《深度学习预算法分析》本硕博打通课程

神经网络和反向传播
NNs and Back Propagation






目录(CONTENT)

大连理工大学 Dalian University Of Technology

大连理工大学
Dalian University Of Technology

01
PART ONE

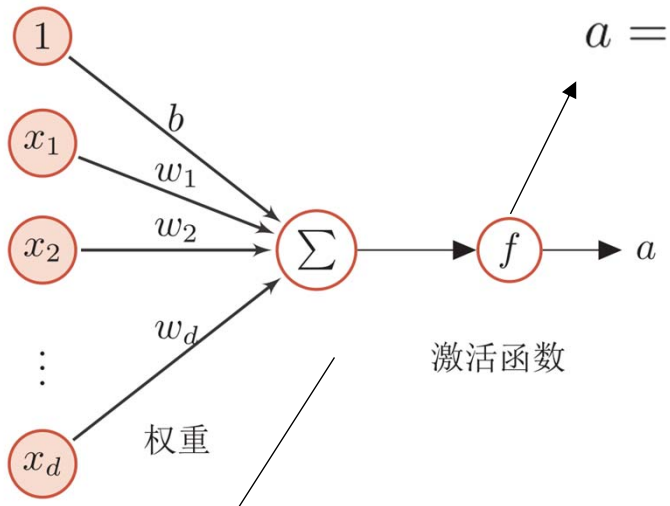
多层感知机

Multi-layer Perceptron

单层感知机-->多层感知机

大连理工大学 Dalian University Of Technology

■ 单层感知机的结构表示如下：

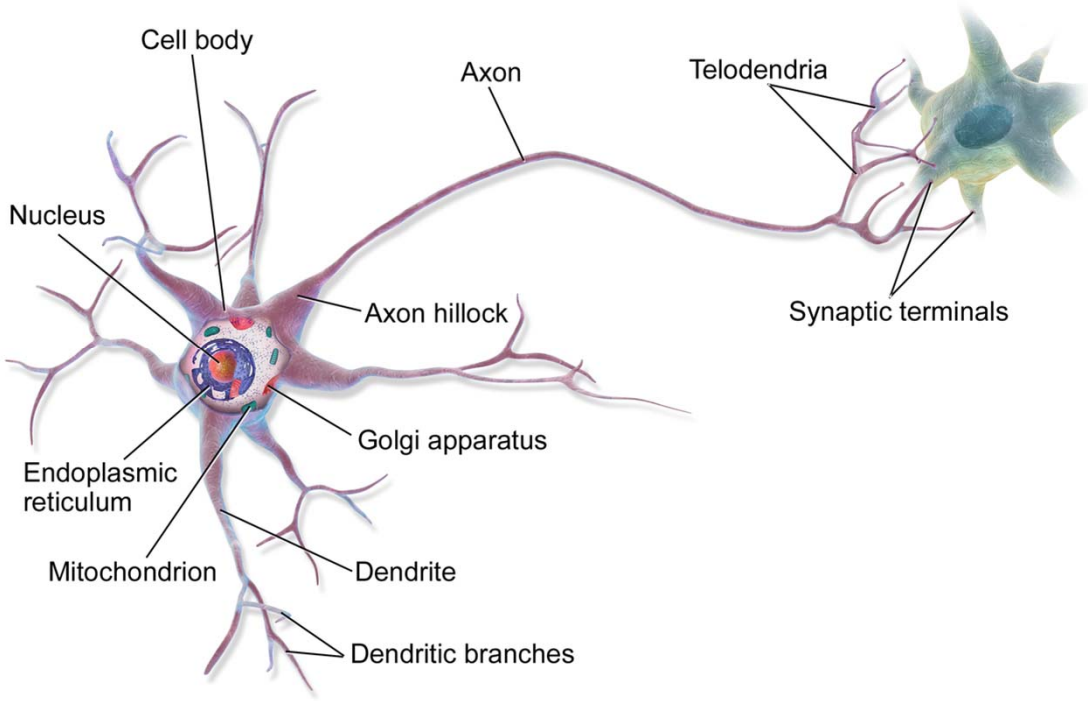


$$z = \sum_{i=1}^{d} w_i x_i + b$$

$$= \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} + b,$$

一个简单的线性模型！

单个神经细胞只有两种状态：兴奋和抑制



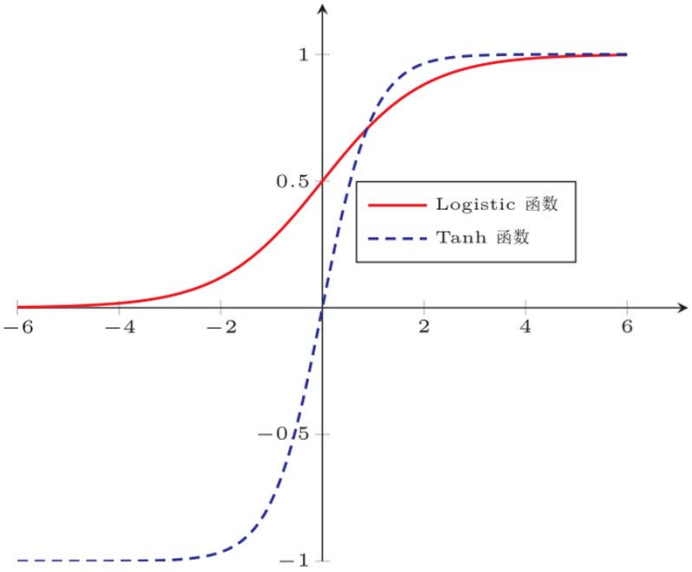
单层感知机-->多层感知机

大连理工大学 Dalian University Of Technology

■ 常见激活函数：

$$\sigma(x) = \frac{1}{1+\exp(-x)}$$

$$\tanh(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$$



性质：

非零中心化的输出会使得其后一层的神经元的输入发生偏置偏移（bias shift），并进一步使得梯度下降的收敛速度变慢。

饱和函数

Tanh函数是零中心化的，而logistic函数的输出恒大于0

单层感知机-->多层感知机

大连理工大学 Dalian University Of Technology

■ 常见激活函数：

$$\mathrm{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

$$= \max(0, x).$$

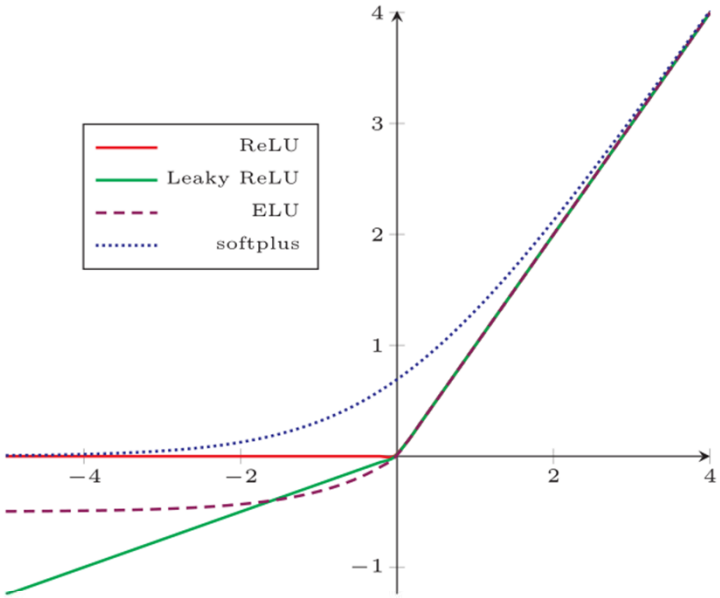
$$\mathrm{LeakyReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \gamma x & \text{if } x \leq 0 \end{cases}$$

$$\mathrm{PReLU}_i(x) = \begin{cases} x & \text{if } x > 0 \\ \gamma_i x & \text{if } x \leq 0 \end{cases}$$

$$\mathrm{ELU}(x) = \begin{cases} x & \text{if } x > 0 \\ \gamma(\exp(x) - 1) & \text{if } x \leq 0 \end{cases}$$

$$= \max(0, x) + \min(0, \gamma(\exp(x) - 1))$$

$$\mathrm{softplus}(x) = \log(1 + \exp(x))$$



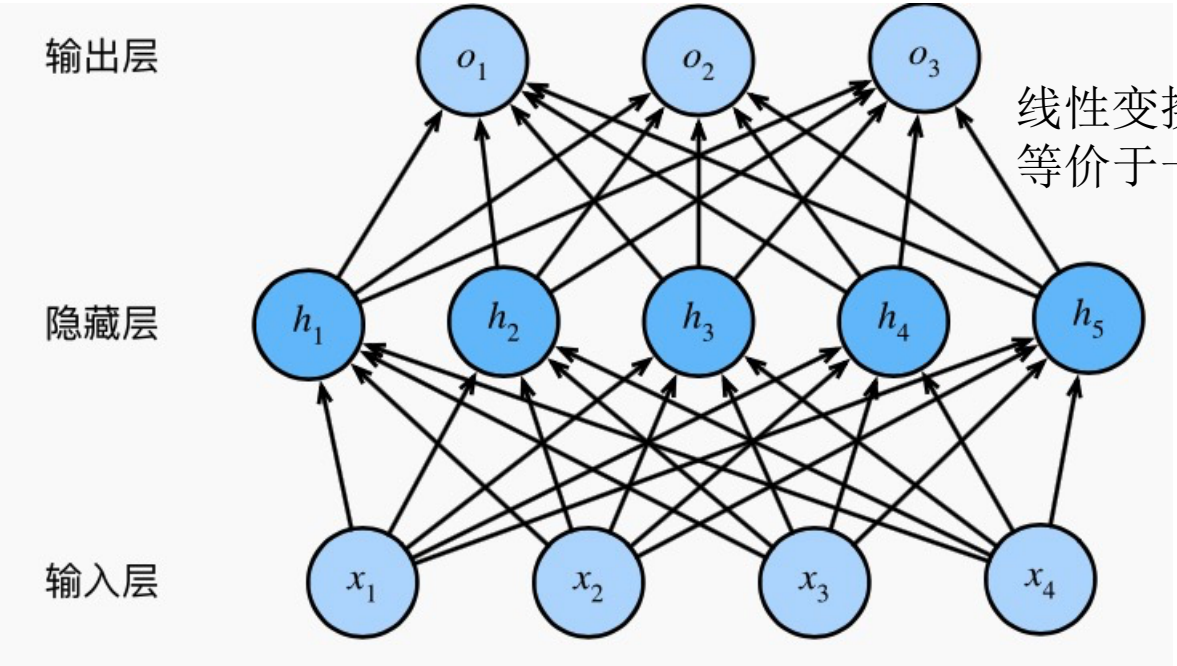
死亡ReLU问题（Dying ReLU Problem）

计算上更加高效
生物学合理性
单侧抑制、宽兴奋边界
在一定程度上缓解梯度消失问题

单层感知机-->多层感知机

大连理工大学 Dalian University Of Technology

■ 多层感知机 (Multilayer Perceptron)



线性变换下:
等价于一个单层感知机

$$\boldsymbol{O} = (\boldsymbol{X}\boldsymbol{W}_h + \boldsymbol{b}_h)\boldsymbol{W}_o + \boldsymbol{b}_o = \boldsymbol{X}\boldsymbol{W}_h\boldsymbol{W}_o + \boldsymbol{b}_h\boldsymbol{W}_o + \boldsymbol{b}_o.$$

多层感知机在单层感知机的基础上引入了一到多个隐藏层(hidden layer)。隐藏层位于输入层和输出层之间。不难发现，即便再添加更多的隐藏层, 以上设计依然只能与仅含输出层的单层感知机等价。

大连理工大学 Dalian University Of Technology

PART TWO 02

前馈神经网络

Feedforward NNs

前馈神经网络

大连理工大学 Dalian University Of Technology

■ 人工神经网络(Artificial Neural Networks)

人工神经网络主要由大量的神经元以及它们之间的有向连接构成。因此考虑三方面：

神经元的激活规则

主要是指神经元输入到输出之间的映射关系，一般为非线性函数。

网络的拓扑结构

不同神经元之间的连接关系。

学习算法

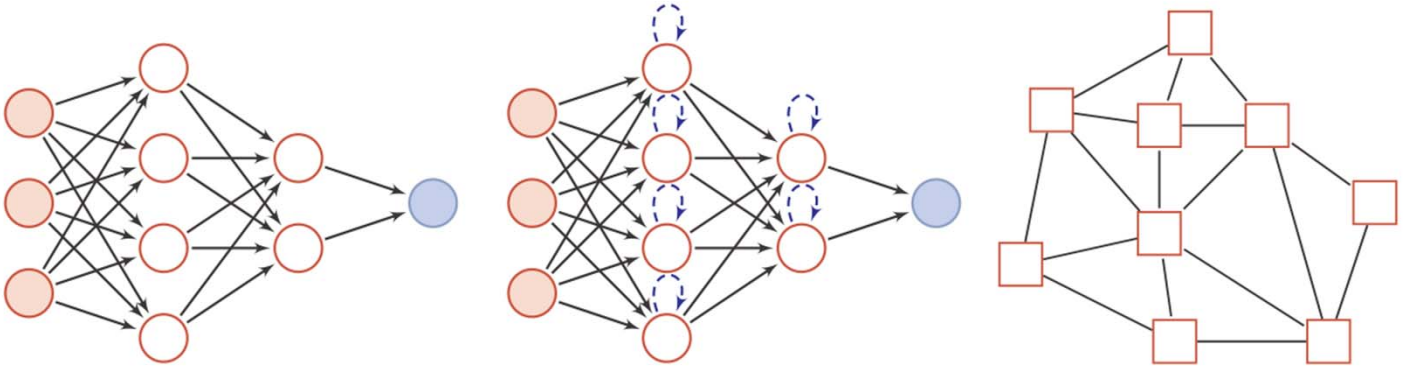
通过训练数据来学习神经网络的参数。

前馈神经网络

大连理工大学 Dalian University Of Technology

■ 人工神经网络(Artificial Neural Networks)

人工神经网络由神经元模型构成，这种由许多神经元组成的信息处理网络具有并行分布结构。



(a) 前馈网络　(b) 记忆网络　(c) 图网络

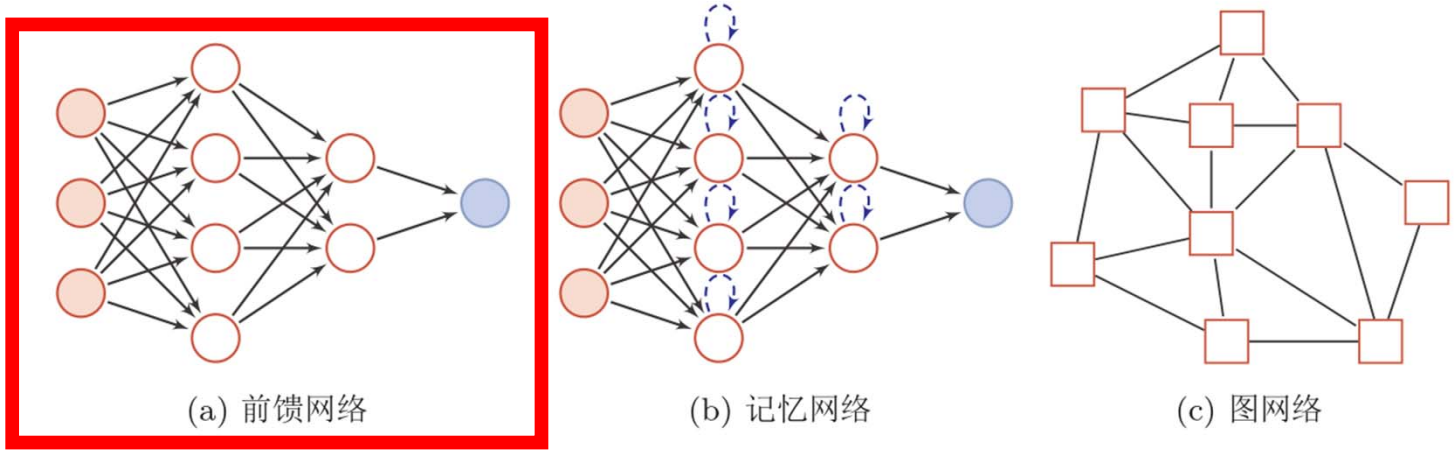
圆形节点表示一个神经元，方形节点表示一组神经元。

前馈神经网络

大连理工大学 Dalian University Of Technology

■ 前馈神经网络(Feedforward Neural Networks)

前馈神经网络是人工神经网络的一种形式，信息前向传递，实现逐层编码处理的网络结构。



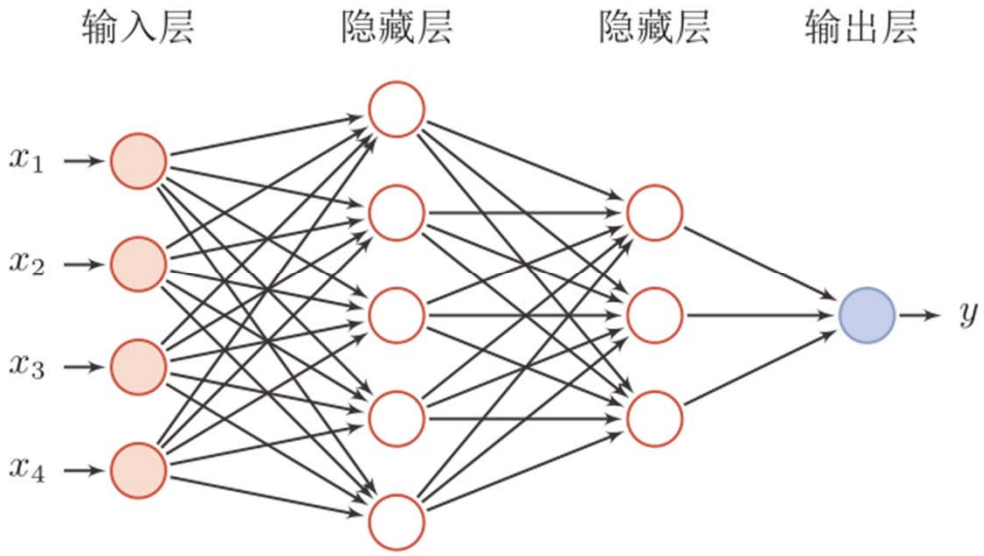
圆形节点表示一个神经元，方形节点表示一组神经元。

前馈神经网络

大连理工大学 Dalian University Of Technology

■ 前馈神经网络(全连接神经网络、多层感知器)

- ➢ 各神经元分别属于不同的层，层内无连接。
- ➢ 相邻两层之间的神经元全部两两连接。
- ➢ 整个网络中无反馈，信号从输入层向输出层单向传播，可用一个有向无环图表示。



记号	含义
L	神经网络的层数
M_l	第 l 层神经元的个数
$f_l(\cdot)$	第 l 层神经元的激活函数
$\boldsymbol{W}^{(l)} \in \mathbb{R}^{M_l \times M_{l-1}}$	第 $l-1$ 层到第 l 层的权重矩阵
$\boldsymbol{b}^{(l)} \in \mathbb{R}^{M_l}$	第 $l-1$ 层到第 l 层的偏置
$\boldsymbol{z}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的净输入（净活性值）
$\boldsymbol{a}^{(l)} \in \mathbb{R}^{M_l}$	第 l 层神经元的输出（活性值）

前馈神经网络

大连理工大学 Dalian University Of Technology

■ 前馈神经网络(Feedforward Neural Networks)

- 前馈神经网络通过下面公式进行信息传播。

$$\boldsymbol{z}^{(l)} = \boldsymbol{W}^{(l)}\boldsymbol{a}^{(l-1)} + \boldsymbol{b}^{(l)},$$
$$\boldsymbol{a}^{(l)} = f_l(\boldsymbol{z}^{(l)}).$$

- 前馈计算：

$$\boldsymbol{x} = \boldsymbol{a}^{(0)} \to \boldsymbol{z}^{(1)} \to \boldsymbol{a}^{(1)} \to \boldsymbol{z}^{(2)} \to \cdots \to \boldsymbol{a}^{(L-1)} \to \boldsymbol{z}^{(L)} \to \boldsymbol{a}^{(L)} = \phi(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{b}))$$

数据前向传递

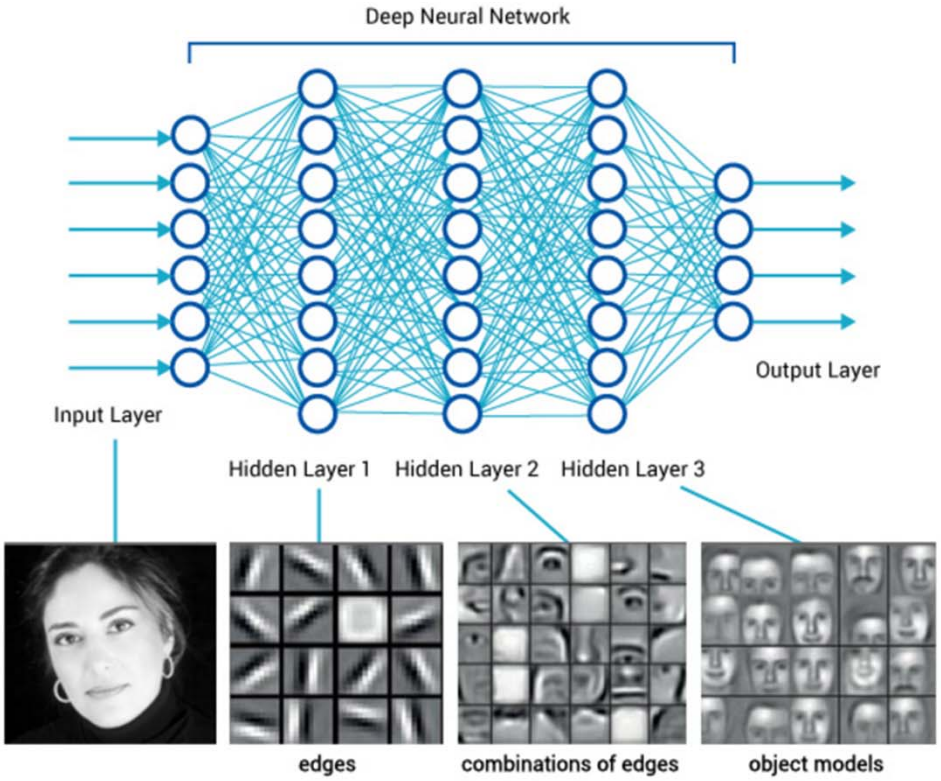
前馈神经网络

■ 前馈神经网络(Feedforward Neural Networks)

- 前馈计算：

$$\boldsymbol{x} = \boldsymbol{a}^{(0)} \rightarrow \boldsymbol{z}^{(1)} \rightarrow \boldsymbol{a}^{(1)} \rightarrow \boldsymbol{z}^{(2)} \rightarrow \cdots \rightarrow \boldsymbol{a}^{(L-1)} \rightarrow \boldsymbol{z}^{(L)} \rightarrow \boldsymbol{a}^{(L)} = \phi(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{b}))$$

深度扩展



前馈神经网络

大连理工大学 Dalian University Of Technology

■ 通用近似定理

定理 4.1 – 通用近似定理（Universal Approximation Theorem）[Cybenko, 1989, Hornik et al., 1989]： 令 $\varphi(\cdot)$ 是一个非常数、有界、单调递增的连续函数，$\mathcal{I}_d$ 是一个 d 维的单位超立方体 $[0,1]^d$，$C(\mathcal{I}_d)$ 是定义在 $\mathcal{I}_d$ 上的连续函数集合。对于任何一个函数 $f \in C(\mathcal{I}_d)$，存在一个整数 m，和一组实数 $v_i, b_i \in \mathbb{R}$ 以及实数向量 $\mathbf{w}_i \in \mathbb{R}^d$，$i = 1, \cdots, m$，以至于我们可以定义函数

$$F(\mathbf{x}) = \sum_{i=1}^{m} v_i \varphi(\mathbf{w}_i^{\mathrm{T}} \mathbf{x} + b_i), \tag{4.33}$$

作为函数 f 的近似实现，即

$$|F(\mathbf{x}) - f(\mathbf{x})| < \epsilon, \forall \mathbf{x} \in \mathcal{I}_d. \tag{4.34}$$

其中 $\epsilon > 0$ 是一个很小的正数。

根据通用近似定理，对于具有线性输出层和至少一个使用“挤压”性质的激活函数的隐藏层组成的前馈神经网络，只要其隐藏层神经元的数量足够，它可以以任意的精度来近似任何从一个定义在实数空间中的有界闭集函数。

前馈神经网络

大连理工大学 Dalian University Of Technology

■ 通用近似定理

- 神经网络可以作为一个“万能”函数来使用，可以用来进行复杂的特征转换，或逼近一个复杂的条件分布。

$$\hat{y} = g(\varphi(\mathbf{x}), \theta)$$

分类器　　神经网络

- 如果$g(\cdot)$为Logistic回归，那么Logistic回归分类器可以看成神经网络的最后一层。

前馈神经网络

■ 分类问题

如果使用Softmax回归分类器，相当于网络最后一层设置C个神经元，其输出经过Softmax函数进行归一化后可以作为每个类的条件概率。

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{z}^{(L)})$$

采用交叉熵损失函数，对于样本(x,y)，其损失函数为

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\mathbf{y}^{\mathrm{T}} \log \hat{\mathbf{y}}$$

大连理工大学 Dalian University Of Technology

03

PART THREE

梯度反向传播

Gradient Back Propagation

梯度反向传播

大连理工大学 Dalian University Of Technology

■ 参数学习 (Parameter Learning)

- 给定训练集为$D = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$，将每个样本$\boldsymbol{x}^{(n)}$输入给前馈神经网络，得到网络输出为$\hat{y}^{(n)}$，其在数据集D上的结构化风险函数为：

$$\mathcal{R}(W, \mathbf{b}) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(\mathbf{y}^{(n)}, \hat{\mathbf{y}}^{(n)}) + \frac{1}{2}\lambda\|W\|_F^2$$

- 梯度下降

$$W^{(l)} \leftarrow W^{(l)} - \alpha\frac{\partial\mathcal{R}(W, \mathbf{b})}{\partial W^{(l)}}$$

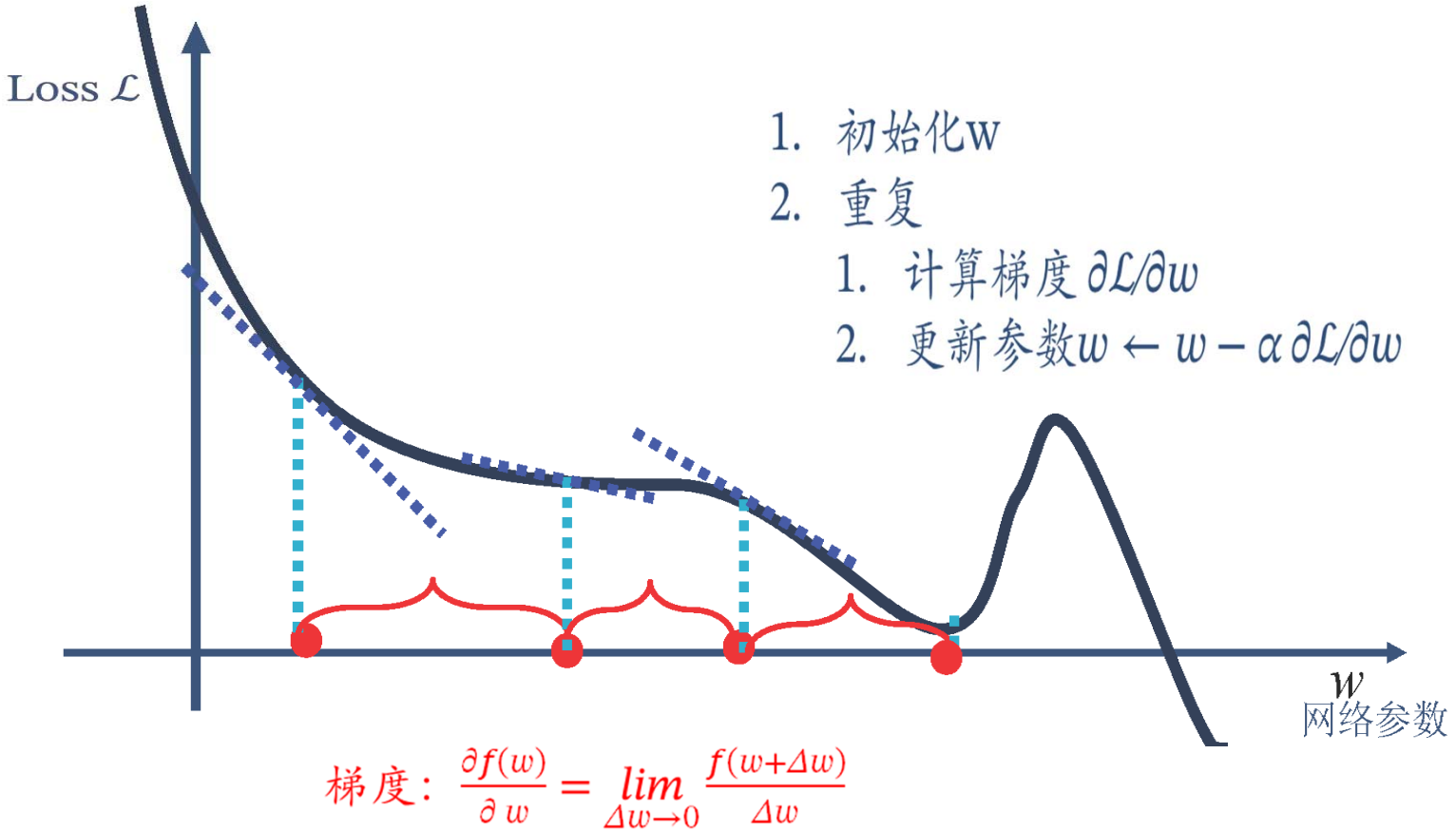
$$\mathbf{b}^{(l)} \leftarrow \mathbf{b}^{(l)} - \alpha\frac{\partial\mathcal{R}(W, \mathbf{b})}{\partial \mathbf{b}^{(l)}}$$

梯度反向传播

大连理工大学 Dalian University Of Technology

■ 梯度下降 (Gradient Descent)

- 梯度下降类似盲人下山



梯度反向传播

大连理工大学 Dalian University Of Technology

■ 梯度下降 (Gradient Descent)

- 神经网络是一个复杂的复合函数(链式法则)

$$y = f^5(f^4(f^3(f^2(f^1(x))))) \rightarrow \frac{\partial y}{\partial x} = \frac{\partial f^1}{\partial x}\frac{\partial f^2}{\partial f^1}\frac{\partial f^3}{\partial f^2}\frac{\partial f^4}{\partial f^3}\frac{\partial f^5}{\partial f^4}$$

- 梯度反向传播(Back Propagation)算法

  根据前馈网络的特点而设计的高效方法

- 更加通用的计算方法

  自动微分(Automatic Differentiation)

梯度反向传播

大连理工大学 Dalian University Of Technology

■ 1.链式法则 (Chain Rule)

- 链式法则(Chain Rule)是在微积分中求复合函数导数的一种常用方法。

(1)若 $x \in \mathbb{R}, \boldsymbol{u} = u(x) \in \mathbb{R}^s, \boldsymbol{g} = g(\boldsymbol{u}) \in \mathbb{R}^t$, 则

$$\frac{\partial \boldsymbol{g}}{\partial x} = \frac{\partial \boldsymbol{u}}{\partial x} \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}} \quad \in \mathbb{R}^{1 \times t}.$$

(2)若 $\boldsymbol{x} \in \mathbb{R}^p, \boldsymbol{y} = g(\boldsymbol{x}) \in \mathbb{R}^s, \boldsymbol{z} = f(\boldsymbol{y}) \in \mathbb{R}^t$, 则

$$\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{y}} \quad \in \mathbb{R}^{p \times t}.$$

(3)若 $X \in \mathbb{R}^{p \times q}$ 为矩阵, $\boldsymbol{y} = g(X) \in \mathbb{R}^s, z = f(\boldsymbol{y}) \in \mathbb{R}$, 则

$$\frac{\partial z}{\partial X_{ij}} = \frac{\partial \boldsymbol{y}}{\partial X_{ij}} \frac{\partial z}{\partial \boldsymbol{y}} \quad \in \mathbb{R}.$$

梯度反向传播

大连理工大学 Dalian University Of Technology

■ 2.反向传播(Back Propagation)

$$\mathbf{z}^{(l)} = W^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}$$

$$\frac{\partial \mathbf{z}^{(l)}}{\partial w_{ij}^{(l)}} = \left[\frac{\partial z_1^{(l)}}{\partial w_{ij}^{(l)}}, \cdots, \frac{\partial z_i^{(l)}}{\partial w_{ij}^{(l)}}, \cdots, \frac{\partial z_{m^{(l)}}^{(l)}}{\partial w_{ij}^{(l)}}\right]$$

$$= \left[0, \cdots, \frac{\partial(\mathbf{w}_{i:}^{(l)}\mathbf{a}^{(l-1)} + b_i^{(l)})}{\partial w_{ij}^{(l)}}, \cdots, 0\right]$$

$$= \left[0, \cdots, a_j^{(l-1)}, \cdots, 0\right]$$

$$\triangleq \mathbb{I}_i(a_j^{(l-1)}) \quad \in \mathbb{R}^{m^{(l)}},$$

权重梯度

$$\frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial w_{ij}^{(l)}} = \frac{\partial \mathbf{z}^{(l)}}{\partial w_{ij}^{(l)}} \frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{z}^{(l)}}$$

$$\frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{b}^{(l)}} = \frac{\partial \mathbf{z}^{(l)}}{\partial \mathbf{b}^{(l)}} \frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{z}^{(l)}}$$

$$\delta^{(l)} = \frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{z}^{(l)}} \quad \in \mathbb{R}^{m^{(l)}}$$

误差项

$$\frac{\partial \mathbf{z}^{(l)}}{\partial \mathbf{b}^{(l)}} = \mathbf{I}_{m^{(l)}} \quad \in \mathbb{R}^{m^{(l)} \times m^{(l)}}$$

偏置梯度

梯度反向传播

大连理工大学 Dalian University Of Technology

■ 2.反向传播(Back Propagation)

- 神经网络结构

$$\mathbf{a}^{(l)} = f_l(\mathbf{z}^{(l)})$$

$$\mathbf{z}^{(l+1)} = W^{(l+1)}\mathbf{a}^{(l)} + \mathbf{b}^{(l+1)}$$

- 逐层反向传播

$$\frac{\partial \mathbf{z}^{(l+1)}}{\partial \mathbf{a}^{(l)}} = (W^{(l+1)})^{\mathrm{T}}$$

$$\delta^{(l)} \triangleq \frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{z}^{(l)}}$$

$$= \frac{\partial \mathbf{a}^{(l)}}{\partial \mathbf{z}^{(l)}} \cdot \frac{\partial \mathbf{z}^{(l+1)}}{\partial \mathbf{a}^{(l)}} \cdot \frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{z}^{(l+1)}}$$

$$= \mathrm{diag}(f_l'(\mathbf{z}^{(l)})) \cdot (W^{(l+1)})^{\mathrm{T}} \cdot \delta^{(l+1)}$$

$$= f_l'(\mathbf{z}^{(l)}) \odot \left((W^{(l+1)})^{\mathrm{T}} \delta^{(l+1)}\right),$$

$$\frac{\partial \mathbf{a}^{(l)}}{\partial \mathbf{z}^{(l)}} = \frac{\partial f_l(\mathbf{z}^{(l)})}{\partial \mathbf{z}^{(l)}}$$

$$= \mathrm{diag}(f_l'(\mathbf{z}^{(l)}))$$

梯度反向传播

大连理工大学 Dalian University Of Technology

■ 2.反向传播(Back Propagation)

在计算出上面三个偏导数之后，公式(4.49)可以写为

$$\frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial w_{ij}^{(l)}} = \mathbb{I}_i(a_j^{(l-1)})\delta^{(l)} = \delta_i^{(l)} a_j^{(l-1)}.$$

进一步，$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$关于第$l$层权重$W^{(l)}$的梯度为

$$\frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial W^{(l)}} = \delta^{(l)}(\mathbf{a}^{(l-1)})^{\mathrm{T}}.$$

同理，$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$关于第$l$层偏置$\mathbf{b}^{(l)}$的梯度为

$$\frac{\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{b}^{(l)}} = \delta^{(l)}.$$

梯度反向传播

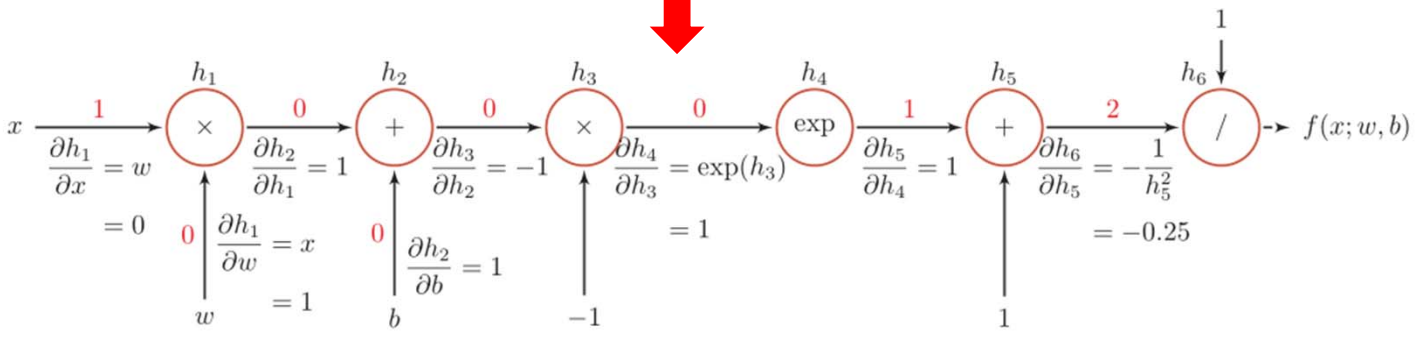
■ 3.自动微分(Automatic Differentiation)

- 自动微分是利用链式法则来自动计算一个复合函数的梯度。

- 例如复合函数

$$f(x;w,b)=\frac{1}{\exp(-(wx+b))+1}.$$

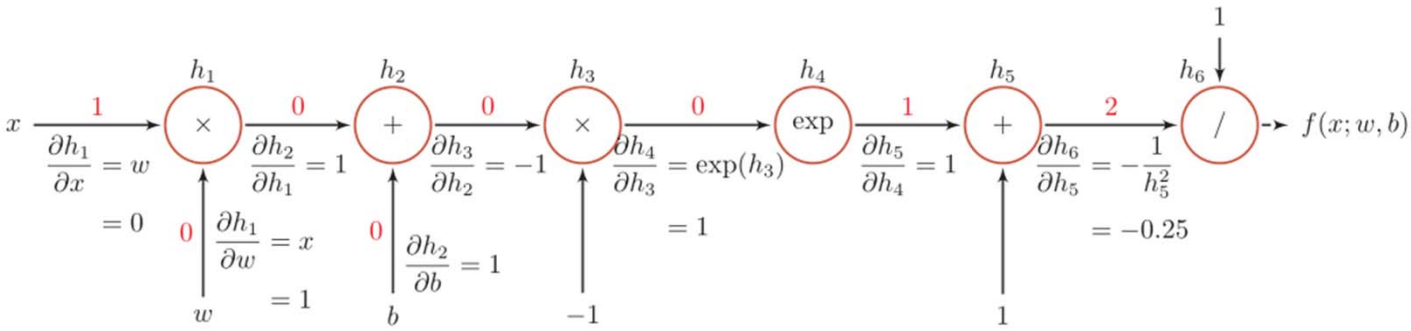
- 计算图



梯度反向传播

■ 3.自动微分(Automatic Differentiation)

➢ 形式化计算



函数	导数	
$h_1 = x \times w$	$\frac{\partial h_1}{\partial w} = x$	$\frac{\partial h_1}{\partial x} = w$
$h_2 = h_1 + b$	$\frac{\partial h_2}{\partial h_1} = 1$	$\frac{\partial h_2}{\partial b} = 1$
$h_3 = h_2 \times -1$	$\frac{\partial h_3}{\partial h_2} = -1$	
$h_4 = \exp(h_3)$	$\frac{\partial h_4}{\partial h_3} = \exp(h_3)$	
$h_5 = h_4 + 1$	$\frac{\partial h_5}{\partial h_4} = 1$	
$h_6 = 1/h_5$	$\frac{\partial h_6}{\partial h_5} = -\frac{1}{h_5^2}$	

当x = 1,w = 0,b = 0时，可以得到

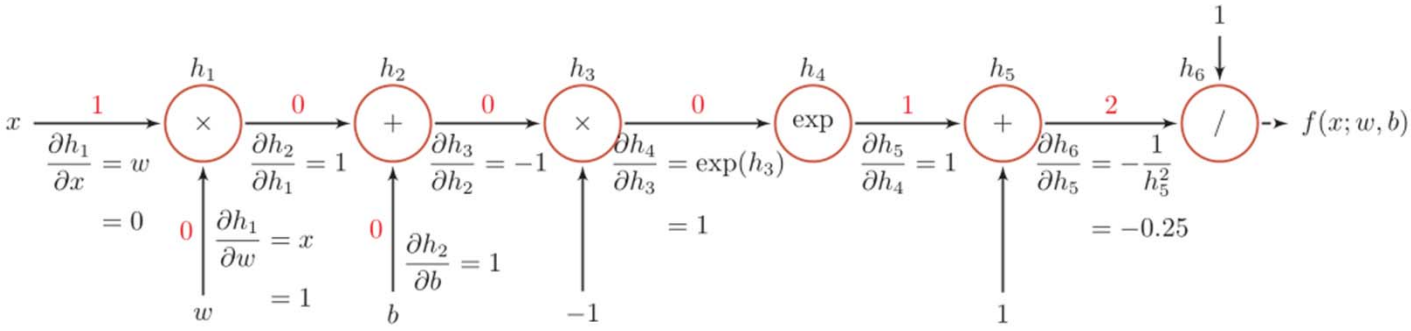
$$\frac{\partial f(x;w,b)}{\partial w}|_{x=1,w=0,b=0} = \frac{\partial f(x;w,b)}{\partial h_6}\frac{\partial h_6}{\partial h_5}\frac{\partial h_5}{\partial h_4}\frac{\partial h_4}{\partial h_3}\frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial w}$$
$$= 1 \times -0.25 \times 1 \times 1 \times -1 \times 1 \times 1$$
$$= 0.25.$$

梯度反向传播

大连理工大学 Dalian University Of Technology

■ 3.自动微分(Automatic Differentiation)

- 前向模式



- 反向模式(与反向传播的计算梯度的方式相同)

  前向计算每一层的状态和激活值，直到最后一层

  反向计算每一层的参数的偏导数

  更新参数

- 如果函数和参数之间有多条路径，可以将这多条路径上的导数再进行相加，得到最终的梯度。

梯度反向传播

大连理工大学 Dalian University Of Technology

■ 3.自动微分(Automatic Differentiation)

- 静态计算图

  在编译时构建计算图，计算图构建好之后在程序运行时不能改变。代表性工具箱Theano和Tensorflow

- 动态计算图

  在程序运行时动态构建，可自动根据函数的结构调整计算图。代表性工具箱DyNet，Chainer和PyTorch

- 两种构建方式各有优缺点。静态计算图在构建时可以进行优化，并行能力强，但灵活性比较差低。动态计算图则不容易优化，当不同输入的网络结构不一致时，难以并行计算，但是灵活性比较高。

大连理工大学
Dalian University Of Technology

04

PART Four

闭形式解

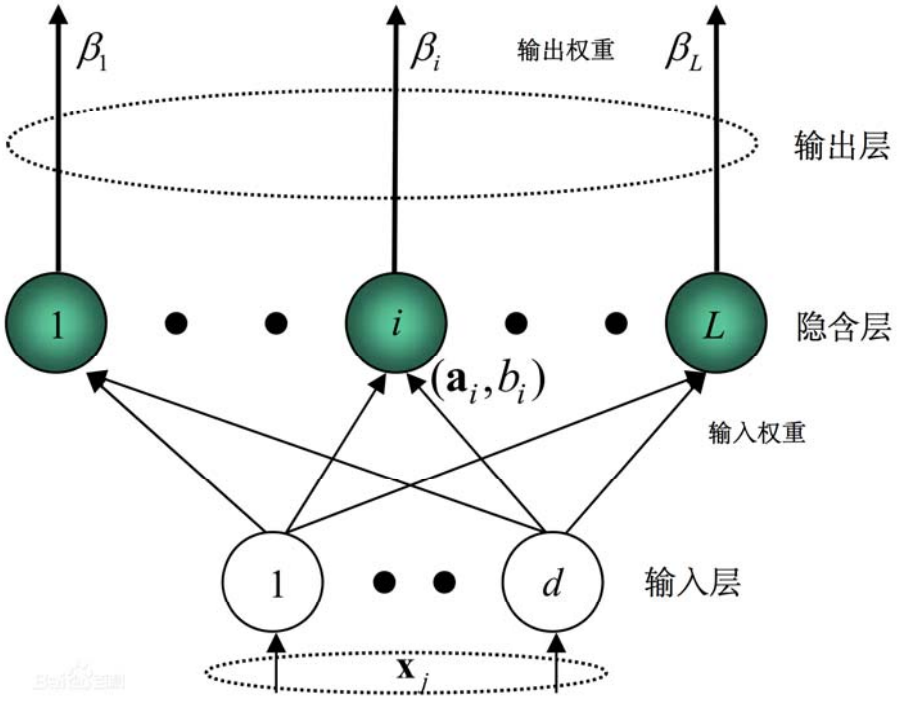
Close-form Solution

闭形式解-->极端学习机

大连理工大学 Dalian University Of Technology

■ 1.极端学习机(Extreme Learning Machine)

- 快速学习算法，对于单隐层神经网络(三层)，ELM可以随机初始化输入权重和偏置，并利用最小二乘得到相应的输出权重。



闭形式解

随机初始化

闭形式解-->极端学习机

大连理工大学 Dalian University Of Technology

■ 1.极端学习机(Extreme Learning Machine)

- 在一个 ELM 的单隐层结构中，第 i 个隐藏节点的输出为：

$$h_i(\mathbf{x}) = G(\mathbf{a}_i, b_i, \mathbf{x})$$

- 整个隐藏层的输出映射为：

$$\mathbf{h}(\mathbf{x}) = [G(h_i(\mathbf{x}), \ldots, h_L(\mathbf{x}))]$$

- 给定 N 个训练样本，ELM 的隐藏层输出矩阵 H 给出为：

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{x}_1) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{x}_N) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix}$$

- 具有 L 个隐藏节点的 ELM 结构的输出为：

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i h_i(\mathbf{x})$$

闭形式解-->极端学习机

大连理工大学 Dalian University Of Technology

■ 1.极端学习机(Extreme Learning Machine)

- 整个隐藏层的输出为： $H\beta$
- 目标矩阵 T (标签矩阵)为： $\mathbf{T} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_N \end{bmatrix}$
- 目标函数为： $\text{Minimize: } \|\beta\|_p^{\sigma_1} + C\|\mathbf{H}\beta - \mathbf{T}\|_q^{\sigma_2}$
- 求解矩阵H的Moore-Penrose广义逆

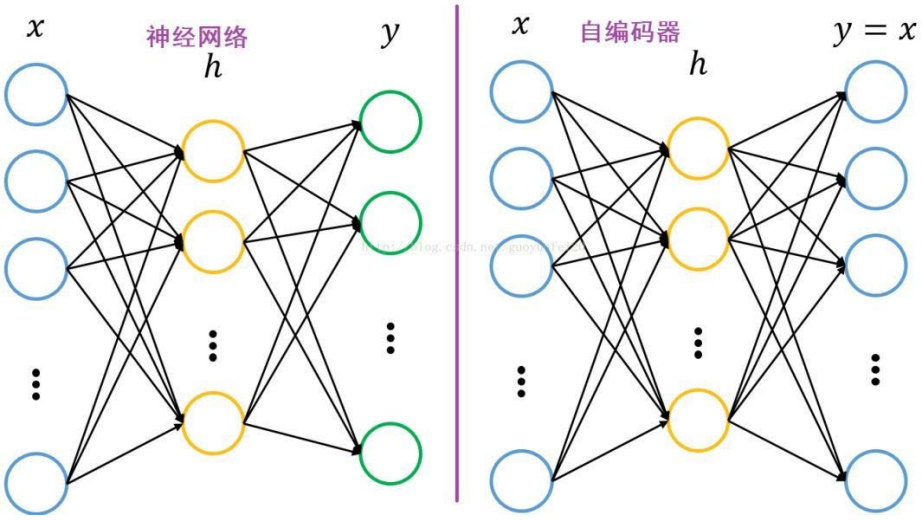
$$\hat{\beta} = H^{+}T$$ 闭形式解!!!

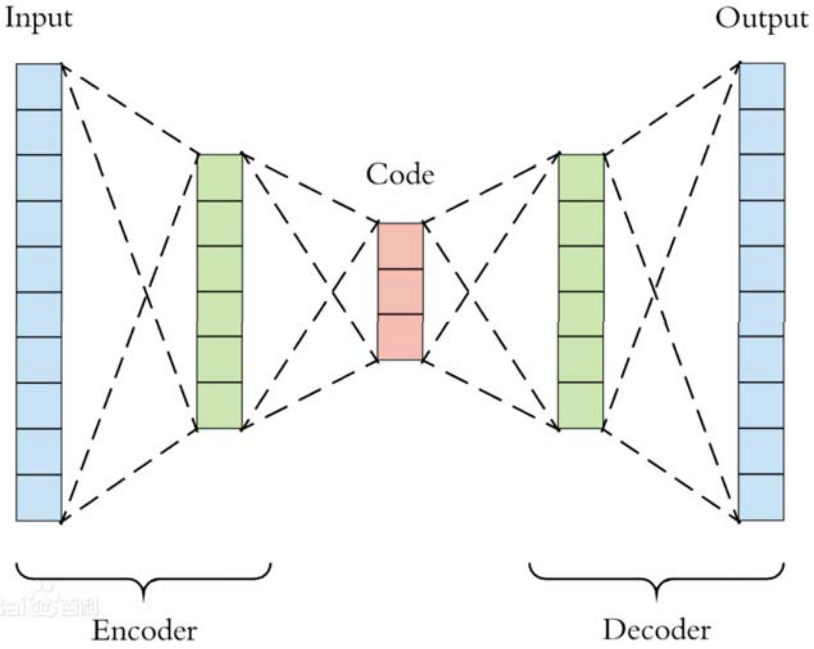
闭形式解-->自编码器

大连理工大学 Dalian University Of Technology

■ 2.自编码器(Auto-Encoder)

- 自编码器是一类在半监督学习和无监督学习中使用的人工神经网络，其功能是通过将输入信息作为学习目标，对输入信息进行表征学习
- 自编码器包含编码器(Encoder)和解码器(Decoder)两部分





闭形式解-->自编码器

大连理工大学 Dalian University Of Technology

■ 2.自编码器(Auto-Encoder)

- 按学习范式，自编码器可以被分为收缩自编码器、正则自编码器和变分自编码器，其中前两者是判别模型、后者是生成模型
- 按构建类型，自编码器可以是前馈结构或递归结构的神经网络
- 自编码器具有一般意义上表征学习算法的功能，被应用于降维和异常值检测
- 包含卷积层结构的自编码器可被应用于计算机视觉问题，包括图像降噪、神经风格迁移等

闭形式解-->自编码器

大连理工大学 Dalian University Of Technology

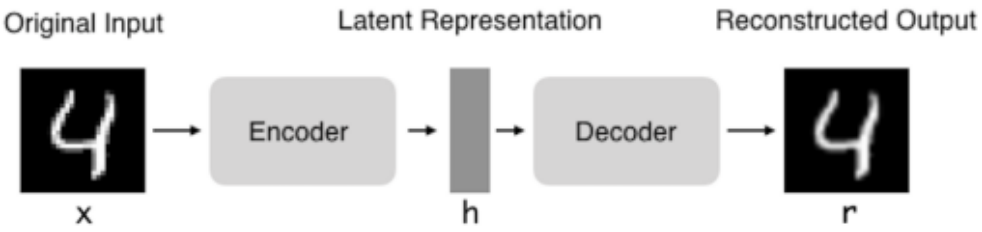
■ 2.自编码器(Auto-Encoder)

- 自编码器是一个输入和输出相同的神经网络,给定输入空间和特征空间，自编码器求解两者的映射使输入特征的重建误差达成最小:

$$y = h(x)$$
$$r = f(y) = f(h(x))$$

重建误差最小

- 求解完成后，由编码器输出的隐含层特征，即“编码特征”可视为输入数据的表征



Architecture of an Autoencoder

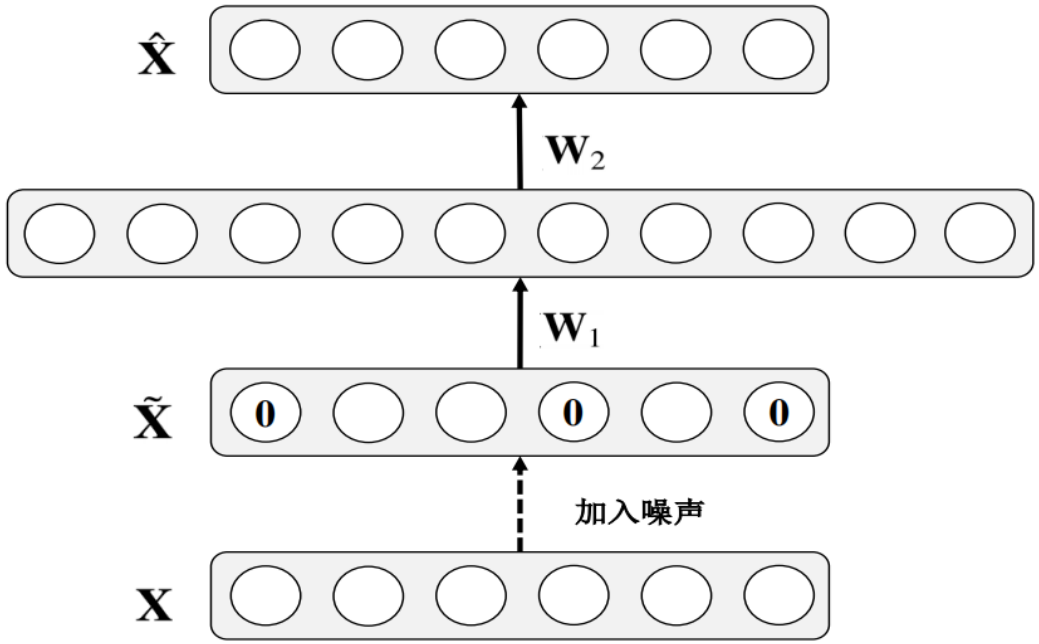
- 当自编码器的编码、解码能力很强时，可以确保对训练样本的完美重建，但却难以发现数据的内在分布和结构规律

闭形式解-->自编码器

大连理工大学 Dalian University Of Technology

■ 2.自编码器(Auto-Encoder)

- 去噪自编码器(Denoising Auto-Encoders, DAE)



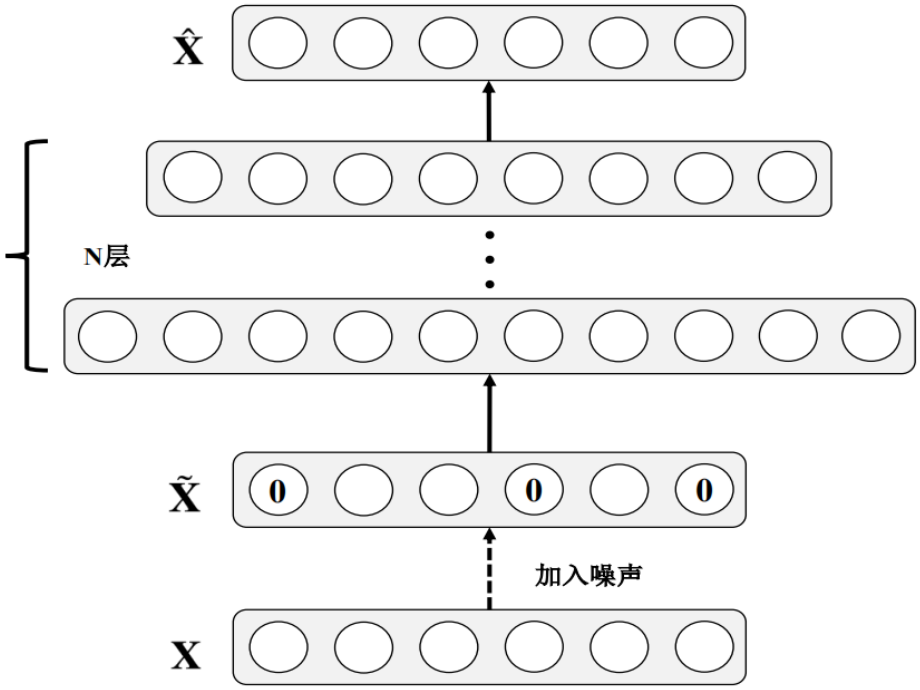
- 通常还可约束编码部分和解码部分的权重保持某种关系，比如捆绑约束(Tied Constraint)、稀疏约束(Sparse Constraint)等。

闭形式解-->自编码器

大连理工大学 Dalian University Of Technology

■ 2.自编码器(Auto-Encoder)

- 堆叠去噪自编码器(Stacked Denoising Auto-Encoders, SDAE)



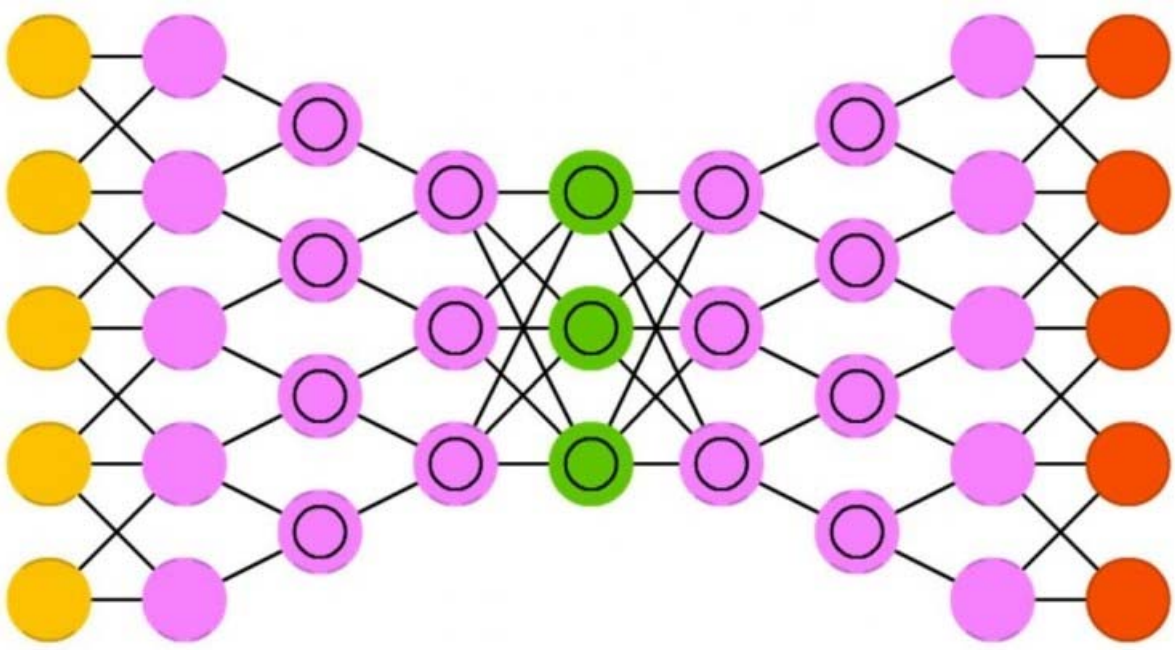
- 引入了更多的隐藏层 ，反复堆叠，从而获得了更多输入数据的深层次和抽象化特 征。

闭形式解-->自编码器

大连理工大学 Dalian University Of Technology

■ 2.自编码器(Auto-Encoder)

- 欠完备自编码器(Unncomplete Auto-Encoders, UAE)



- 编码维度小于输入维度的自编码器。学习欠完备的表示将强制自编码器捕捉训练数据中最显著的特征。

大连理工大学
Dalian University Of Technology

05
PART FIVE

小结

Summary

小结

大连理工大学 Dalian University Of Technology

■小结

- ➢多层感知机
  - ✓单层感知机 (连接方式, 激活函数)
  - ✓多层感知机 (等价形态, 多层扩展)
- ➢前馈神经网络
  - ✓ 人工神经网络
  - ✓ 前馈神经网络(连接模式,数据传递)
  - ✓ 通用近似定理
  - ✓ 分类问题
- ➢梯度反向传播
  - ✓参数学习
  - ✓梯度下降(链式法则,反向传播,自动微分)
- ➢闭形式解
  - ✓极端学习机
  - ✓自编码器(降噪自编码器, 堆叠降噪自编码器, 欠完备降噪自编码器)