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

监督学习

监督式学习（Supervised learning），是一个机器学习中的方法，可以由训练资料中学到或建立一个模式（函数 / learning model），并依此模式推测新的实例。训练资料是由输入物件（通常是向量）和预期输出所组成。函数的输出可以是一个连续的值（称为回归分析），或是预测一个分类标签（称作分类）。

Author:WangQi
Email:wangqikaixin@gmail.com

自学习目标

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

Author:WangQi
Email:wangqikaixin@gmail.com

自学习目标

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

最小化代价函数即需要找到一组参数（w，b）使代价函数最小。

假设函数预测 → 参数是w、b，变量是样本输入

代价函数最小化 → 参数是样本输入与标记，变量w、b

Author:WangQi
Email:wangqikaixin@gmail.com

如何求得J的最小值

1. 解析法：找到一个公式能求得极小值。

不可行。神经网络的参数数量巨大，几乎或根本无法找到解析解。

2. 最小二乘法求解。

通常不可行。线性模型可用正规方程求解，计算量大。无通用求解公式。

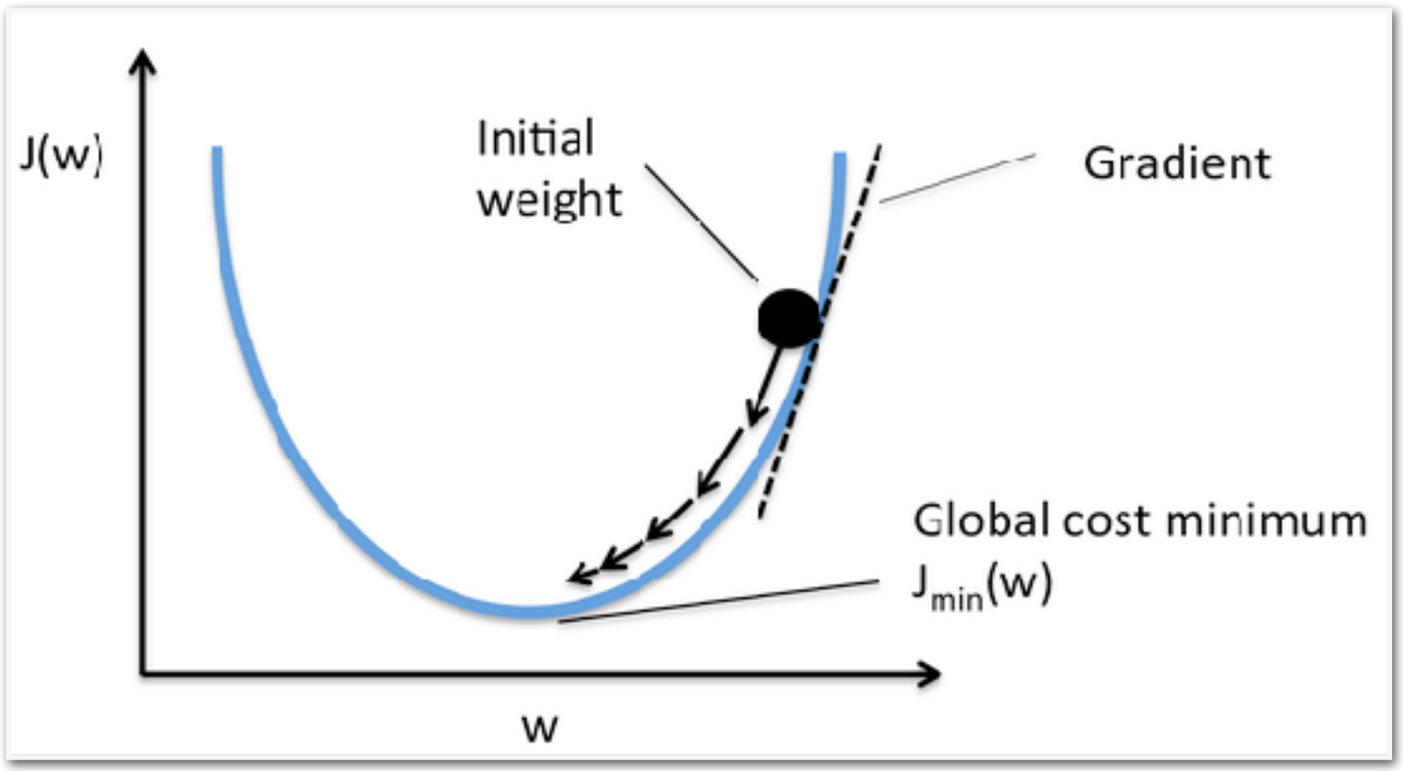
3. 最优化方法。

可行。常用梯度下降、牛顿法等方法。

Author:WangQi
Email:wangqikaixin@gmail.com

梯度下降法

梯度下降法(Gradient Descent, GD)是一个一阶最优化算法，通常也称为**最速下降法**。 其用负梯度方向为搜索方向进行迭代搜索。

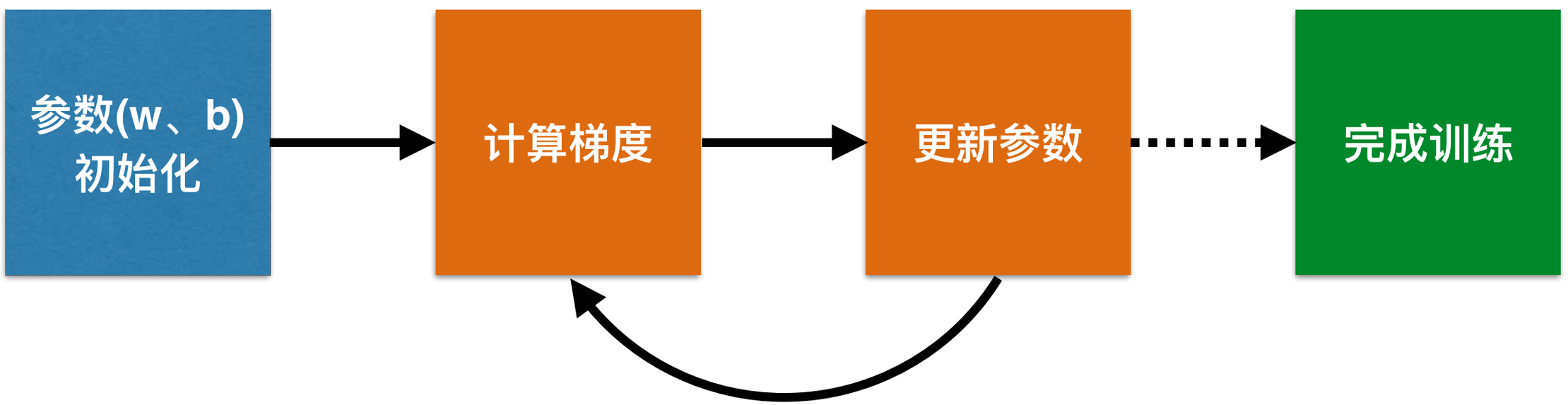


- 导数符号的方向就是下降的反方向
- 导数的大小对我们意义不大
- 越接近极小值迭代步长越小
- 迭代的步长太大可能导致无法下降
- 往往只能得到极小值，而非最小值

Email:wangqikaixin@gmail.com

训练神经网络

GD是一种依靠迭代运算逐步逼近局部极小值的算法，我们把迭代的过程叫做**训练（Train）**



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。

3. 使用服从正态分布的数值初始化

可行。并且效果较好。

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

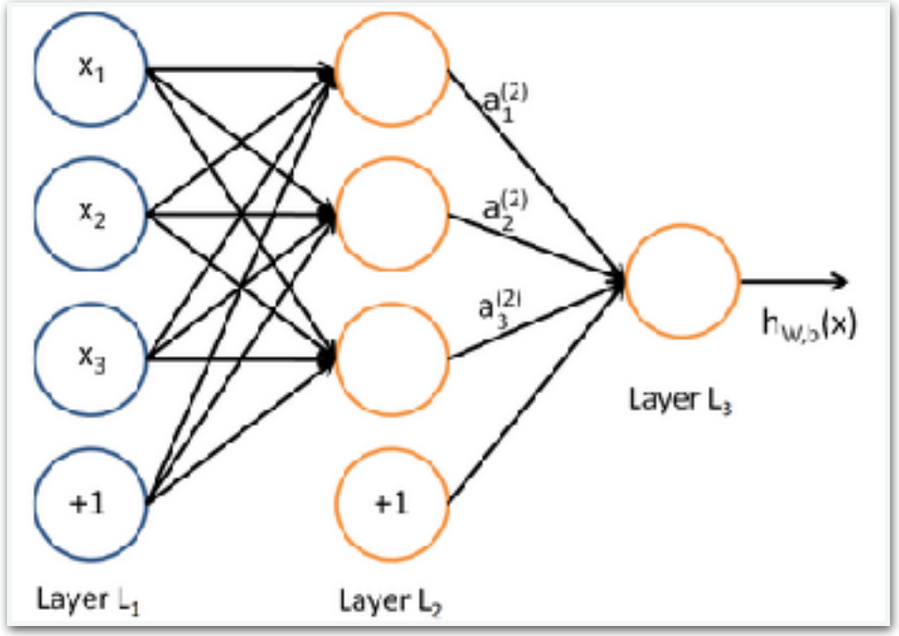
$$b_{i}^{(l)} = b_{i}^{(l)} - \alpha \frac{\partial}{\partial b_{i}^{(l)}} J(W,b)$$

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

求偏导数的方法

ANN的本质是复合函数，可以使用链式求导法则来求得参数的导数。



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

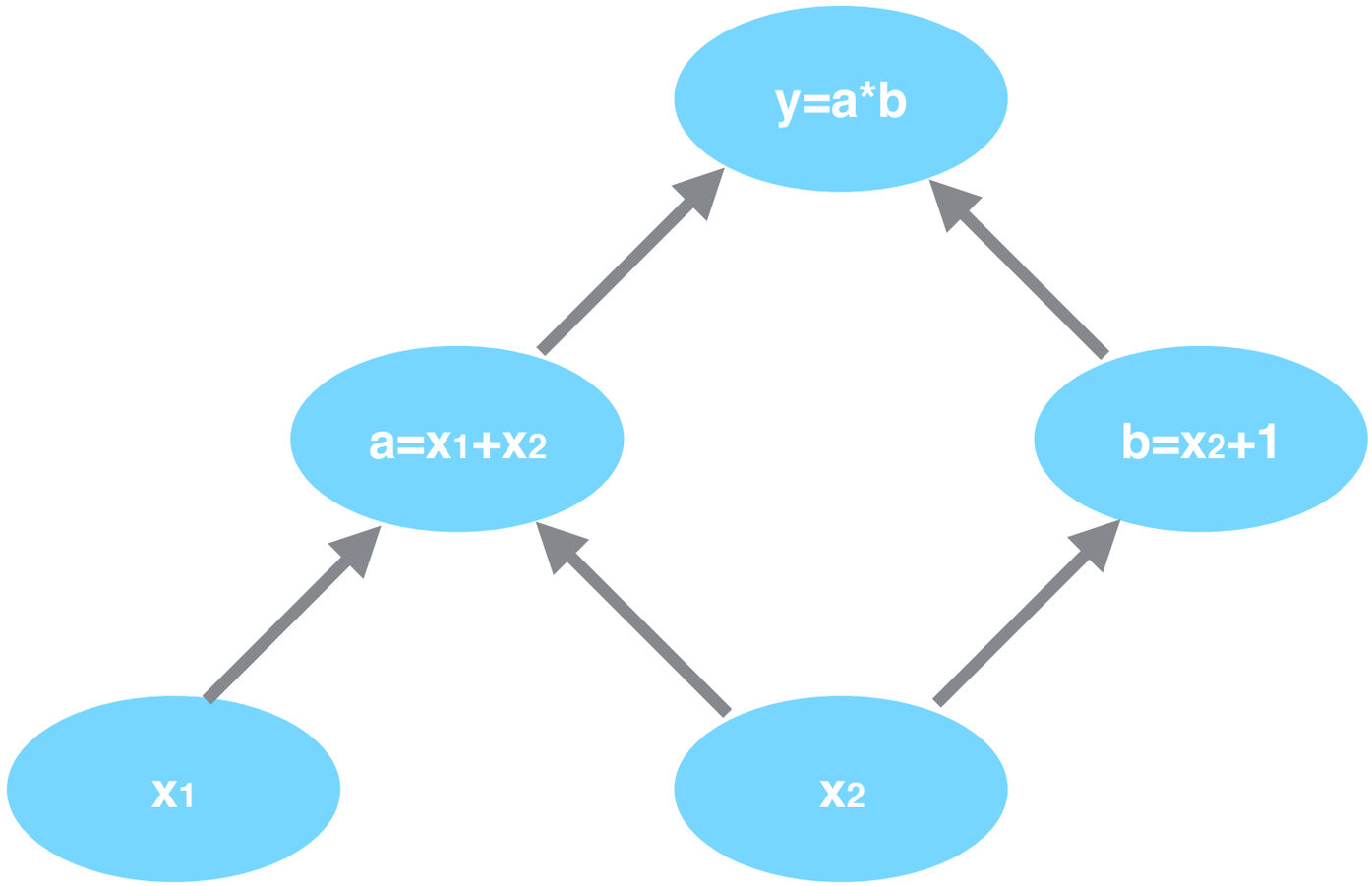
$$a_2^{(2)} = f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)})$$

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

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

Author:WangQi
Email:wangqikaixin@gmail.com

以求 $y=(x_1+x_2)\times(x_2+1)$ 的偏导数为例



将式子拆开如上图

对每条通路求偏导

$$\frac{\partial y}{\partial x_1}=\frac{\partial y}{\partial a}\cdot\frac{\partial a}{\partial x_1}$$

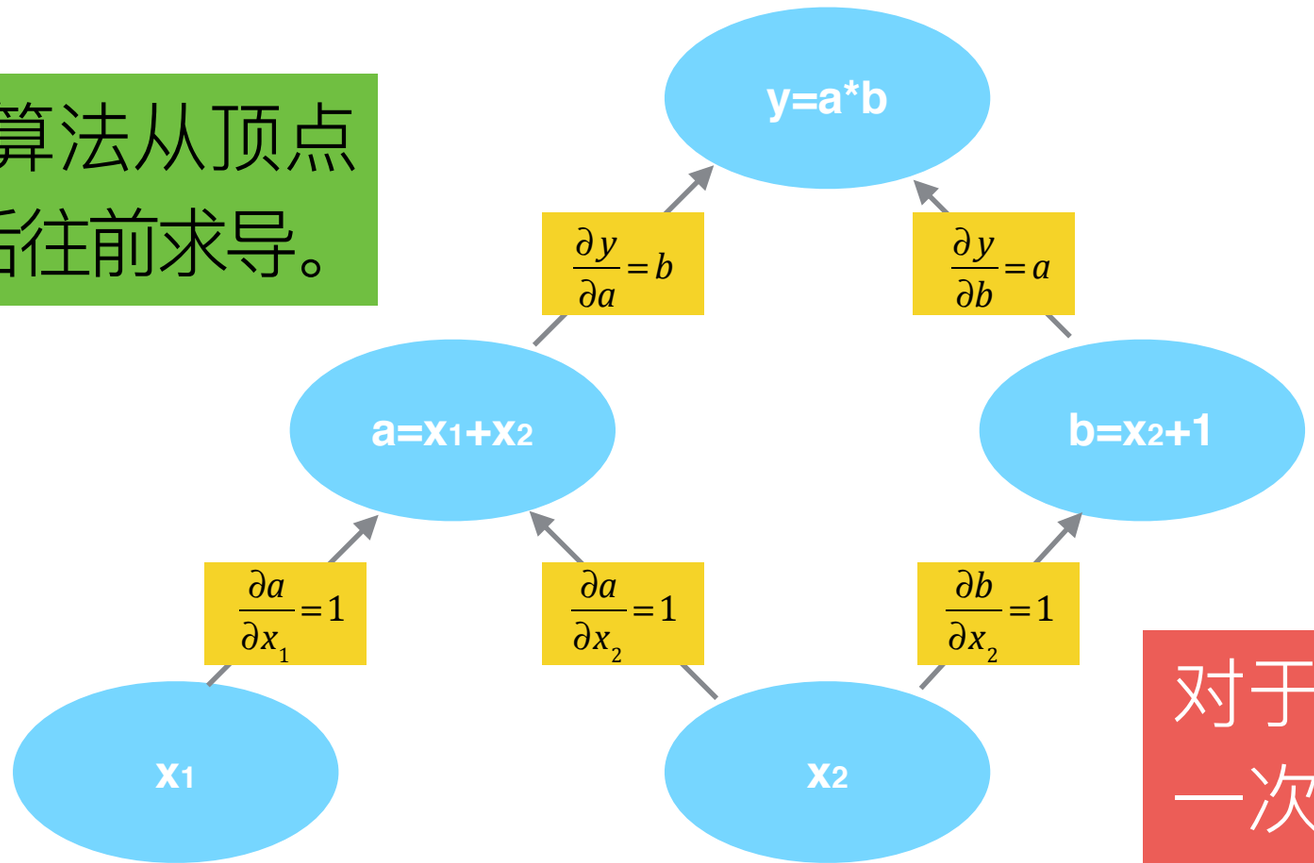
$$\frac{\partial y}{\partial x_2}=\frac{\partial y}{\partial a}\cdot\frac{\partial a}{\partial x_2}+\frac{\partial y}{\partial b}\cdot\frac{\partial b}{\partial x_2}$$

重复计算了对a的偏导

Author:WangQi
Email:wangqikaixin@gmail.com

反向传播算法的优势

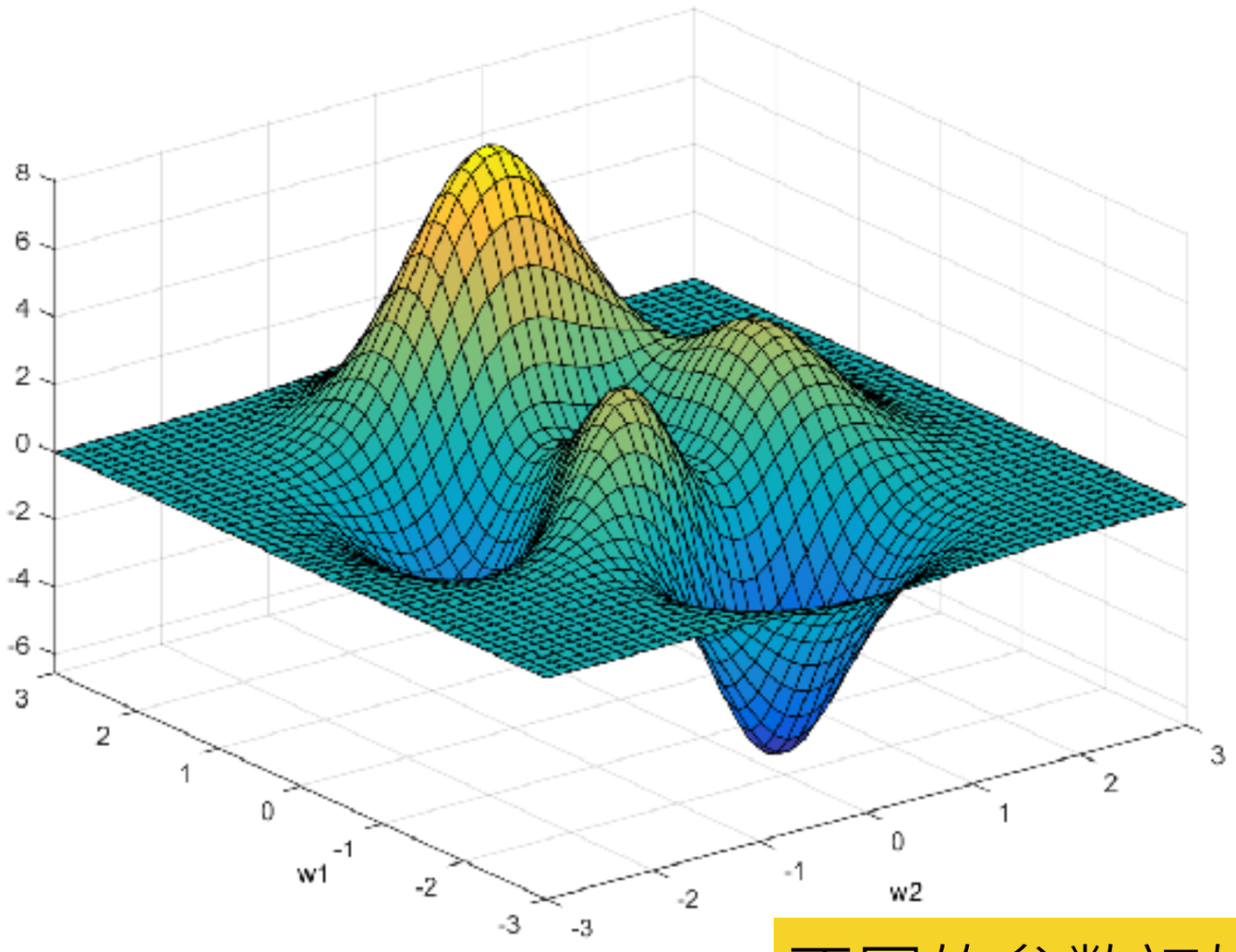
反向传播算法从顶点开始，从后往前求导。



对于每一个路径只访问一次就能求顶点对所有下层节点的偏导值。

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

非凸函数优化与局部最小值

反向传播算法实现了一种对可能的网络权值空间的梯度下降搜索，而对于多层网络来说，误差的曲面可能包含多个局部极小值（而不是像我们上图那样只有局部最小值就是全局最小值），**反向传播算法只能保证收敛到局部最小值，但在现实应用中，人们发现局部最小值得问题并没有那么严重**。这可以理解为网络维度越大，也就越为梯度下降提供更多的逃逸路线，让梯度下降离开相对该权值的局部最小值。

Author:WangQi
Email:wangqikaixin@gmail.com

神经网络的输入层与输出层

输入层与数据直接相连接

图像数据，将每一个像素与输入层相连接。

语音数据，将语音波形量化后与输入层相连接。

文本数据，将文字索引与输入层相连接。

输出层根据情况设置不同的单元。

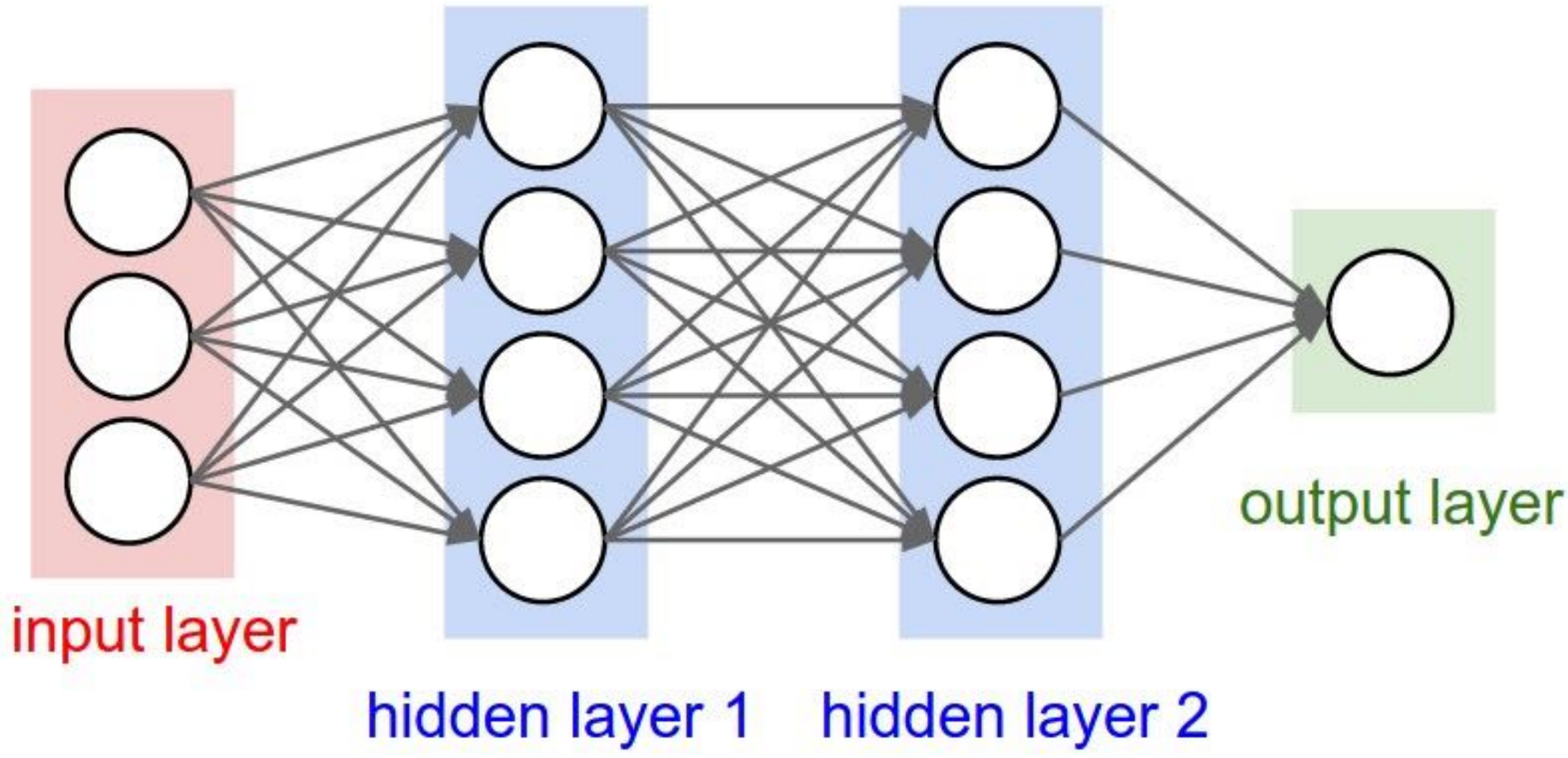
二分类以单个sigmoid神经元作为输出层。

多分类（包括二分类）以多个softmax单元作为输出层。

回归任务可以设置线性单元作为输出层。

Author:WangQi
Email:wangqikaixin@gmail.com

神经网络的参数数量



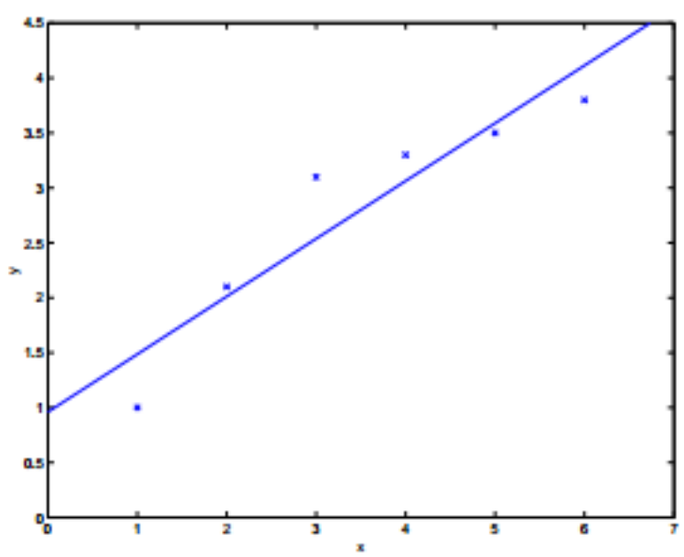
$$num\_of\_\arg s = \sum_{i=1}^{l-1} L_i \cdot L_{i+1} + \sum_{j=2}^{l} L_j$$

Author:WangQi
Email:wangqikaixin@gmail.com

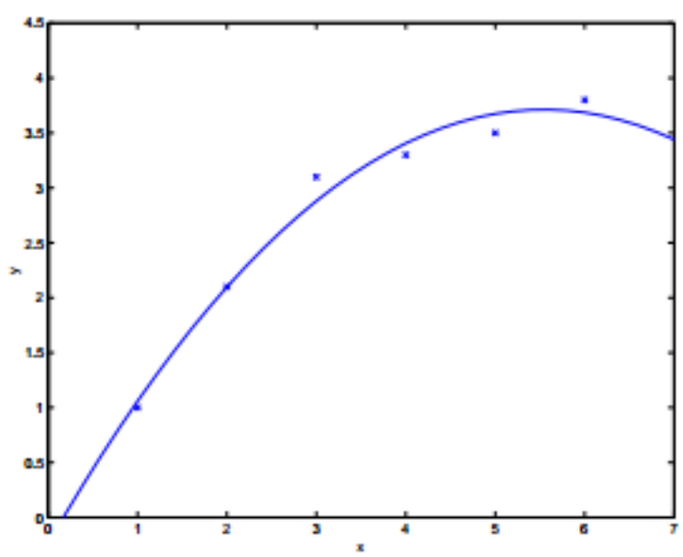
过拟合与欠拟合

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。

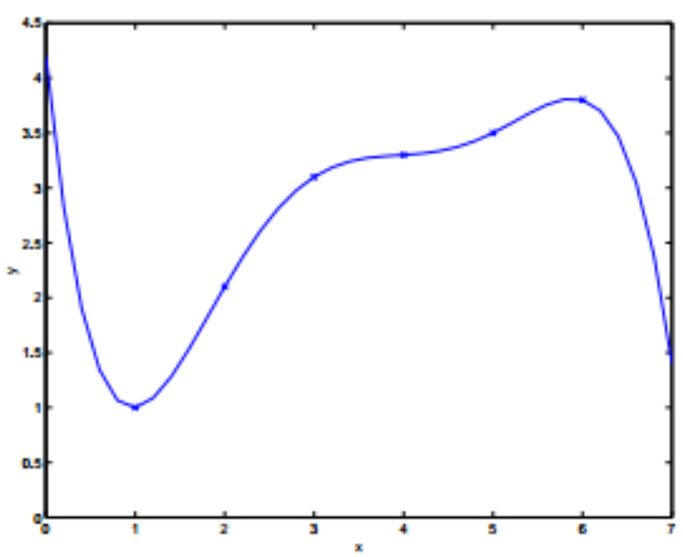
神经网络的参数规模太大，可能导致过拟合，即模型复杂度过高，模型学习到了噪声特征。



欠拟合



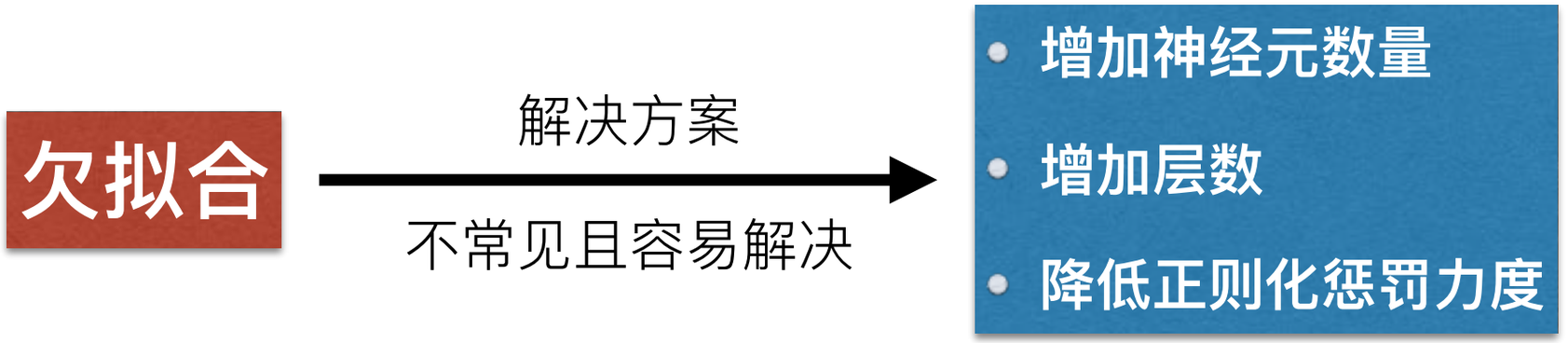
拟合

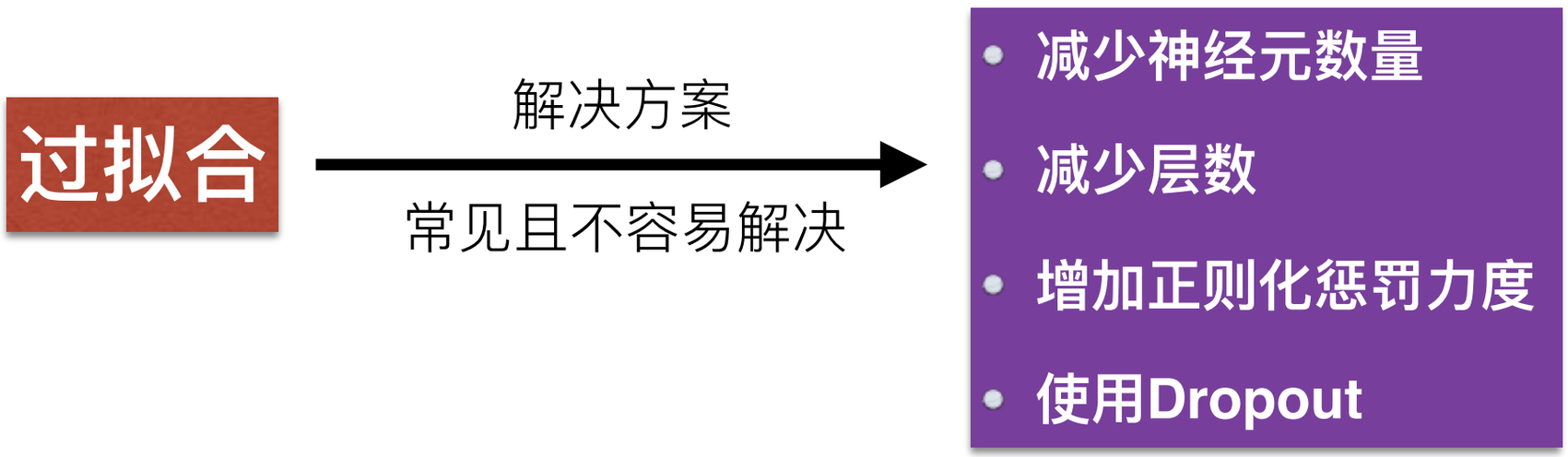


过拟合

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

- 监督学习是指样本带有标记的机器学习算法。
- 代价函数：通过使用均方误差来衡量预测值与样本标记的差距。
- 神经网络的自学习问题可以转化为最小化代价函数的问题。
- 使用反向传播算法作为最优化算法求解梯度，并用来更新参数。
- 由非凸代价函数带来了局部极小值问题。但通常不影响算法获得良好的结果。
- 神经网络的输入层与输出层的结构决定了其可以进行端到端的学习。
- 神经网络的参数包括连接权重与偏置值。
- 过拟合与欠拟合需要注意。过拟合是常见的影响神经网络训练的问题。

Author:WangQi
Email:wangqikaixin@gmail.com

THANKS

Author:WangQi
Email:wangqikaixin@gmail.com