

## 1. BP 算法讨论

### \* 常用准则函数:

$$\text{平方误差准则: } E(w) = \frac{1}{2} \|y - a^{(L)}\|^2$$

适合连续输出，且最后一层不含 Sigmoid 或 Softmax 激活函数的神经网络

$$\text{注意: } \delta^{(L)} = -(y - a^{(L)}) \cdot f'(z^{(L)})$$

而对 + sigmoid 或 softmax 函数而言若  $|z^{(L)}|$  很大，则  $f'(z^{(L)}) \rightarrow 0$ ，进而  $\delta^{(L)}$  很小，反向传播过程中更新缓慢。

$$\text{交叉熵准则: } E_{ce}(w) = -\sum_{k=1}^K y_k \ln f(z_k^{(L)})$$

适合二分类或多分类场景，二分类时，设类别  $y_k$  取值 0, 1，则

$$E_{ce}(w) = -\ln f(z_k^{(L)}) \quad \text{取 } f \text{ 为 softmax，有}$$

$$\Rightarrow \delta^{(L)} = f(z_k^{(L)}) - 1$$

此时可认为误差为残差，不会有学习过慢的问题。

$$\text{Minkowski 误差准则: } E_{mink}(w) = \left( \sum_{k=1}^K |y_k - f(z_k^{(L)})|^R \right)^{1/R}, \quad 1 < R < 2.$$

### \* 激励函数选取:

基本要求：非线性、连续可导、最好单调（否则增加局部最小值点）

### \* 隐含层数量：

- 重要结论：
- 各结点具有不同阈值时，具一个隐含层的网络可以表示任意函数。
  - 采用 S型函数时，一个隐含层可实现任意判别分类问题，两个隐含层可实现向量输入的任意输出函数。
  - 隐含层数量  $\Rightarrow$  网络表达能力。

### \* 结点个数：

通常隐含层结点个数较多，网络训练结束后再考虑减少结点，可使用

压缩神经网络

稀疏连接的神经网络

dropout 技术

### \* 初始权重：

会影响网络收敛速度和最终收敛位置

通常在一个均匀分布中随机选择： $-w_0 < w < w_0$  ( $w_0$  不能太小)

### \* 正则化技术：

防止网络过拟合而采用的技术

$$2\text{-norm: } E_{new}(w) = E(w) + \frac{\lambda}{2} \sum_j w_j^2$$

$$\text{启发式: } E_{new}(w) = E(w) + \frac{\lambda}{2} \sum_j \frac{w_j^2 / w_j^2}{1 + w_j^2 / w_j^2}$$

\* 学习率：太小收敛太慢，太大训练不稳定，一般而言训练过程中会衰减。

$$w(t+1) = w(t) + (1-\alpha) \Delta_{bp} w(t) + \alpha \Delta(t-1)$$

$\Delta_{bp} w(t)$ : BP 算法获取更新量

$$\Delta(t-1): w(t) - w(t-1)$$

优点：

1. 迭代序列更平滑

2. 调整以可逃离饱和区

\*停止训练准则：没有固定准则，但过度训练会导致 overfitting

\*BP算法存在的问题：

### 1. 网络的麻痺现象

误差  $\delta$  正比于  $f'(x)$ ，若权重较大， $\delta$  型函数在饱和区训练，可能训练很慢

### 2. 梯度消失

每一层收集误差时都需要乘上该层导数，而导数通常小于 1。网各层教很深时，下层误差衰减严重

### 3. 局部极小

由于网络训练通常非凸，存在众多局部极小，训练过程很可能陷入其中。

### 4. 对于复杂问题，训练时间较长。

## 2. 径向基函数网络 (RBF)

使用场景：模式分类，函数近似（可对任意连续非线性函数近似）

问题背景：d 维空间有 n 个样本  $\{x_1, \dots, x_n\}$

样本经函数映射至 p 维空间  $\{t_1, \dots, t_p\}$

现需还原映射函数。

思路：用径向基函数的组合来作近似

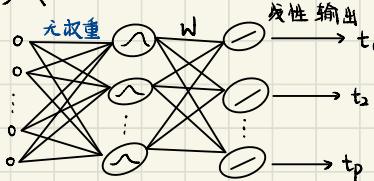
$$g_j(x) = \sum_{k=1}^p w_{kj} \phi_k(x), \quad j=1, \dots, p$$

每一维对应不同的加权求和。

矩阵形式：

$$\begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_n(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_n(x_2) \\ \vdots & \vdots & & \vdots \\ \phi_1(x_n) & \phi_2(x_n) & \dots & \phi_n(x_n) \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1p} \\ w_{21} & w_{22} & \dots & w_{2p} \\ \vdots & \vdots & & \vdots \\ w_{n1} & w_{n2} & \dots & w_{np} \end{bmatrix} = \begin{bmatrix} t_1^1 & t_1^2 & \dots & t_1^p \\ t_2^1 & t_2^2 & \dots & t_2^p \\ \vdots & \vdots & & \vdots \\ t_p^1 & t_p^2 & \dots & t_p^n \end{bmatrix}$$

网络形式：



n 个径向基函数 加权求和

问题：由于 n 个样本就需要 n 个径向基函数。对大规模数据时，可用聚类中心替代，再用非线性输出。这样还可以增加泛化能力。

常用径向基函数：

$$\text{Gaussian: } \phi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right)$$

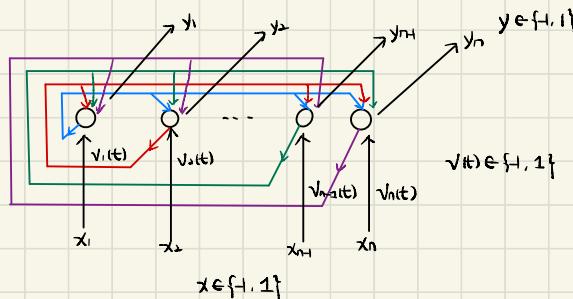
$$\text{Reflected Sigmoid: } \phi(r) = \frac{1}{1 + \exp\left(\frac{r}{2\sigma}\right)}$$

$$\text{Inverse multi-quadratics: } \phi(r) = \frac{1}{\sqrt{r^2 + \sigma^2}}$$

### 3. 反馈神经网络

通过神经元状态变迁，最终稳定于某一状态，具有稳定性、各种类型吸引子以及混沌现象

单层网络  
结构：各  
节点地位  
相同



#### Hopfield 网络

网络输入： $x = (x_1, x_2, \dots, x_d)$ ， $x_i \in \{-1, 1\}$

网络输出： $y = (y_1, y_2, \dots, y_d)$ ， $y_i \in \{-1, 1\}$

网络在 t 时刻状态： $v(t) = (v_1(t), v_2(t), \dots, v_d(t))$ ， $v_i(t) \in \{-1, 1\}$

结点间连接权重：Hopfield 网络对称连接（保证稳定性）， $w_{ij} = w_{ji}$ ， $w_{ii} = 0$

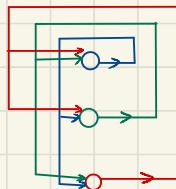
网络运行方式：

- ② 输入初始双极型数据  $x = (x_1, \dots, x_d)$ 。

③ 计算下一状态：
$$\begin{cases} v_j(0) = x_j \\ v_j(t+1) = f_j\left(\sum_i w_{ij} v_i(t) + w_j\right) \quad f_j = \text{sgn}(x) \end{cases}$$

④ 将状态作为输入，继续计算，直到网络状态不再改变。

⑤ 将稳定状态输出。



信息循环往复  
传递，直至到达  
稳态。

#### Boltzmann 机：

一种随机的 Hopfield 网络，具有隐单元的反馈互连网络。

BM 有一部分神经元与外部结点连接，受到外部条件约束。

但 BM 网络结构复杂，训练代价大，存在局部极小。