模式识别与机器学习

第五章 神经网络

模式识别与机器学习

5. 1、 神经网络的概念

引言

人工神经网络(Artificial neural network)

- 由多个神经元组成的一种模仿生物神经网络结构和功能的数学模型。

神经网络的主要组件

- 神经元(neuron)： 神经网络的基本处理单元。
- 神经网络结构：神经元之间的连接结构。

01 神经元

神经元

神经元计算模型

- 针对神经元j： $z_j(\boldsymbol{x}) = g(\boldsymbol{w}_j^T\boldsymbol{x} + w_{0j}) = g\left(\sum_{i=1}^{P} w_{ij}x_i + w_{0j}\right)$

- 神经元输入信号：

$$\boldsymbol{x} = [x_1, x_2, \ldots, x_P]^T$$

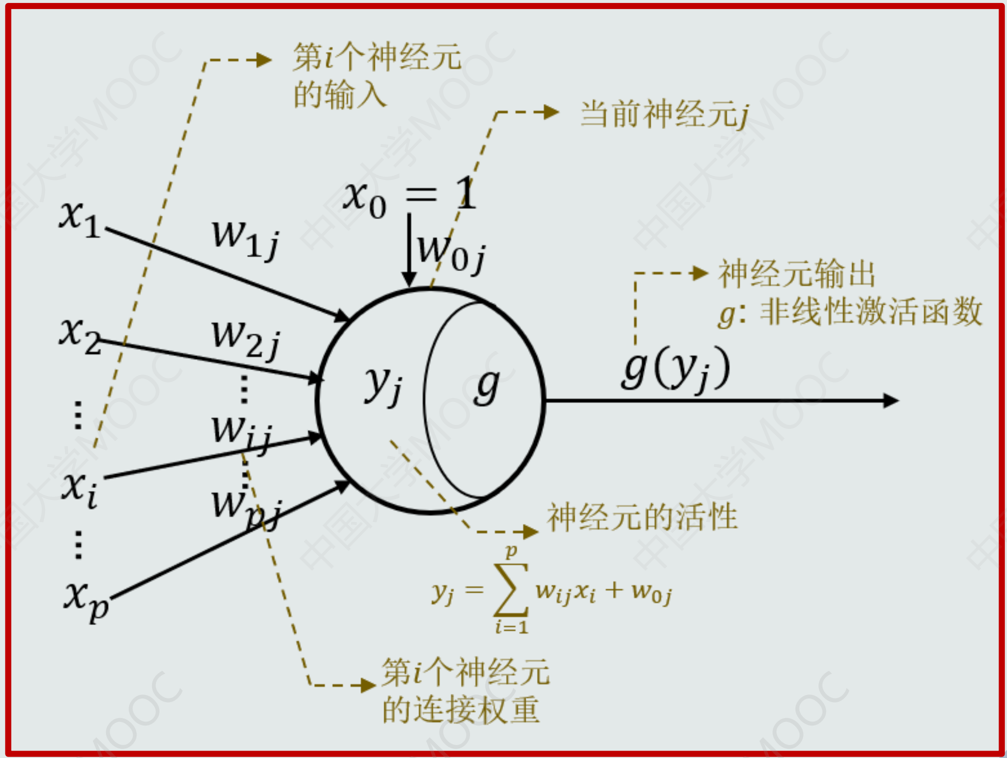
- 网络权重：

$$\boldsymbol{w}_j = [w_{1j}, w_{2j}, \ldots, w_{Pj}]^T$$

- 偏置量： w_{0j}

- 设$x_0 = 1$：

$$z_j(\boldsymbol{x}) = g\left(\sum_{i=0}^{P} w_{ij}x_i\right)$$



神经元

神经元计算模型

- 针对神经元j：

$$z_j(\boldsymbol{x}) = g(\boldsymbol{w}_j^T \boldsymbol{x} + w_{0j})$$

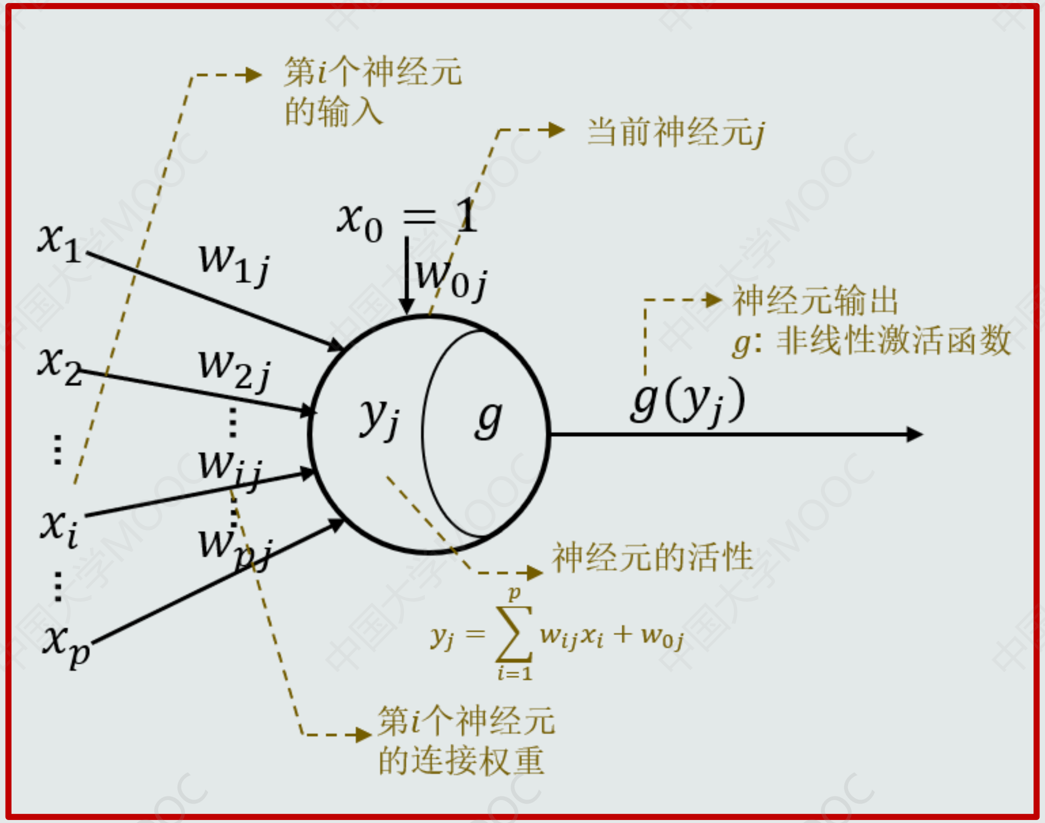
- 神经元活性(activation)：

$$y_j = \boldsymbol{w}_j^T \boldsymbol{x} + w_{0j}$$

- 神经元激活函数g（activation function）：非线性函数

$$g(y_j)$$

- 神经元输出：$z_j = g(y_j)$



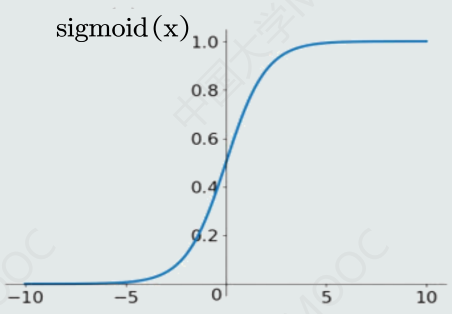
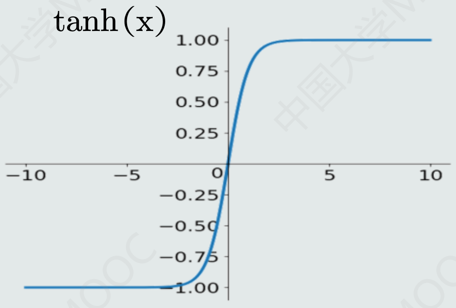
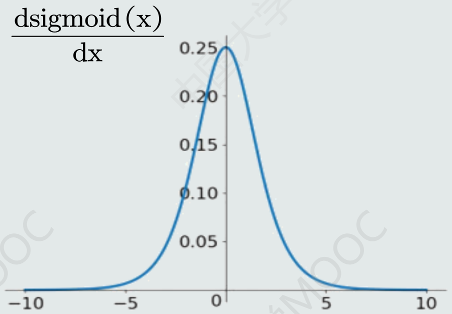
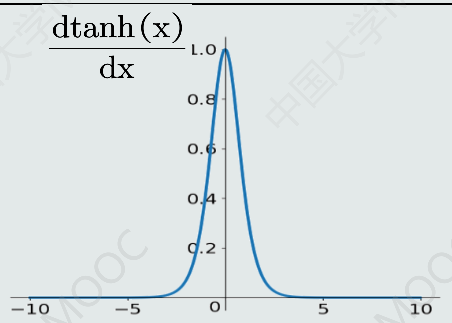
神经元

神经元与线性判据

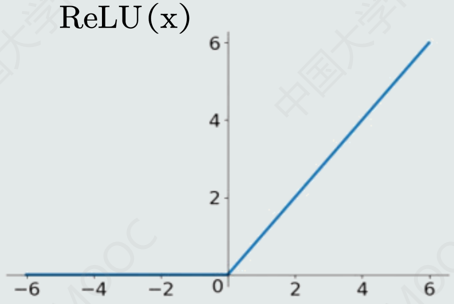
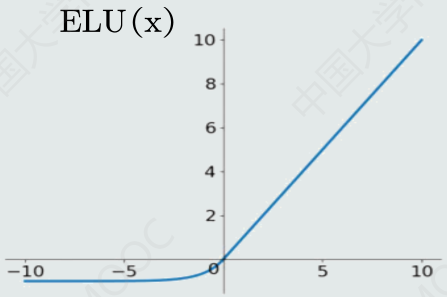
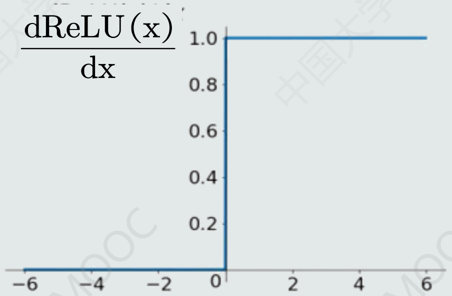
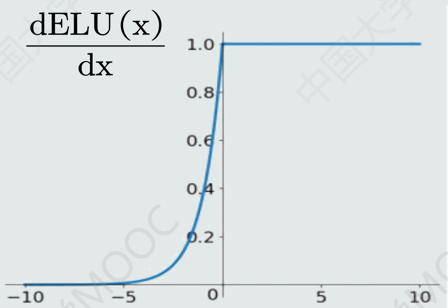
$$z_j(\boldsymbol{x}) = g(\boldsymbol{w}_j^T \boldsymbol{x} + w_{0j})$$

- 神经元活性计算相当于输入信号向量和权重向量之间的相似度测量。
- 如果激活函数是符号函数(sign函数)，则神经元相当于一个线性判据。
- 如果激活函数是Sigmoid函数，则神经元相当于一个Logistic回归模型。

常用的激活函数

	sigmoid	tanh
函数表达式	$\mathrm{sigmoid}(x)=\dfrac{1}{1+e^{-x}}$	$\tanh(x)=\dfrac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
梯度表达式 (可微分)	$\dfrac{d\,\mathrm{sigmoid}(x)}{dx}=\mathrm{sigmoid}(x)\,(1-\mathrm{sigmoid}(x))$	$\dfrac{d\tanh(x)}{dx}=1-(\tanh(x))^{2}$
函数波形表达		
梯度波形表达		

常用的激活函数

	ReLU (Rectified Linear Units)	ELU (Exponential Linear Units)
函数表达式	$\mathrm{ReLU}(x) = \max(0, x)$	$\mathrm{ELU}(x) = \begin{cases} x, & \text{if } x > 0 \\ a(e^x - 1), & \text{if } x \leq 0 \end{cases}$
梯度表达式 (不可微)	$\frac{d\mathrm{ReLU}(x)}{dx} = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases}$	$\frac{d\mathrm{ELU}(x)}{dx} = \begin{cases} 1, & \text{if } x > 0 \\ ae^x, & \text{if } x \leq 0 \end{cases}$
函数波形表达		
梯度波形表达		

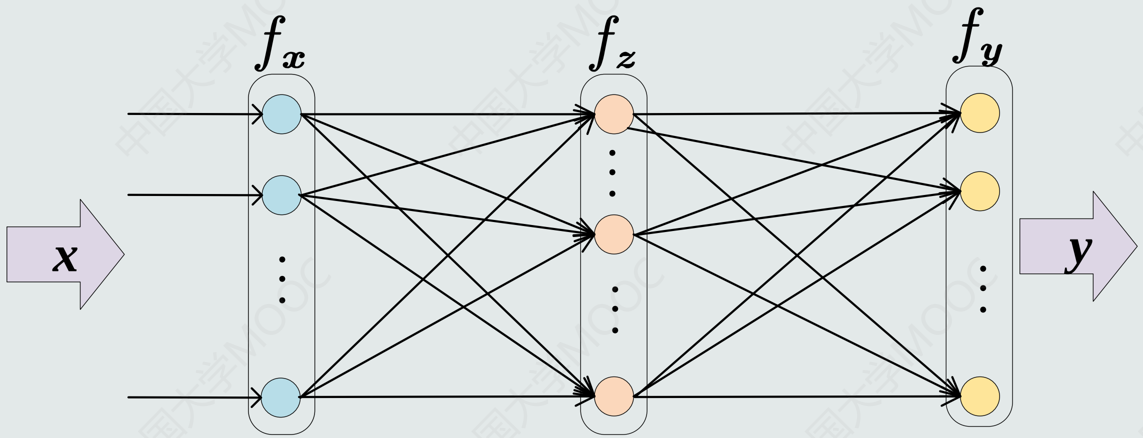
02

神经网络

神经网络结构类型

前馈（feedforward）神经网络

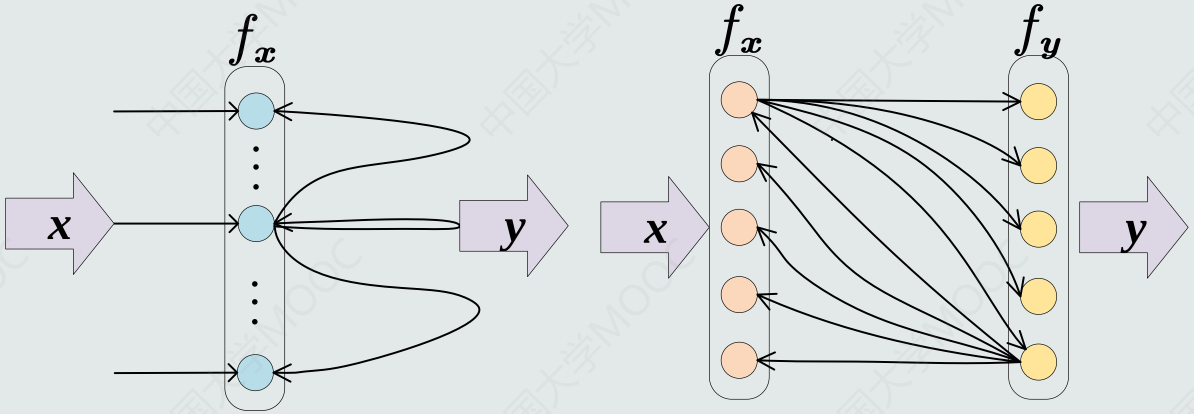
- 整个网络中，信号不构成回路。



神经网络结构类型

反馈（feedback）神经网络

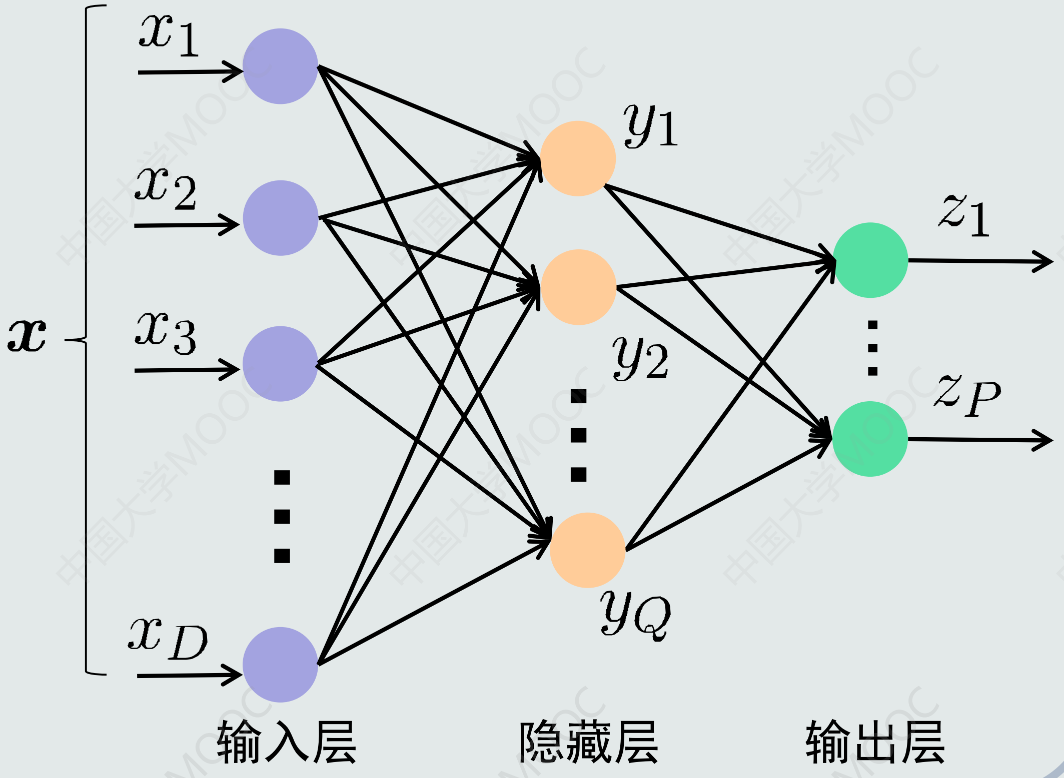
- 网络中存在信号回路。



单隐层神经网络

输入层

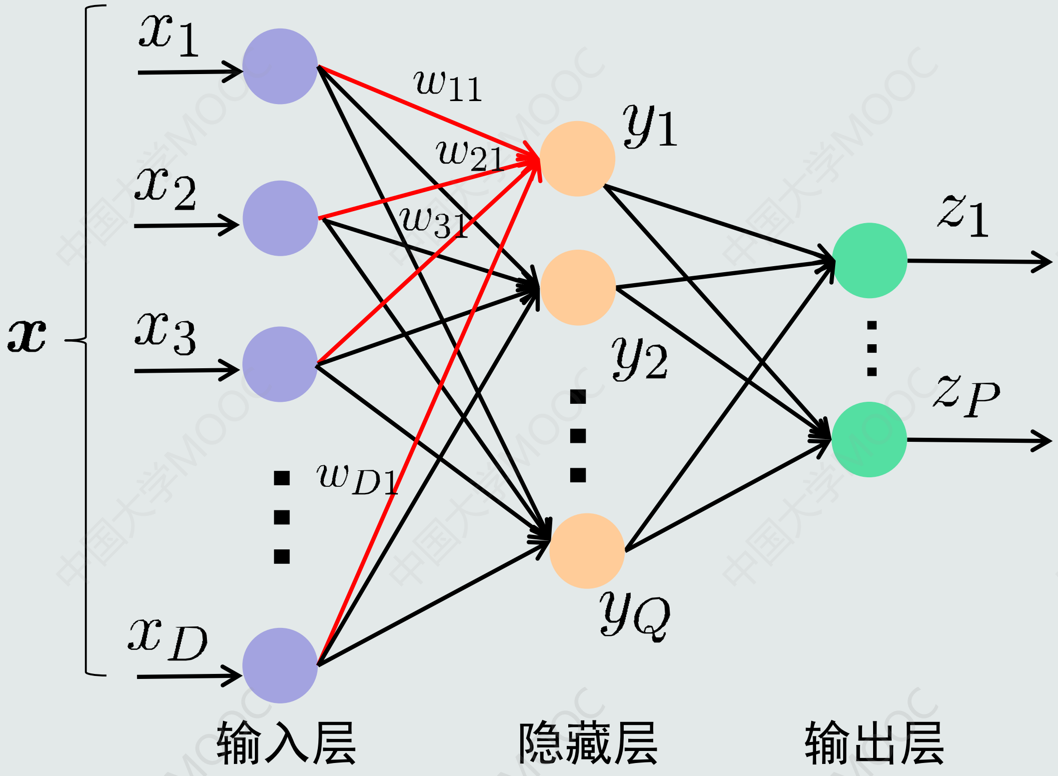
- 输入层：由D个线性神经元组成。该层神经元无权重表达。
- 单个神经元d的输入为：
  $$x_d$$
- 输入层的输入为：
  $$\boldsymbol{x} = [x_1, \ldots, x_D]^T$$
- 单个神经元d的输出为：
  $$x_d$$
- 输入层的输出为：
  $$\boldsymbol{x} = [x_1, \ldots, x_D]^T$$



单隐层神经网络

隐藏层

- 隐藏层：由Q个非线性神经元组成。
- 单个神经元q的输入为：
  $$\boldsymbol{x} = [x_1, \ldots, x_D]^T$$
- 单个神经元q的权重为：
  $$\boldsymbol{w}_q = [w_{1q}, \ldots, w_{Dq}]^T$$
- 单个神经元q的输出为：
  $$y_q = g(\boldsymbol{w}_q^T \boldsymbol{x})$$
- 隐藏层所有神经元的输出：
  $$\boldsymbol{y} = [y_1, \ldots, y_Q]^T$$



单隐层神经网络

输出层

- 输出层：由P个非线性神经元组成。
- 单个神经元p的输入为：

$$\boldsymbol{y} = [y_1, \ldots, y_Q]^T$$

- 单个神经元p的权重为：

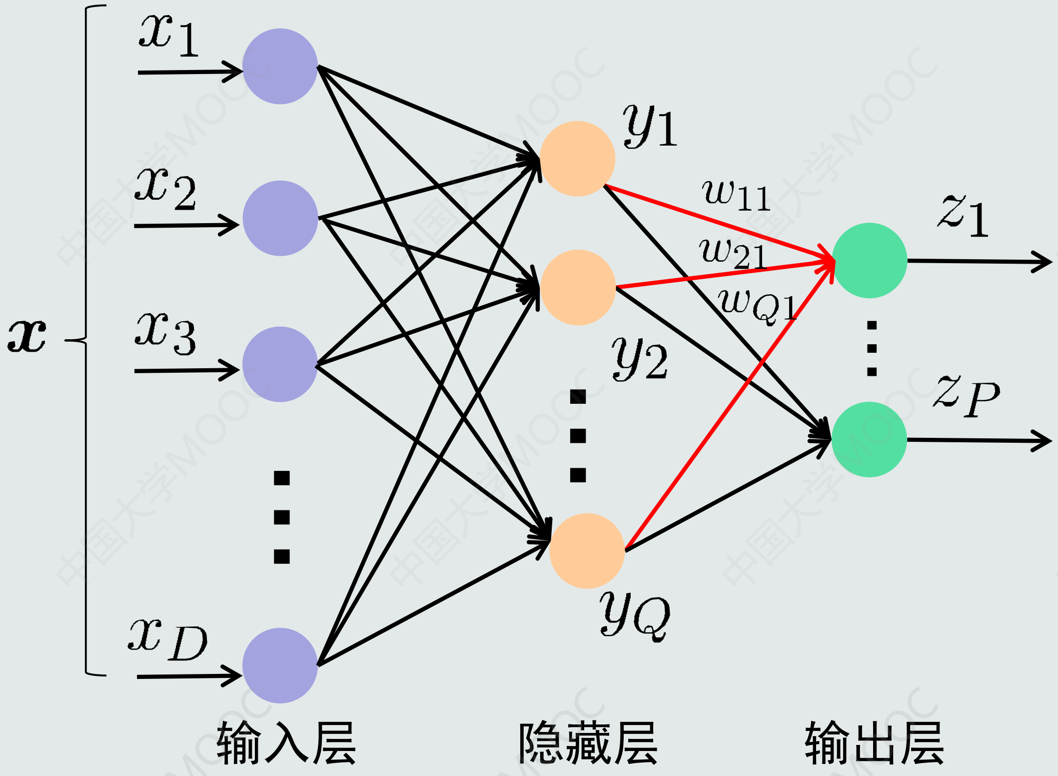
$$\boldsymbol{w}_p = [w_{1p}, \ldots, w_{Qp}]^T$$

- 单个神经元p的输出为：

$$z_p = g(\boldsymbol{w}_p^T \boldsymbol{y})$$

- 输出层所有神经元的输出：

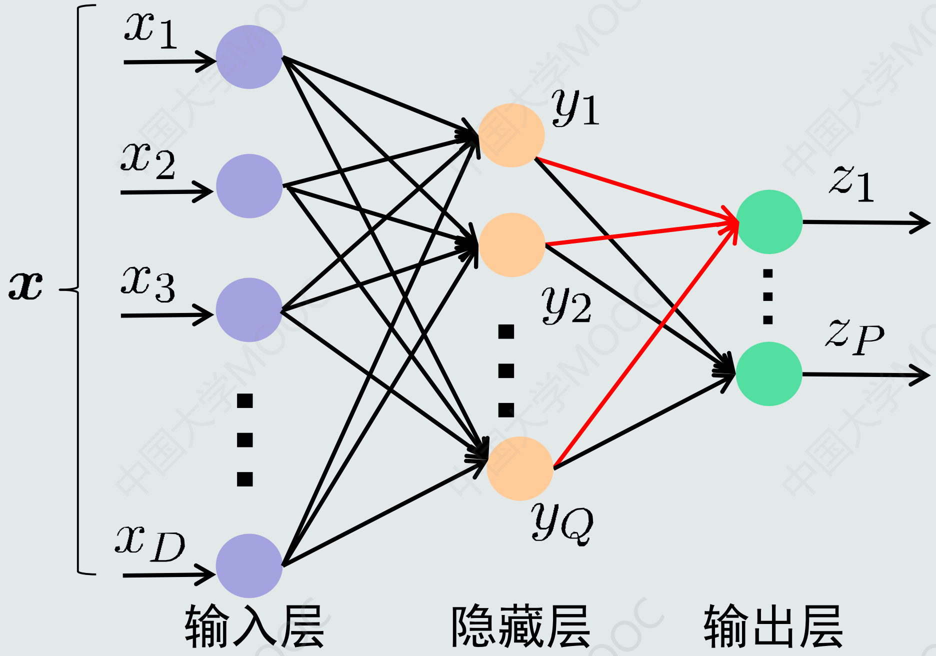
$$\boldsymbol{z} = [z_1, \ldots, z_P]^T$$



全连接

全连接

- 全连接（full connection）：当前层每个神经元都与前后层的每个神经元存在连接关系。



隐藏层的数学解释

线性变换 + 非线性激活函数

- 每个隐藏层所有神经元的计算相当于矩阵W与该层输入向量相乘：

$$y_q = g(\boldsymbol{w}_q^T \boldsymbol{x}) \quad \Rightarrow \quad \boldsymbol{y} = g(\boldsymbol{W}^T \boldsymbol{x})$$

$$\text{where } \boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m, \ldots, \boldsymbol{w}_Q]$$

- 每个隐藏层相当于：线性变换+非线性激活函数。
- 因此，对于分类而言，隐藏层的作用就是将输入信号通过线性变换和非线性激活函数，投影到输出空间，力图使得在输入空间线性不可分的数据经过投影后，在输出空间线性可分。

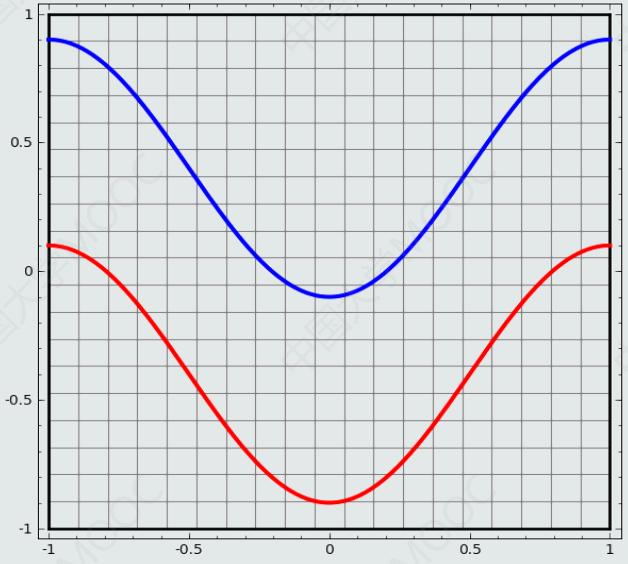
隐藏层的几何解释

神经网络 vs. Kernel SVM

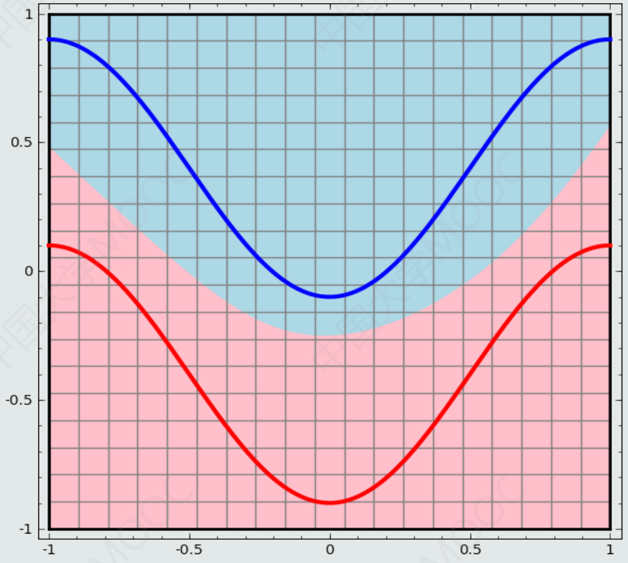
- 线性映射矩阵W+非线性激活函数，以解析函数的形式表达输入空间到输出空间的非线性映射。
- 映射矩阵W由训练算法学习出来的。

- 没有直接表达输入空间到输出空间的非线性映射，而是通过核函数，定义了输入空间向量之间的点积与映射到输出空间后的点积之间的关系。
- 手动指定核函数。
- 支持向量由训练算法得到。

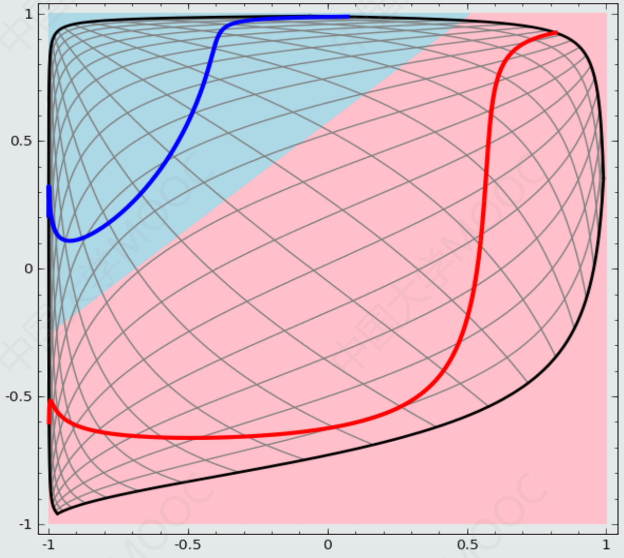
隐藏层的几何解释



输入空间
（C_1 类：红线，C_2 类：蓝线）



输入空间分类边界（真值）



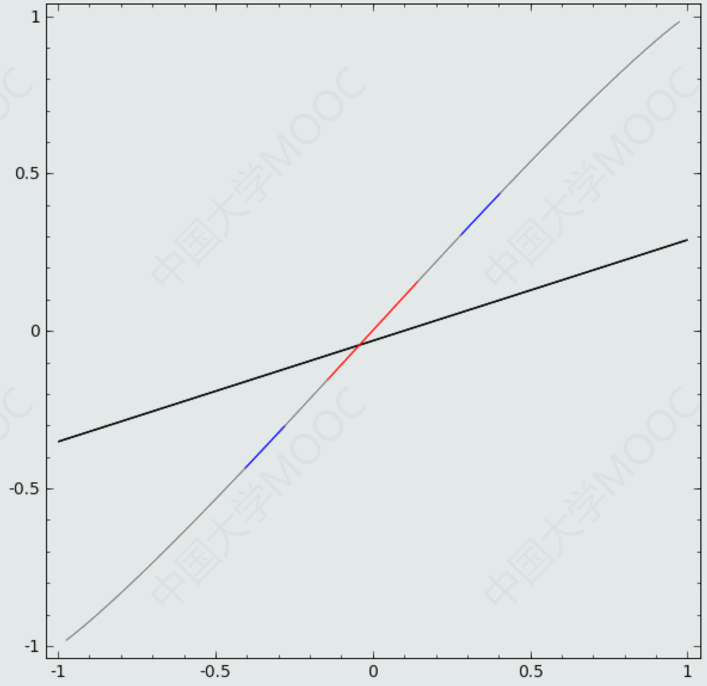
隐藏层

隐藏层的几何解释

线性变换：低维→高维

- 为了实现线性不可分到线性可分的映射，通常$\boldsymbol{W}$矩阵是从低维到高维：

输入空间（1维）
（C_1类：红线，C_2类：蓝线）

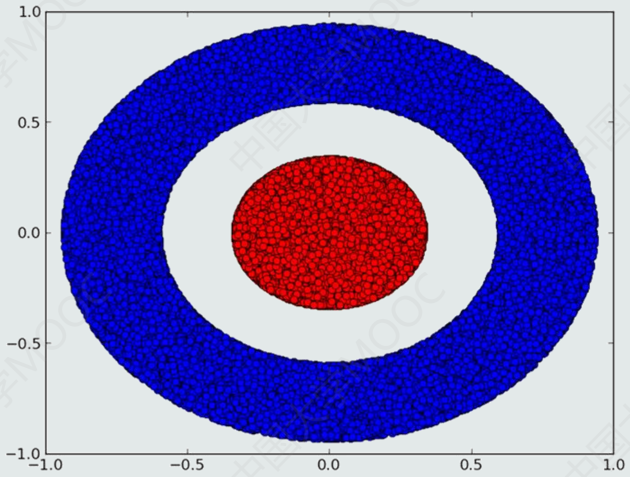


隐藏层（1维→2维）

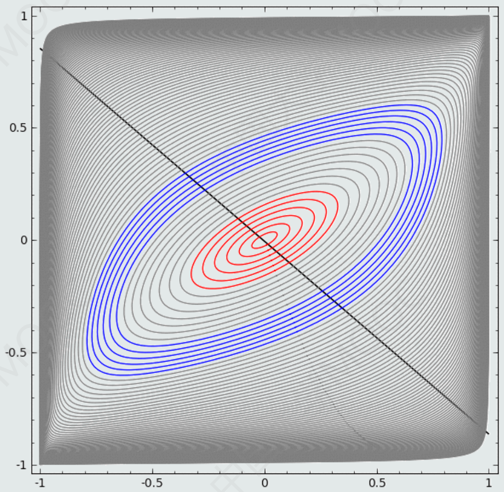
隐藏层的几何解释

线性转换：低维→高维

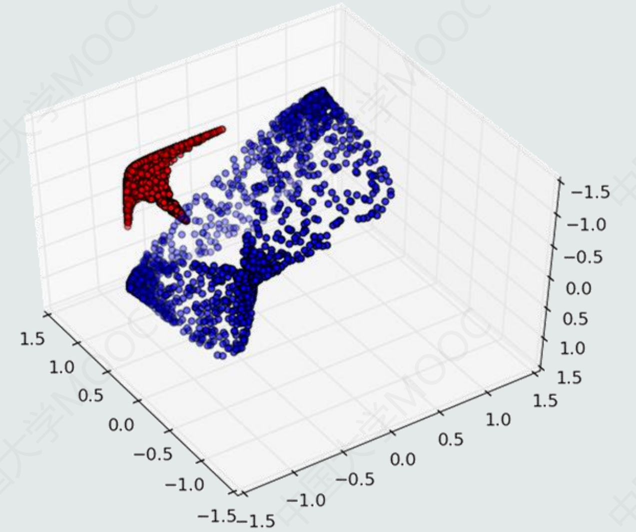
- 为了实现线性不可分到线性可分的映射，通常W矩阵是从低维到高维：



输入空间
（C_1类：红线，C_2类：蓝线）



隐藏层（2维→2维）



隐藏层（2维→3维）

总结

神经网络用途

分类 & 回归

(非线性)

总结

非线性

- 每一层有多个神经元。
- 隐藏层神经元和输出层神经元有非线性激活函数。
- 因此，多层神经网络相当于通过串行+并行的方式来组合多个非线性函数，达到输入到输出信号的非线性映射，实现非线性函数拟合以及非线性分类。

如何学习神经网络?