

NN

大纲

1. ANN 人工神经网络
2. MLP 多层感知机
3. Hopfield Network Hopfield 网络
4. Self-Organizing Feature Map 自组织映射网

ANN

1. 联结主义

1. 概念

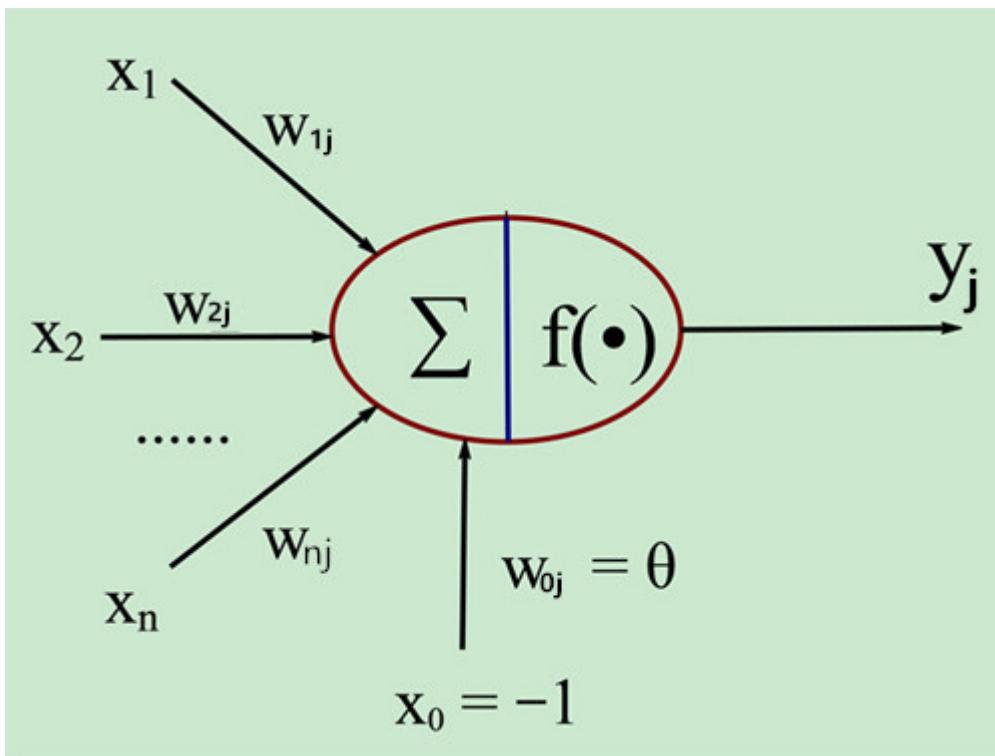
独立的个体连接形成图，图有多种结构

- 有环
- 无环
- 有向
- 无向

2. 目的

使用生物神经元的仿真结构激励形成仿真功能

2. M-P神经元模型



○ 输入

- X : 输入向量
- W : 权值矩阵
- θ : 偏置

○ 输出

- Y : 输出标量

○ 函数变换

- f: 激活函数

- Sigmoid
- Softmax
- Hardlim
- Gaussian

- g: 整合函数

1. 线性变换的权值求和

3. 神经网络的主要问题

1. 网络结构
2. 学习方式

上面的两个方面的不同就导致了我们的神经网络的不同

4. 网络结构

- 前馈网络

- 静态
- 无循环
- 后层神经元的输出对前层的神经元没有影响

- 反馈网络

- 动态(非线性动态系统)
- 存在循环
- 后曾的神经元的输出可以对前层的神经元产生影响(甚至同层之间也会存在权值连接关系), 已经变成纯粹的网络结构

5. 学习方式

- 学习思想

1. 通过环境的影响来调整学习的参数
2. 增量式学习 / 批处理学习
3. 监督学习 / 无监督学习

- 学习策略

1. 误差校正

- 监督式
- 最小化误差
- 学习模式

$$w = \arg \min_{\theta} \frac{1}{K} \sum_{k=1}^K error(Exp, Y)$$
$$w = w + \eta \delta$$

2. Hebbrian学习

- 无监督
- 强化两个神经元之间的联系
 - 同一时间被激发的神经元间的联系会被强化
 - 如果两个神经元总是不能同步激发, 那么它们间的联系将会越来越弱
- 学习模式

$$W_{ij}(t+1) = W_{ij}(t) + a \cdot y_i \cdot y_j$$

- W_{ij} : i,j神经元之间的连接的权值
- a : 学习率
- y_i : i神经元的输出
- y_j : j神经元的输出

3. 竞争学习

- 无监督
- 竞争学习是指网络单元群体中所有单元相互竞争对外界刺激模式响应的权利
- 竞争取胜的单元抑制了竞争失败单元对刺激模式的响应

MLP

说在前面

- 线性函数的组合还是线性函数
 - 1. $ax+b, cx+d, ex+f, \dots$ 合成之后还是线性函数，只不过是系数的累加
 - 2. 激活函数如果也是线性的话，相当于是对一个线性函数套上了 $y=kx+b$ 的幌子，还是线性函数
 - 3. 线性函数表达的函数集有限，非线性函数可以拟合和表达更多的函数形式(只要我们的精度足够高)

1. 单层感知

- 二元数值型激活函数
- 输入层，输出层
- 学习方式
 - 监督学习
 - 最小化误差
- 异或问题不能解决?
 - 1. 我们单纯从拟合函数的角度说(这里不考虑线性分类问题)
 - 2. 我们将问题升华一下?单层感知机拟合非线性函数?
 - 3. 可以的(A Fast Algorithm For Training Single Layered Perception Neural Networks)

2. 陷入低潮期(激活函数的非线性构造思路是偏门没人去研究)

3. 引入多层感知机对非线性函数拟合

- 三层神经网络可以拟合所有的非线性函数，四层网络可以表达一切函数
- 更多曾表示的函数集合更多，在高维特征下的表现更加的简单

4. 深度网络的好处

- 1. 学习特征的层次结构，高层次的特征是从低层次的结构中抽取出来的
- 2. 深度网络可以更简单的表示我们的非线性的函数形式(相对于浅层来说)
 - K层网络秒数的函数形式，在K-1层中可能需要指数形式的节点个数才可以有效的表示

3. 结论

- 1. 指数个数的网络的节点个数的计算代价不可容忍
- 2. 层次低的网络可能描述的模型框架可能存在较低的泛化能力

5. 学习模式

B-P网络

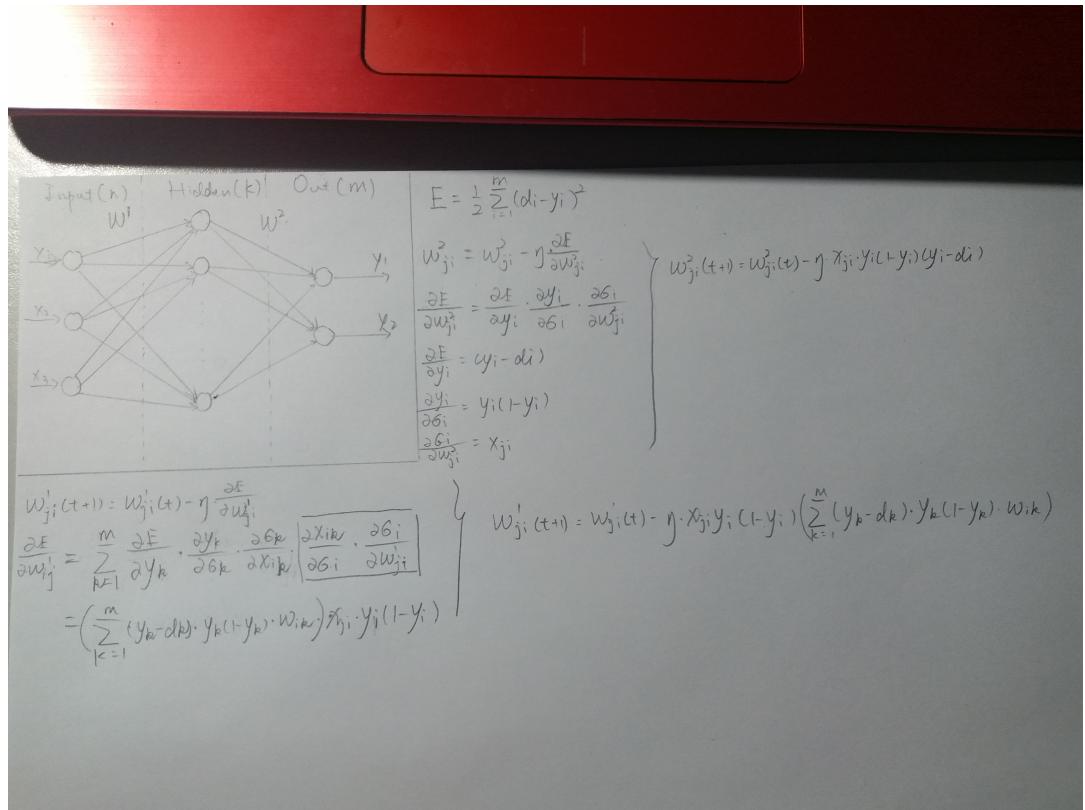
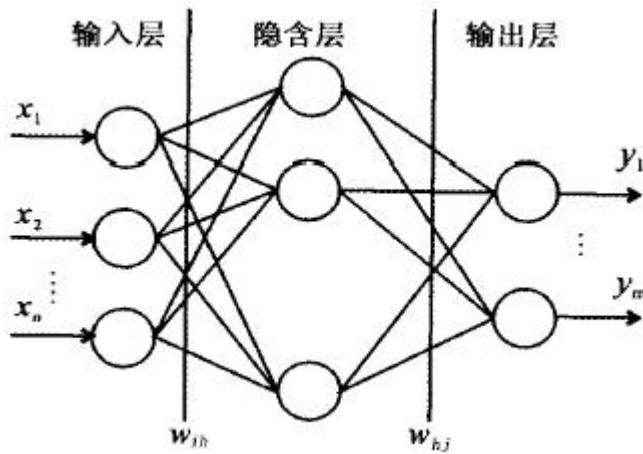
- 使用Sigmoid非线性激活函数的多层感知器
- 使用误差反向传播的思路动态调整参数
- 算法
 - 输入->输出向量的前馈计算
 - 误差计算公式(MSE常用)
 - 误差反向传播到前面的权重矩阵进行调整(梯度下降)
 - 迭代调整误差
- 算法的推导
 - 1. 误差计算公式 MSE

$$E = \frac{1}{2} \sum_{i=1}^n (d_i - y_i)^2$$

2. GD - BGD / SGD

$$w = w + \eta \frac{\partial E}{\partial w}$$

3. 推导



◦ 应用

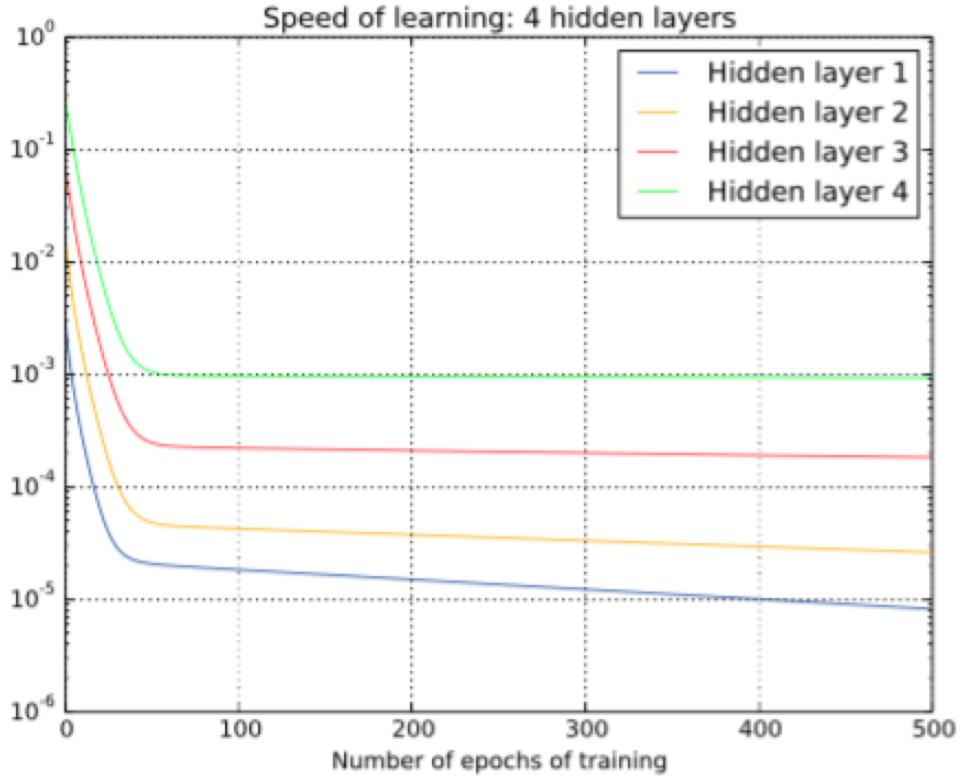
1. 人脸识别
2. 字符识别

◦ 深度BP网络

1. 问题
 - BP网络中我们的经常使用的是3,4层网络
 - 深层的网络照理说我们会得到更加优秀的结果，但是结果往往不如人意
2. MLP早期层次没有得到更好的序列训练
3. 后层的网络的权值调整并不是合理的，因为前层的权值的调整并没有到位
4. 更容易收敛到局部最小

5. 梯度消失 / 梯度爆炸

1. 梯度消失



$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \underbrace{w_2 \sigma'(z_2)}_{<\frac{1}{4}} \underbrace{w_3 \sigma'(z_3)}_{<\frac{1}{4}} \underbrace{w_4 \sigma'(z_4)}_{\text{common terms}} \frac{\partial C}{\partial a_4}$$

$$\frac{\partial C}{\partial b_3} = \sigma'(z_3) w_4 \sigma'(z_4) \frac{\partial C}{\partial a_4}$$

■ 前提我们的权值初始化的时候使用(0-1)高斯分布

2. 梯度爆炸

如果我们的权值初始化的值过大或者相对大，在我们的求Sigmoid的函数的倒数相乘之后会得到一个相对很大的值，导致梯度暴增，算法迭代过程中会出现局部震荡

DBN / Autoencoders

1. DBN

- 隐藏层引入Sigmoid非线性变换
- 输出层引入Softmax输出分类概率
- 使用概率生成模型

- 包含多层隐含层以及大量的隐含参数，扩大表达的函数的空间
- 无监督学习初始化
- 学习模式
 - MLE : 最大似然法
 - 反向传播
- RBM(受限玻尔兹曼机)

2. Autoencoders

Hopfield Network

- 反馈，动态结构
- 工作模式
 - 并行工作模式：多个神经元同时更新状态
 - 串行工作模式：一次只有一个神经元更新状态
- 引入能量函数决定我们的反馈系统是稳定的

$$E = -\frac{1}{2} \sum_{i=0}^n \sum_{j=0}^n w_{ij} s_i s_j - \sum_{i=1}^n I_i s_i$$

$$E = -\frac{1}{2} \sum_{i=0}^n \sum_{j=0}^n w_{ij} s_i s_j$$

- 已经证明，如果权重矩阵满足下面条件，网络结构就是稳定的

$$\begin{aligned} E &= -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j + \sum_{i=1}^n \theta_i x_i \\ &= -\frac{1}{2} \left(x_i \sum_{j=1}^n w_{ij} x_j + \sum_{k \neq i} \sum_{j=1}^n w_{kj} x_k x_j \right) + \theta_i x_i + \sum_{k \neq i} \theta_k x_k \\ &= -\frac{1}{2} \left(x_i \left(w_{ii} x_i + \sum_{j \neq i} w_{ij} x_j \right) + \sum_{k \neq i} \left(w_{ki} x_k x_i + \sum_{j \neq i} w_{kj} x_k x_j \right) \right) + \theta_i x_i + \sum_{k \neq i} \theta_k x_k \\ &= -\frac{1}{2} \left(w_{ii} x_i^2 + x_i \sum_{j \neq i} w_{ij} x_j + x_i \sum_{k \neq i} w_{ki} x_k + \sum_{k \neq i} \sum_{j \neq i} w_{kj} x_k x_j \right) + \theta_i x_i + \sum_{k \neq i} \theta_k x_k \\ \Delta E &= -\frac{1}{2} \left(\Delta x_i \sum_{j \neq i} w_{ij} x_j + \Delta x_i \sum_{k \neq i} w_{ki} x_k \right) + \theta_i \Delta x_i \\ &= -\frac{1}{2} \left(\Delta x_i \sum_{j=1}^n w_{ij} x_j + \Delta x_i \sum_{k=1}^n w_{ki} x_k \right) + \theta_i \Delta x_i \\ &= \boxed{-\Delta x_i \left(\sum_{j=1}^n w_{ij} x_j - \theta_i \right)} \end{aligned}$$

$$w_{ij} = \begin{cases} w_{ji}, i \neq j \\ 0, i = j \end{cases}$$

无论初始值是多少，网络最终都会收敛(不一定是全局最优解)[变化率一定是负的，能量函数一定递减]

- 联想记忆(离散Hopfield网络)？？？？

基于与数据本身部分相似的输入来回忆数据，联想记忆具有记忆的容错能力，可以在有噪声的情况下回复我们的存储的稳定状态(但是有可能是局部最优的状态)

1. 训练权值 \mathbf{W}

- 利用输入的数据监督式的训练 \mathbf{W} 权重

$$\mathbf{w} = \sum_{k=1}^K \mathbf{X}_k \mathbf{X}_k^T - K \mathbf{I}$$

- 将训练后的 \mathbf{W} 矩阵加入能量函数中进行迭代(串行工作模式，一次只改变一个节点的状态，改变状态到一个能量函数最低的状态[必收敛]，直到收敛)

2. 根据输入迭代求出联想的结果

- 最优化(连续Hopfield网络)
 - Hopfield收敛到极小值的能力决定了该网络可以执行最优化计算
 - 首先将最优化问题的目标函数转变成Hopfield的能量函数即可
 - 举例TSP问题
 - 目标函数转化成能量函数

$$\begin{aligned} E = & \frac{\lambda_1}{2} \sum_{x=1}^n \sum_{y=1}^n \sum_{i=1}^n \sum_{j=1}^n s_{xi} d_{xy} s_{yj} + \\ & \frac{\lambda_2}{2} \sum_{x=1}^n \sum_{i=1}^n \sum_{j=1}^n s_{xi} s_{xj} + \frac{\lambda_3}{2} \sum_{i=1}^n \sum_{x=1}^n \sum_{y=1}^n s_{xi} s_{yi} + \frac{\lambda_4}{2} (\sum_{x=1}^n \sum_{i=1}^n s_{xi} - n)^2 \end{aligned}$$

SOM

- 无监督学习算法，类似于聚类算法(但是无需超参数)
- 降维但是保持信息的结构(拓扑结构)不变[天然的降维方法]
 - 在二维或者低维空间中有效的表示高维数据
- 学习原则
 - 自我强化
 - 竞争
 - 合作
- 典型SOM网共有两层，输入层模拟感知外界输入信息的视网膜，输出层模拟做出响应的大脑皮层
- 输入层
 - 信号传递作用
 - 维数任意
- 输出层
 - 工作层，代表大脑皮层的特定功能区
 - 加权整合函数
 - 激活函数是线性函数
 - 维数一般是1-2维
- 连接方式

- 输入层神经元和输出层神经元的前向连接
 - 目的是为了特征提取
 - 每一个输出层神经元只响应特殊的输入信号
- 输出层神经元之间的侧向连接
 - 考虑神经元之间的竞争合作关系
 - 根据神经元之间的举例来加强和抑制神经元的兴奋
 - 根据神经生物学的研究成果，侧向连接的权重演示墨西哥草帽函数
 - 协同区(重点)：如果神经元处在协同区的激活气泡(中心激活神经元的邻域)中处于激活状态
否则是抑制状态
 - 抑制区
 - 弱激励区
- 运行机制
 - 学习什么：学习连接的权重矩阵
 - 竞争 / 合作
 - 每个神经元竞争对输入模式的响应，并且出现协同现象
 - 竞争可以使用欧式距离来计算相似度，选择最合适的神经元激活
 1. 怎么竞争？
 - 每一个输入层神经元和输出层神经元都存在权值连接
 - 当获得输入向量之后，将每一个神经元的权重向量和输入向量进行比较(相似度计算)
 - 选取最小距离(最相似)的神经元激活(获胜)
 - 邻域形状可以根据需要指定(一般使用高斯邻域)
 - 协同邻域应该随着时间收缩(甚至只包含获胜神经元本身)
 - 权重更新
 - 邻域内的神经元的权值被更新
 - 调整权重

$$w_j(t+1) = \begin{cases} w_j(t) + \eta(t)(X(t) - w_j(t)), & j \in \Lambda_i(t) \\ w_j(t), & j \notin \Lambda_i(t) \end{cases}$$
 - 只有竞争获胜的神经元及其邻域神经元会被更新权重，调整是使得权重向量尽可能靠近输入向量
 - 算法
 1. 初始化随机设置输出层连接权值(随机的思想和我们没有开始学习的状态是一致的)
 2. 随机采样构建输入向量
 3. 计算相似性获得获胜神经元及其邻域
 4. 根据上述的调整区中公式更新权重
 5. 更新邻域大小?????
 6. 迭代知道权重无显著变化为止
 - 理解
 1. 开始的SOM网络是混沌的，在不断的学习之后我们就会在构建出有拓扑结构的权值矩阵，构成了输入到输出的映射关系
 2. 变换的特性
 1. 拓扑排序特性：数据的模式被学习和存储下来
 2. 密度匹配特性
 3. 输入空间逼近特性
 - 输出空间是输入空间的一种近似表达
 - 实现了数据压缩和维数压缩存储的特性
 - 数据压缩：将大量的数据信息压缩在权重矩阵中

- 维数压缩：高维数据被压缩成离散的1-2维数据(大脑的仿生)

4. 特征选择特性