Week 05.

5.1 神经元模型

$y_i = f(\sum_i w_i x_i - \theta_j)$

其中 f 为激活函数

5.2 感知机

与： $y = f(x_1 + x_2 - 2)$

或： $y = f(x_1 + x_2 - 0.5)$

非： $y = f(-0.6 x_1 + 0.5)$

调整原则： $w_i \leftarrow w_i + \Delta w_i$

$\Delta w_i = \eta (y - \hat{y}) x_i$

5.3 BP 算法

$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ $x_i \in R^d$. y_i , R^l

$\alpha_h = \sum_{i=1}^{d} w_{ih} x_i$, $b_h = \delta(\alpha_h)$

$\beta_j = \sum_{j=1}^{q} w_{hj} x_j$, $\hat{y} = \delta(\beta_j)$ δ : sigmoid 函数.

α_h 为第h个隐层的输入， $\beta_j \Rightarrow$ 第j个输出层的输入， $b_h \Rightarrow$ 第h个隐层的输出

对于 (x_k, y_k) 此时 $E_k = \frac{1}{2} \sum_{j=1}^{l} (\hat{y}_j^k - y_j^k)^2$

参数： $d \times q$, $q \times l$, d, q

$v \leftarrow v - \eta\, dv$

$dw_{hj} = \frac{\partial E_k}{\partial w_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}} = (\hat{y} - y) \cdot \hat{y}(1 - \hat{y}) x_j$

缓解 BP 网络过拟合
- 早停： 训练.验证， e_{train} 下降， e_{valid} 上升则停止
- 正则化： $E = \lambda \frac{1}{m} \sum_{k=1}^{m} E_k + (1-\lambda) \sum_i w_i^2$

5.4. 跳出局部极小的方法

① 不同初始点

② 模拟退火：接受次优解

③ sgd

④ 遗传算法.

5.5 其它常见网络.

(1) RBF

输入：d 维 x，输出：实值

$$\varphi(x)=\sum_{i=1}^{q} w_i \rho(x, c_i)$$

$$\rho(x, c_i)=e^{-\beta_i \|x-c_i\|^2}$$

训练方法：1. 采样聚类 $\rightarrow c_i$

2. BP算法 $\rightarrow w_i, \beta_i$

(2) ART：自适应谐振网络，

比较层，识别层，识别层阈值，重置模块.

$\Rightarrow$ 竞争 $\Longrightarrow$ 计算相似度

计算对应模式距离

小者胜.

大于阈值，更新权值

不大于，增设新神经元

(3) SOM：自组织映射

高维 $\Rightarrow$ 低维，保持拓扑结构.

输入 → 计算最佳匹配单元 → 更新权重 至收敛.

(4) 级联网络：结构自适应

(5) Elman 网络：递归网络

t时刻输出 有关 $\begin{cases} t\text{时刻输入} \\ t-1\text{时刻网络状态} \end{cases}$

深度学习

无监督逐层训练：

1.预训练：上层输出作下层输入

2.BP训练

或 CNN 权值共享

(6) Boltzmann 机

最小化能量 $$E(s)=-\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w_{ij} s_i s_j-\sum_{i=1}^{n}\theta_i s_i$$

神经元任意分布：$$p(s)=\frac{e^{-E(s)}}{\sum_t e^{-E(t)}}$$

受限 Boltzmann 机：对比散度训练

$$p(v|h)=\prod_{i=1}^{d} p(v_i|h),\quad p(h|v)=\prod_{j=1}^{q} p(h_j|v)$$

互相迭代 $$\Delta w=\eta(vh^T-v'h'^T)$$

w_1, w_2, x_1, x_2, b $\rightarrow$ $z = w_1 x_1 + w_2 x_2 + b \longrightarrow \hat{y} = \sigma(z) \longrightarrow L(\hat{y}, y)$

$-[y \log \hat{y} + (1-y)\log(1-\hat{y})]$

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_1}$$

$$= \left[-\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}}\right] \cdot \hat{y}(1-\hat{y}) \cdot x_1$$

$$= (\hat{y} - y)\, x_1$$

$$\frac{\partial L}{\partial w_2} = (\hat{y} - y)\, x_2, \qquad \frac{\partial L}{\partial b} = \hat{y} - y$$

$dw = 0$, $db = 0$

$z = np.dot(w^T, x) + b$ $\Rightarrow$ $\begin{cases} w^T & (1, n_x) \\ x & (n_x, m) \end{cases}$

$\hat{y} = 1 / (1 + np.exp(z))$

b使用广播.

$dz = \hat{y} - y$ $\Rightarrow$ $(1, m)$

$db = \frac{1}{m} \cdot np.sum(dz.T)$

$dw = \frac{1}{m} \cdot np.sum(np.dot(x, (dz).T))$

db : m, 1
x : n_x, m
dz : 1, m
dw : n_x, 1

$w = w - \alpha\, dw$

$b = b - \alpha\, db$

① np.random.randn((5,1)) √

尽量不要用 (5)

② 用了请 a.reshape(5,1)

③ 进一步确认

assert(a.shape == (5,1))