No.

Date.

chap-4 前馈神经网络

4.1 神经元

净输入 z 一个神经元获得输入信号 $\vec{x}$ 的加权和

$$z = \sum w_i x_i + b$$

$$= \vec{w}^T \vec{x} + b$$

活性值

$$a = f(z)$$

⇒ 4.1.1 Sigmoid 型激活函数

- Logistic
- tanh

$$\sigma'(x) = \sigma(x)(1-\sigma(x))$$

$$\sigma'(\vec{x}) = \text{diag}(\sigma(\vec{x})) \odot (1-\sigma(\vec{x}))$$

Logistic 函数

$$\sigma(x) = \frac{1}{1+\exp(-x)} \quad \in [0,1]$$

Tanh 函数

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \quad \in (-1,1)$$

$$\tanh(x) = 2\sigma(2x) - 1$$

No.

Date.

The Tanh 的输出是零中心化的. logistic 不是, 总是大于0.

非零中心化 使得其后一层神经网络会产生偏移(bias shift), 并进一步使SGD收敛速度变慢.

偏置

⇒⇒ 4.1.1.1 Hard-logistic 和 Hard-Tanh 函数

用分段函数近似

$$\text{hard-logistic}(x) = \max(\min(0.25x + 0.5, 1), 0)$$

$$\text{hard-tanh}(x) = \max(\min(x, 1), -1)$$

⇒ 4.1.1 修正线性单元

$$\text{ReLU}(x) = \max(0, x)$$

优点 1. 计算高效

2. 生物角度: 单侧抑制, 宽兴奋边界

3. 稀疏性: 50%为0或很小.

4. 缓解梯度消失

缺点 1. 非零中心化, 偏置偏移

2. 死亡 ReLU 问题

No.

Date.

⇒⇒ 带泄露的 ReLU

$$\text{Leaky ReLU}(x) = \begin{cases} x & \text{if } (x>0) \\ rx & \text{if } (x<0) \end{cases}$$

$$= \max(0, x) + r\min(0, x)$$

当 $r<1$ 时，r 一般等于 0.01。

$$\text{Leaky ReLU}(x) = \max(x, rx)$$

相当于一个简单的 maxout 单元

⇒⇒ ~~P.T.~~ 2 带参数的 ReLU

引入一个可学习的参数，不同神经元可以有不同的参数。

$$\text{PReLU}_i(x) = \begin{cases} x & \text{if } x>0 \\ r_i x & \text{if } x<0 \end{cases}$$

$$= \max(0, x) + r_i \min(0, x)$$

⇒⇒ ELU 指数线性单元

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

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

No.

Date.

⇒⇒ softplus 函数

rectifier 的平滑版本，具有采样统计意义

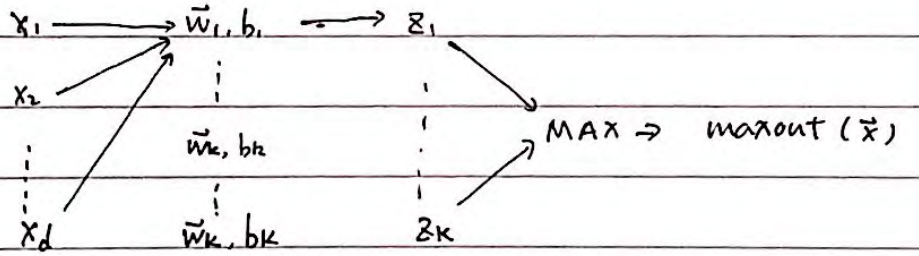
$$\text{softplus}(x) = \log(1+\exp(x))$$

6.1.3 Maxout 单元

maxout 的输入是上层神经元的全部输出 $\vec{x} = [x_1; \cdots; x_d]$

k 个权重向量 $\vec{w}_k \in \mathbb{R}^d$ 和偏置 b_k，k 个净输出 z_k

$$z_k = \vec{w}_k^T \vec{x} + b_k$$



$$\text{maxout}(\vec{x}) = \max(z_k)$$

整体学习输入到输出间的非线性映射关系.

No.

Date. . .

6.2 网络结构

⇒ 6.2.1 前馈网络

全连接前馈网络

卷积神经网络.

⇒ 6.2.2 反馈网络.

循环神经网络

Hopfield网络. 玻尔兹曼机

记忆增强网络.

⇒ 6.2.3 图网络.

No.

Date. . .

P.) 前馈神经网络.

L: 神经网络的层数

$m^{(l)}$: 第 l 层神经网络的个数

$f_{(l)}$: 激活函数

$W^{(l)}$: $l-1$ 到 l 层的权重矩阵

$\vec{b}^{(l)}$: ------

$\vec{z}^{(l)}$: l 层神经元的净输入

$\vec{a}^{(l)}$: (---- 净输出

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

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

$a^{(l-1)} \longrightarrow z^{(l)} \longrightarrow a^{(l)}$

⇒ P.3.1 通用近似定理.

只需一个包含足够多神经元的隐藏层，多层前馈网络能以任意精度逼近任意复杂度的函数。

⇒ P.3.2 应用到机器学习

特征抽取 $x \to \phi(x)$

No.

Date. . .

⇒ 4.3.3 参数学习

$$L(\vec{y}, \hat{\vec{y}}) = -\vec{y}^T \log \hat{\vec{y}}$$

EMF $$R(W, \vec{b}) = \frac{1}{N} \sum L(\vec{y}^{(m)}, \hat{\vec{y}}^{(m)}) + \frac{1}{2} \lambda \|W\|_F^2$$

Frobenius 范数

$$\|W\|_F^2 = \sum_{l} \sum_{i}^{m^{(l)}} \sum_{j}^{n^{(l)}} (w_{ij}^{(l)})^2$$

第 l 层参数更新方式

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

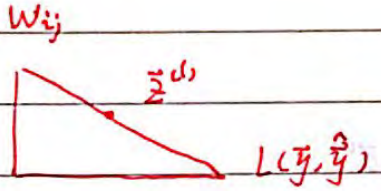
$$\vec{b}^{(l)} \leftarrow \vec{b}^{(l)} - \alpha \cdot \frac{\partial R(W, \vec{b})}{\partial \vec{b}^{(l)}}$$

No.

Date.

4.4 反向传播算法

因为 $\frac{\partial L(\vec{y},\hat{\vec{y}})}{\partial W^{(l)}}$ 计算涉及矩阵微分，较为繁琐，故我们先计算 $\frac{\partial L(\vec{y},\hat{\vec{y}})}{\partial W_{ij}^{(l)}}$.



$X \to \vec{y} \to s$

$$\frac{\partial s}{\partial x_{ij}} = \left(\frac{\partial \vec{y}}{\partial x_{ij}}\right)^T \frac{\partial s}{\partial \vec{y}}$$

$$\frac{\partial L(\vec{y},\hat{\vec{y}})}{\partial W_{ij}^{(l)}} = \left(\frac{\partial \vec{z}^{(l)}}{\partial W_{ij}^{(l)}}\right)^T \cdot \frac{\partial L(\vec{y},\hat{\vec{y}})}{\partial \vec{z}^{(l)}} \qquad (4.46)$$

$$\frac{\partial L(\vec{y},\hat{\vec{y}})}{\partial \vec{b}^{(l)}} = \left(\frac{\partial \vec{z}^{(l)}}{\partial \vec{b}^{(l)}}\right)^T \cdot \frac{\partial L(\vec{y},\hat{\vec{y}})}{\partial \vec{z}^{(l)}} \qquad (4.47)$$

$\frac{\partial L(\vec{y},\hat{\vec{y}})}{\partial \vec{z}^{(l)}}$ 目标函数关于第 l 层神经元 $z^{(l)}$ 的偏导数

称为 误差项

$\frac{\partial \vec{z}^{(l)}}{\partial \vec{b}^{(l)}}$

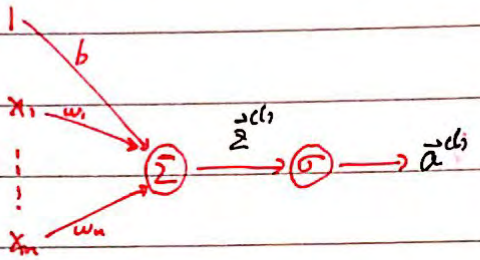
$\frac{\partial L(\vec{y},\hat{\vec{y}})}{\partial \vec{z}^{(l)}} \cdot \frac{\partial \vec{z}^{(l)}}{\partial W_{ij}^{(l)}}$

No.

Date.

⇒⇒⇒ 计算偏导数 $\frac{\partial \vec{z}^{(l)}}{\partial w_{ij}^{(l)}}$

因 $\vec{z}^{(l)}$ 与 $w_{ij}^{(l)}$ 的函数关系为 $\vec{z}^{(l)} = w^{(l)}\vec{a}^{(l-1)} + \vec{b}^{(l)}$



$$\frac{\partial \vec{z}^{(l)}}{\partial w_{ij}^{(l)}} = \frac{\partial (w^{(l)}\vec{a}^{(l)} + \vec{b}^{(l)})}{\partial w_{ij}^{(l)}}$$

$$= \begin{bmatrix} \frac{\partial (w_{1:}^{(l)}\vec{a}^{(l-1)} + \vec{b}^{(l)})}{\partial w_{ij}^{(l)}} \\ \vdots \\ \frac{\partial (w_{i:}^{(l)}\vec{a}^{(l-1)} + \vec{b}^{(l)})}{\partial w_{ij}^{(l)}} \\ \vdots \\ \frac{\partial (w_{m:}^{(l)}\vec{a}^{(l-1)} + \vec{b}^{(l)})}{\partial w_{ij}^{(l)}} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ a_j^{(l-1)} \\ \vdots \\ 0 \end{bmatrix} = \mathbb{I}_i(a_j^{(l-1)})$$

⇒⇒⇒ 计算偏导数 $\frac{\partial \vec{z}^{(l)}}{\partial \vec{b}^{(l)}}$

因 $\vec{z}^{(l)} = w^{(l)}\vec{a}^{(l-1)} + \vec{b}^{(l)}$

$\frac{\partial \vec{z}^{(l)}}{\partial \vec{b}^{(l)}} = I_{m^{(l)}}$ 为 $m^{(l)} \times m^{(l)}$ 的单位矩阵

No.

Date. . .

⇒⇒⇒ 计算误差项 $\frac{\partial L(\vec{y},\hat{\vec{y}})}{\partial \vec{z}^{(l)}}$

第 l 层神经元的误差项 表示 l 层神经元对最终误差的影响.

$$\delta^{(l)} = \frac{\partial L(\vec{y},\hat{\vec{y}})}{\partial \vec{z}^{(l)}}$$

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

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

根据 $\vec{a}^{(l)} = f(\vec{z}^{(l)})$, $f(\cdot)$ 为逐位计算函数

$$\frac{\partial \vec{a}^{(l)}}{\partial \vec{z}^{(l)}} = \frac{\partial f(\vec{z}^{(l)})}{\partial \vec{z}^{(l)}}$$

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

根据链式法则

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

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

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

$$= f_l'(\vec{z}^{(l)}) \odot ((W^{(l+1)})^T \delta^{(l+1)}) \quad (4.59)$$

No.

Date.

$$\frac{\partial L(\vec{y},\hat{\vec{y}})}{\partial W_{ij}^{(l)}} = \prod_i (a_j^{(l)})^T \delta^{(l)} = \delta_i^{(l)} a_j^{(l-1)}$$

第 l 层梯度 $W^{(l)}$ 的梯度为

$$\frac{\partial L(\vec{y},\hat{\vec{y}})}{\partial W^{(l)}} = \delta^{(l)} \cdot (\vec{a}^{(l-1)})^T$$

同理

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

No.

Date. . .

4.5 自动梯度计算

⇒ 4.5.1 数值微分

计算不准确

计算复杂

⇒ 4.5.2 符号微分

优点：平台无关　　处理数学表达式

缺点：编译时间长

查询语言

难以调试

⇒ 4.5.3 自动微分

处理一个元数字一段程序

$$f(x;w,b) = \frac{1}{\exp(-(wx+b)) + 1}$$ ⇒ 分解成计算图

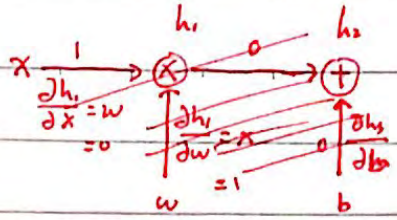
非叶子节点表示一个基本操作

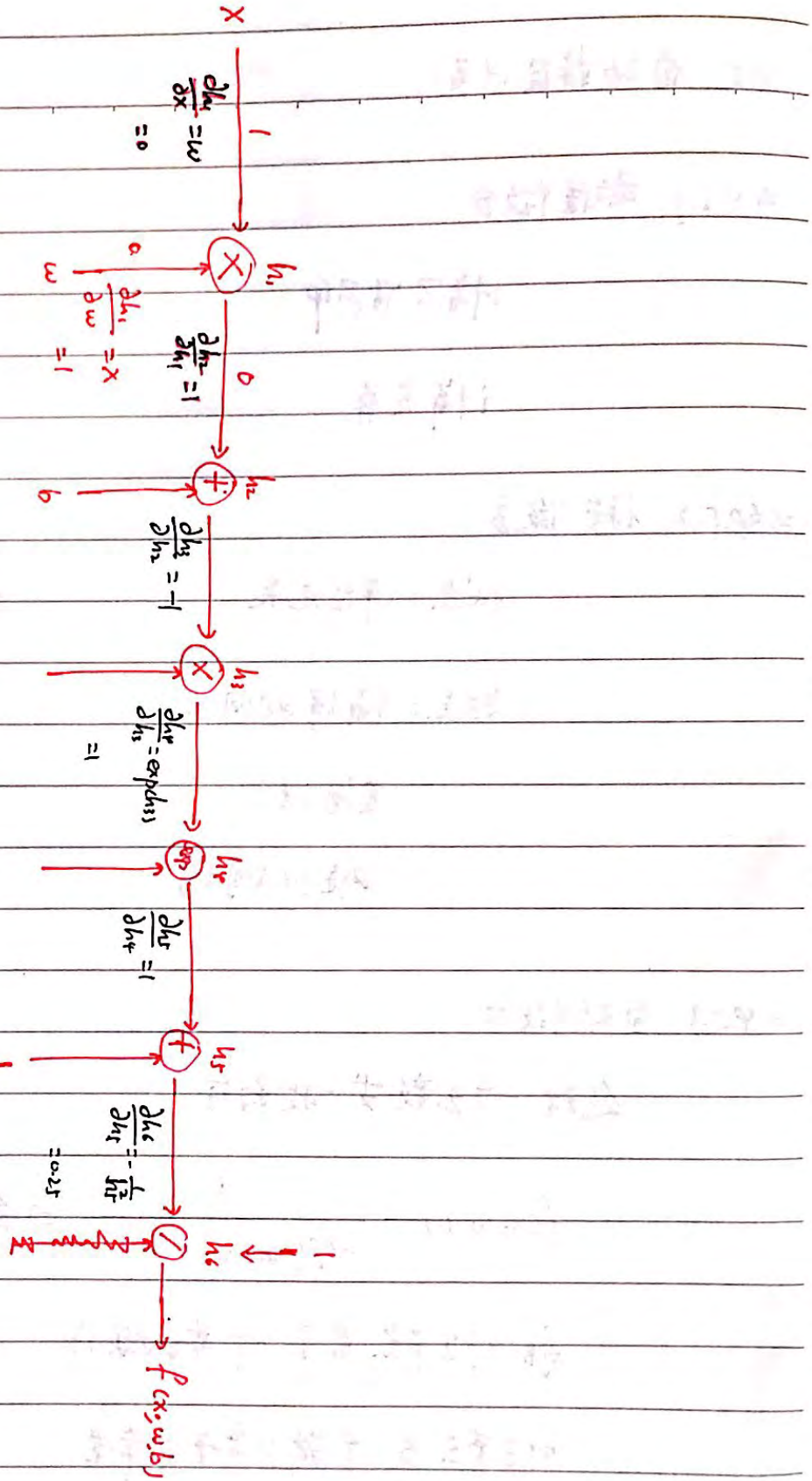
叶子节点为一个输入变量或常量

复合函数 $f(x;w,b)$ 由所基本函数构成

No.

Date.





No.

Date. . .

$$\frac{\partial f(x;w,b)}{\partial w} = \frac{\partial f(x;w,b)}{\partial h_6}\cdot\frac{\partial h_6}{\partial h_5}\cdot\frac{\partial h_5}{\partial h_4}\cdot\frac{\partial h_4}{\partial h_3}\cdot\frac{\partial h_3}{\partial h_2}\cdot\frac{\partial h_2}{\partial h_1}\cdot\frac{\partial h_1}{\partial w}$$

$$\frac{\partial f(x;w,b)}{\partial b} = \frac{\partial f(x;w,b)}{\partial h_6}\cdot\frac{\partial h_6}{\partial h_5}\cdot\frac{\partial h_5}{\partial h_4}\cdot\frac{\partial h_4}{\partial h_3}\cdot\frac{\partial h_3}{\partial h_2}\cdot\frac{\partial h_2}{\partial b}$$

如果函数和变量之间有多条路径，可以将多条路径上的导数再相加得到最终导数.

前向模式 如左图. $\frac{\partial h_1}{\partial w} \rightarrow \frac{\partial f(x;w,b)}{\partial w}$

反向模式 $\frac{\partial f(x;w,b)}{\partial w} \rightarrow \frac{\partial h_1}{\partial w}$

符号微分和自动微分

静态计算图和动态计算图.

No.

Date. . .

4.6 优化问题

⇒ 4.6.1 非凸优化问题

⇒ 4.6.2 梯度消失问题

$$\delta^{(l)} = f_l'(\vec{z}^{(l)}) \odot (W^{(l+1)})^T \delta^{(l+1)}$$

sigmoid 饱和导数接近 0.