第五课——网络优化-线性神经网络，delta学习规则，梯度下降法

51CTO学院

edu.51cto.com

线性神经网络

- 线性神经网络与感知器的主要区别在于，感知器的激活函数只能输出两种可能的值，而线性神经网络的输出可以取任意值，其激活函数是线性函数。线性神经网络采用Widrow-Hoff学习规则，即LMS（Least Mean Square）算法来调整网络的权值和偏置。

- 线性神经网络在结构上与感知器非常相似，只是神经元激活函数不同。在模型训练时把原来的sign函数改为了purelin函数（y=x）。

LMS学习规则

2.4.4 LMS 学习规则

1962 年，Bernard Widrow 和 Marcian Hoff 提出了 Widrow-Hoff 学习规则。因为它能使神经元实际输出与期望输出之间的平方差最小，所以又称为最小均方规则（LMS）。LMS 学习规则的学习信号为：

$$r=d_j-\boldsymbol{W}_j{}^{\mathrm{T}}\boldsymbol{X} \tag{2.25}$$

权向量调整量为：

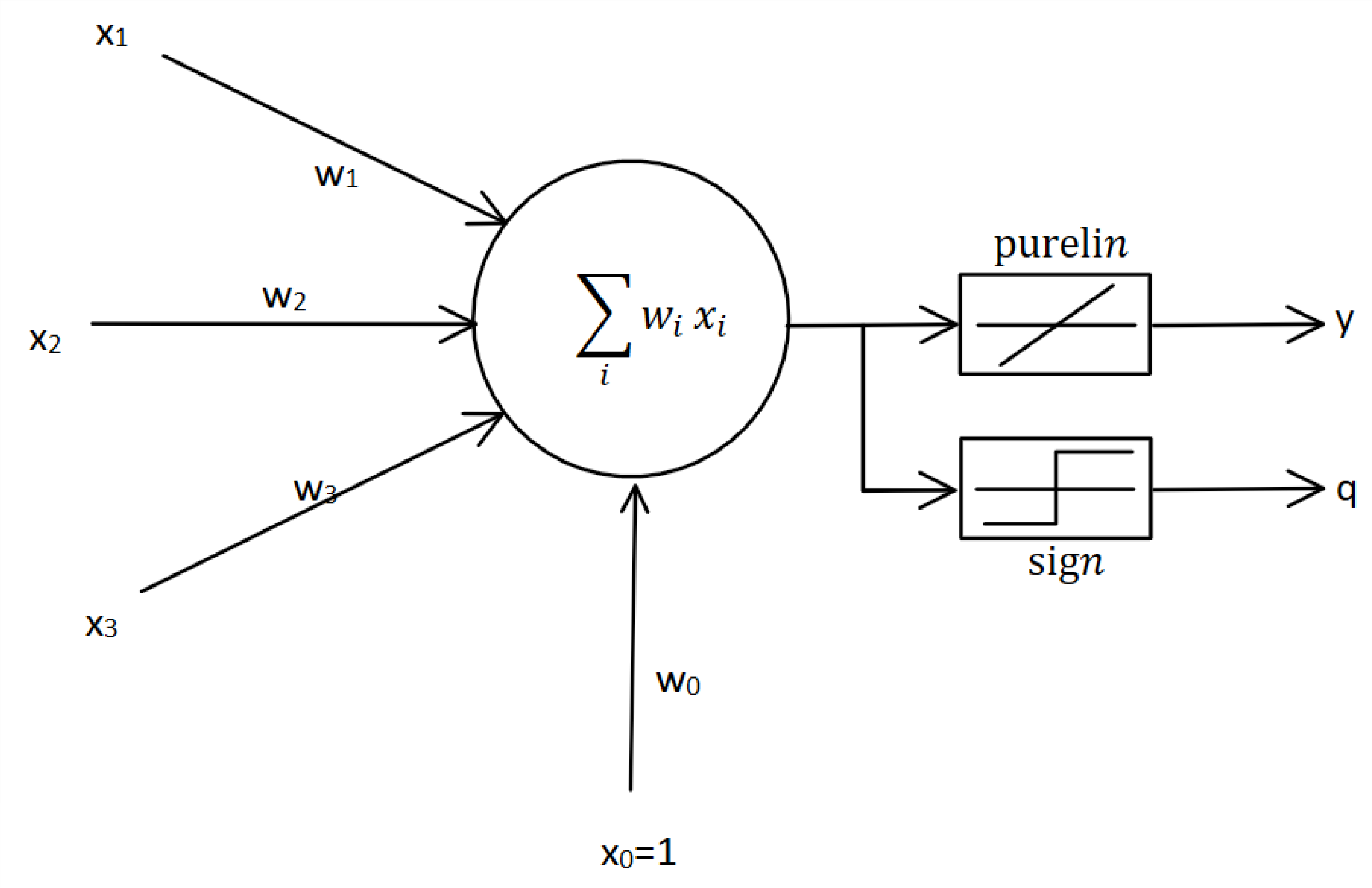
$$\Delta\boldsymbol{W}_j=\eta(d_j-\boldsymbol{W}_j^{\mathrm{T}}\boldsymbol{X})\boldsymbol{X} \tag{2.26a}$$

$\Delta\boldsymbol{W}_j$ 的各分量为：

$$\Delta w_{ij}=\eta(d_j-\boldsymbol{W}_j{}^{\mathrm{T}}\boldsymbol{X})x_j \qquad i=0,1,\cdots,n \tag{2.26b}$$

实际上，如果在 δ 学习规则中假定神经元转移函数为 $f(\boldsymbol{W}_j{}^{\mathrm{T}}\boldsymbol{X})=\boldsymbol{W}_j{}^{\mathrm{T}}\boldsymbol{X}$，则有 $f'(\boldsymbol{W}_j{}^{\mathrm{T}}\boldsymbol{X})=1$，此时式(2.20) 与式(2.25) 相同。因此，LMS 学习规则可以看成是 δ 学习规则的一个特殊情况。该学习规则与神经元采用的转移函数无关，因而不需要对转移函数求导数，不仅学习速度较快，而且具有较高的精度。权值可初始化为任意值。

线性神经网络结构



Delta学习规则

- 1986年，认知心理学家McClelland和Rumelhart在神经网络训练中引入了δ规则，该规则也可以称为连续感知器学习规则。

- δ学习规则是一种利用梯度下降法的一般性的学习规则。

Delta学习规则

代价函数（损失函数）（Cost Function,Lost Function）：

$$E=\frac{1}{2}(d_j-o_j)^2$$

$$=\frac{1}{2}[d_j-f(\mathbf{W}_j{}^{\mathrm{T}}\mathbf{X})]^2 \qquad (2.21)$$

其中，误差 E 是权向量 $\mathbf{W}_j$ 的函数。欲使误差 E 最小，$\mathbf{W}_j$ 应与误差的负梯度成正比，即：

$$\Delta\mathbf{W}_j=-\eta\nabla E \qquad (2.22)$$

式中，比例系数 η 是一个正常数。由式(2.21)，误差梯度为：

$$\nabla E=-(d_j-o_j)f'(\mathbf{W}_j^{\mathrm{T}}\mathbf{X})\mathbf{X} \qquad (2.23)$$

将此结果代入式(2.22)，可得权值调整计算式：

$$\Delta\mathbf{W}_j=\eta(d_j-o_j)f'(\mathrm{net}_j)\mathbf{X} \qquad (2.24a)$$

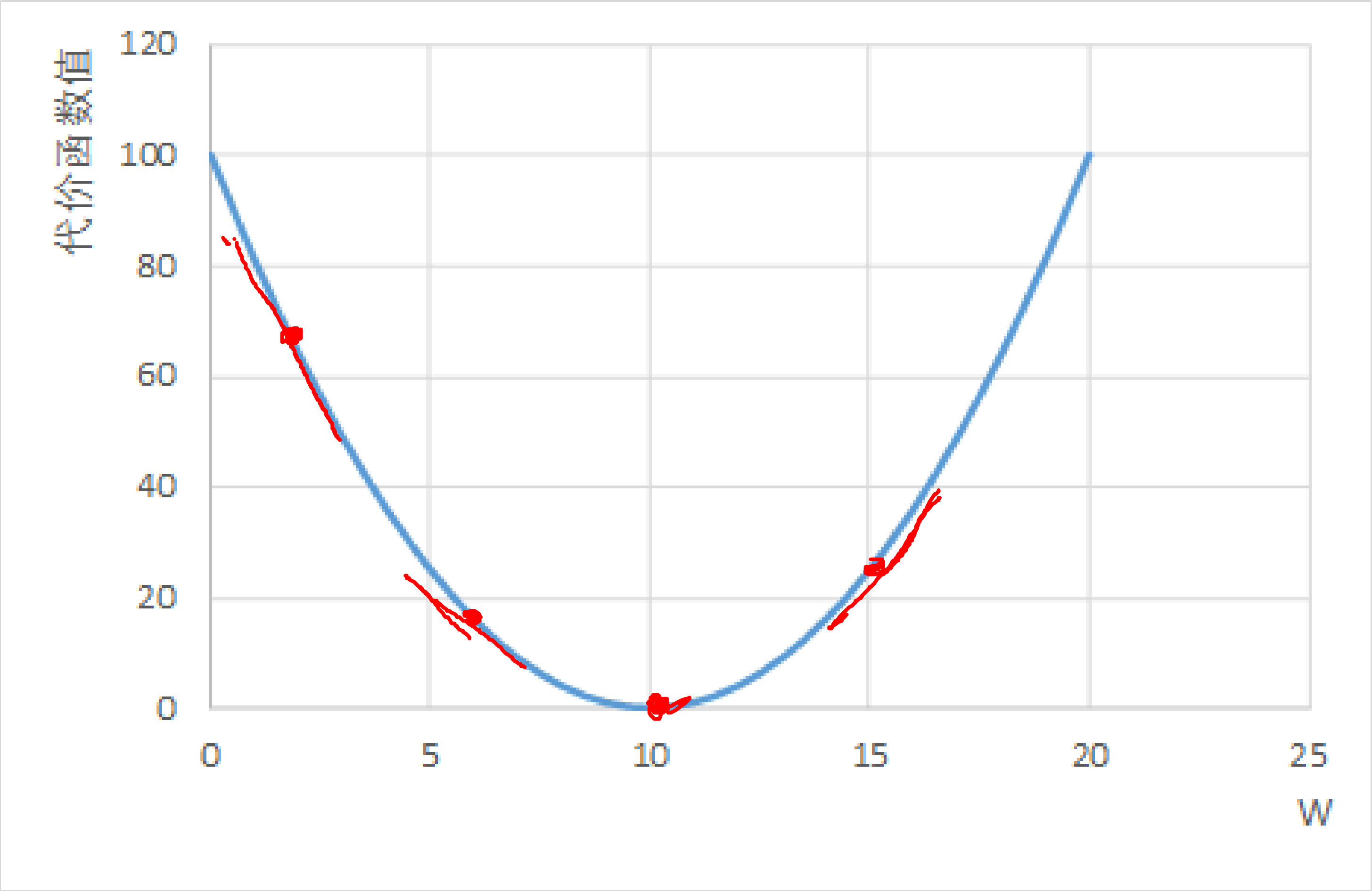
可以看出，上式中 η 与 $\mathbf{X}$ 之间的部分正是式(2.20)中定义的学习信号 δ。$\Delta\mathbf{W}_j$ 中每个分量的调整由下式计算：

$$\Delta w_{ij}=\eta(d_j-o_j)f'(\mathrm{net}_j)x_i \qquad i=0,1,\cdots,n \qquad (2.24b)$$

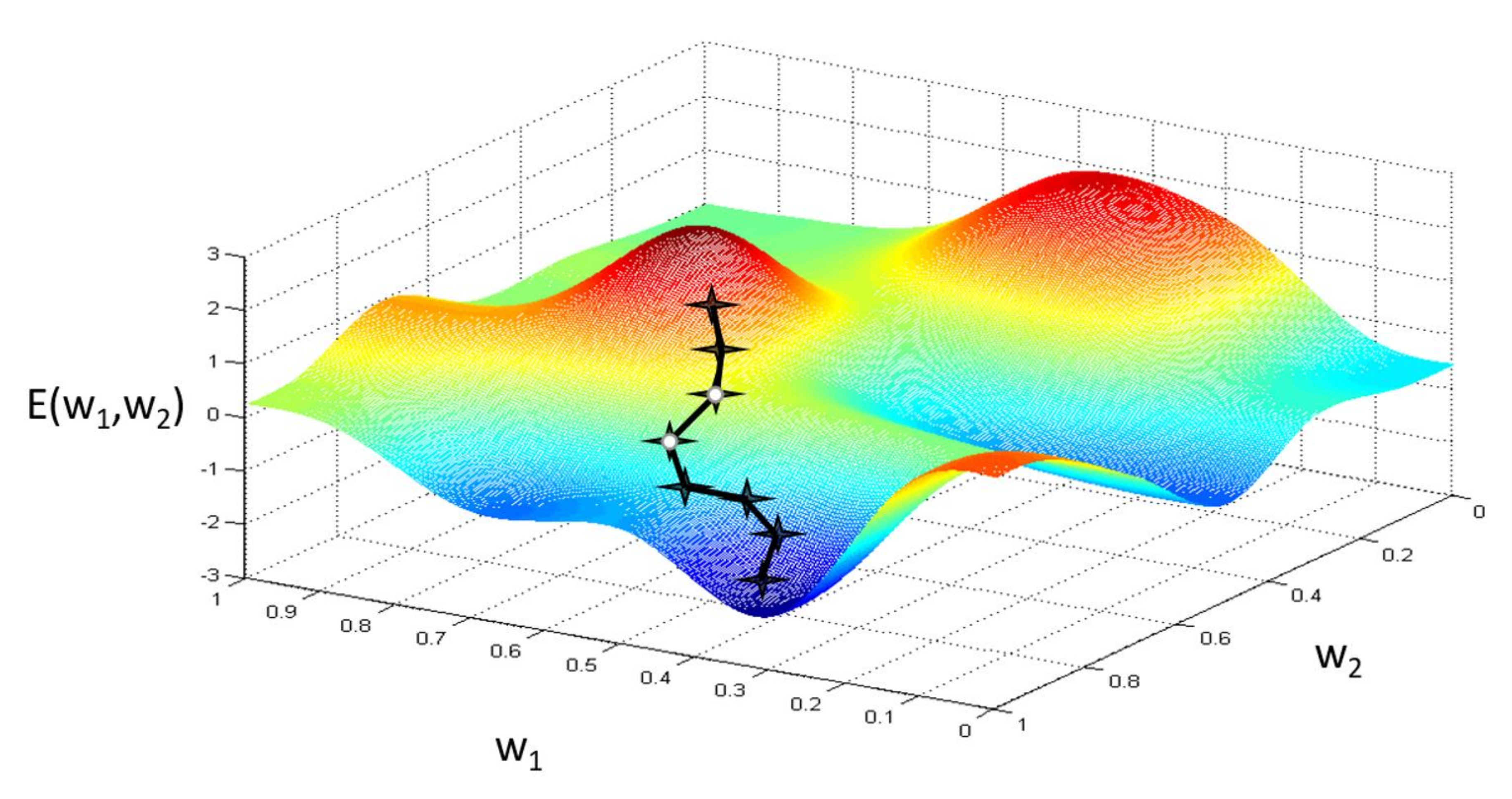
δ 学习规则可推广到多层前馈网络中，权值可初始化为任意值。

梯度下降法-一维情况

$W = W + \Delta W$

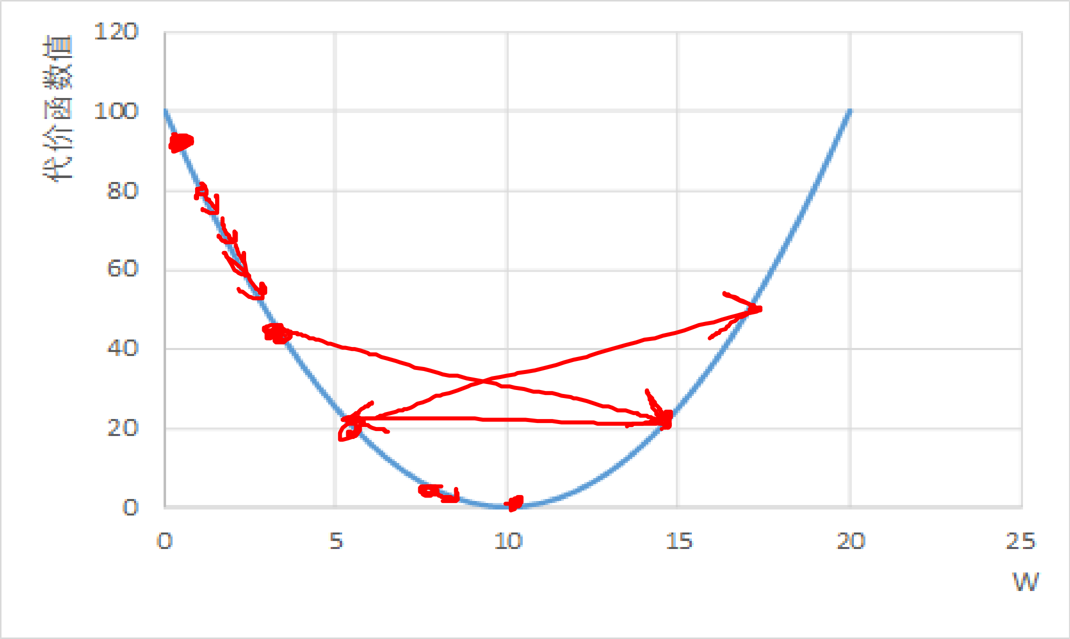


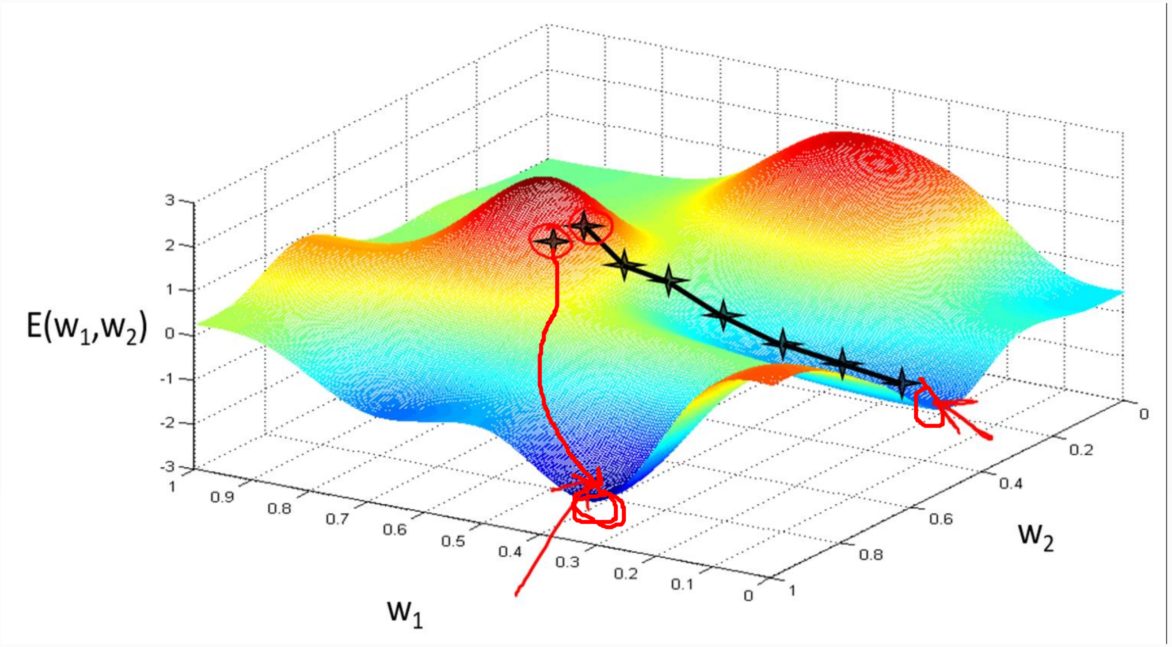
梯度下降法-二维情况



梯度下降法的问题

- 学习率难以选取，太大会产生震荡，太小收敛缓慢
- 容易陷入局部最优解





解决异或问题

Madaline 可以用一种间接的方式解决线性不可分的问题，方法是用多个线性函数对区域进行划分，然后对各个神经元的输出做逻辑运算。如图 5-3 所示，Madaline 用两条直线实现了异或逻辑。

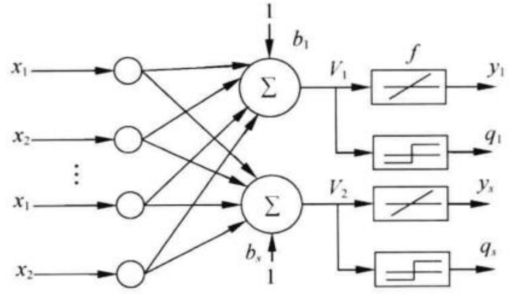


图 5-2　Madaline 结构图

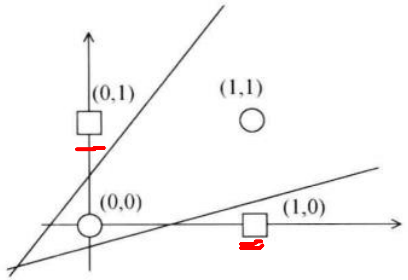


图 5-3　Madaline 实现异或

线性神经网络解决线性不可分问题的另一个方法是，对神经元添加非线性输入，从而引入非线性成分，这样做会使等效的输入维度变大，如图 5-4 所示。

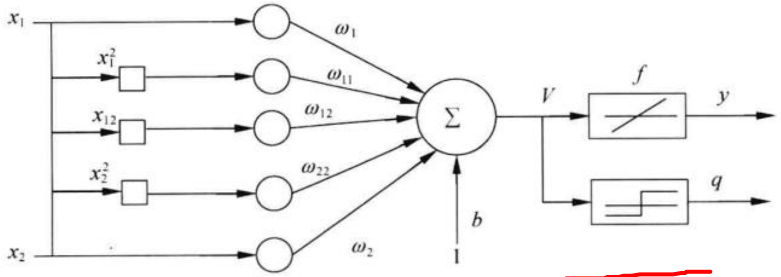


图 5-4　线性网络解决非线性问题

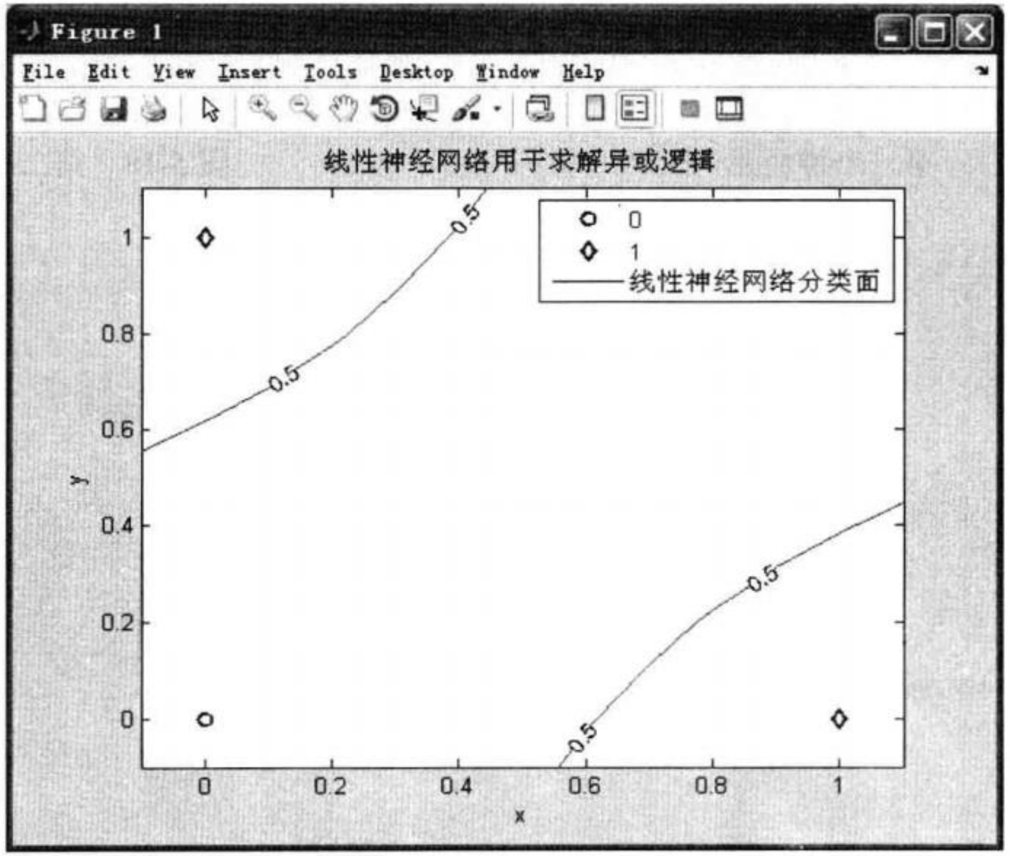


图 5-15　异或问题的决策面

微信公众号：深度学习与神经网络

QQ群：616043628

51CTO学院

Thank You !

为梦想增值！