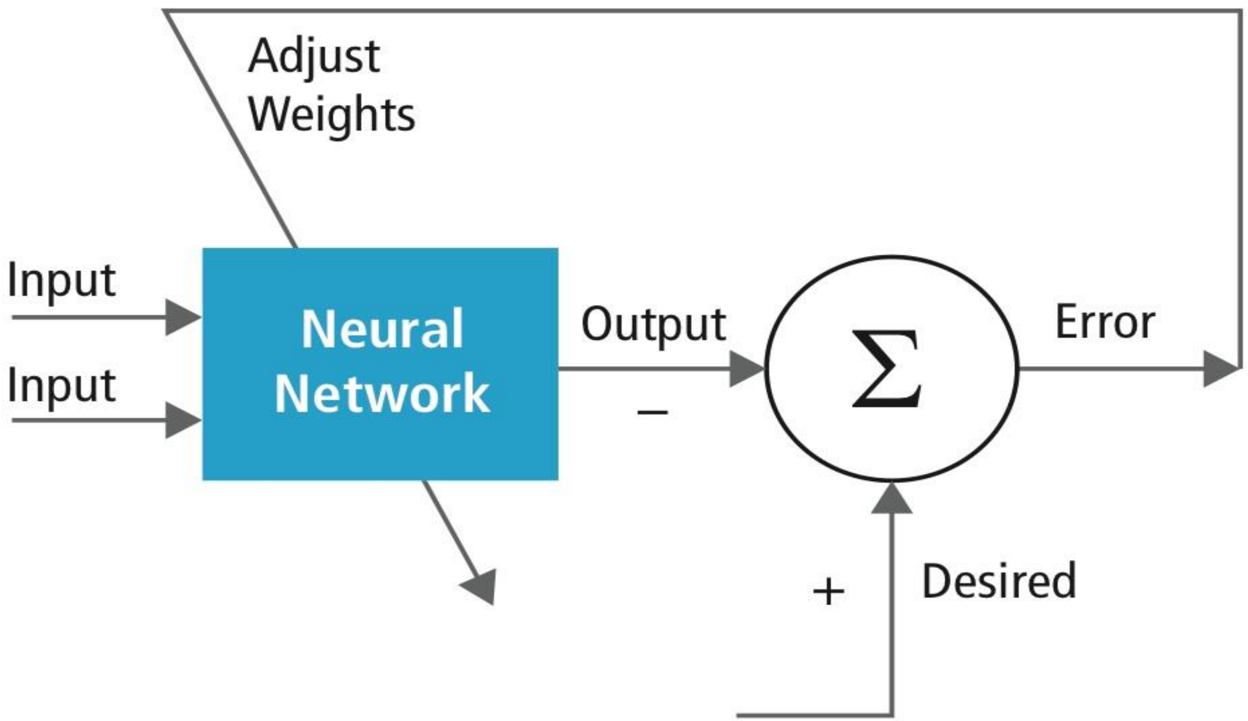
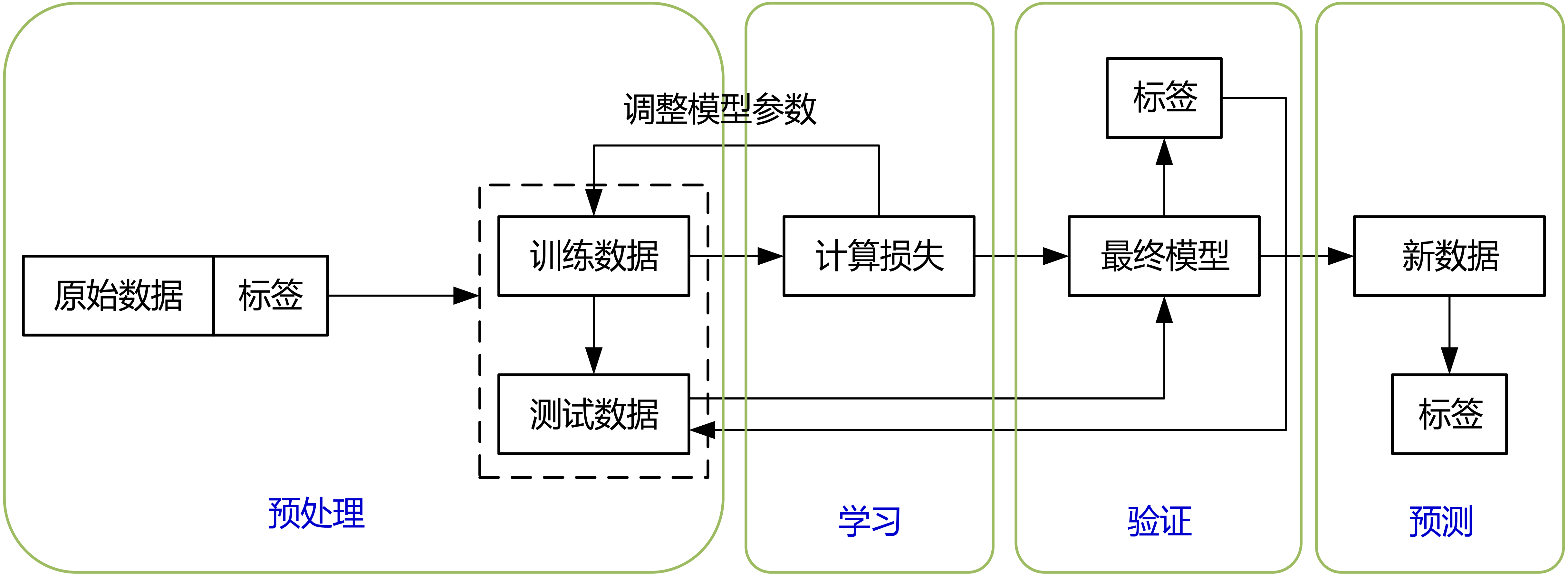
神经网络的学习



通过调整神经元的参数，使得网络对给定输入可产生期望输出

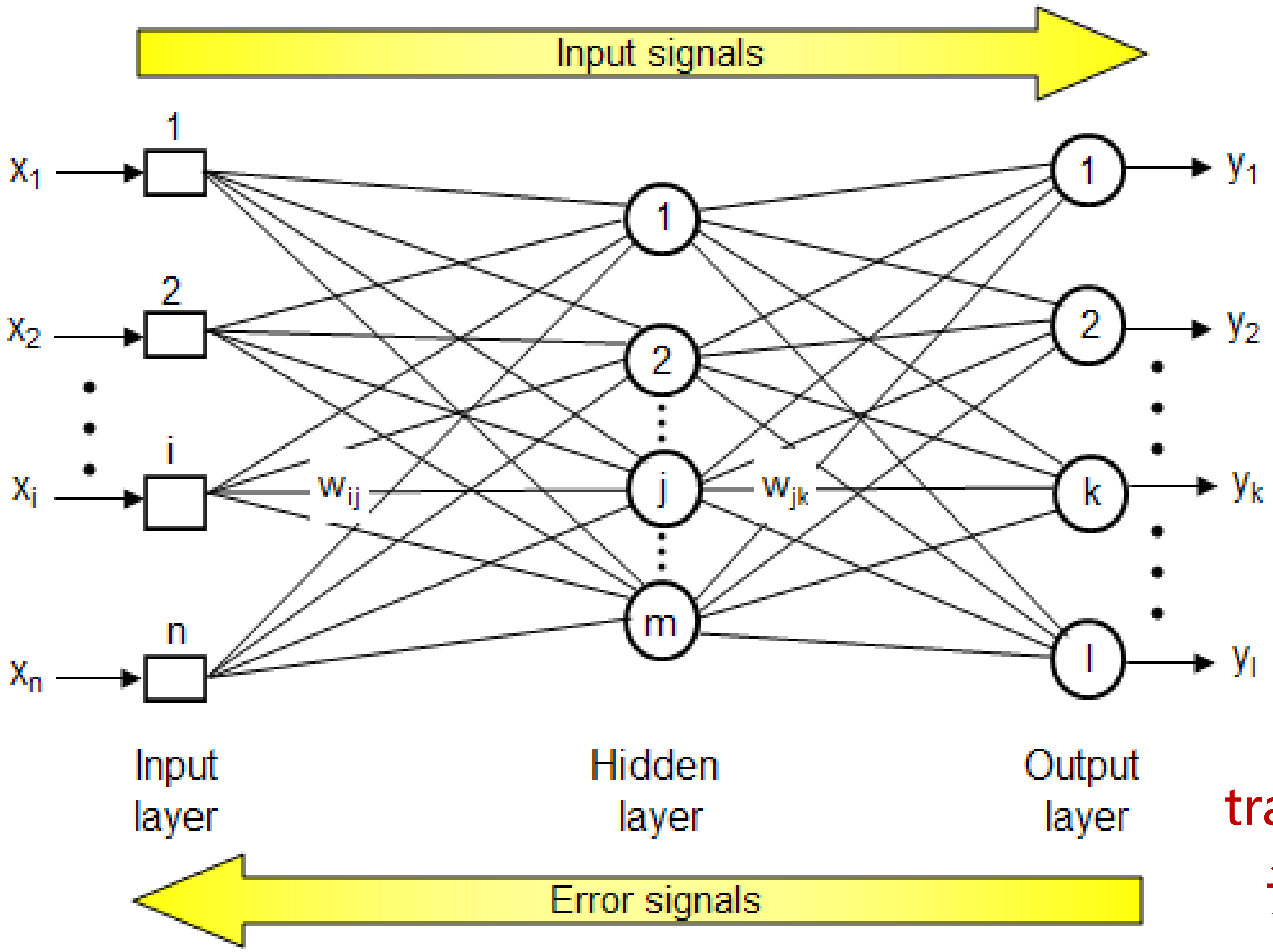
监督学习过程



前馈运算与反向传播

Inference/prediction

推理与预测



training /learning

训练与学习

损失函数（Loss Function）

- 损失函数，又称目标函数，或误差函数，用来度量网络实际输出与期望输出之间的不一致程度，指导网络的参数学习和表示学习。

- 损失函数是一个非负实值函数。
- 针对不同的问题，会采用不同的损失函数
  - 回归问题(连续型)：平方损失等
  - 分类问题(离散型)：对数损失、交叉熵等
- 不同的损失函数会影响网络的训练速度和网络的泛化性能

损失函数-连续型输出

- 平方损失函数(Square Loss)

$$L(y, f(x)) = (y - f(x))^2$$

- 绝对值损失函数(Absolute Value Loss)

$$L(y, f(x)) = \left| y - f(x) \right|$$

损失函数-离散型输出

交叉熵损失(Cross-Entropy Loss； Log Loss)

熵：用于度量变量的不确定性程度

熵越大，随机变量或系统的不确定性就越大。

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

$$\{A, B, C, D\}$$

$$\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\}$$

$$H(X) = \frac{1}{2}\log(2) + \frac{1}{4}\log(4) + \frac{1}{8}\log(8) + \frac{1}{8}\log(8) = \frac{1}{2} + \frac{1}{2} + \frac{3}{8} + \frac{3}{8} = \frac{7}{4}.$$

根据真实分布，我们能够找到一个最优策略，以最小的代价消除系统的不确定性，而这个代价大小就是信息熵。

交叉熵：主要用于度量两个概率分布间的差异性信息

$$H(p,q) = -\sum_{x_i} p(x_i)\log q(x_i)$$

$$p_k = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$$

$$q_k = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$$

$$H(p,q) = \frac{1}{2} * \log_2 4 + \frac{1}{4} * \log_2 4 + \frac{1}{8} * \log_2 4 + \frac{1}{8} * \log_2 4 = 2$$

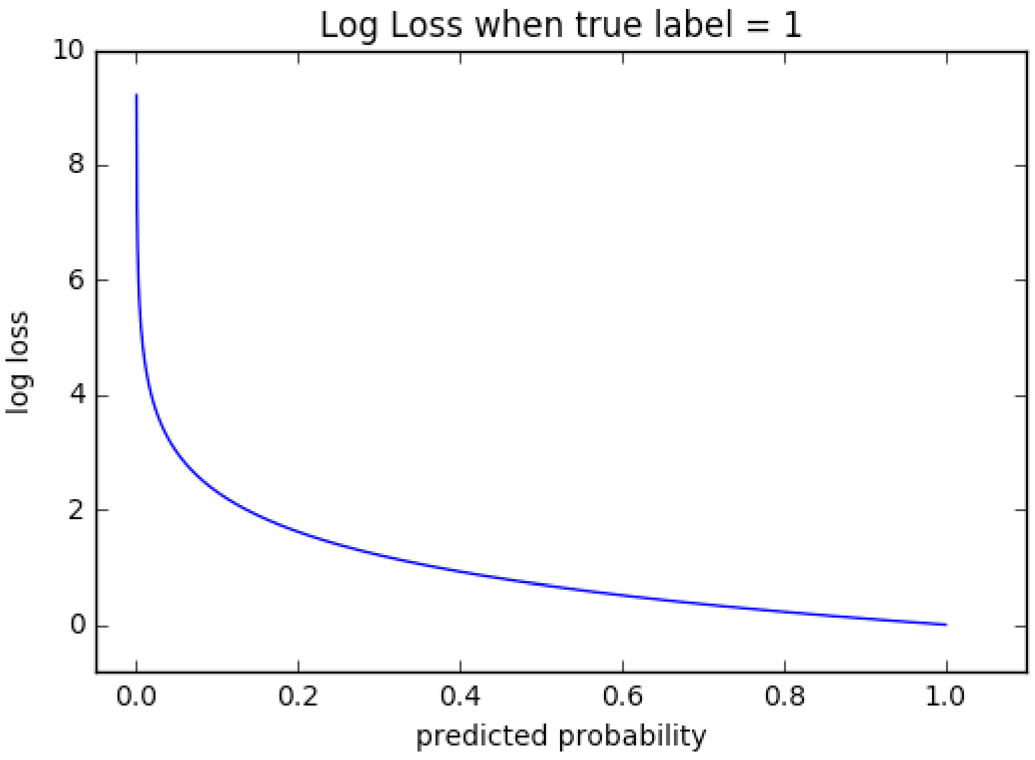
交叉熵用来衡量在给定的真实分布下，使用非真实分布所指定的策略消除系统的不确定性所需要付出的努力的大小。

交叉熵损失(Cross-Entropy Loss； Log Loss)

二分类问题的交叉熵损失函数：

对于样本（x, y），x为样本， y为对应的标签, 在二分类问题中，其取值的集合可能为{0，1}。

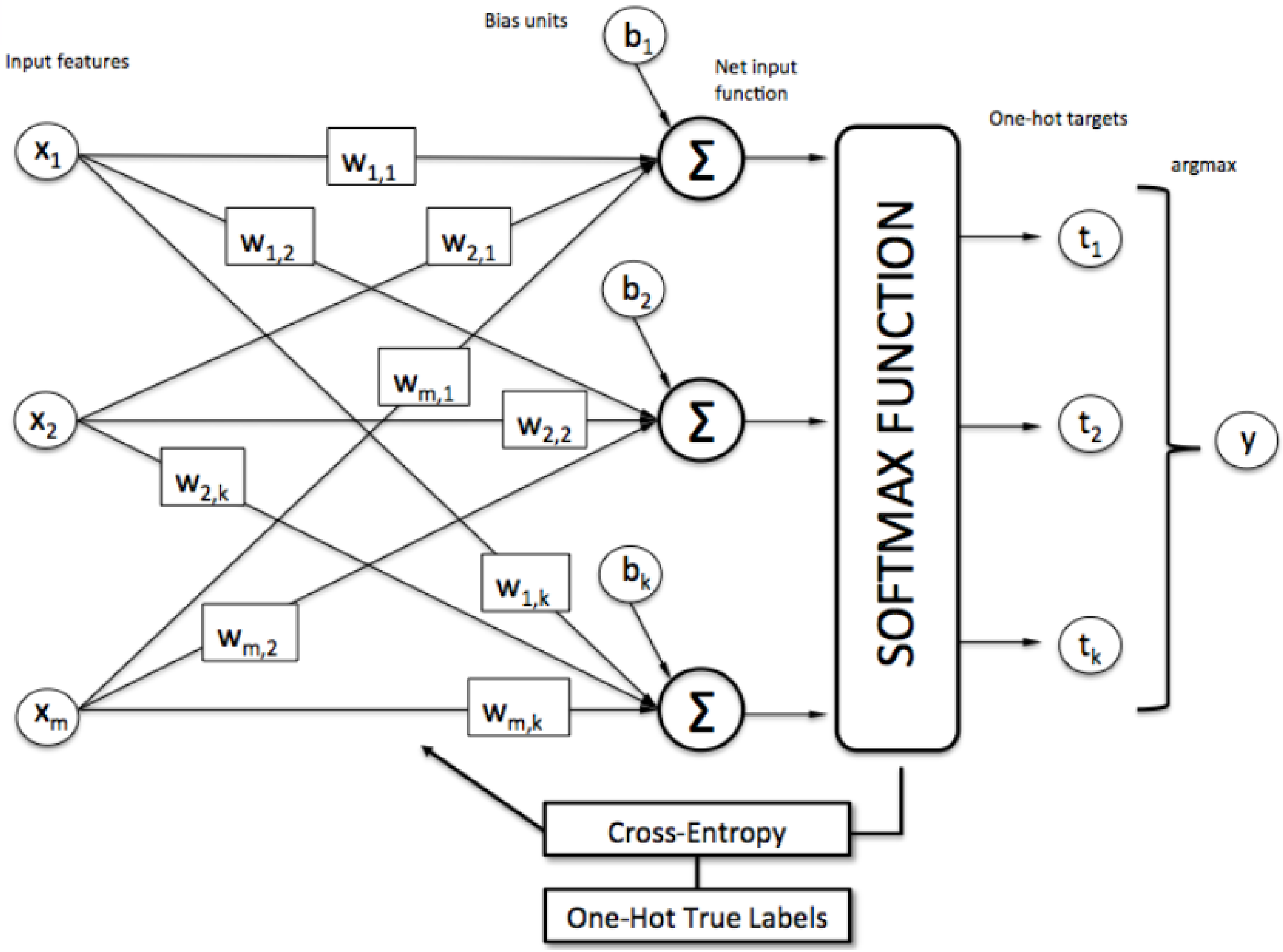
假设某个样本的真实标签为y，该样本的$y=1$的概率为p，则该样本的损失函数为：$-(ylog(p)+(1-y)\log(1-p))$。



- 交叉熵越低，就说明由算法所产生的策略最接近最优策略，也间接说明算法所算出的非真实分布越接近真实分布。

- 预测输出越接近真实样本标签 1，损失函数 L 越小；预测输出越接近 0，L 越大。

Softmax回归 (Softmax Regression)



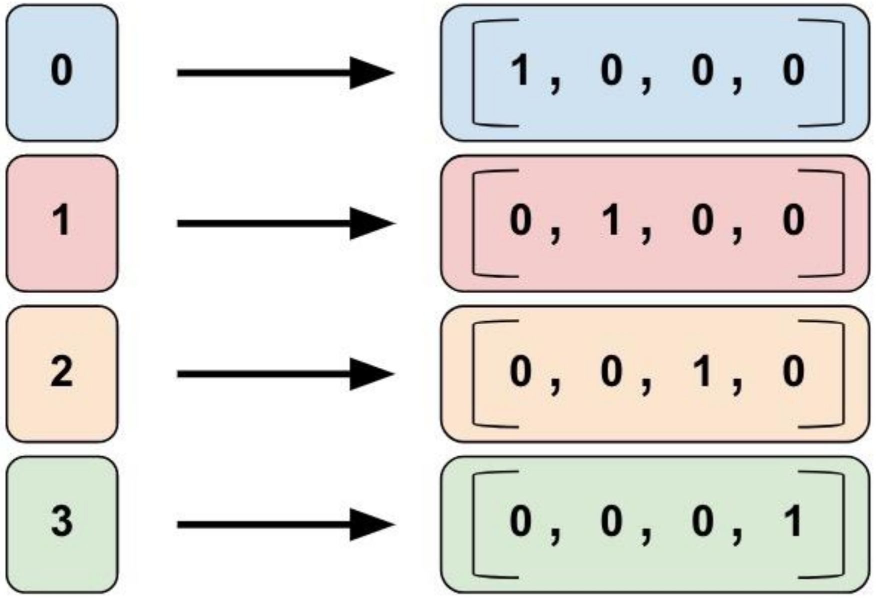
- 交叉熵和Softmax在多分类问题的结合应用
- 交叉熵可用于比较softmax输出和独热编码(one-hot encoding)输出之间的距离

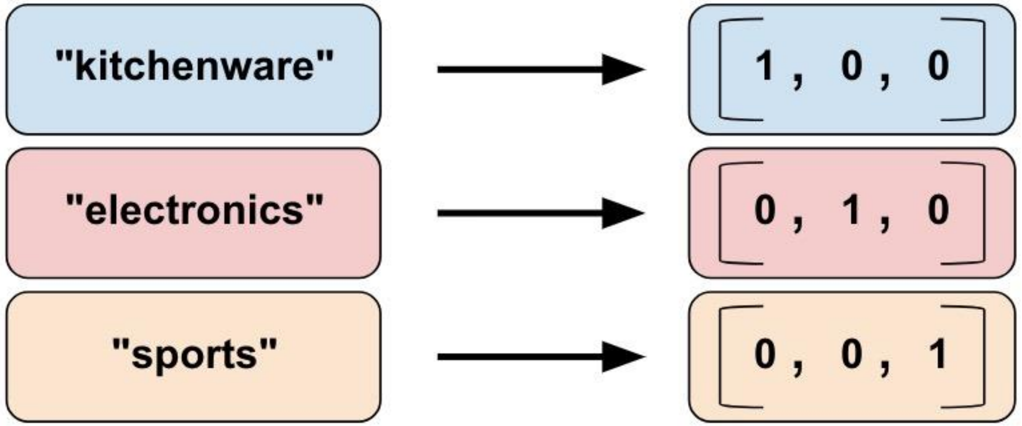
独热编码 (One-hot Encoding)

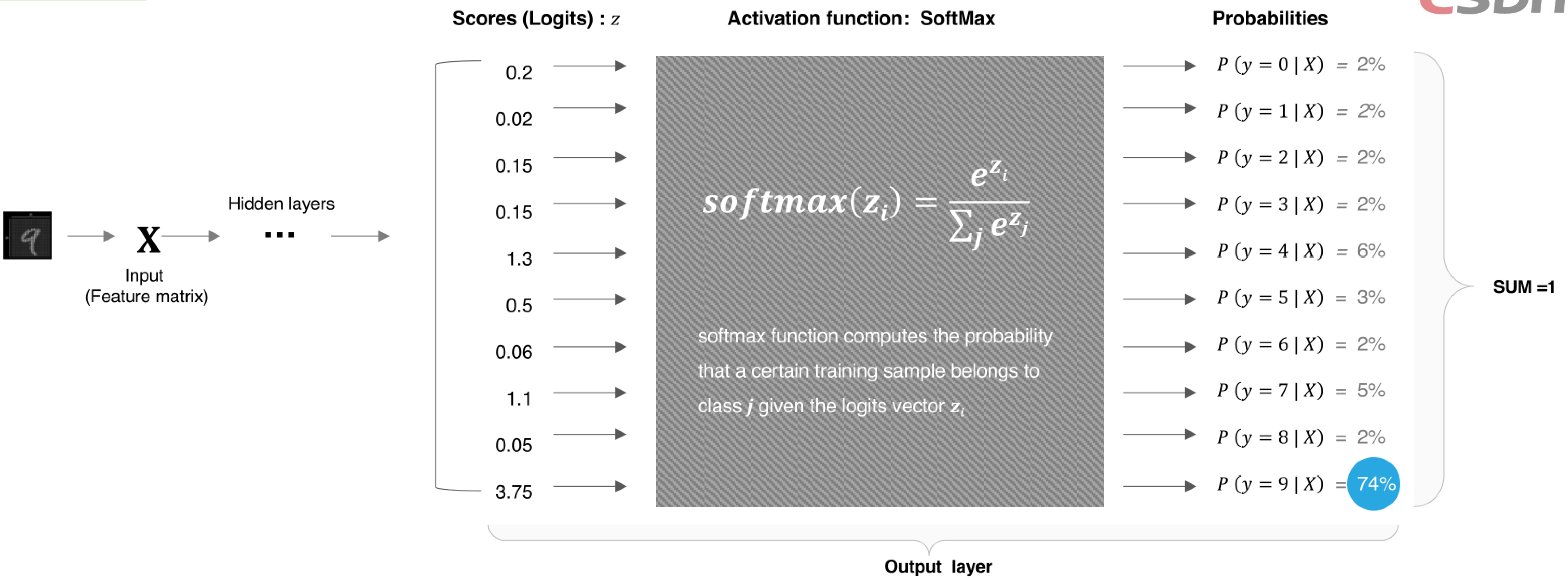
- 在分类问题中，独热编码是一种表示目标变量或类别的方法。

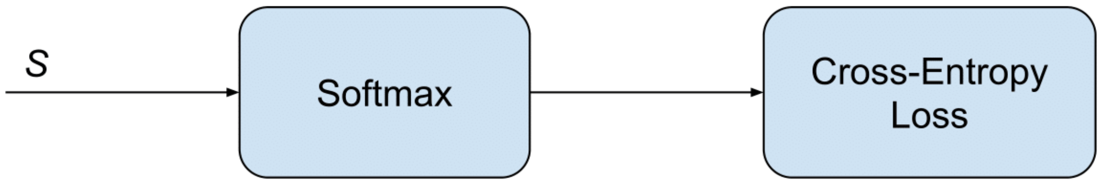
  目标变量可以把字符串标签转换为独热编码向量。

  一个独热向量在目标类别的索引处填充1，在其他地方填充0。例如，如果目标类别是猫和狗，它们可以分别用 [1,0]和[0,1]表示。对于1000个类别，一个独热编码向量的大小为1000个整数，其中除一个数为1外全为0。









$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$

Softmax回归 (Softmax Regression)

假设有一个三分类问题，某个样例的正确答案是（1,0,0）。某模型经过Softmax回归之后的预测答案是（0.5,0.4,0.1)，那么这个预测和正确答案之间的交叉熵是：

$$H((1,0,0),(0.5,0.4,0.1)) = -(1\times\log 0.5 + 0\times\log 0.4 + 0\times\log 0.1) \approx 0.3$$

如果另外一个模型的预测是（0.8,0.1,0.1），那么这个预测值和真实值之间的交叉熵是：

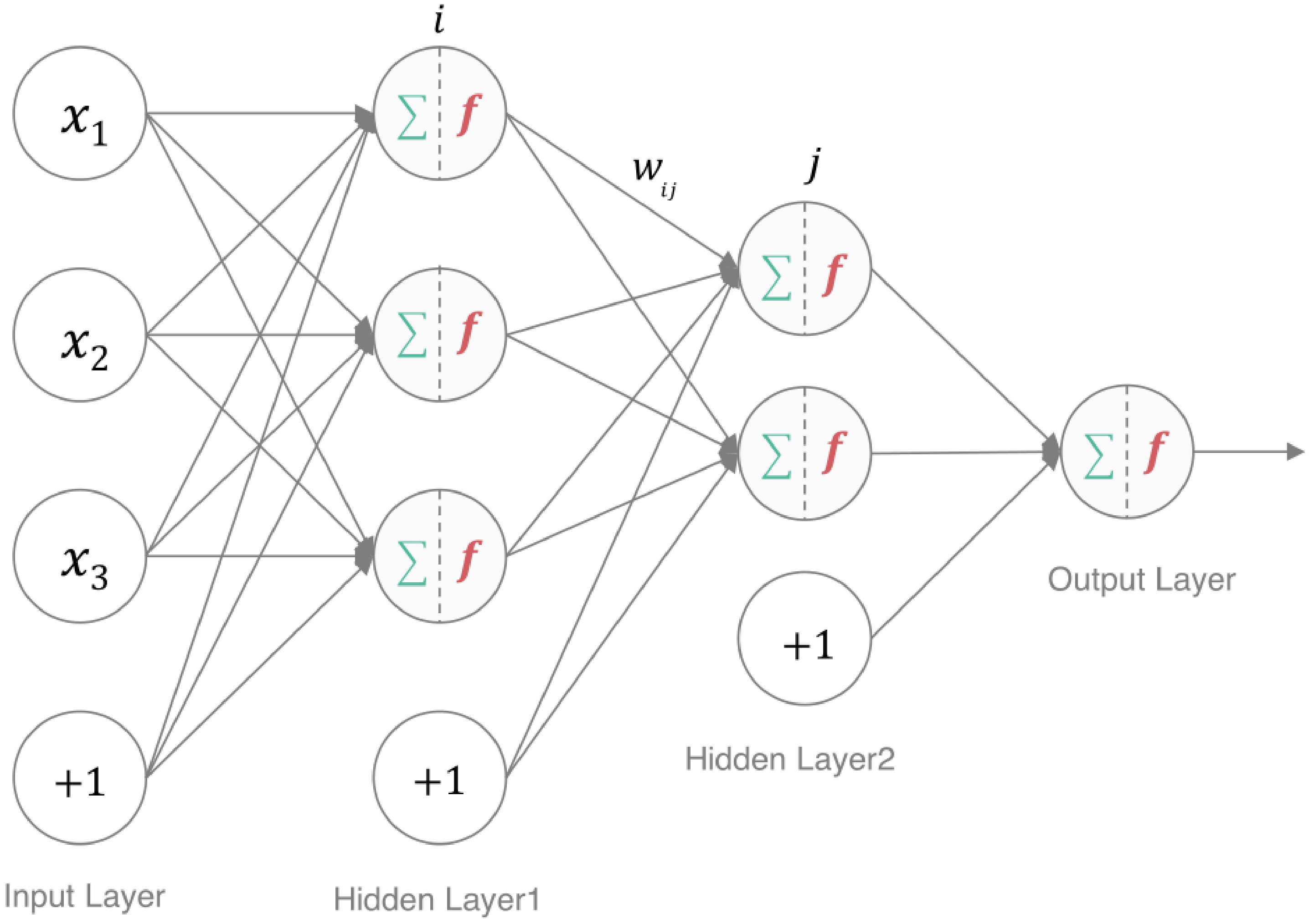
$$H((1,0,0),(0.8,0.1,0.1)) = -(1\times\log 0.8 + 0\times\log 0.1 + 0\times\log 0.1) \approx 0.16$$

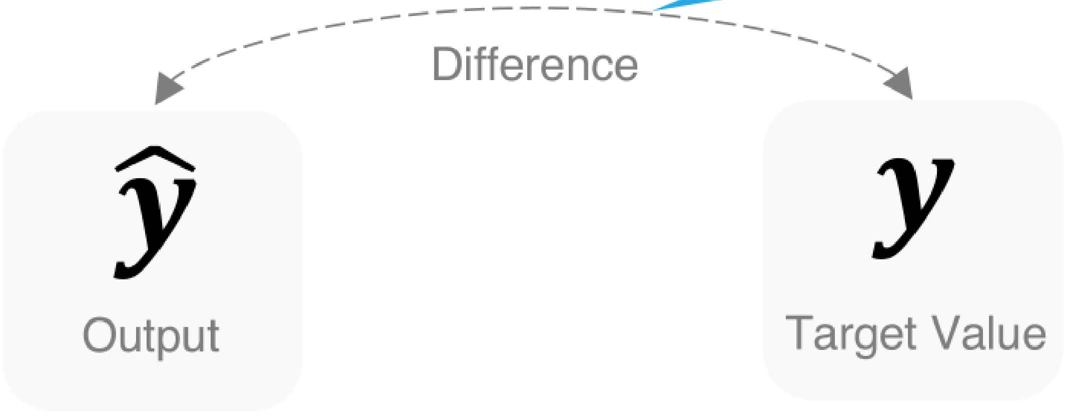
从直观上可以很容易知道第二个答案要优于第二个。通过交叉熵计算得到的结果也是一致的（第二个交叉熵的值更小）。

神经网络训练的优化目标

$$w_{ij}^{new} = w_{ij}^{old} - \eta \frac{\partial E}{\partial w_{ij}}$$

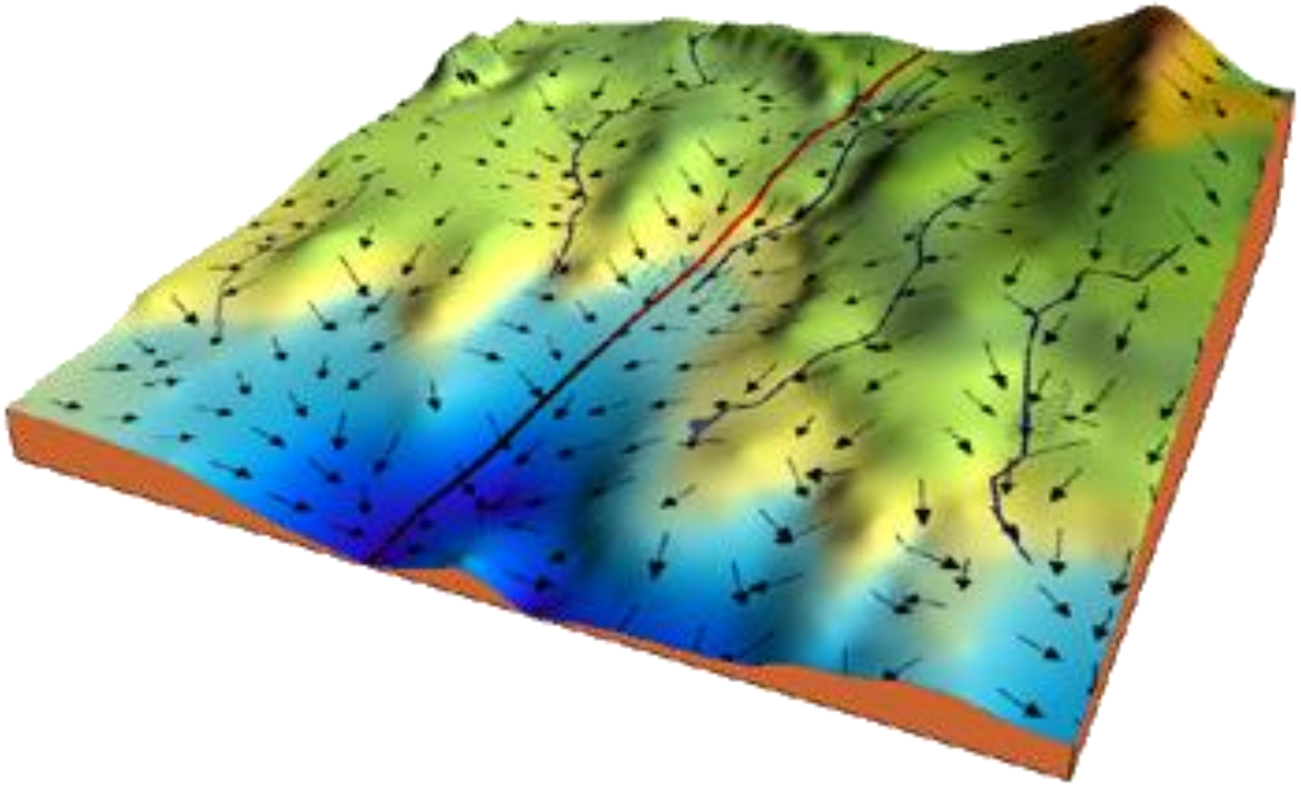
η 是学习率





$$Error = (output - target)^2$$

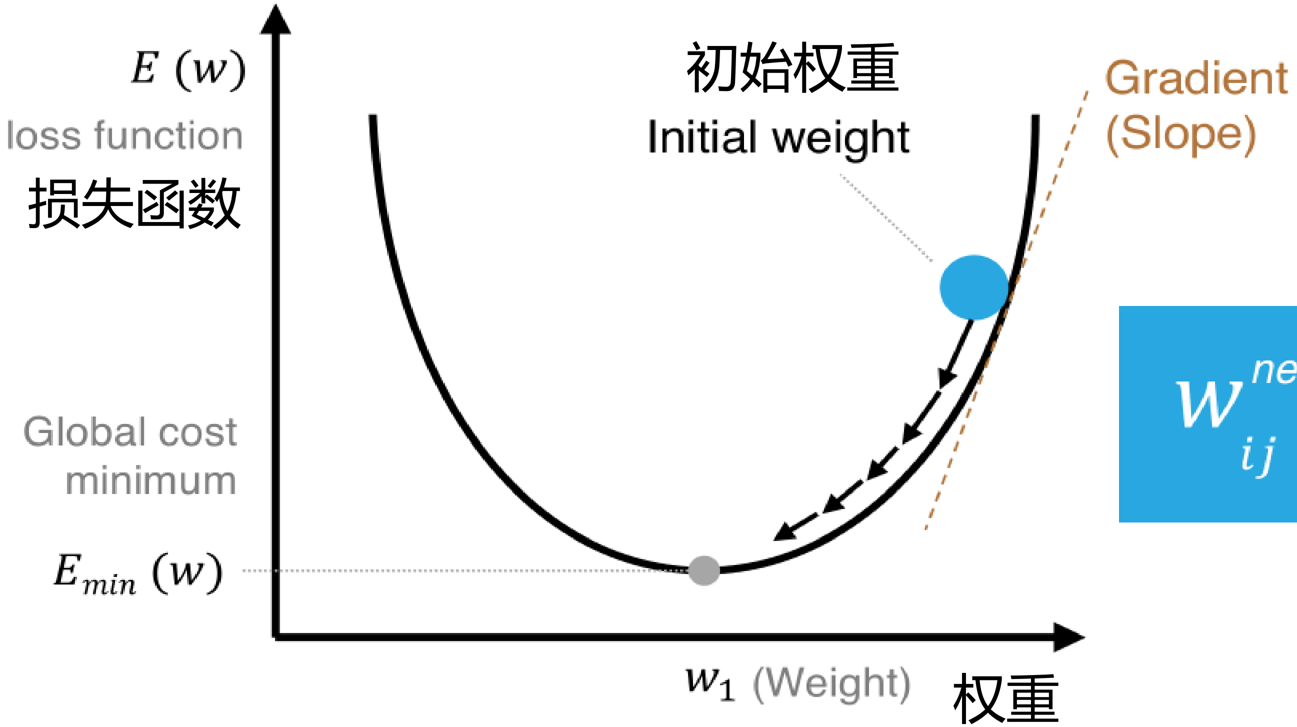
梯度下降 （Gradient Descent）



- 求解非线性无约束优化问题的最基本方法; 最小化损失函数的一种常用的一阶优化方法

梯度下降

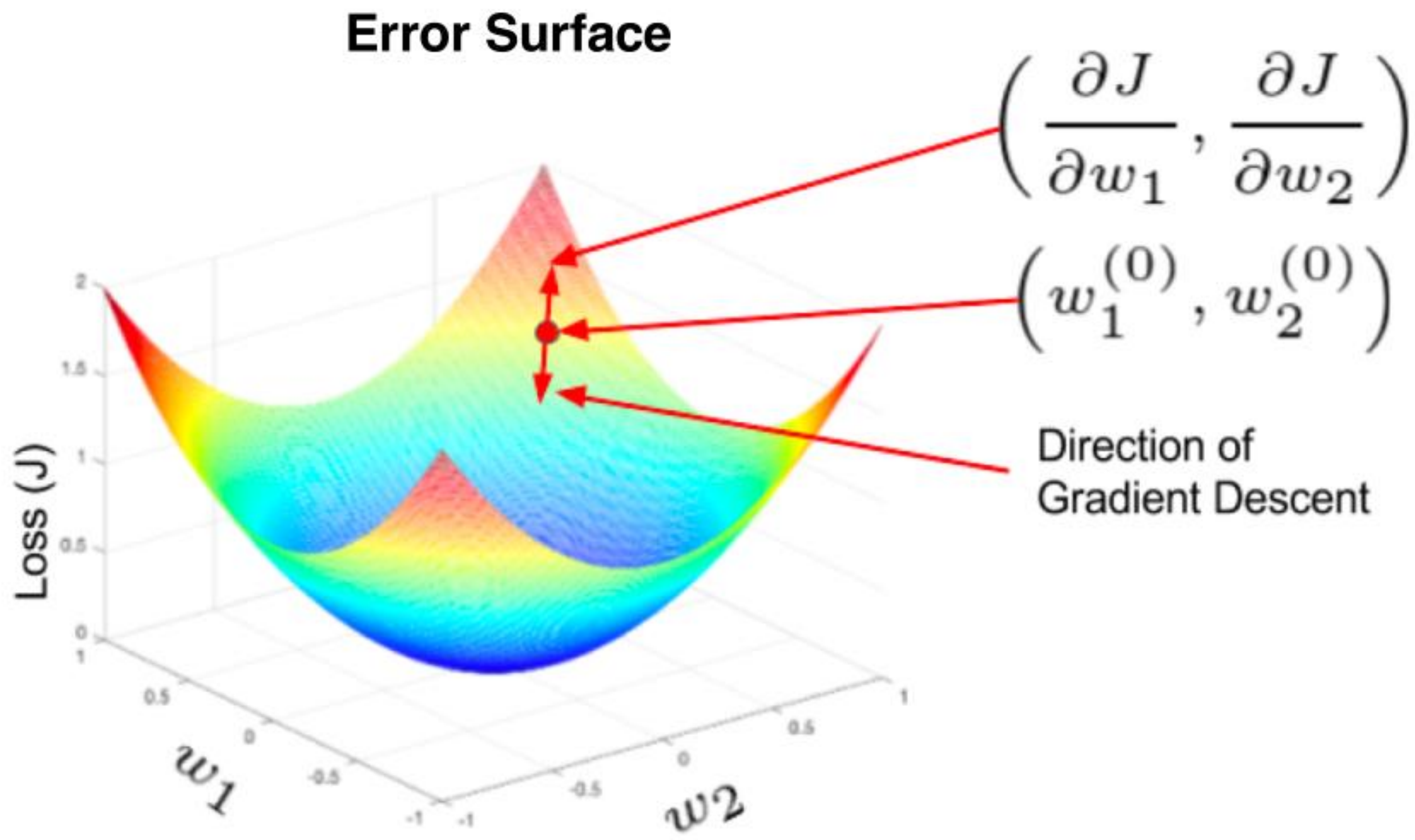
Error Surface 误差曲面



$$w_{ij}^{new} = w_{ij}^{old} - \eta \frac{\partial E}{\partial w_{ij}}$$

Error surface of a linear neuron with one weight

沿负梯度方向，函数值下降最快



Error surface of a linear neuron with two input weights

非凸误差曲面