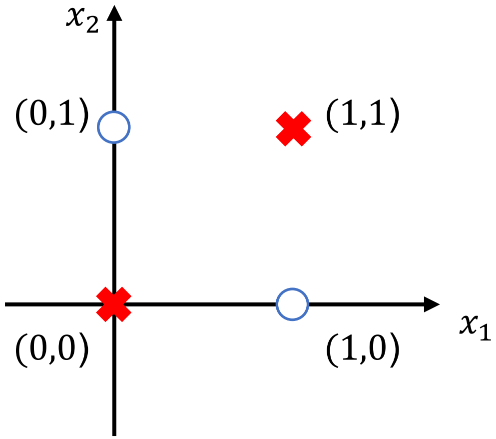
例子：异或问题

不光滑毛毛虫 

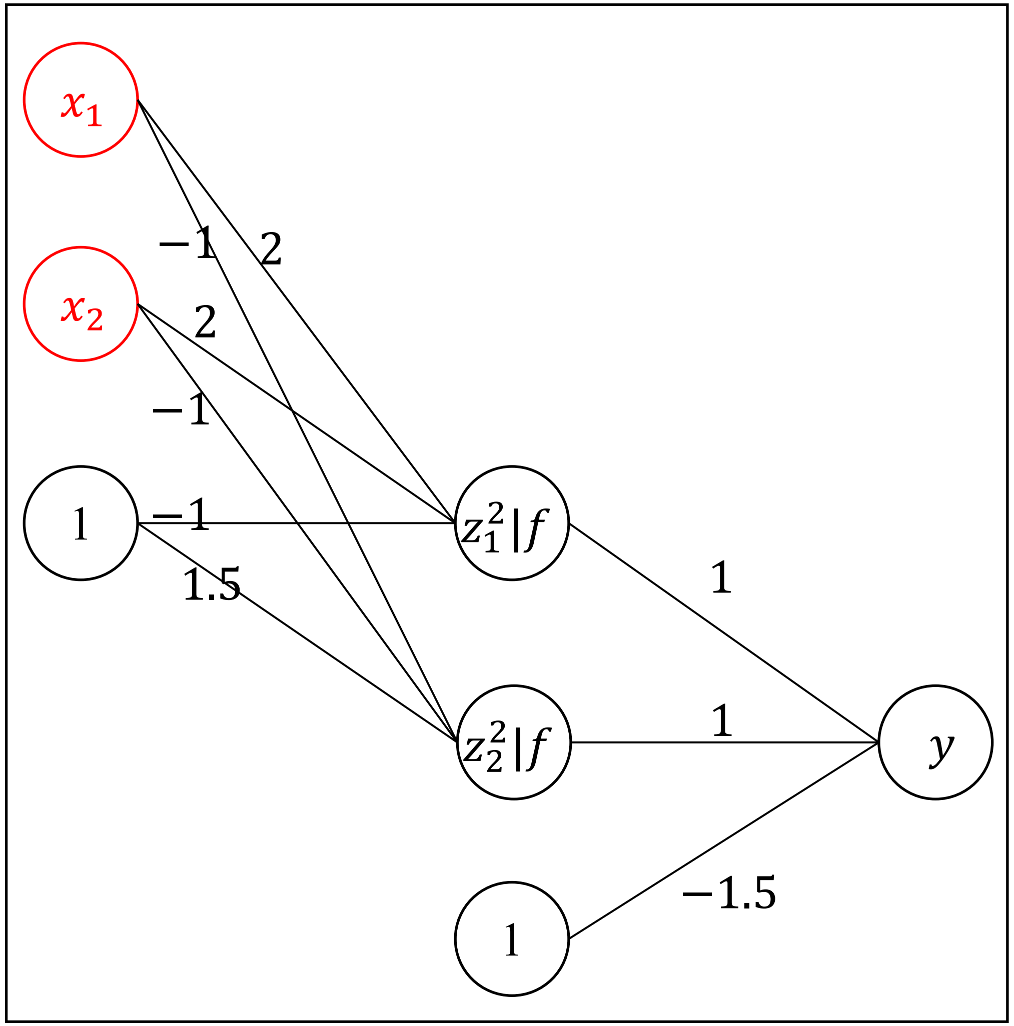
$\begin{bmatrix}0\\1\end{bmatrix}$ $\begin{bmatrix}1\\0\end{bmatrix}$

光滑毛毛虫 

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



三层网络可以解决异或问题



1

0

网络深度问题

梯度消失问题

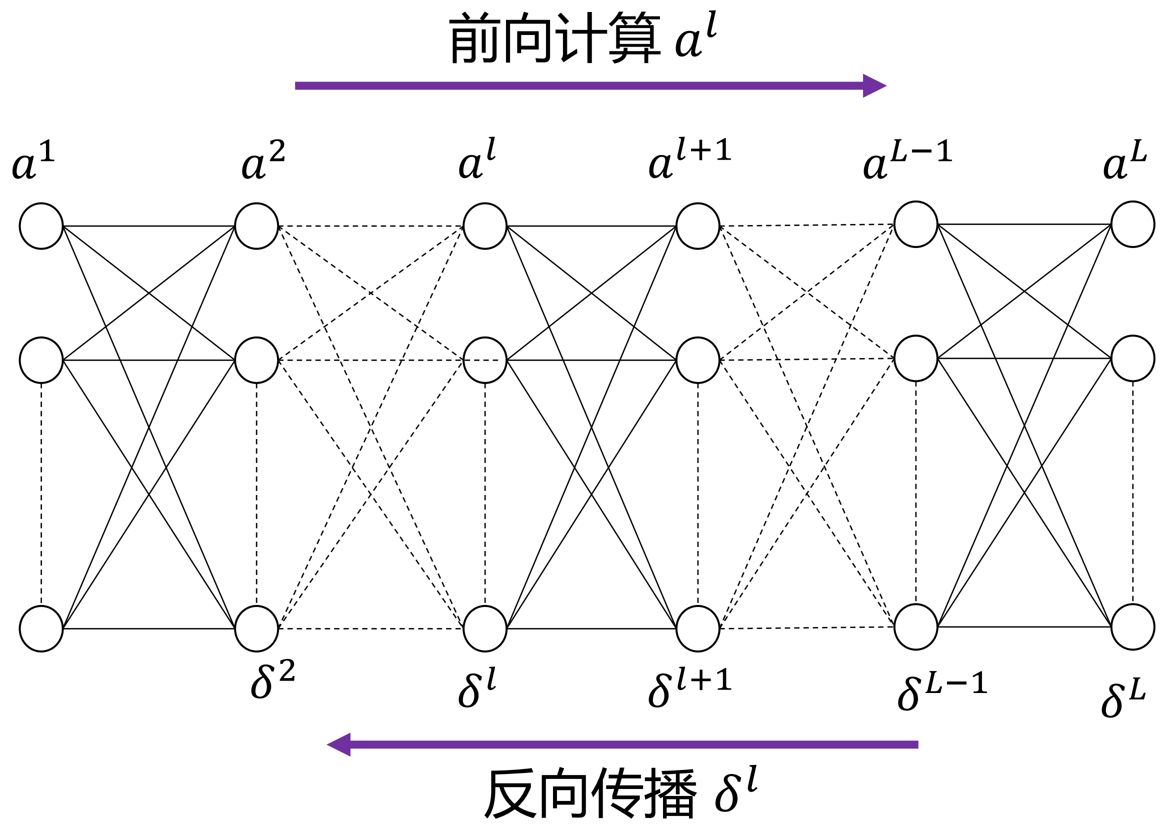
代价函数: $J(w^1, \cdots, w^{L-1})$

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

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

关键:

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



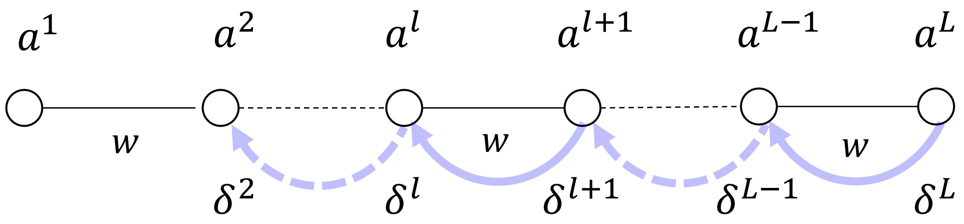
网络深度问题

梯度消失问题

一个简单的例子

$$w = w^{l}$$

$$\delta^{l} = \dot{f}\left(z^{l}\right) \cdot w \cdot \delta^{l+1}$$

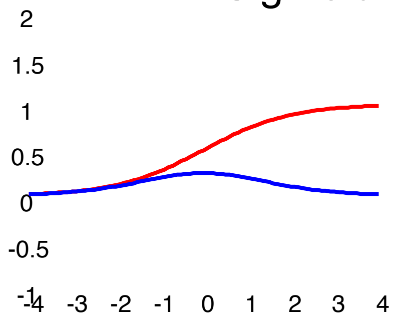


$$\begin{aligned}
\delta^{l} &= \dot{f}\left(z^{l}\right) \cdot w \cdot \delta^{l+1} \\
&= \dot{f}\left(z^{l}\right) \cdot w \cdot \dot{f}\left(z^{l+1}\right) \cdot w \cdot \delta^{l+2} \\
&= w \cdot \dot{f}\left(z^{l}\right) \cdot w \cdot \dot{f}\left(z^{l+1}\right) \cdots w \cdot \dot{f}\left(z^{L-1}\right) \cdot \delta^{L} \\
&= \prod_{m=L-1}^{l} \left(w \cdot \dot{f}(z^{m})\right) \cdot \delta^{L} \to 0
\end{aligned}$$

$$\left|\frac{\partial \delta^{l}}{\partial \delta^{L}}\right| = \prod_{m=L-1}^{l} \left|w \cdot \dot{f}(z^{m})\right| \leq |w|^{L-l+1} \cdot (0.25)^{L-l+1}$$

$\dot{f}(z^{m}) \leq 0.25$

Sigmoid

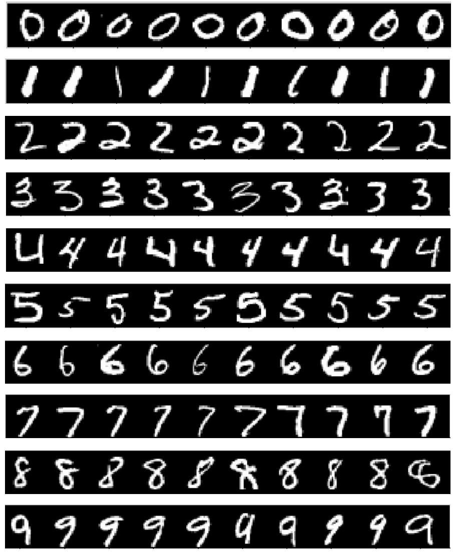


Notes：

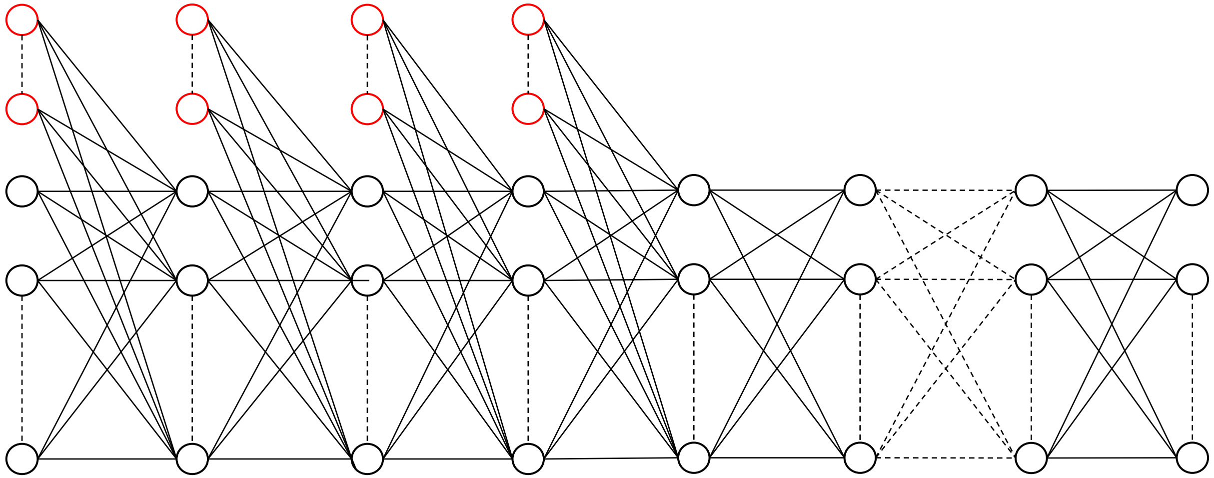
δ^{l} 的指数下降导致梯度消失问题。

网络深度问题

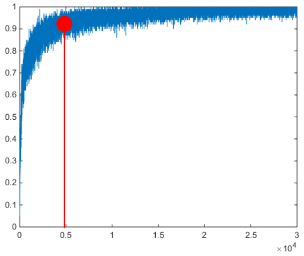
网络的深度与具体问题相关



手写体数字识别问题



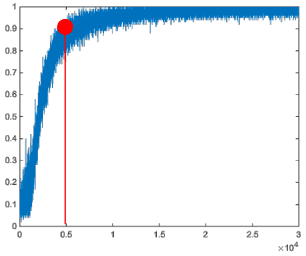
5 layers



准确率

- Training=97.55%
- Testing=95.25%

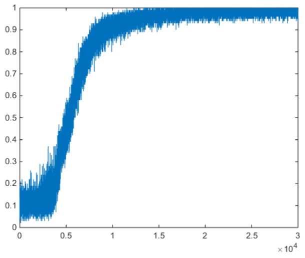
8 layers



准确率

- Training= 98.65%
- Testing= 95.10%

9 layers



准确率

- Training=98.45%
- Testing=93.20%

提纲

I

- 网络结构问题
- 学习算法问题
- 目标输出问题
- 网络输入问题

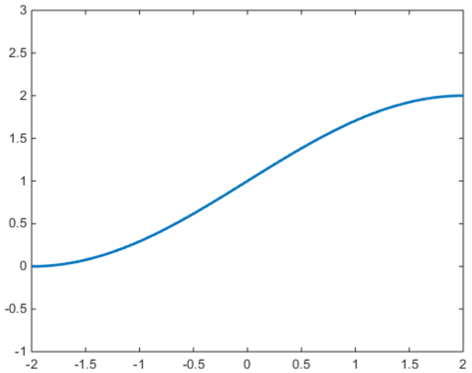
II

- 网络预测问题
- 性能函数问题
- 网络深度问题
- 训练数据问题

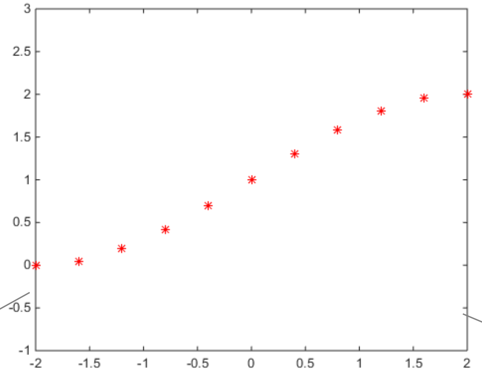
训练数据问题

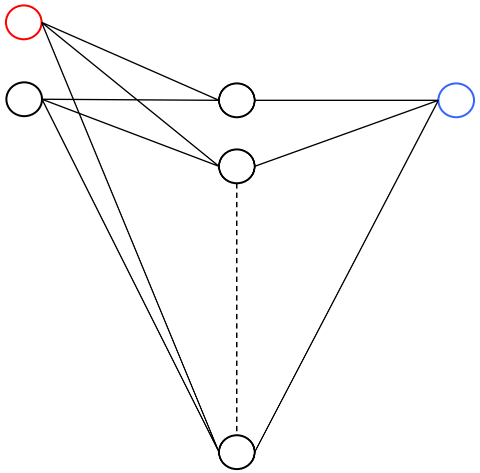
用 2-9-1 网络拟合一个部分正弦曲线

$$y = g(x) = 1 + \sin\left(\frac{\pi}{4}x\right), x \in [-2,2]$$



采样





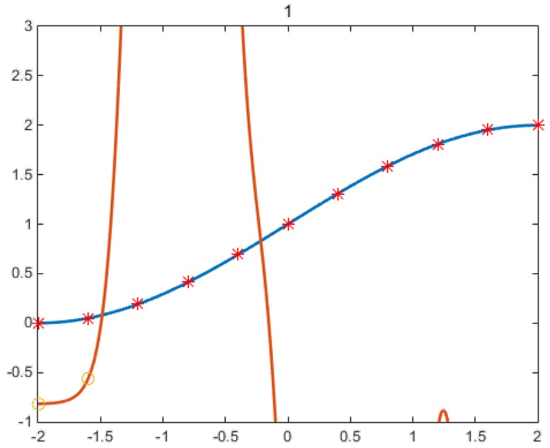
(1,1) → (0,9) → (0,1)

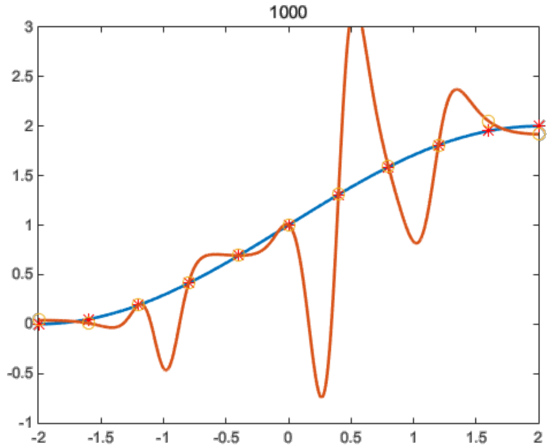
	1	2	3	4	5	6	7	8	9	10	11
x	-2	-1.6000	-1.2000	-0.8000	-0.4000	0	0.4000	0.8000	1.2000	1.6000	2
y	0	0.0489	0.1910	0.4122	0.6910	1	1.3090	1.5878	1.8090	1.9511	2

11 samples

训练数据问题

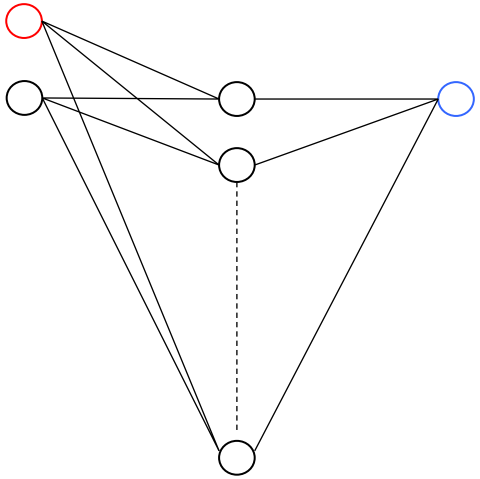
过拟合





11 个数据样本

用2-9-1网络拟合一个部分正弦曲线

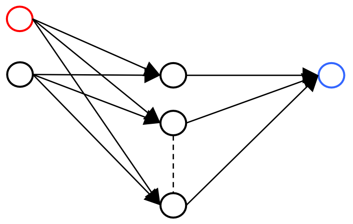


2-9-1 网络有 27 个权重需要调优。一般来说，需要比参数数量更多的样本来进行训练。

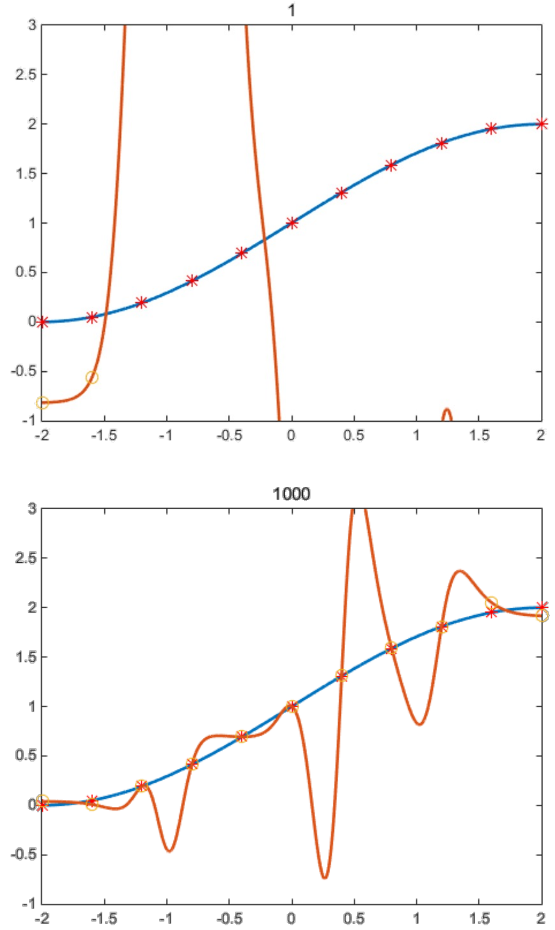
网络与数据样本拟合得很好，但在曲线的其他部分表现不好！过拟合!

- 在训练数据上拟合的很好
- 无法良好拟合测试数据
- 需要**更多**的数据！

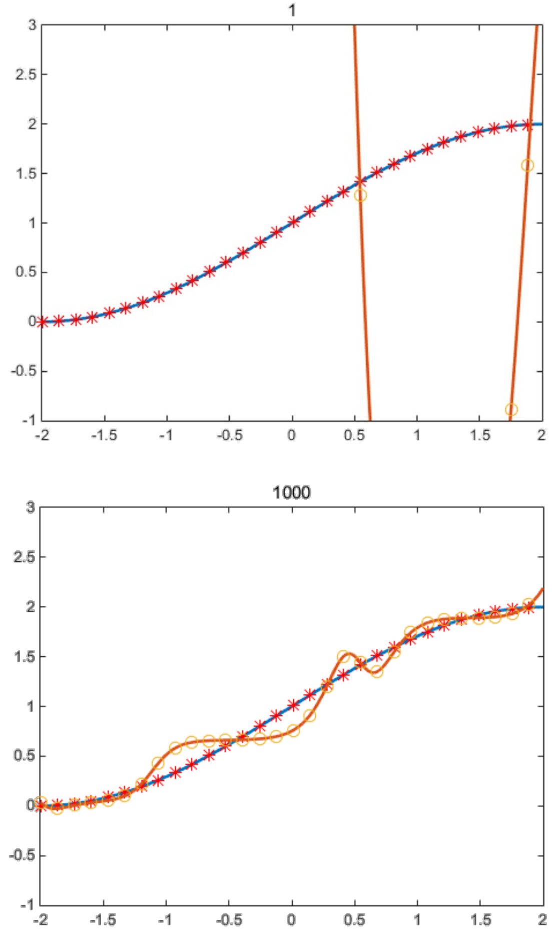
训练数据问题



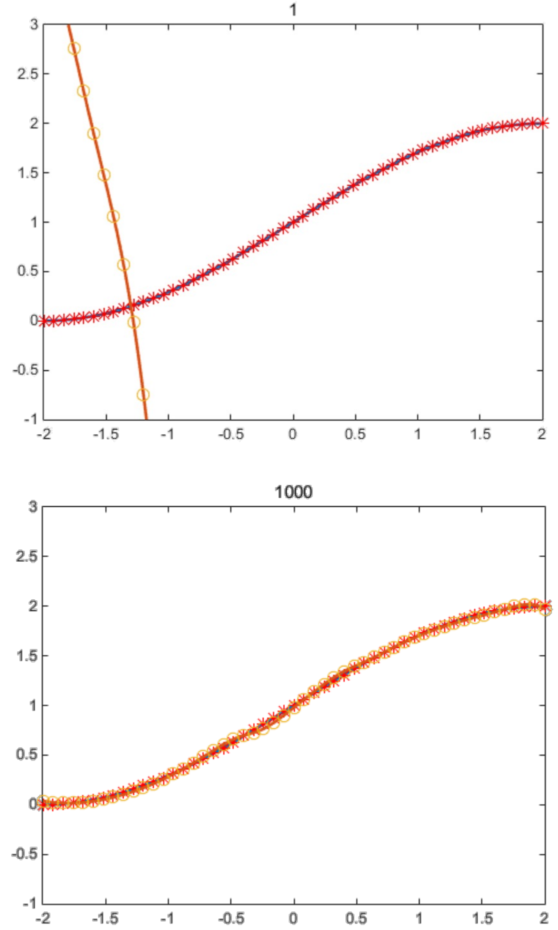
(1,1)→ (0,9)→ (0,1)



11 个数据样本

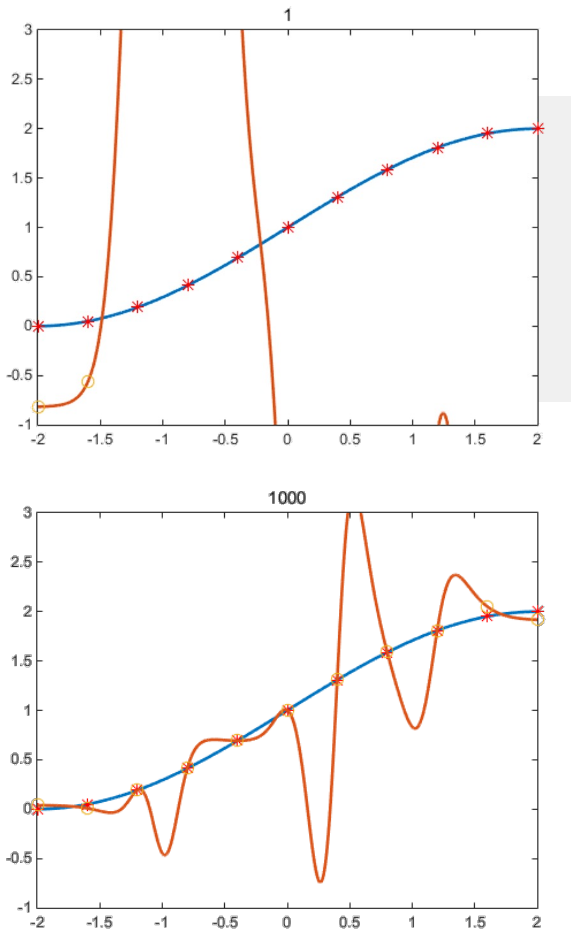


23 个数据样本

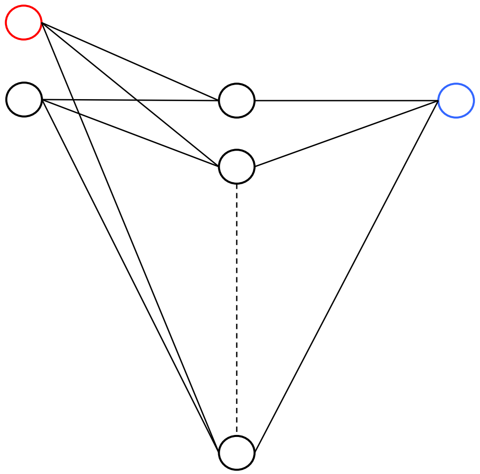


51 个数据样本

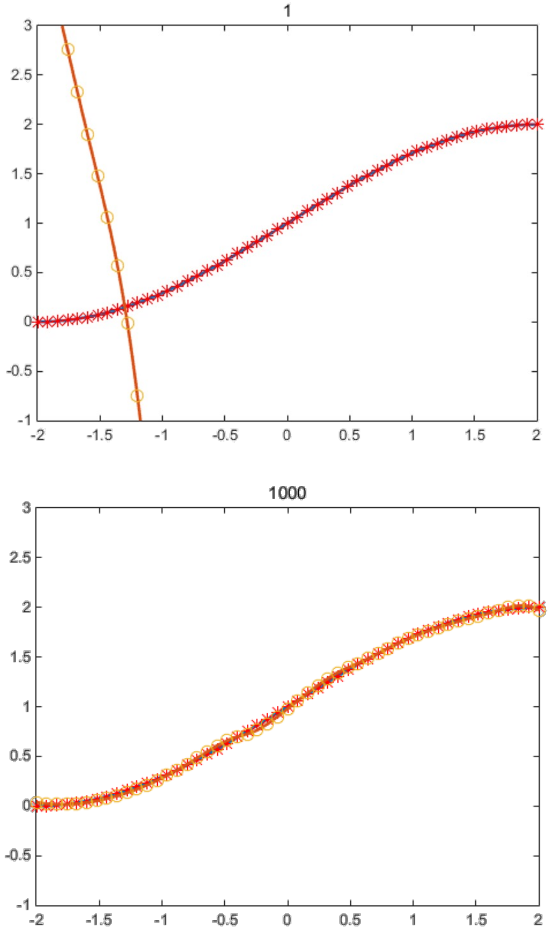
训练数据问题



11 个数据样本



为了使网络具有泛化性，网络参数个数应该少于训练集中样本数据的个数。



51 个数据样本

训练数据问题

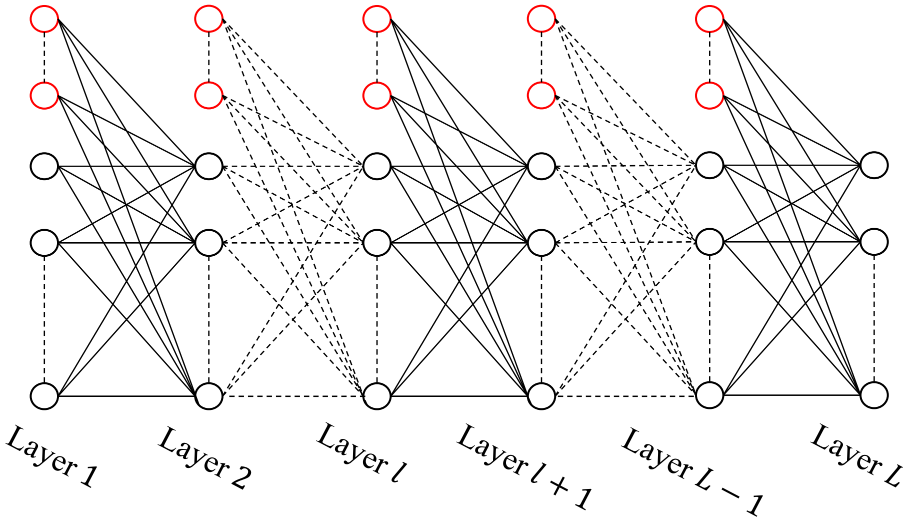
大数据

大数据中的复杂模式需要复杂的模型来处理。

丰富的数据可以用于模型训练。(样本)

高度非线性，灵活，可训练的模型。(复杂性)

深度神经网络



深度神经网络模型中有大量的参数需要训练。

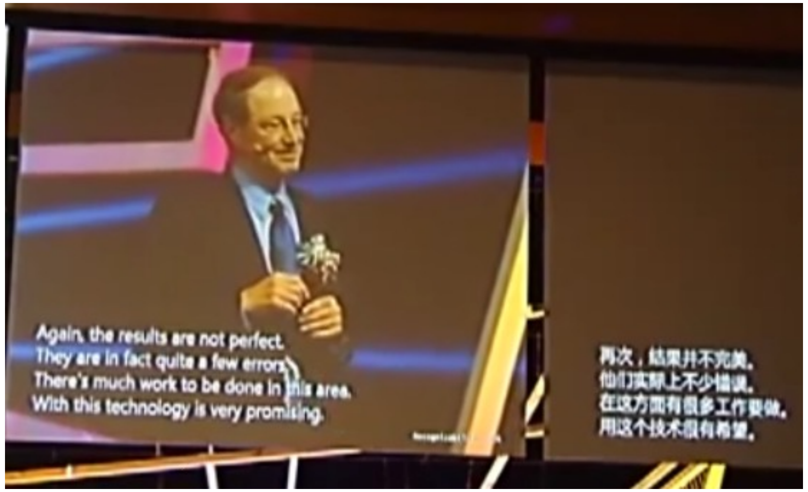
大数据+ 深度神经网示例

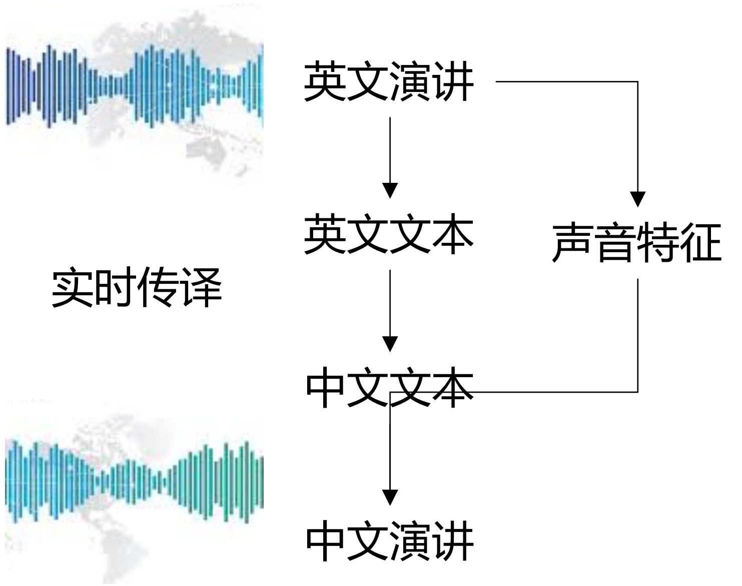
语音识别

1950s　语音波形 + 模式识别 = 识别少量词汇

1970s　高斯混合模型 + 隐马尔可夫模型 = ~80%的识别率

2011　深度神经网络建模的语音模型 = 令人惊讶的实时识别!





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