深度学习

之ANN的自我学习

Author:WangQi
Email:wangqikaixin@gmail.com

“学习：透过外界教授或从自身经验提高能力的过程。学习必须专心致志并持之以恒。”

–Wikipedia

Author:WangQi
Email:wangqikaixin@gmail.com

概览

1. 神经元的自学习。

2. 无约束最优化。

3. 最速下降法。

4. 牛顿法。

5. 最速下降法与牛顿法。

6. 人工神经网络的自学习。

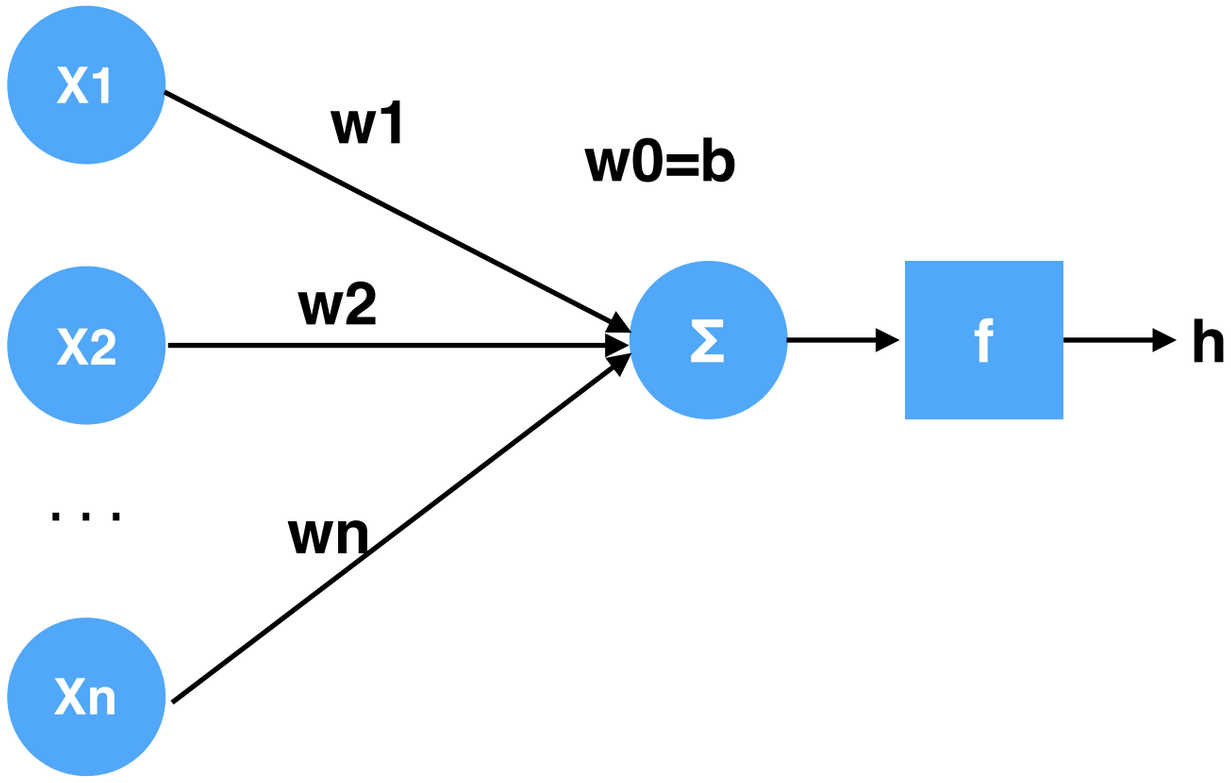
7. 过拟合与欠拟合。

Author:WangQi
Email:wangqikaixin@gmail.com

1. 神经元的自学习

Author:WangQi
Email:wangqikaixin@gmail.com

线性神经元



$$f(z)=z$$

$$h=\sum_{i=0}^{n} w_i x_i$$

当一个信号S输入神经元时，伴随着噪声信号V共同传入，即输入信号X=(S+V)，我们希望输出的信号包含尽量少的噪声，即我们希望神经元可以进行自适应滤波，思考如何去做?

Author:WangQi
Email:wangqikaixin@gmail.com

LMS算法

最小均方算法（Least Mean Square，LMS）是最早解决如预测和信道均等化问题的自适应滤波算法。

$$e = y - h \qquad J = \frac{1}{2}e^2 \longrightarrow \min J = \frac{1}{2}e^2$$

误差　　代价函数　　最小化代价函数

Author:WangQi
Email:wangqikaixin@gmail.com

线性神经元的自适应滤波

- 神经元对输入信号x产生一个输出h。
- 计算期望信号y与输出信号h之间的误差e。
- 根据误差对神经元参数进行自动调整。

调整方法：使用最优化法逼近代价函数极小值。

Author:WangQi
Email:wangqikaixin@gmail.com

线性神经元的最优化目标

假设函数：$$h\left(x\right)=\sum_{i=0}^{n}w_{i}x_{i}$$

代价函数：$$J\left(W\right)=\frac{1}{2}e^{2}=\frac{1}{2}\left(y-h\right)^{2}=\frac{1}{2}\left(y-\sum_{i=0}^{n}x_{i}w_{i}\right)^{2}$$

目标：$$W=\arg\min J$$

注意：假设函数自变量为样本输入**X**，代价函数的自变量为参数**W**。

Author:WangQi
Email:wangqikaixin@gmail.com

2. 无约束最优化

Author:WangQi
Email:wangqikaixin@gmail.com

无约束最优化

考虑代价函数J(**W**)，它是对未知参数向量**W**连续可微的函数。我们需要选择适当的权值向量**W**使得最小化代价函数J(**W**)。

即希望找到一个最优解$\mathbf{W}^*$，满足：$J(W^*) \leq J(W)$

最优性的必要条件是：$\nabla J(W^*) = 0$

Author:WangQi
Email:wangqikaixin@gmail.com

局部迭代下降

一类特别适合自适应滤波器设计的无约束最优化法是以局部迭代下降思想为基础的：

1. 随机设置一个初始参数值**W**(0)。

2. 产生一系列的参数值**W**(1)，**W**(2)，…，使得：

$J(W(n+1)) < J(W(n))$　直到算法收敛到最优解**W***

注意：n表示步数，不是参数。

Author:WangQi
Email:wangqikaixin@gmail.com

3. 最速下降法

Author:WangQi
Email:wangqikaixin@gmail.com

最速下降法

最速下降法（Steepest Descent），也称为**梯度下降法（Gradient Descent）**，是一个一阶最优化算法。通过对函数某一点对应的梯度的反方向以规定步长距离进行迭代搜索以找到函数的局部极小值。

Author:WangQi
Email:wangqikaixin@gmail.com

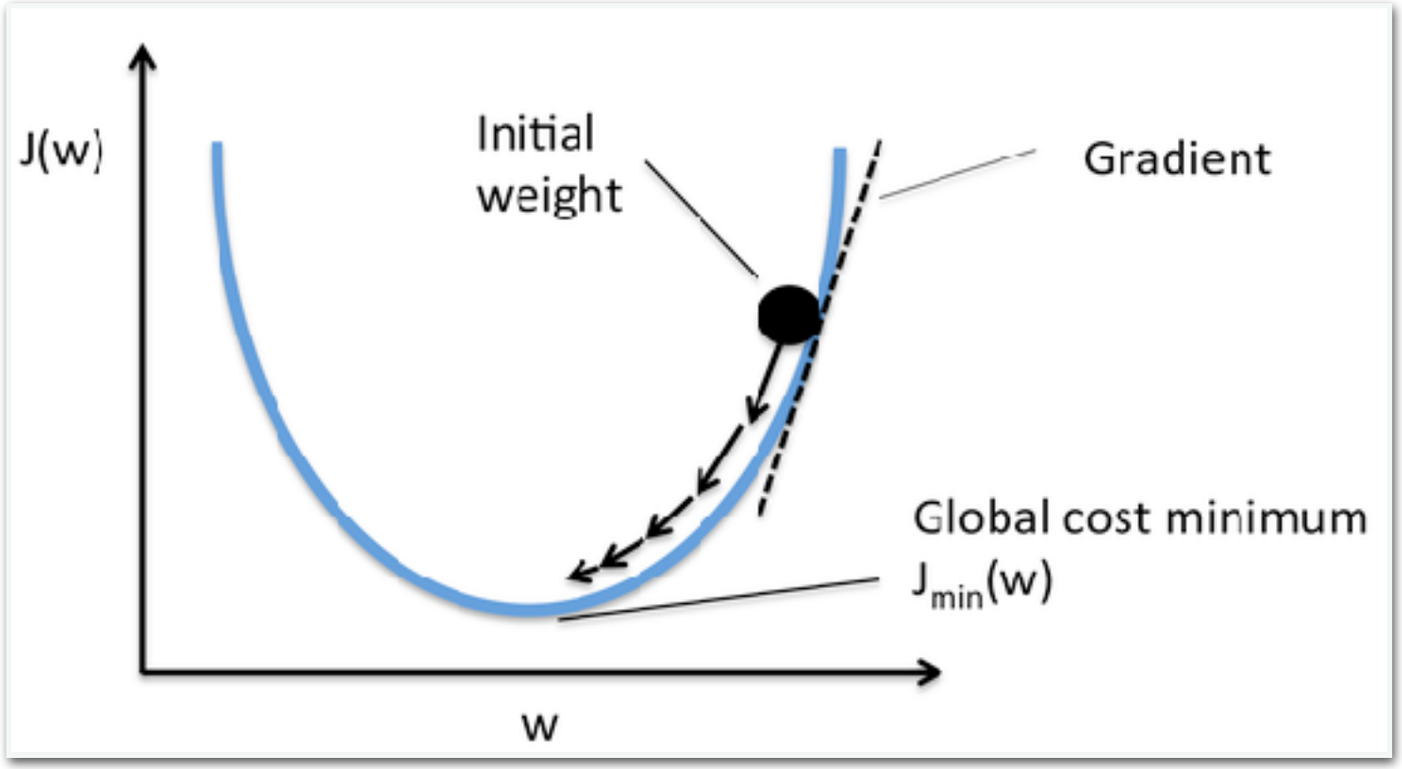
最速下降法

迭代规则： $W(n+1)=W(n)-\alpha\nabla J(W)$

其中 α 一般被称为步长（Stepsize）或学习率（Learning-rate），用来控制迭代速率。

Author:WangQi
Email:wangqikaixin@gmail.com

最速下降法



导数符号的方向就是下降的反方向

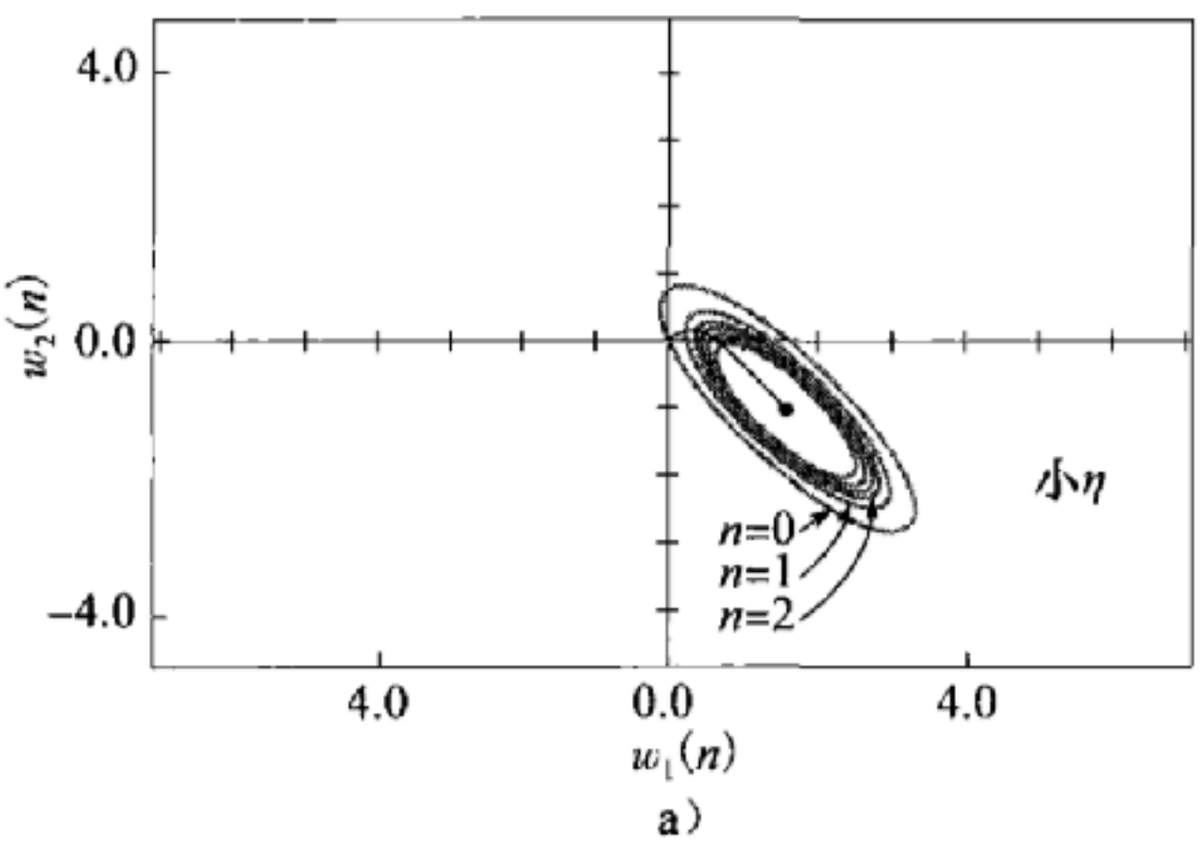
越接近极小值迭代步长越小

迭代的步长太大可能导致无法下降

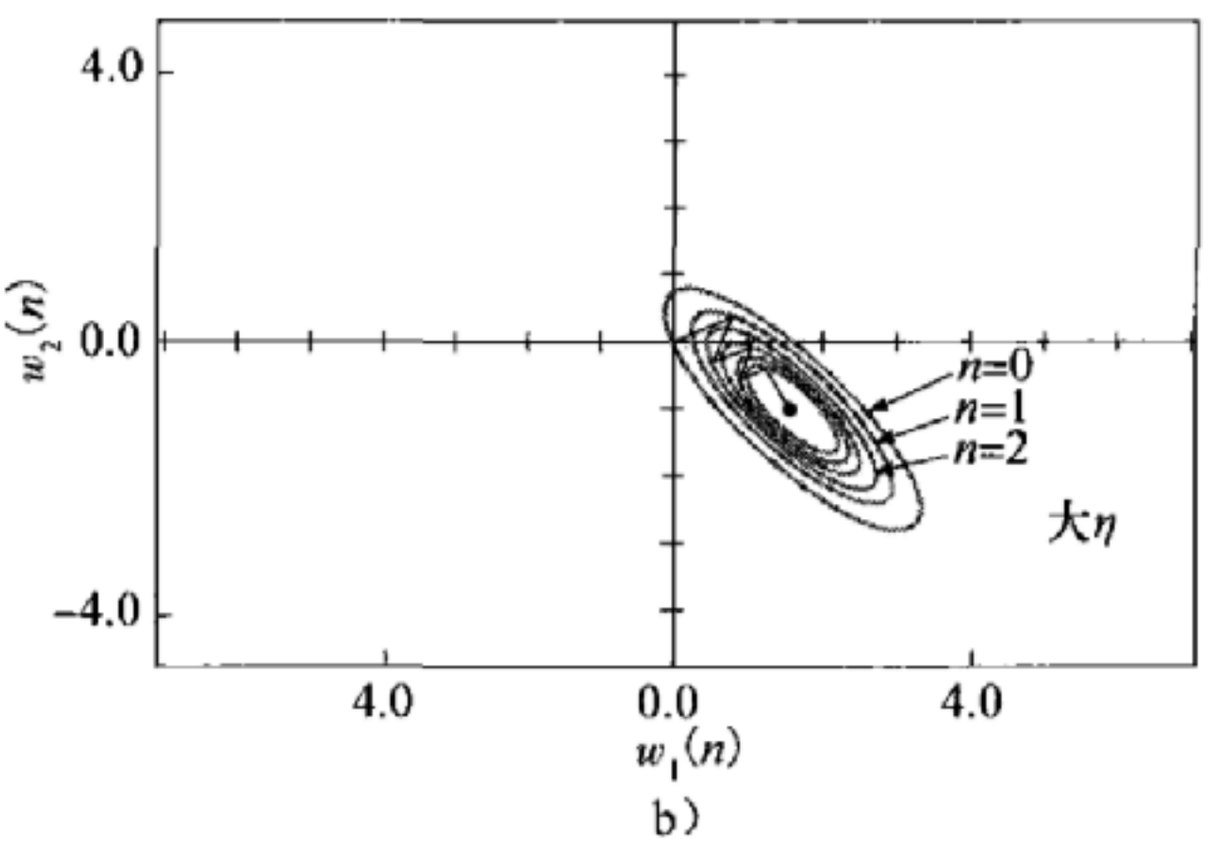
往往只能得到极小值，而非最小值

Author:WangQi
Email:wangqikaixin@gmail.com

学习率



a)



b)

Author:WangQi
Email:wangqikaixin@gmail.com

线性神经元自适应滤波

假设函数：$$h(x)=\sum_{i=0}^{n}w_i x_i$$

代价函数：$$J(W)=\frac{1}{2}\left(y-\sum_{i=0}^{n}x_i w_i\right)^2=\frac{1}{2}\left(y-W^T X\right)$$

梯度：$$\frac{\partial J}{\partial W_i}=-X_i\left(y-W^T X\right)$$

参数更新：$$W_i=W_i-X_i\left(y-W^T X\right)$$

Author:WangQi
Email:wangqikaixin@gmail.com

线性神经元的自适应滤波

LMS算法流程

训练样本：输入信号向量$=\mathbf{x}(n)$

期望响应$=y(n)$

用户选择参数：α

初始化：设$\hat{\mathbf{w}}(0)=\mathbf{0}$

计算：当$n=1, 2, \cdots$，计算

$$e(n)=y(n)-\hat{\mathbf{w}}^T(n)\mathbf{x}(n)$$

$$\hat{\mathbf{w}}(n+1)=\hat{\mathbf{w}}(n)+\alpha\mathbf{x}(n)e(n)$$

Author:WangQi
Email:wangqikaixin@gmail.com

Python实现自适应滤波器

Author:WangQi
Email:wangqikaixin@gmail.com

4. 牛顿法

Author:WangQi
Email:wangqikaixin@gmail.com

牛顿法

牛顿法（Newton’s Method）是二阶最优化算法。基本思想是最小化代价函数在当前点周围的二阶近似值。

Author:WangQi
Email:wangqikaixin@gmail.com

牛顿法

迭代规则： $W(n+1)=W(n)-\mathrm{H}_{W(n)}^{-1}g_{W(n)}$

H表示海森矩阵（Hessian Matrix），是多元函数的二阶偏导数。g表示梯度。

Author:WangQi
Email:wangqikaixin@gmail.com

牛顿法推导

使用二次**泰勒级数**展开式逼近代价函数：

$$J(W) \approx J(C) + \nabla J(C) \cdot (W - C) + \frac{1}{2} \cdot (W - C)^T \cdot \nabla^2 J(C) \cdot (W - C)$$

C表示J上的一点**W**(n)，即使用**W**(n)以及其附近的二阶泰勒展开式来逼近J。

若使用g表示一阶偏导（梯度），H表示海森矩阵，则：

$$J(W) \approx J(C) + g_c \cdot (W - C) + \frac{1}{2} \cdot (W - C)^T \cdot H_c \cdot (W - C)$$

Author:WangQi
Email:wangqikaixin@gmail.com

牛顿法推导

令： $j(W)=J(C)+g_c\cdot(W-C)+\frac{1}{2}\cdot(W-C)^T\cdot H_c\cdot(W-C)$

即： $J(W)\approx j(W)$

求J的最值，需满足： $\nabla J(W^*)=0$

这里使： $\nabla j(W)=0$ **可以得到W的迭代法：**

$$W=C-\mathrm{H}_C^{-1}g_c \quad \textbf{即：} \quad W(n+1)=W(n)-\mathrm{H}_{W(n)}^{-1}g_{W(n)}$$

Author:WangQi
Email:wangqikaixin@gmail.com

5. 最速下降法与牛顿法

Author:WangQi
Email:wangqikaixin@gmail.com

在线学习

在线学习（online learning），即每次训练时使用一个样本计算误差。可用于追踪样本的微小改变。其训练速度快，但容易陷入局部极小值。梯度下降法中的**随机梯度下降**为一种在线学习的方式。

Author:WangQi
Email:wangqikaixin@gmail.com

批量学习

批量学习（batch learning），即每次训练模型时，均把所有样本全部输入，误差由平均误差定义。批量学习有着存储要求，面对大规模数据集时很难训练。**批量梯度下降法**是一种批量学习的方式。

Author:WangQi
Email:wangqikaixin@gmail.com

小批量学习

小批量学习（min-batch learning），即每次训练模型时，把一部分样本输入，误差由平均误差定义。与批量学习相比，其收敛速度快，与在线学习相比其不容易收敛到局部极小值。**小批量梯度下降法**是一种小批量学习的方式。通常我们采用小批量学习的方式去训练模型

Author:WangQi
Email:wangqikaixin@gmail.com

收敛准则

通常，不能证明反向传播算法是收敛的，并且没有明确定义的算法停止准则。

通常使用如下方法对是否收敛进行判断：

1. 当梯度向量的欧几里得范数达到一个充分小的阈值时。

2. 当迭代的每一个回合的均方误差变化的绝对速率足够小时。

Author:WangQi
Email:wangqikaixin@gmail.com

最速下降法的优缺点

- 计算复杂度低。
- 梯度方向下降速率高。
- 学习率不恰当可能导致下降速度缓慢或者不收敛（可设置学习率退火缓和）。

Author:WangQi
Email:wangqikaixin@gmail.com

牛顿法的优缺点

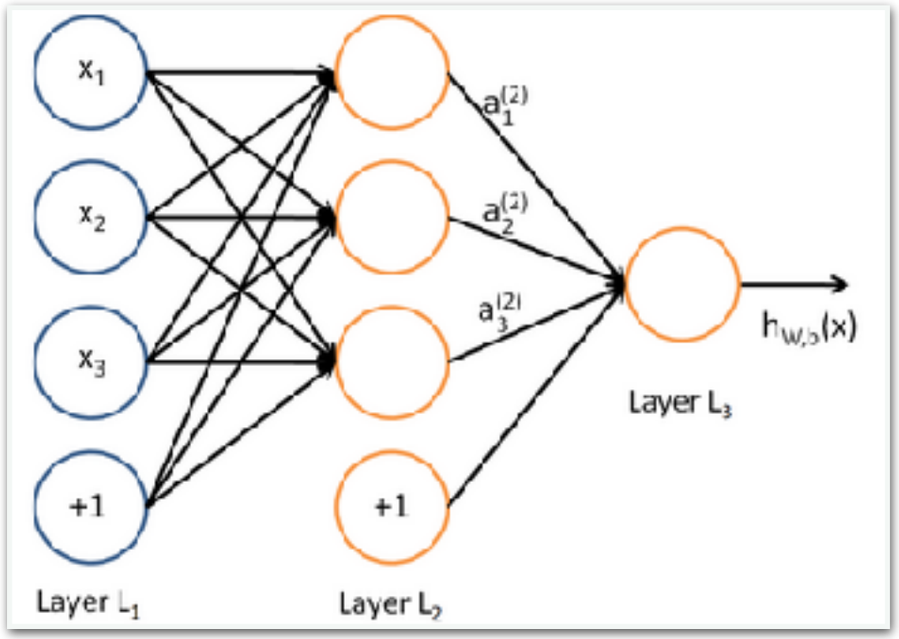
- 收敛速度快于最速下降法。
- 要求海森矩阵必须为正定矩阵。
- 计算复杂度高。
- 不能保证模型收敛或收敛到全局最小值。

Author:WangQi
Email:wangqikaixin@gmail.com

6. 神经网络自学习

ANN的本质

ANN的本质是复合函数。



$$a_1^{(2)} = f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)})$$

$$a_2^{(2)} = f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)})$$

$$a_3^{(2)} = f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)})$$

$$h_{W,b}(x) = a_1^{(3)} = f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)})$$

Author:WangQi
Email:wangqikaixin@gmail.com

ANN的输出误差定义

如果我们把神经网络用函数h(x)表示，用y表示我们期望神经网络的输出，那么我们可以使用均方误差来表示神经网络的输出误差，即代价函数（Cost function）：

$$J = \frac{1}{2m}\sum_{i=1}^{m} \| y^{(i)} - h_{w,b}(x^{(i)}) \|^2$$

目标：最小化代价函数

方法：反向传播算法

Author:WangQi
Email:wangqikaixin@gmail.com

参数更新规则

连接权重更新规则

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W,b)$$

偏置值更新规则

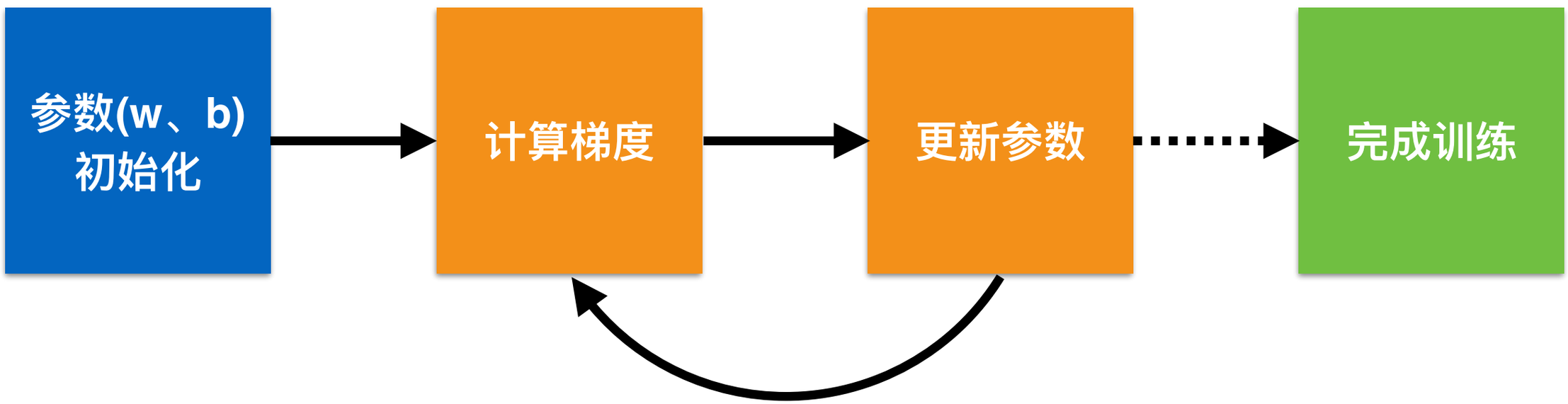
$$b_{i}^{(l)} = b_{i}^{(l)} - \alpha \frac{\partial}{\partial b_{i}^{(l)}} J(W,b)$$

两种变量的更新规则是一致，但求得的梯度公式不同。

Author:WangQi
Email:wangqikaixin@gmail.com

训练神经网络

神经网络的训练与LMS算法的训练类似，也是不断迭代更新参数，**训练过程如下：**



Author:WangQi
Email:wangqikaixin@gmail.com

参数初始化方法

1. 使用特定常数初始化。例如全0初始化。

不一定可行。例如使用ReLU神经元时，梯度均为0，无意义。

2. 使用随机数初始化

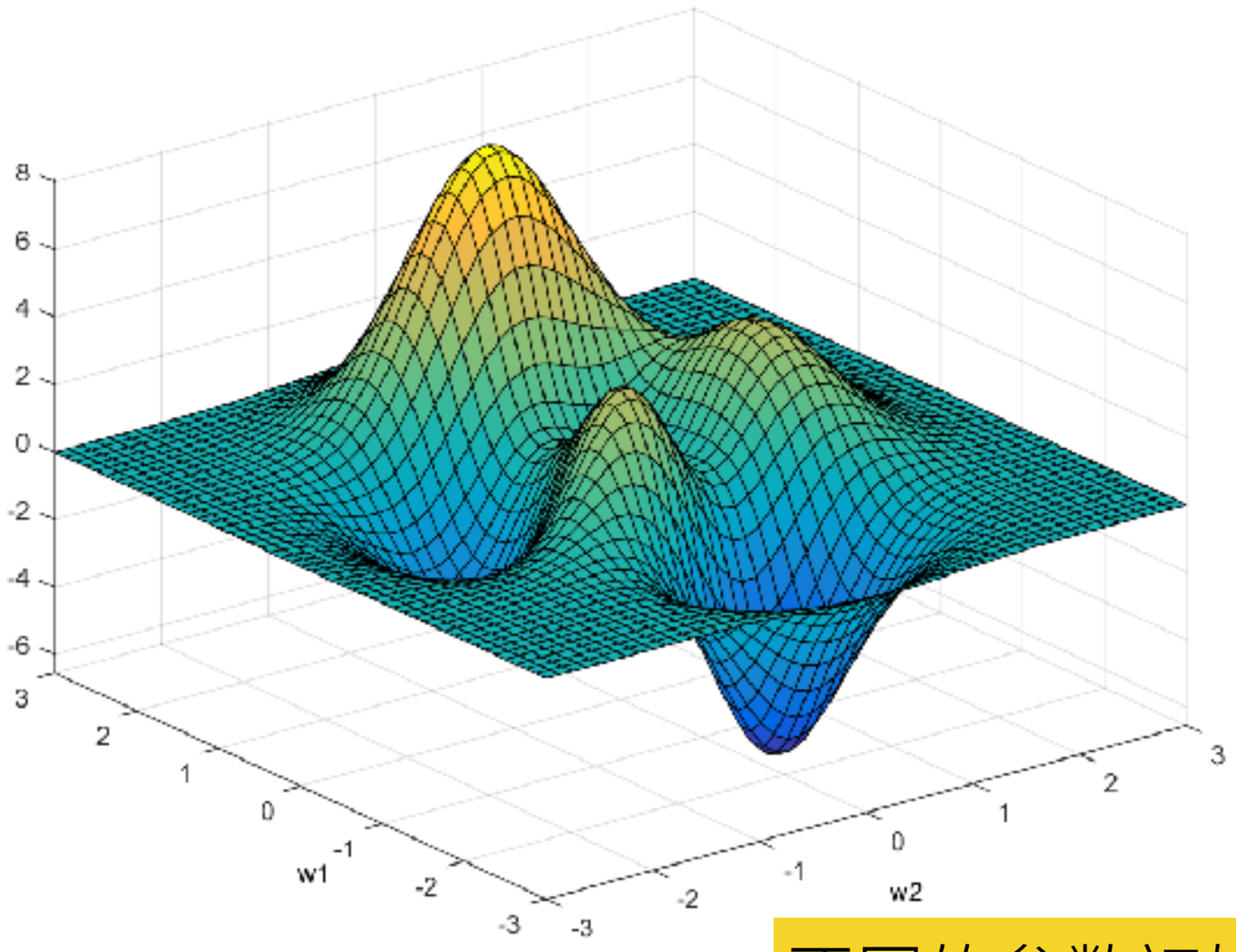
可行。但效果一般，即影响反向传播算法的运行效果。

3. 使用服从正态分布的数值初始化

可行。效果较好。其它优秀的方法如Xavier等。

Author:WangQi
Email:wangqikaixin@gmail.com

非凸函数优化与局部最小值



神经网络往往是非凸函数

梯度下降只能求得极小值

通常极小值近似于最小值

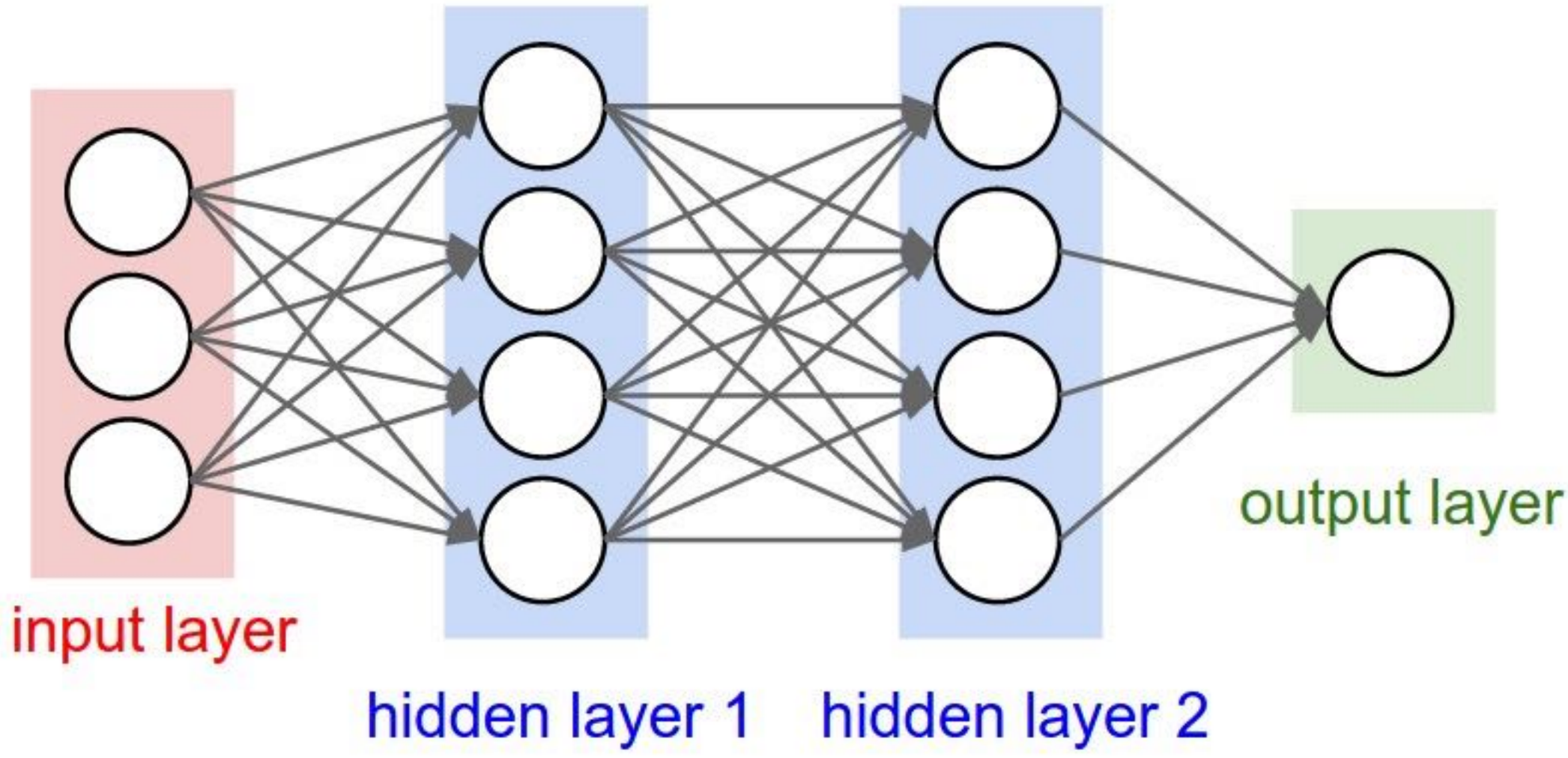
不同的参数初始化可以得到不同但近似的结果

Author:WangQi
Email:wangqikaixin@gmail.com

7. 过拟合与欠拟合

Author:WangQi
Email:wangqikaixin@gmail.com

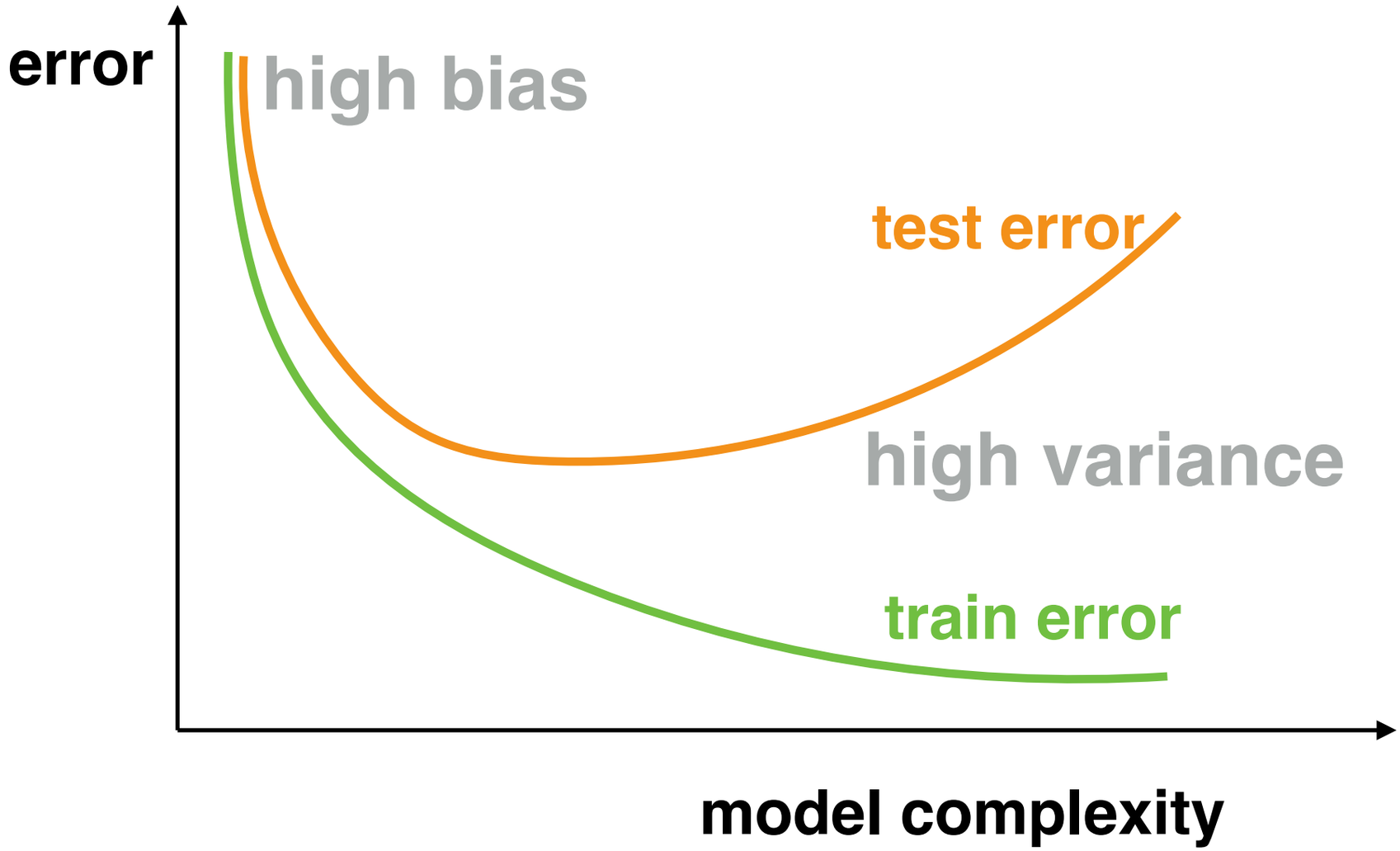
神经网络的参数数量



$$num\_of\_\arg s = \sum_{i=1}^{l-1} L_i \cdot L_{i+1} + \sum_{j=2}^{l} L_j$$

Author:WangQi
Email:wangqikaixin@gmail.com

偏差与方差

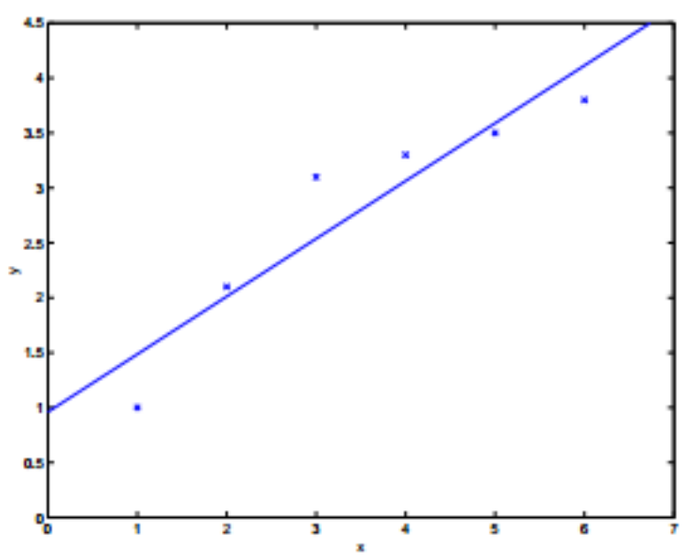


Author:WangQi
Email:wangqikaixin@gmail.com

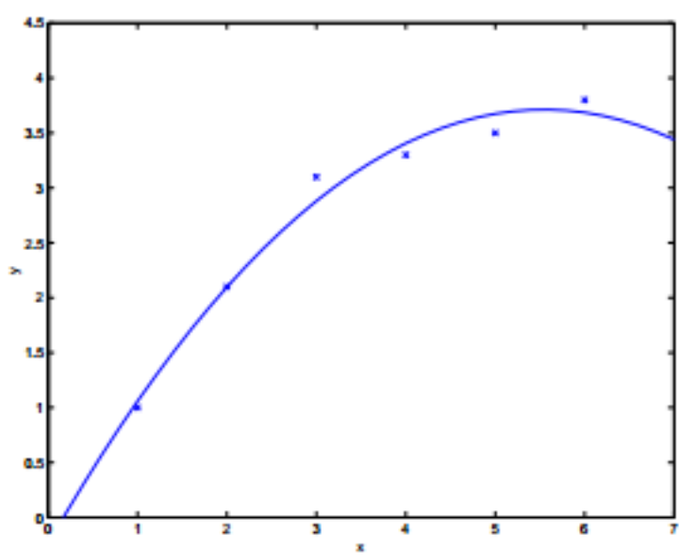
过拟合与欠拟合

神经网络的参数规模太小，可能导致欠拟合，即模型复杂度过低，不足以刻画样本的特征。

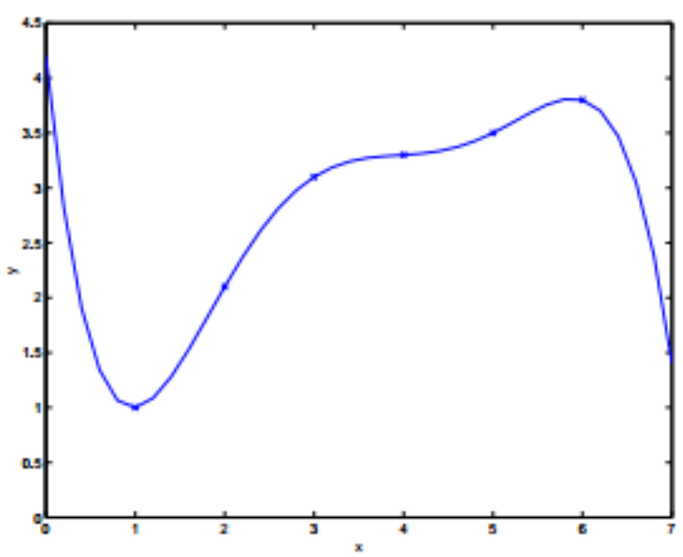
神经网络的参数规模太大，可能导致过拟合，即模型复杂度过高，模型学习到了噪声特征。



欠拟合



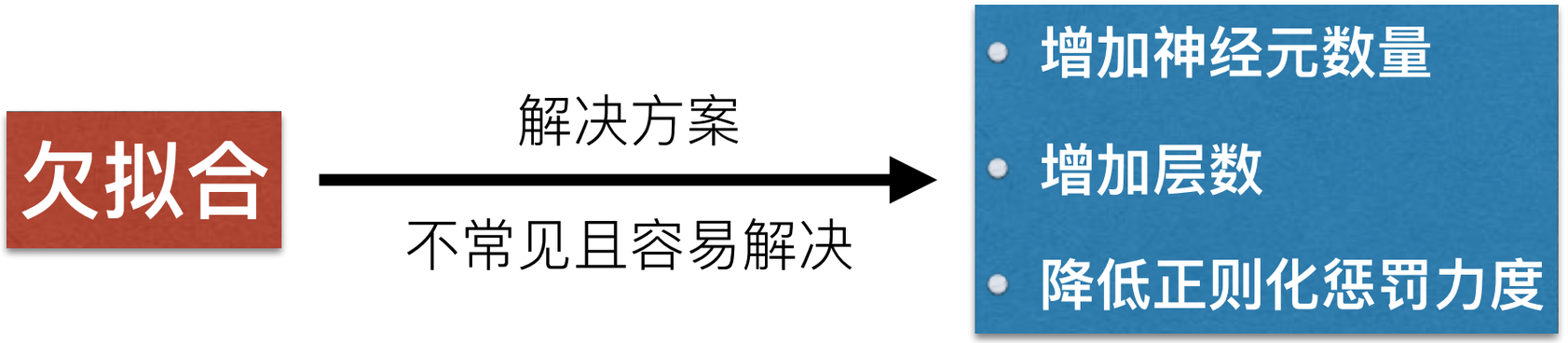
拟合

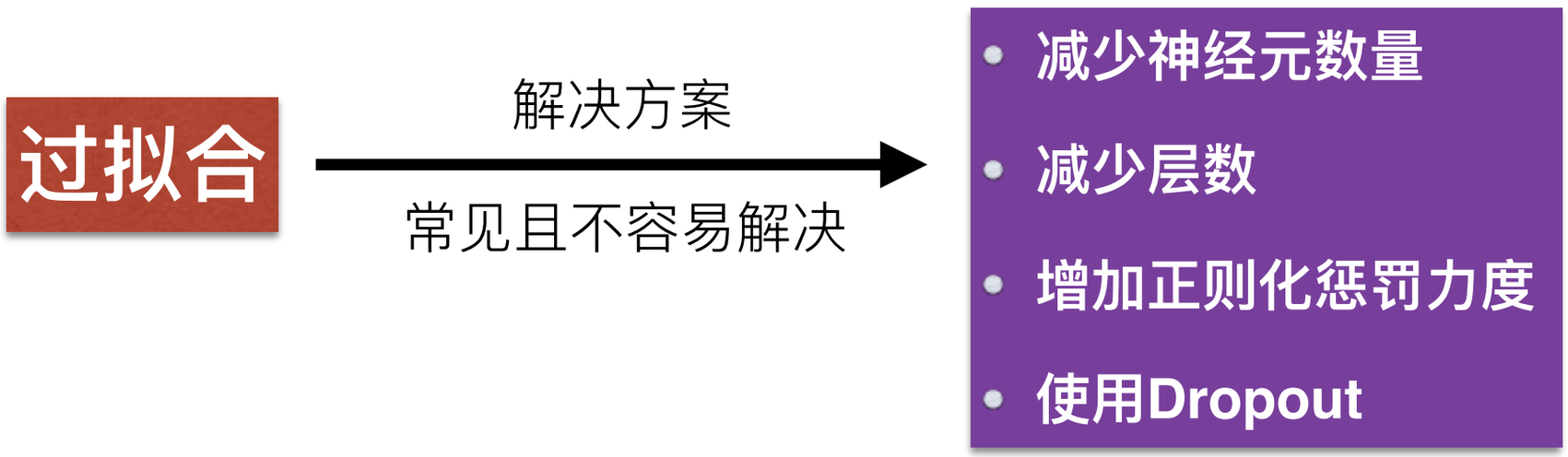


过拟合

Author:WangQi
Email:wangqikaixin@gmail.com

过拟合与欠拟合解决方案





Author:WangQi
Email:wangqikaixin@gmail.com

小节

- 自适应滤波器与LMS算法的训练过程。
- 最速下降法以及迭代流程。
- 牛顿法的推导以及迭代流程。
- 牛顿法与最速下降法的优缺点。
- 在线学习、批量学习、小批量学习的特点。
- 神经网络的误差定义与参数更新规则。
- 过拟合与欠拟合出现的原因与解决方法。

Author:WangQi
Email:wangqikaixin@gmail.com

THANKS

Author:WangQi
Email:wangqikaixin@gmail.com