人工智能与机器学习

第十二讲 神经网络（下）

倪东 叶琦
工业控制研究所

杭州 • 浙江大学 • 2022

提纲

- 讨论训练单个单元的学习算法
- 介绍组成神经网络的几种主要单元
  - 感知器（perceptron）
  - 线性单元（linear unit）
  - sigmoid单元（sigmoid unit）
- 给出训练多层网络的反向传播算法
- 讨论几个一般性问题
  - ANN的表征能力
  - 假设空间搜索的本质特征
  - 过度拟合问题
  - 反向传播算法的变体

小结（2）

- 梯度下降收敛到训练误差相对网络权值的局部极小值。只要训练误差是假设参数的可微函数，梯度下降可用来搜索很多连续参数构成的假设空间
- 反向传播算法能够创造出网络输入中没有明确出现的特征。
- 交叉验证方法可以用来估计梯度下降搜索的合适终止点，从而最小化过度拟合的风险
- 其他ANN学习算法，递归网络方法训练包含有向环的网络，级联相关算法改变权和网络结构

补充读物

- 本书其他与ANN学习相关的章节
  - 第6章给出了选择最小化误差平方和的贝叶斯论证，以及在某些情况下，用最小化交叉熵代替最小化误差平方和的方法
  - 第7章讨论了为可靠学习布尔函数所需要的训练实例数量的理论结果，以及某些类型网络的VC维
  - 第5章讨论了过度拟合和避免过度拟合的方法
  - 第12章讨论了使用以前知识来提高泛化精度的方法

补充读物（2）

- 发展历程
  - McCulloch & Pitts
  - Widrow & Hoff
  - Rosenblatt
  - Minsky & Papert
  - Rumelhart & McClelland; Parker
- 教科书
  - Duda & Hart
  - Windrow & Stearns
  - Rumelhart & McClelland

收敛性和局部极小值

- 对于多层网络，误差曲面可能含有多个不同的局部极小值，梯度下降可能陷入这些局部极小值中的任何一个
- 对于多层网络，反向传播算法仅能保证收敛到误差E的某个局部极小值，不一定收敛到全局最小误差
- 尽管缺乏对收敛到全局最小误差的保证，反向传播算法在实践中仍是非常有效的函数逼近算法

收敛性和局部极小值（2）

- 网络的权越多，误差曲面的维数越多，也就越可能为梯度下降**提供更多的逃逸路线**
- 考虑随着训练中迭代次数的增加网络权值的演化方式
  - 如果把网络的权值初始化为接近于0的值，那么在早期的梯度下降步骤中，网络将表现为一个**非常平滑的函数**，近似为输入的线性函数，这是因为sigmoid函数本身在**权值靠近0时接近线性**
  - 仅当权值增长一定时间后，它们才会到达可以表示**高度非线性网络函数的程度**，可以预期在这个能表示更复杂函数的权空间区域存在更多的局部极小值
  - 但是当权到达这一点时，它们已经足够靠近全局最小值，即便它是这个区域的局部最小值也是**可以接受**的

收敛性和局部极小值（3）

- 用来缓解局部极小值问题的启发式规则
  - 为梯度更新法则加一个冲量，可以带动梯度下降过程，冲过狭窄的局部极小值（原则上，也可能冲过狭窄的全局最小值）
  - 使用随机的梯度下降而不是真正的梯度下降。随机近似对于每个训练样例沿一个不同的误差曲面有效下降，这些不同的误差曲面通常有不同的局部极小值，这使得下降过程不太可能陷入一个局部极小值
  - 使用同样的数据训练多个网络，但用不同的随机权值初始化每个网络。如果不同的训练产生不同的局部极小值，那么对分离的验证集合性能最好的那个网络将被选中，或者保留所有的网络，输出是所有网络输出的平均值

前馈网络的表征能力

- **布尔函数**：任何布尔函数可以被具有两层单元的网络准确表示，尽管在最坏情况下所需隐层单元的数量随着网络输入数量的增加成指数级增长。
  - 考虑下面的通用方案：对于每一个可能的输入向量，创建不同的隐层单元，并设置它的权值使**当且仅当这个特定的向量输入到网络时该单元被激活**，这样就产生了一个对于任意输入仅有一个单元被激活的隐藏层，然后把输出单元实现为一个仅由所希望的输入模式激活的或门。

前馈网络的表征能力（2）

- **连续函数**：每个有界的连续函数可以由一个**两层的网络**以任意小的误差逼近。这个结论适用于在隐藏层使用sigmoid单元、在输出层使用（非阈值）线性单元的网络。所需的隐层单元数量依赖于要逼近的函数。
- **任意函数**：任意函数可以被一个有**三层单元的网络**以任意精度逼近。两个隐藏层使用sigmoid单元，输出层使用线性单元，每层所需单元数不确定。
  - 证明方法：首先说明任意函数可以被许多局部化函数的线性组合逼近，这些局部化函数的值除了某个小范围外都为0；然后说明两层的sigmoid单元足以产生良好的局部逼近
- 注意：梯度下降从一个初始值开始，因此搜索范围里的网络权向量可能不包含所有的权向量

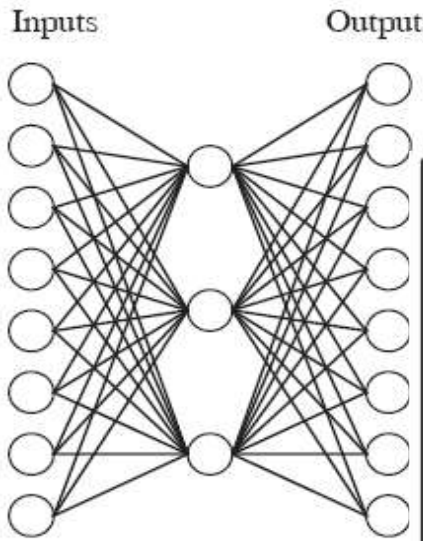
假设空间搜索和归纳偏置

- 反向传播算法的假设空间是n个网络权值形成的n维欧氏空间。这个空间是连续的，与决策树学习和其它基于离散表示的方法的假设空间不同
- 假设空间的连续性以及误差E关于假设的连续参数可微，导致了一个定义良好的误差梯度，为最佳假设的搜索提供了一个非常有用的结构。
- 精确地刻画出反向传播学习的归纳偏置是有难度的，它依赖于梯度下降搜索和权空间覆盖可表征函数空间的方式的相互作用性
- 把这一偏置粗略地刻画为在数据点之间平滑插值。如果给定两个正例，它们之间没有反例，反向传播算法会倾向于把这两点之间的点也标记为正例

隐藏层表示

- 反向传播算法的一个迷人特性是：它能够在网络内部的隐藏层发现有用的中间表示
  - 训练样例仅包含网络输入和输出，权值调节的过程可以自由地设置权值，来定义任何隐层单元表示，这些隐层单元表示在使误差E达到最小时最有效。
  - 引导反向传播算法定义新的隐藏层特征，这些特征在输入中没有明确表示出来，但能捕捉输入实例中与学习目标函数最相关的特征
- 多层网络在隐藏层自动发现有用表示的能力是ANN学习的一个关键特性。允许学习器创造出设计者没有明确引入的特征。
- 网络中使用的单元层越多，就可以创造出越复杂的特征

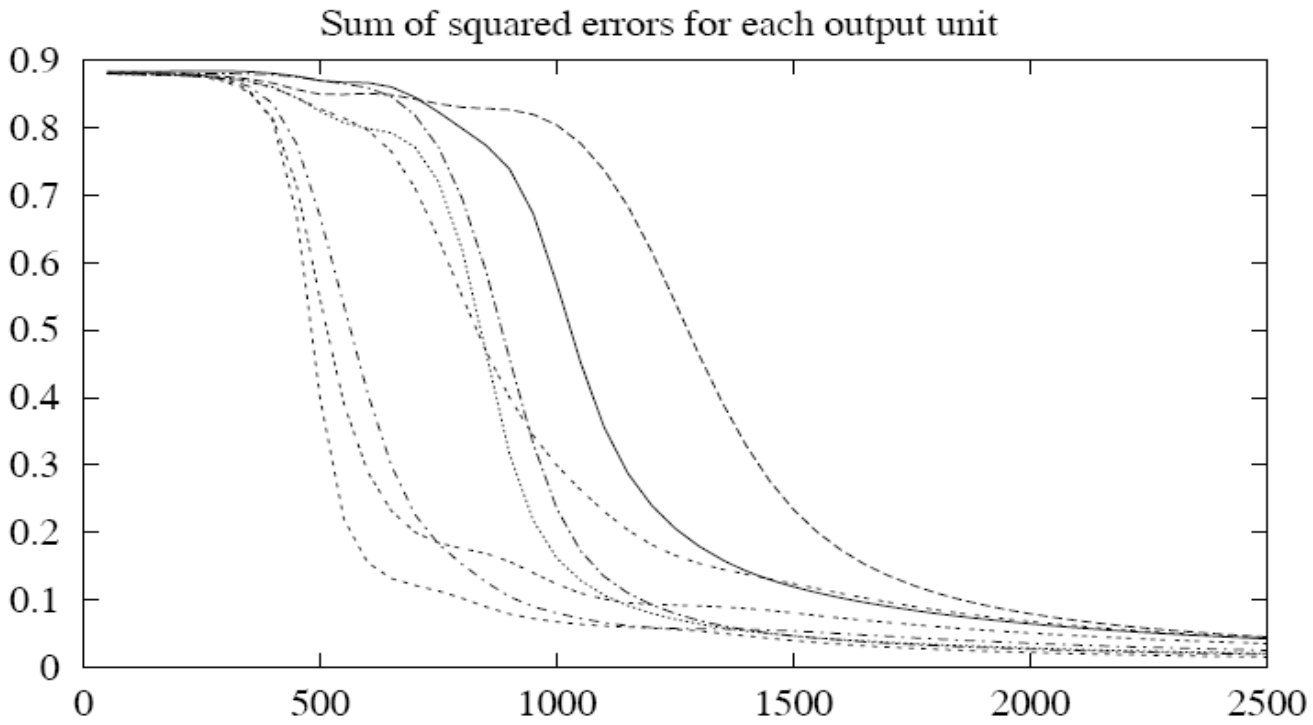
目标函数



Input		Output
10000000	→	10000000

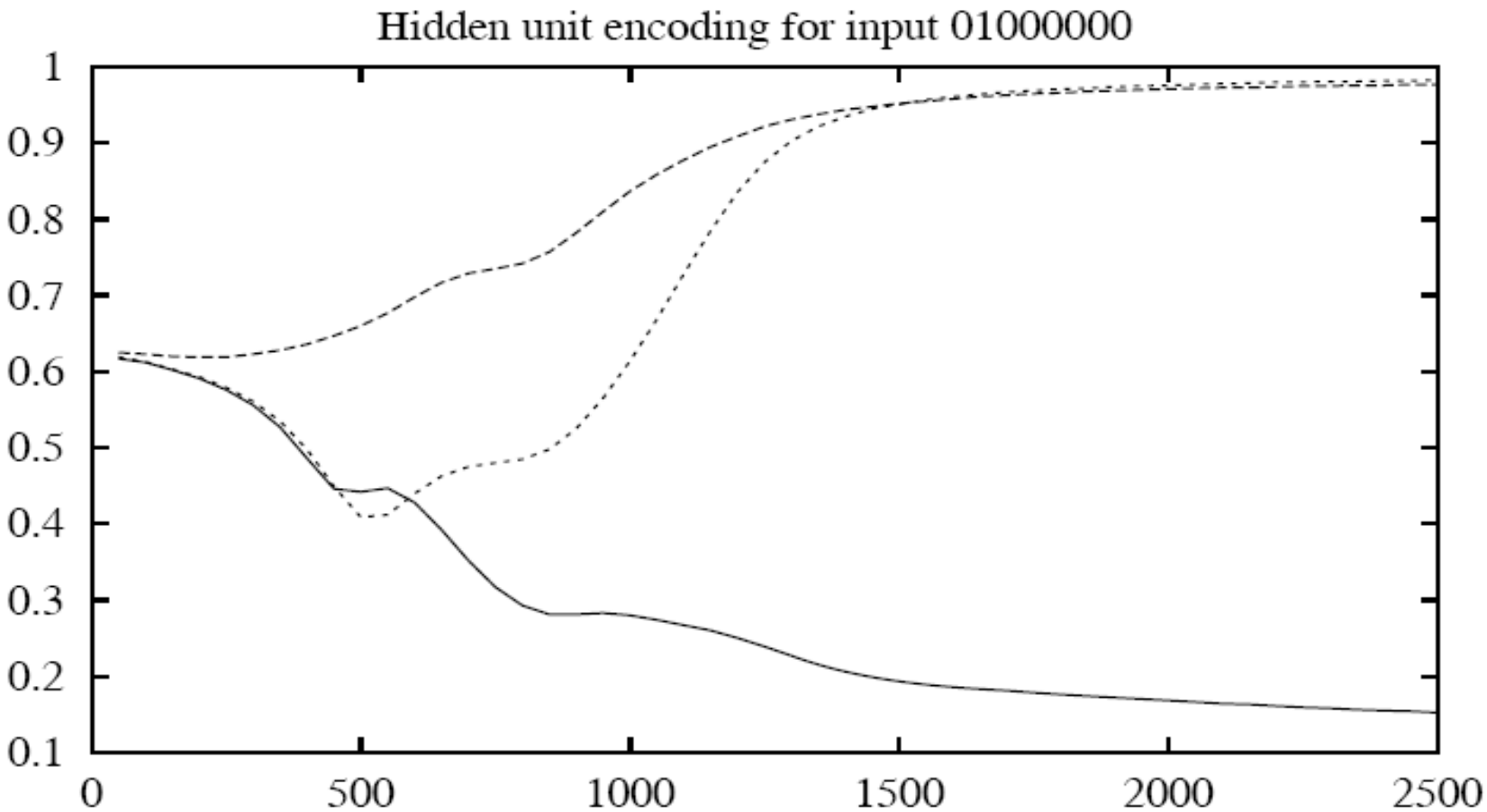
Input		Hidden Values				Output
10000000	→	.89	.04	.08	→	10000000
01000000	→	.01	.11	.88	→	01000000
00100000	→	.01	.97	.27	→	00100000
00010000	→	.99	.97	.71	→	00010000
00001000	→	.03	.05	.02	→	00001000
00000100	→	.22	.99	.99	→	00000100
00000010	→	.80	.01	.98	→	00000010
00000001	→	.60	.94	.01	→	00000001

输出单元的误差平方和



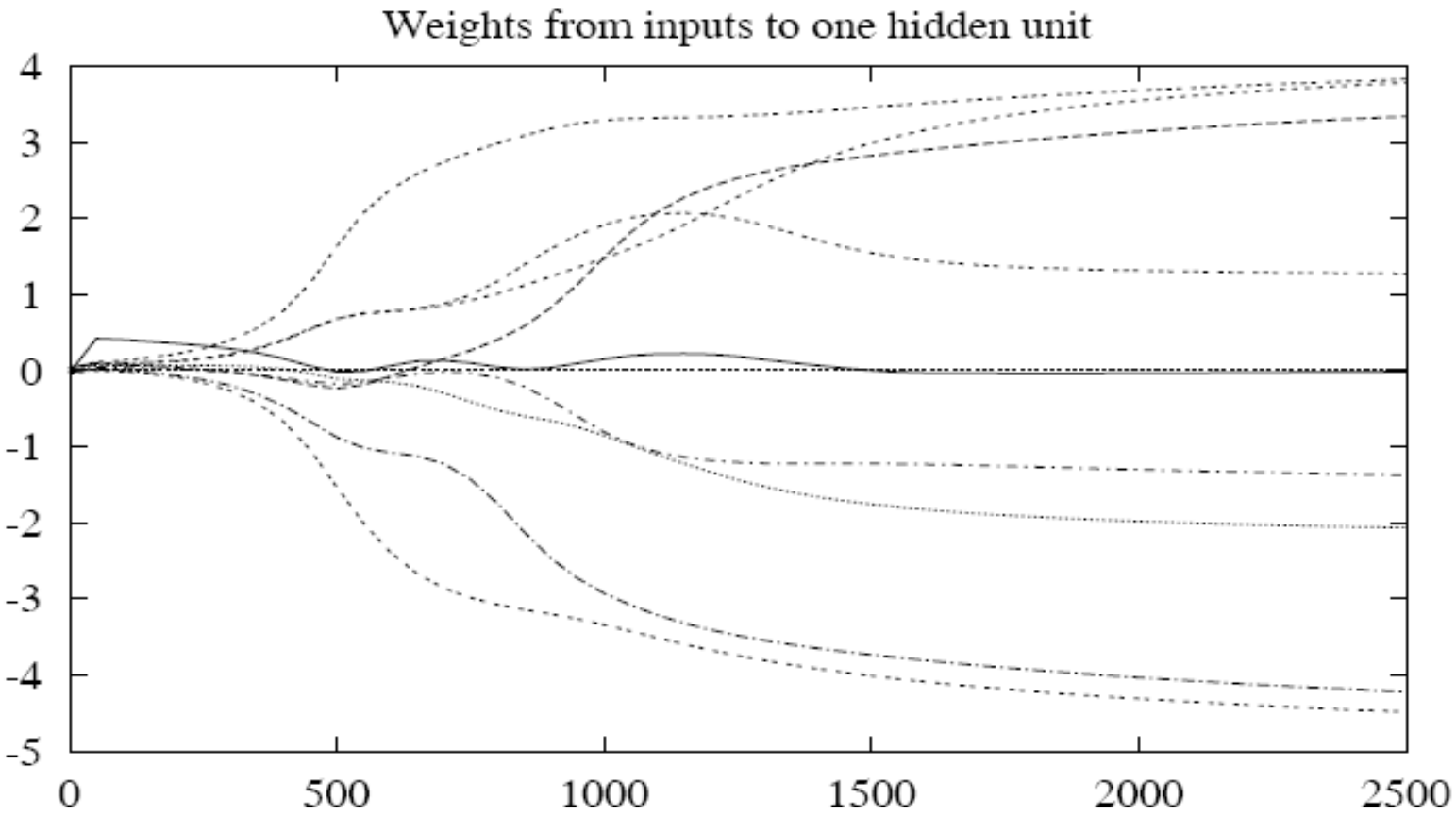
输出的误差平方和随着梯度下降而下降
某些单元快，某些单元慢

输入01000000的隐藏单元编码



隐藏单元编码在收敛之前经历不同的编码

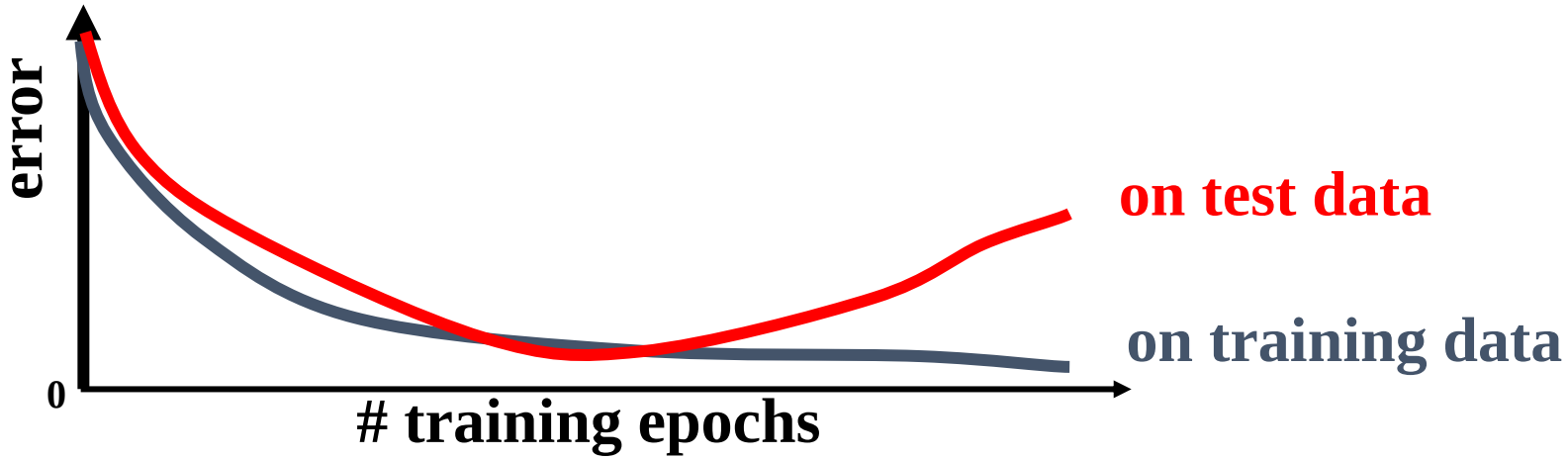
输入到一个隐藏单元的权



权值的变化和隐藏层编码和输出误差平方和的变化一致

泛化、过度拟合和停止判据

- 权值更新算法的终止条件
  - 一种选择是，对训练样例的误差降低至某个预先定义的阈值之下
    - 这不是一个好的策略，因为反向传播算法容易过度拟合训练样例，降低对于其他未见实例的泛化精度
- 泛化精度：网络拟合训练数据外的实例的精度
- 必须小心不要过早停止训练



尽管在训练样例上的误差持续下降，但在验证样例上测量到的误差先下降，后上升。

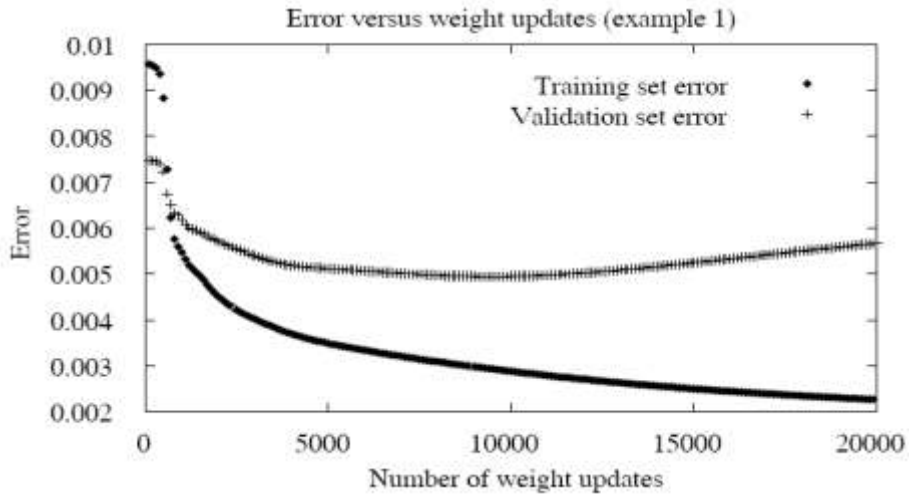
因为这些权值拟合了训练样例的“特异性”，而这个特异性对于样例的一般分布没有代表性。ANN中大量的权值参数为拟合这样的“特异性”提供了很大的自由度

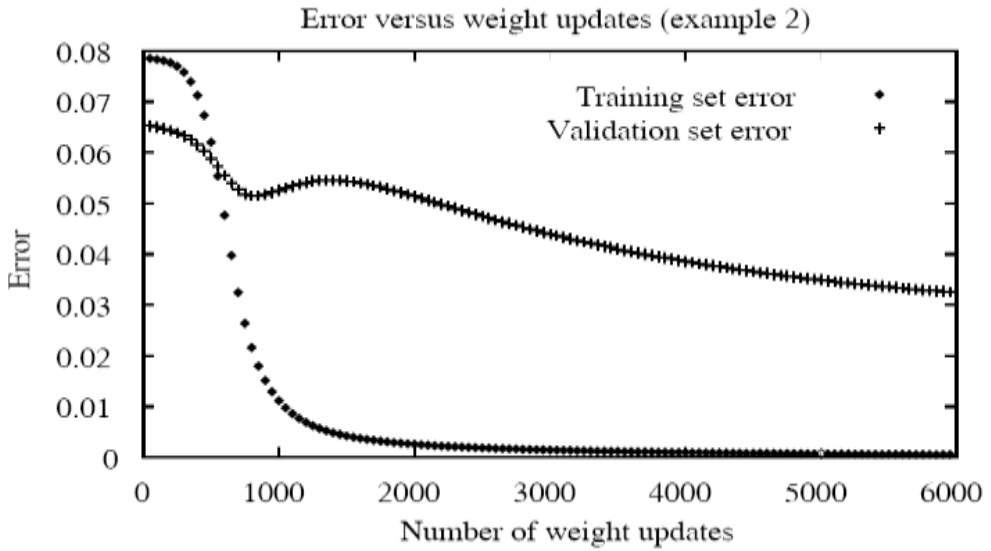
过度拟合

- 为什么过度拟合发生在迭代的后期，而不是早期?
  - 设想网络的权值是被初始化为小随机值的，使用这些几乎一样的权值仅能描述非常平滑的决策面
  - 随着训练的进行，一些权值开始增长，以降低在训练数据上的误差，同时学习到的决策面的复杂度也在增加
  - 如果权值调整迭代次数足够多，反向传播算法可能会产生过度复杂的决策面，拟合了训练数据中的噪声和训练样例中没有代表性的特征

过度拟合解决方法

- 权值衰减
  - 它在每次迭代过程中以某个小因子降低每个权值，这等效于修改E的定义，加入一个与网络权值的总量相应的惩罚项，此方法的动机是保持权值较小，从而使学习过程向着复杂决策面的反方向偏置
- 验证数据
  - 一个最成功的方法是在训练数据外再为算法提供一套验证数据，应该使用在验证集合上产生最小误差的迭代次数，不是总能明显地确定验证集合何时达到最小误差


Error versus weight updates (example 1)
Training set error
Validation set error
Error
Number of weight updates
0.01
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0
5000
10000
15000
20000

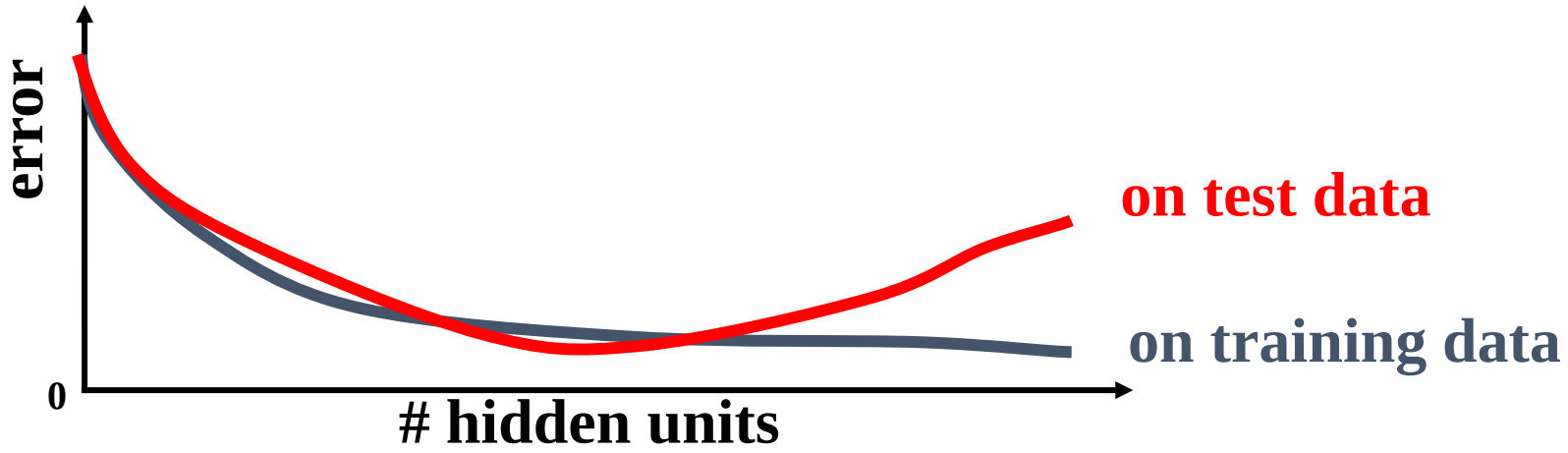

Error versus weight updates (example 2)
Training set error
Validation set error
Error
Number of weight updates
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
1000
2000
3000
4000
5000
6000

过度拟合解决方法（2）

- 一般而言，过度拟合是一个棘手的问题
- 交叉验证方法在可获得额外的数据提供验证集合时工作得很好，但是小训练集合的过度拟合问题更为严重
- k-fold交叉验证
  - 把训练样例分成k份，然后进行k次交叉验证过程，每次使用不同的一份作为验证集合，其余k-1份合并作为训练集合。
  - 每个样例会在一次实验中被用作验证样例，在k-1次实验中被用作训练样例
  - 每次实验中，使用上面讨论的交叉验证过程来决定在验证集合上取得最佳性能的迭代次数，然后计算这些迭代次数的均值 $\bar{i}$
  - 最后，运行一次反向传播算法，训练所有m个实例并迭代 $\bar{i}$ 次

隐层单元个数的确定

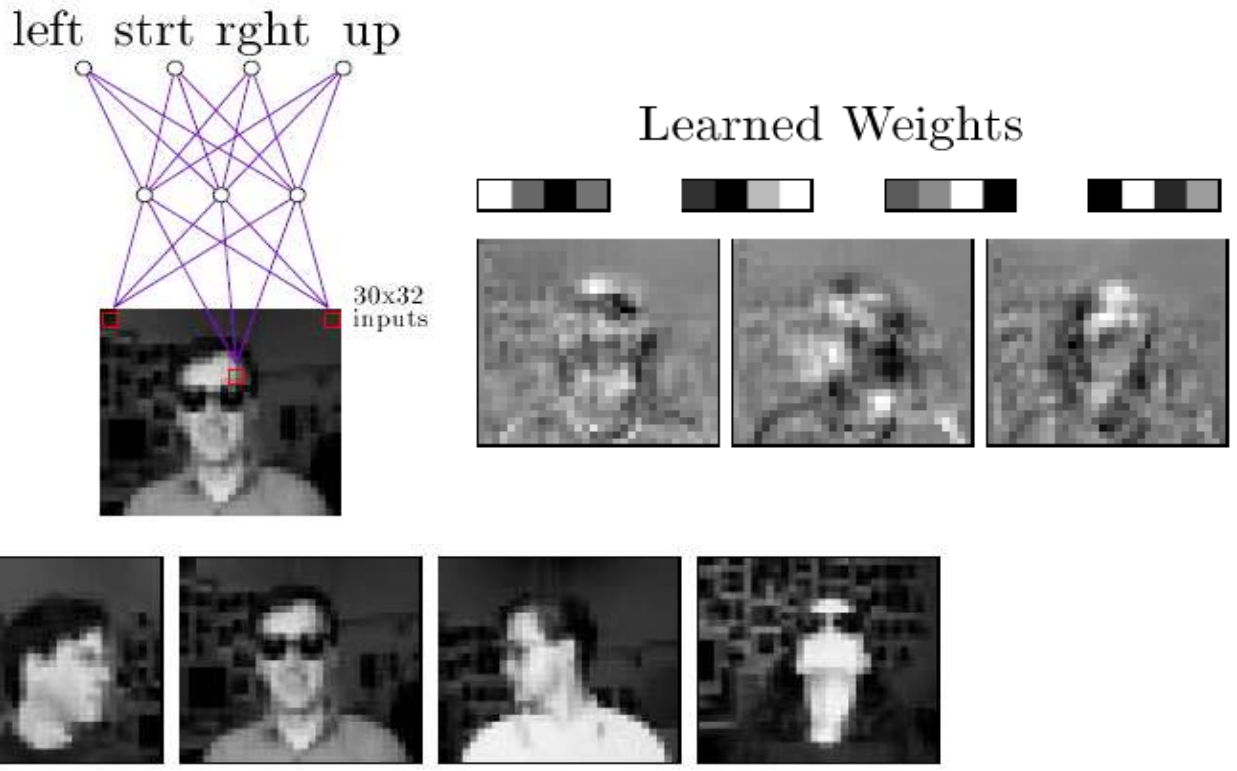
- 较少的隐层单元可防止网络过度拟合数据



- 应用交叉验证方法确定隐层单元数

举例：人脸识别

- 训练样例
  - 20个不同人的摄影图像
  - 每个人大约32张图像
    - 不同的表情
      - 快乐、沮丧、愤怒、中性
    - 不同的方向
      - 左、右、正前、上
    - 不同的穿戴
      - 是否带眼镜
  - 共624幅灰度图像
    - 分辨率为120×128，每个像素使用0（黑）到255（白）的灰度值描述
- 任务：学习图像中人脸的朝向



共搜集624幅灰度图像，训练了240幅图像样例后，独立测试集合的精度达90%，20个不同的人

人脸识别——设计要素

- 输入编码
  - ANN的输入必然是图像的某种表示，那么设计的关键是如何编码这幅图像
  - 例如，可以对图像进行预处理，分解出边缘、亮度一致的区域或其他局部图像特征，然后把这些特征输入网络，问题是导致每幅图像有不同数量的**特征参数**，而ANN具有固定数量的输入单元
  - 把图像编码成固定的30×32像素的亮度值，每个像素对应一个网络输入，**把范围是0到255的亮度值按比例线性缩放到0到1的区间内**，以使网络输入和隐层单元、输出单元在同样的区间取值。

人脸识别——设计要素（2）

- 输出编码
  - ANN必须输出4个值中的一个来表示输入图像中人脸的朝向
  - 可以使用单一的输出单元来编码这4种情况
  - 这里使用4个不同的输出单元，每一个对应4种可能朝向中的一种，取具有最高值的输出作为网络的预测值。称为1-of-n输出编码
- 选择1-of-n的原因
  - 为网络表示目标函数提供了更大的自由度
  - 最高值输出和次高值输出间的差异可以作为对网络预测的置信度

人脸识别——设计要素（3）

- 输出单元的目标值
  - 一个显而易见的方法，<1,0,0,0>...
  - 这里使用的方法，<0.9,0.1,0.1,0.1>...
  - 避免使用0和1作为目标值的原因
    - sigmoid单元对于有限权值不能产生这样的输出
    - 如果企图训练网络来准确匹配目标值0和1，梯度下降将会迫使权值无限增长
    - 0.1和0.9是sigmoid单元在有限权值情况下可以完成的

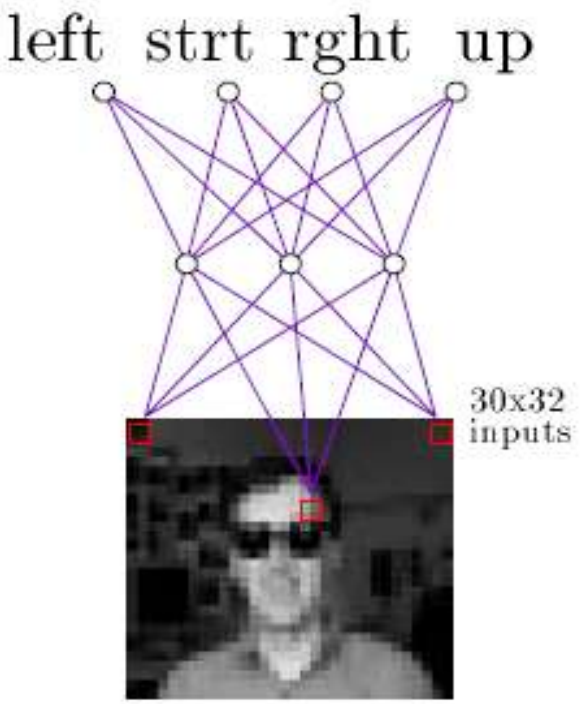
人脸识别——设计要素（4）

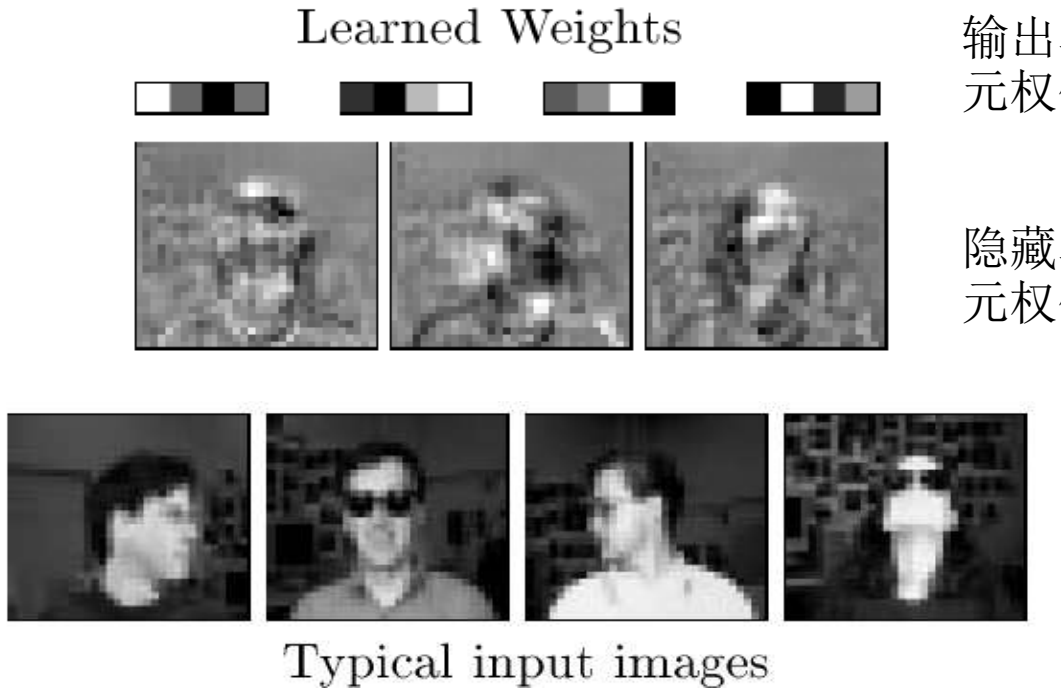
- 网络结构图
  - 网络包含多少个单元以及如何互连?
  - 最普遍的结构是分层网络，一层的每个单元向前连接到下一层的每一个单元
  - 目前采用了包含两层sigmoid单元的标准结构
  - 隐藏单元的数量
    - 3个，达到90%的精度，训练时间约5分钟
    - 30个，提高1~2个百分点，训练时间约1个小时
  - 实践发现，需要某个最小数量的隐藏单元来精确地学习目标函数，并且超过这个数量的多余的隐藏单元不会显著地提高泛化精度
  - 如果没有使用交叉验证，那么增加隐藏单元数量经常会增加过度拟合训练数据的倾向，从而降低泛化精度

人脸识别——设计要素（5）

- 学习算法的其他参数
  - 学习速率设定为0.3，冲量设定为0.3
  - 太小会产生大体相当的泛化精度，但需要更长的训练时间
  - 太大，训练将不能收敛到一个具有可接受误差的网络
  - 使用完全的梯度下降
  - 输出单元的权值被初始化为小的随机值
  - 输入单元的权值被初始化为0
  - 训练的迭代次数的选择可以通过分割可用的数据为训练集合和验证集合来实现
  - 最终选择的网络是对验证集合精度最高的网络
  - 最终报告的精度是在没有对训练产生任何影响的第三个集合——测试集合上测量得到的

学习到的隐藏层表示





输出单元权值

隐藏单元权值

以向右看为例说明隐藏层表示

学习到的隐藏层表示

- 图中紧挨人脸图像下的4个矩形，每个矩形描绘了网络中4个输出单元中的一个权值，每个矩形中的4个小方形表示和这个输出单元关联的4个权值
- 隐藏单元的权值显示在输出单元的下边，每个隐藏单元接受所有30×32个像素输入。与这些输入关联的30×32个权值被显示在它们对应的像素的位置
- 针对每一个训练样例，梯度下降迭代100次后的网络权值显示在图的下部。

人工神经的其他课题

其它可选的误差函数

- 为权值增加一个**惩罚项**
  - 把一个随着权向量幅度增长的项加入到E中，这导致梯度下降搜寻较小的权值向量，从而减小过度拟合的风险，等价于使用权衰减策略

$$E(\vec{w}) = \frac{1}{2}\sum_{d\in D}\sum_{k\in outputs}(t_{kd} - o_{kd})^2 + \gamma\sum_{i,j} w_{ji}^2$$

- 对误差增加一项**目标函数的斜率或导数**
  - 某些情况下，训练信息中不仅有目标值，而且还有关于目标函数的导数

$$E(\vec{w}) = \frac{1}{2}\sum_{d\in D}\sum_{k\in outputs}\left[(t_{kd} - o_{kd})^2 + \mu\sum_{j\in inputs}\left(\frac{\partial t_{kd}}{\partial x_d^j} - \frac{\partial o_{kd}}{\partial x_d^j}\right)^2\right]$$

其它可选的误差函数（2）

- 使网络对目标值的交叉熵最小化
  - 比如根据借贷申请者的年龄和存款余额，预测他是否会还贷，目标函数最好以申请者还贷的概率的形式输出，而不是输出明确的0和1。在这种情况下，可以证明最小化交叉熵的网络可以给出最好的概率估计。交叉熵定义如下：

$$-\sum_{d\in D} t_d \log o_d + (1-t_d)\log(1-o_d)$$

  - 后续章节将讨论何时及为什么最可能的网络假设就是使交叉熵最小化的假设，并推导了相应的sigmoid单元的梯度下降权值调整法则，还描述了在什么条件下最可能的假设就是使误差平方和最小化的假设。

其他可选的误差函数（3）

- 通过权值共享改变有效误差函数
  - 把与不同单元或输入相关联的权“捆绑在一起”，强迫不同的网络权值取一致的值，通常是为了实施人类设计者事先知道的某个约束
  - 约束了假设的潜在空间，减小了过度拟合的风险
  - 实现方法，首先在共享权值的每个单元分别更新各个权值，然后取这些权值的平均，再用这个平均值替换每个需要共享的权值。
  - 被共享的权值比没有共享的权值更有效地适应一个不同的误差函数

其它可选的误差最小化过程

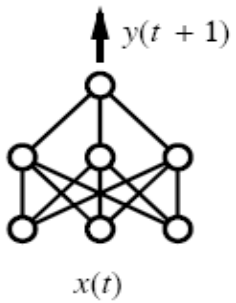
- **梯度下降**是搜寻使误差函数最小化的假设的最通用的方法之一，但不是最高效的
- 不妨把权值更新方法看作是要决定这样两个问题：
  - 选择一个改变当前**权值向量的方向**（梯度的负值）
  - 选择要**移动的距离**（学习速率）
- **线搜索**，每当选定了一条确定权值更新方向的路线，那么权更新的距离是通过沿这条线**寻找误差函数的最小值**来选择的
- **共轭梯度**，进行一系列线搜索来搜索误差曲面的最小值，这一系列搜索的第一步仍然使用梯度的反方向，在后来的每一步中，选择**使误差梯度分量刚好为0并保持为0的方向**
- 像共轭梯度这样的方法对最终网络的泛化误差没有明显的影响，唯一可能的影响是，不同的误差最小化过程会陷入不同的局部最小值

递归网络

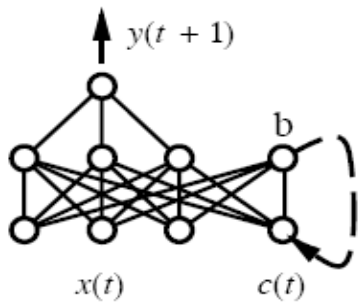
- 递归网络是有如下特征的人工神经网络
  - 适用于时序数据
  - 使用网络单元在时间t的输出作为其它单元在时间t+1的输入
  - 递归网络支持在网络中使用某种形式的有向环
- 考虑一个时序预测任务
  - 根据当天的经济指标x(t)，预测下一天的股票平均市值y(t+1)
  - 训练一个前馈网络预测输出y(t+1)，图4-11a

递归网络（2）

- 考虑一个时序预测任务
  - 根据当天的经济指标x(t)，预测下一天的股票平均市值y(t+1)
  - 训练一个前馈网络预测输出y(t+1)，图4-11a
- 预测y(t+1)时，考虑任意过去的时间窗内的信息，图4-11b
- 图4-11b那样的递归网络可以使用反向传播算法的简单变体来训练



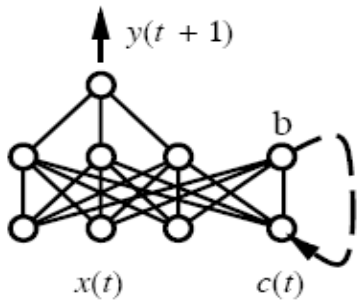
(a) Feedforward network



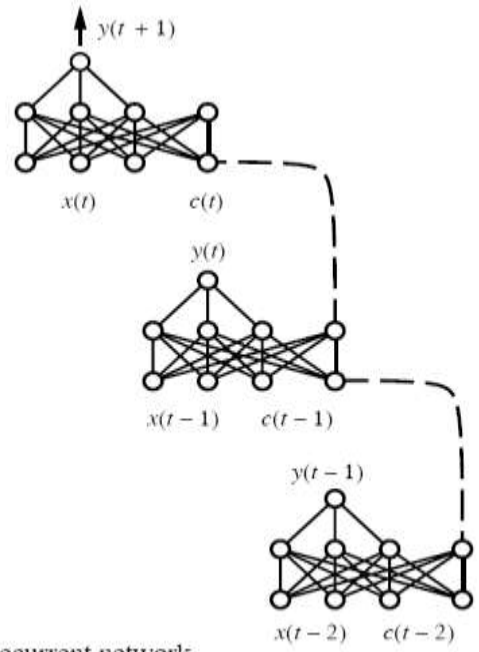
(b) Recurrent network

递归网络（3）

- 把递归网络拷贝成几份，**用不同拷贝间的连接替换掉反馈环，这个大的网络不再包含回路**，所以可以直接使用反向传播算法来训练
- 实践中，我们仅保留一份递归网络和权值集合的拷贝，在训练了展开的网络后，**可以取不同拷贝中权值的平均值作为最终网络的对应的权值**
- 实践中，递归网络比没有反馈环的网络**更难以训练**，泛化的可靠性也不如后者，然而它们仍因**较强的表征力**而保持着重要性



(b) Recurrent network



(c) Recurrent network unfolded in time

动态修改网络结构

- 动态增长或压缩网络单元和单元间连接的数量
- 从一个不包含隐藏单元的网络开始，然后根据需要增加隐藏单元来增长网络，直到训练误差下降到某个可接受的水平
  - 级联相关算法，每当加入一个新的隐藏单元，它的输入包括所有原始的网络输入和已经存在的隐藏单元的输出，网络以这种方式增长，积聚隐藏单元，直到网络的残余误差下降到某个可接受的水平
  - 由于每一步仅有一层网络在被训练，级联相关算法显著减少了训练时间
  - 算法的一个实际困难是，因为算法可以无限制地增加单元，很容易过度拟合训练数据。

动态修改网络结构

- 从一个复杂的网络开始修剪掉某些连接
  - 判断某个权是否无关紧要的一种方法是看它的值**是否接近0**
  - 在实践中更加成功的方法是考虑这个权值的一个小的变化对误差的影响（**连接的显著性**）
  - 最不显著的连接被拆除，重复这个过程，直到遇到某个终止条件为止（**最优脑损伤法**）
- 一般而言，动态修改网络结构的方法能否稳定地提高反向传播算法的泛化精度还有待研究

小结（2）

- 梯度下降收敛到训练误差相对网络权值的局部极小值。只要训练误差是假设参数的可微函数，梯度下降可用来搜索很多连续参数构成的假设空间
- 反向传播算法能够创造出网络输入中没有明确出现的特征。
- 交叉验证方法可以用来估计梯度下降搜索的合适终止点，从而最小化过度拟合的风险
- 其他ANN学习算法，递归网络方法训练包含有向环的网络，级联相关算法改变权和网络结构

补充读物

- 其他与ANN学习相关的内容
  - 贝叶斯学习给出了选择最小化误差平方和的贝叶斯论证，以及在某些情况下，用最小化交叉熵代替最小化误差平方和的方法
  - 计算学习讨论了为可靠学习布尔函数所需要的训练实例数量的理论结果，以及某些类型网络的VC维
  - 评估假设中讨论了过度拟合和避免过度拟合的方法
  - 归纳和分析学习也讨论了使用以前知识来提高泛化精度的方法

补充读物（2）

- 发展历程
  - McCulloch & Pitts
  - Widrow & Hoff
  - Rosenblatt
  - Minsky & Papert
  - Rumelhart & McClelland; Parker
- 教科书
  - Duda & Hart
  - Windrow & Stearns
  - Rumelhart & McClelland