人工智能学习笔记二——神经网络

人工神经网络（Artificial Neural Networks，简写为ANNs）也简称为神经网络（NNs）或称作连接模型（Connection Model），它是一种模仿动物神经网络行为特征，进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度，通过调整内部大量节点之间相互连接的关系，从而达到处理信息的目的。

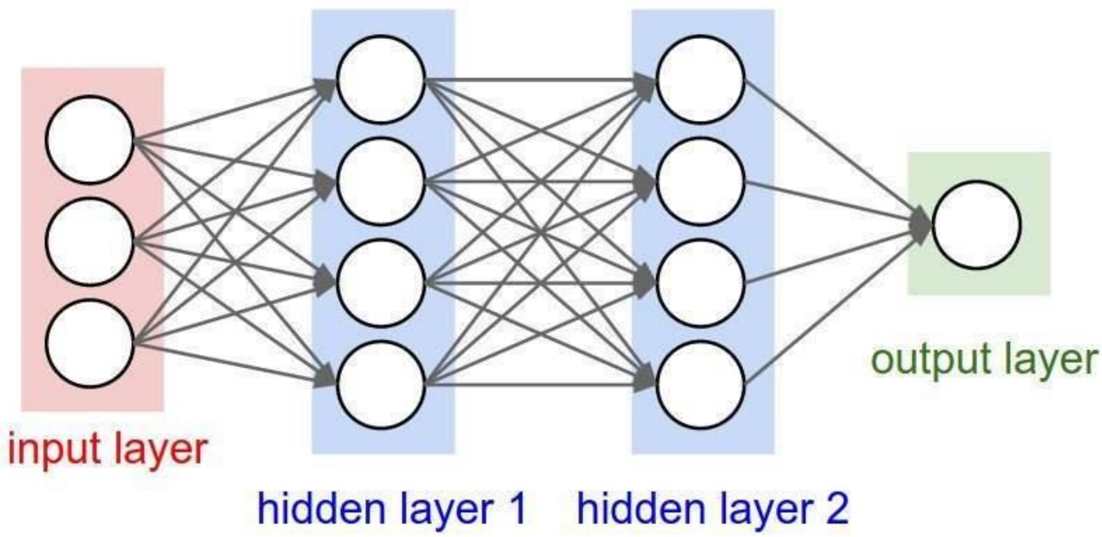

图1 神经网络示意图

人工神经网络的基本结构是神经元。图 2 的左边展示了一个生物学的神经元，右边展示了一个常用的数学模型。乍一看还是有点相似的，事实上也是，人工神经网络中的神经元也有受到生物神经元的启发。

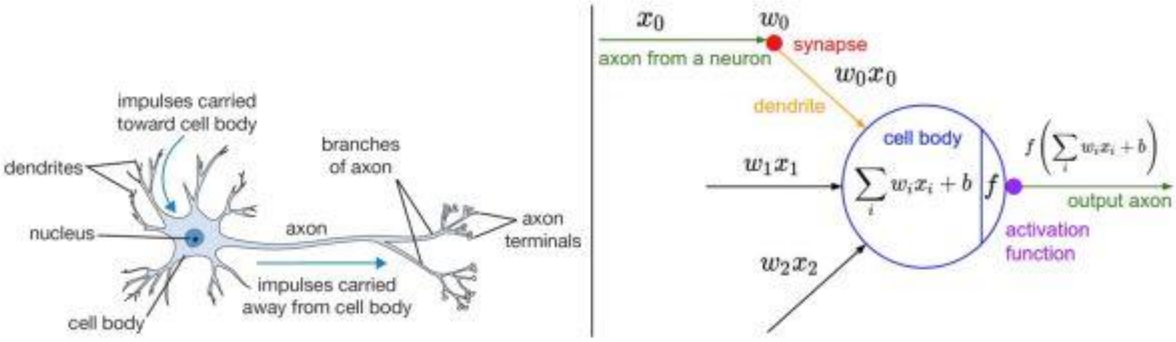

图 2 神经元示意图

它的计算过程是将所有的输入数据$x_1, x_2, x_3 \ldots\ldots x_n$分别乘以一个对应的权重参数$w_1, w_2, w_3 \ldots\ldots w_n$，（权重参数一开始是随机化赋值的，后续会通过梯度下降法进行更新，下文会讲），再加上一个偏置项 b（一开始也是随机赋值，后续梯度下降更新），最后进行激活函数的非线性操作。这样，一个神经元的工作就完成了。具体公式为$f(\sum_i^n x_i w_i + b)$。$f(x)$为激活函数

那么，为什么需要激活函数呢？

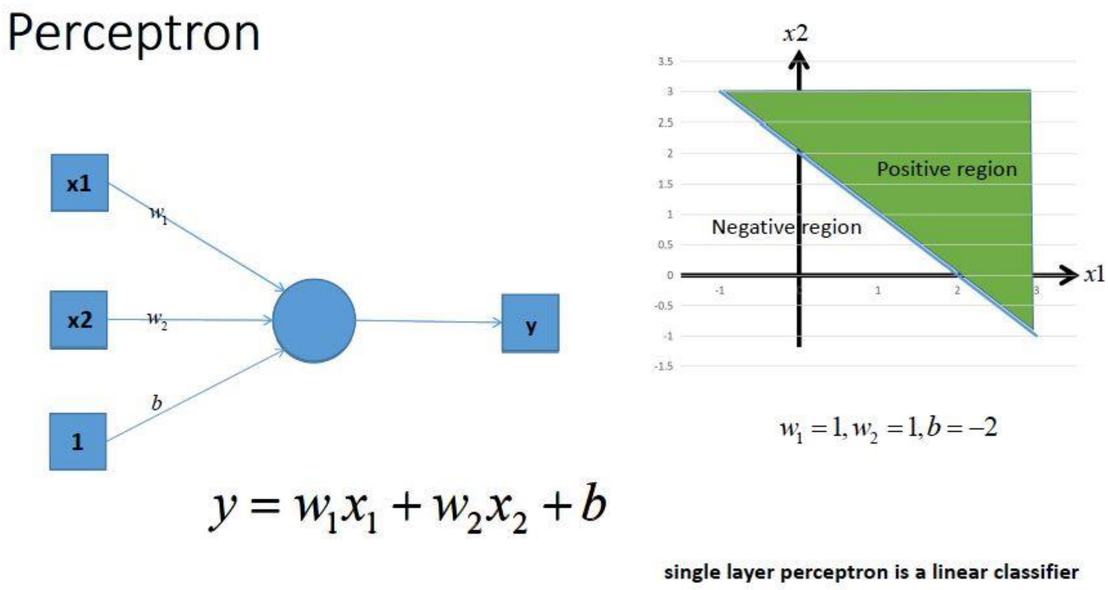

图 3 单层神经元工作原理图

举个例子，图 3 可看做普通的线性分类器，也就是线性回归方程。这个比较基础，效果在右边。当然有时候我们发现这样的线性分类器不符合我们要求时，我们很自然的想到那我们就加多一层，这样可以拟合更加复杂的函数，如下图 4：

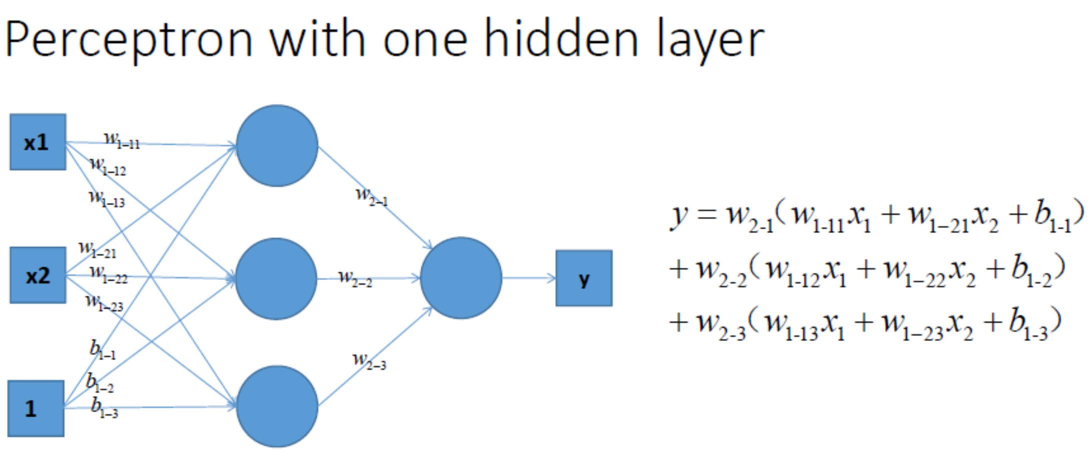

图 4 多层神经元工作原理图

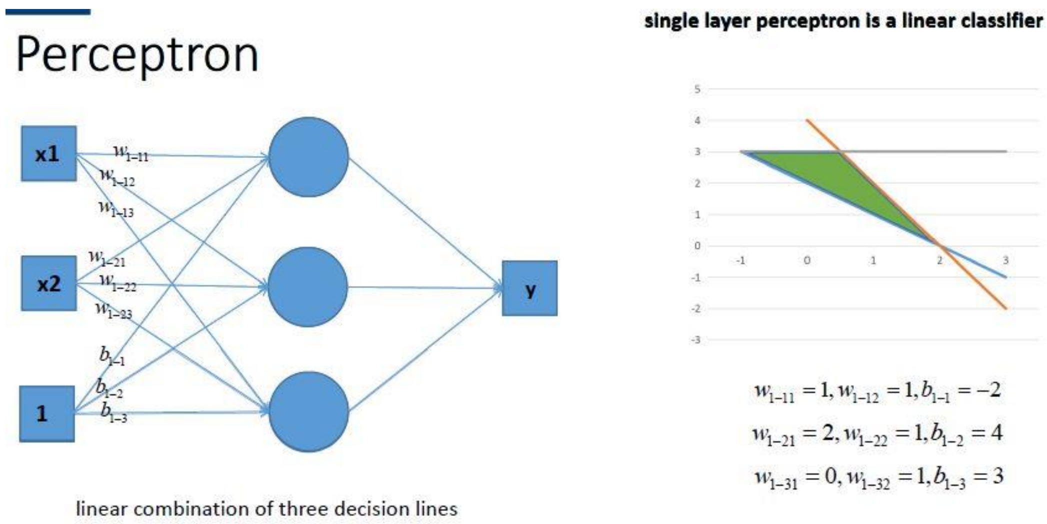

图 5 多层神经元工作结果图

但当我们动笔算下， 就会发现， 这样一个神经网络组合起来,输出的时候无论如何都还是一个线性方程。如上图 5 右边，就只能这样分类。(那也太蠢了吧)。下图表示加激活函数的情况！

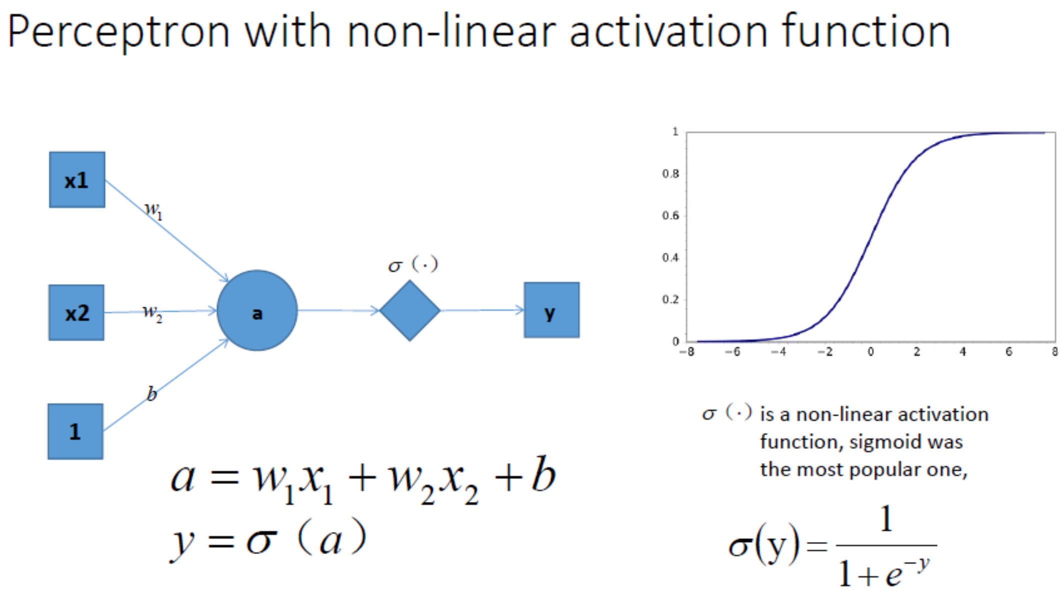

图 6 单层有激活函数的神经元工作图

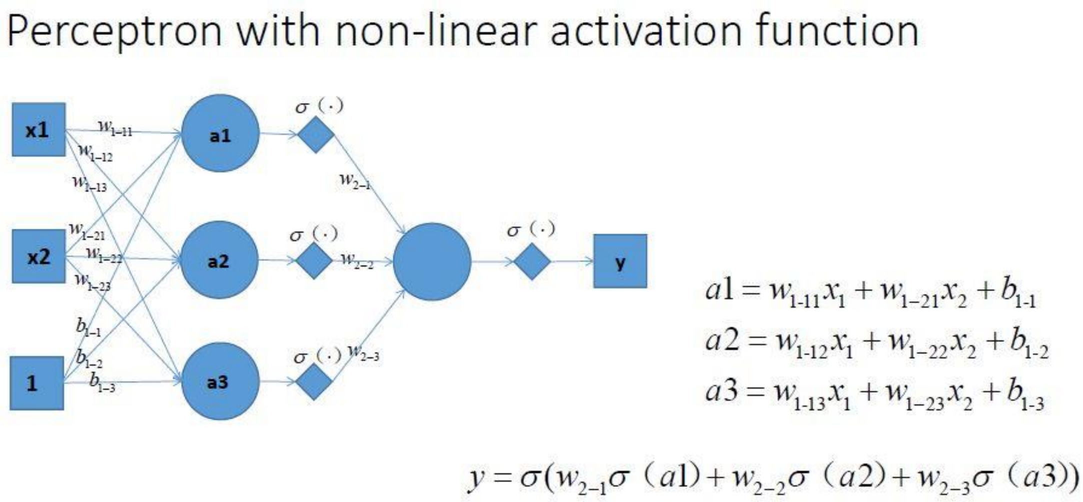

图 7 多层有激活函数的神经元工作图

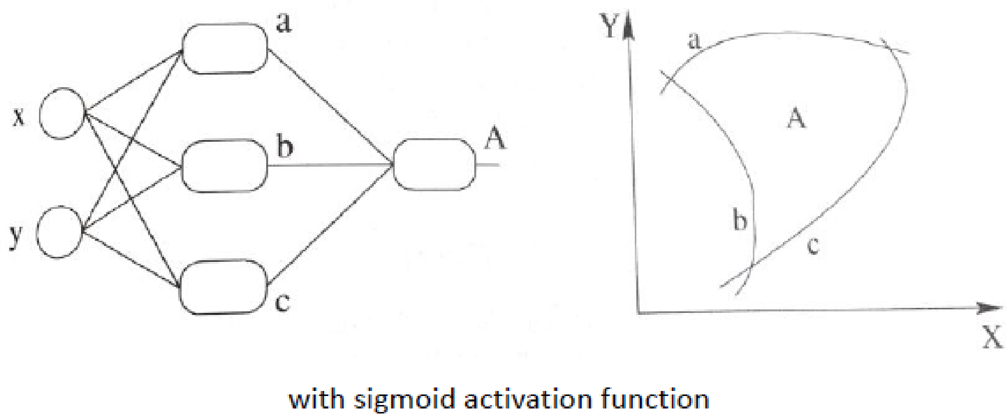

图 8 多层有激活函数的神经元结果图

图 8 的右边简单表示左图的可视化，那么对比之前的无激活函数的图，很明显是更加的非线性，拟合能力也会更强，同时可以想到，当层数更多，其能力也会越来越强！

简单来说：就是使得神经网络具有拟合非线性函数的能力，使得其具有强大的表达能力！

接着来说一下损失函数，一言以蔽之，损失函数就是用来度量模型的预测值 f(x)与真实值 Y 的差异程度的运算函数，它是一个非负实值函数，通常使用 L(Y, f(x))来表示，损失函数越小，模型的鲁棒性就越好。

损失函数的使用主要是在模型的训练阶段，每个批次的训练数据送入模型后，通过一层层的神经元输出预测值，然后损失函数会计算出预测值和真实值之间的差异值，也就是损失值。得到损失值之后，模型通过反向传播（梯度下降法）去更新各个参数，来降低真实值与预测值之间的损失，使得模型生成的预测值往真实值方向靠拢，从而达到学习的目的。

常见的损失函数有：

1， 均方误差损失函数（MSE）

计算公式为：$L(Y,f(x))=\frac{1}{n}\sum_{i=1}^{N}(f(x_i)-Y_i)^2$（n 表示样本，即输入

数据的数量，N 表示输出数据的大小规模，下文同）

在回归问题中，均方误差损失函数用于度量样本点到回归曲线的距离，通过最小化平方损失使样本点可以更好地拟合回归曲线。均方误差损失函数（MSE）的值越小，表示预测模型描述的样本数据具有越好的精确度。由于无参数、计算成本低和具有明确物理意义等优点，MSE 已成为一种优秀的距离度量方法。尽管 MSE 在图像和语音处理方面表现较弱，但它仍是评价信号质量的标准，在回归问题中，MSE 常被作为模型的经验损失或算法的性能指标。

2， 平均绝对误差函数（MAE）

计算公式为：$\boldsymbol{L(Y, f(x)) = \frac{1}{n}\sum_{i=1}^{N}|f(x_i) - Y_i|}$

MAE 相比 MSE 有个优点就是 MAE 对离群点不那么敏感，更有包容性。因为 MAE 计算的是误差 y-f(x) 的绝对值，无论是 y-f(x)>1 还是 y-f(x)<1，没有平方项的作用，惩罚力度都是一样的，所占权重一样。

实际应用中，我们应该选择 MSE 还是 MAE 呢？从计算机求解梯度的复杂

度来说，MSE 要优于 MAE，而且梯度也是动态变化的，能较快准确达到收敛。但是从离群点角度来看，如果离群点是实际数据或重要数据，而且是应该被检测到的异常值，那么我们应该使用MSE。另一方面，离群点仅仅代表数据损坏或者错误采样，无须给予过多关注，那么我们应该选择MAE 作为损失。

3， 交叉熵损失函数（CrossEntropy Loss）

计算公式为：$L(Y,f(x)) = -\sum_{i=1}^{N} f(x_i) \ln Y_i$

交叉熵是信息论中的一个概念，最初用于估算平均编码长度，引入机器学习后，用于评估当前训练得到的概率分布与真实分布的差异情况。为了使神经网络的每一层输出从线性组合转为非线性逼近，以提高模型的预测精度，在以交叉熵为损失函数的神经网络模型中一般选用 tanh、sigmoid、softmax 或 ReLU 作为激活函数。

交叉熵损失函数刻画了实际输出概率与期望输出概率之间的相似度，也就是交叉熵的值越小，两个概率分布就越接近，特别是在正负样本不均衡的分类问题中，常用交叉熵作为损失函数。目前，交叉熵

损失函数是卷积神经网络中最常使用的分类损失函数，它可以有效避免梯度消散。在二分类情况下也叫做对数损失函数。

4， softmax 损失函数

计算公式为：$L(\boldsymbol{Y},\boldsymbol{f}(\boldsymbol{x})) = -\frac{1}{n}\sum_{i=1}^{N}\ln\frac{e^{Y_i}}{\sum_{j=1}^{c}e^{f(x_{ij})}}$

从标准形式上看，softmax 损失函数应归到对数损失的范畴，在监督学习中，由于它被广泛使用，所以单独形成一个类别。softmax 损失函数本质上是逻辑回归模型在多分类任务上的一种延伸，常作为 CNN 模型的损失函数。softmax 损失函数的本质是将一个 k 维的任意实数向量 x 映射成另一个 k 维的实数向量，其中，输出向量中的每个元素的取值范围都是(0,1)，即 softmax 损失函数输出每个类别的预测概率。由于 softmax 损失函数具有类间可分性，被广泛用于分类、分割、人脸识别、图像自动标注和人脸验证等问题中，其特点是类间距离的优化效果非常好，但类内距离的优化效果比较差。

softmax 损失函数具有类间可分性，在多分类和图像标注问题中，常用它解决特征分离问题。在基于卷积神经网络的分类问题中，一般

使用 softmax 损失函数作为损失函数，但是 softmax 损失函数学习到的特征不具有足够的区分性，因此它常与对比损失或中心损失组合使用，以增强区分能力。

有了损失函数，我们就可以更新权重参数和偏置项的数据了，具体的方法为梯度下降法。

梯度下降是机器学习中的常用算法，通过不断迭代计算函数的梯度，判断该点的某一方向和目标之间的距离，最终求得最小的损失函数和相关参数，为建立线性模型提供支持。

梯度下降是一种广泛用于求解线性和非线性模型最优解的迭代算法，它的中心思想在于通过迭代次数的递增，调整使得损失函数最小化的权重。

它的作用是用于优化一个目标函数，如果要最小化一个损失函数，使用的就是梯度下降法，如果要最大化一个效用函数，使用的是梯度上升法。

简而言之：

1，梯度下降就是用来求某个函数最小值时自变量对应取值

2，损失函数就是一个自变量为算法的参数，函数值为误差值的函数。所以

梯度下降就是找让误差值最小时候算法取的参数。

梯度下降的公式为：$w_{ij} = w_{ij} - \alpha\frac{\partial L}{\partial w_{ij}}$，其中$\alpha$称作学习率，通常设置为一个很小的数值。

例如：求函数$f(x) = x^2 + y^2 + 3x - xy$的最小值

则可以轻易求出$\frac{\partial f(x)}{\partial x} = 2x + 3 - y$，$\frac{\partial f(x)}{\partial y} = 2y - x$

不妨令学习率$\alpha = 0.2$，初始值$x = 3$，$y = 3$，则不断代入梯度下降的公式，就可以得到如下表格：

迭代次数	$\frac{\partial f(x)}{\partial x}$的值	$\frac{\partial f(x)}{\partial y}$的值	x的值	y的值	$f(x)$的值
0	6	3	3	3	18
1	4.2	3	1.8	2.4	10.08
2	3.12	2.64	0.336	1.272	5.313
……	……	……	……	…….	…‥
19	0.065	0.065	-1.948	-0.948	-2.996

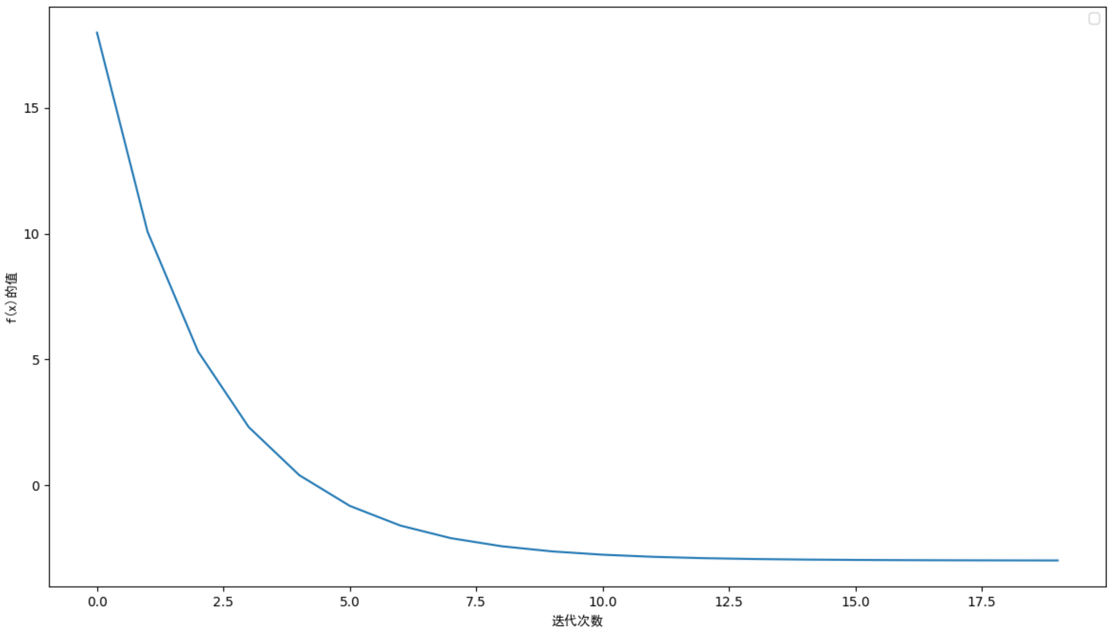

图 9 f(x)变化曲线

那么到这里，神经网络的工作原理就已经讲完了，下面举一个例子，来演示神经网络的学习过程：

假设现在我们有三组输入数据[1,2,3],[4,5,6],[2,3,4]，和三组输出数据[1,0],[0,1],[0.5,0.5]，构建如图 10 的神经网络模型。

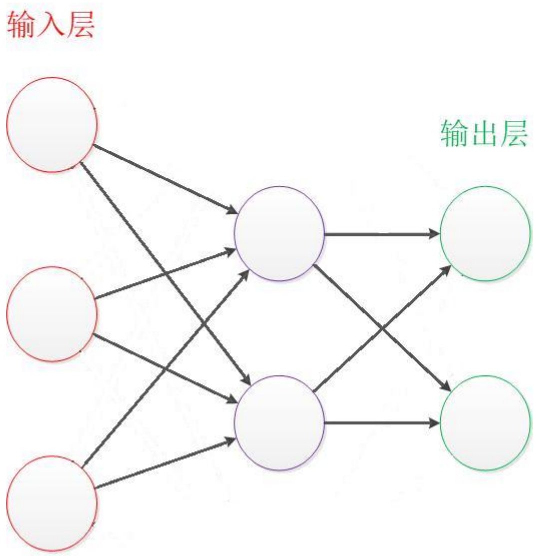

图 10 神经网络模型图

第一层选用的激活函数是 relu，第二层选用的激活函数是 sigmoid，损失函数用的是 MSE。第一层的权重参数初始值$[w_{111},w_{112},w_{113}]$，$[w_{121},w_{122},w_{123}]$为[1,0,-1],[-2,1,0]，偏置值$b_1$为 1；第二层的权重参数初始值$[w_{211},w_{212}]$，$[w_{221},w_{222}]$为[1,-1],[-1,1]，偏置值$b_2$为 1。学习率设置为 0.5。那么就可以得到如下表格：

迭代次数	w_{111}的值	w_{112}的值	w_{113}的值	w_{121}的值	w_{122}的值	w_{123}的值	w_{211}的值	w_{212}的值	w_{221}的值	w_{222}的值	b_1的值	b_2的值	L 的值

0	1	0	−1	−2	1	0	1	−1	−1	1	1	1	0.29
1	0.85	−0.19	−1.23	−2.05	0.9	−0.15	1	−0.98	−1	0.97	0.91	0.95	0.23
2	0.70	−0.38	−1.45	−2.09	0.82	−0.27	1	−0.98	−1	0.96	0.83	0.91	0.21
……	……	……	……	……	……	……	……	……	……	……	……	……	……
50	−6.93	−9.91	−12.89	−3.94	−2.88	−5.82	1	−0.98	−1	0.96	−2.92	0.06	0.17

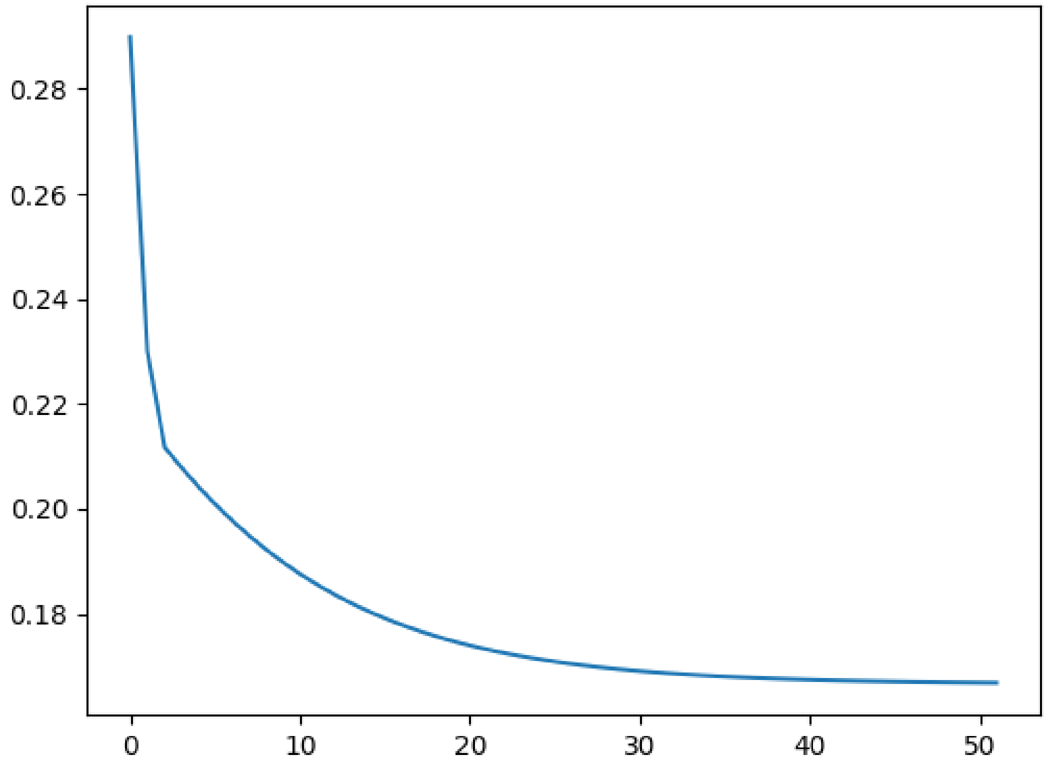

图 11 损失函数变化曲线

参考文章：[神经网络（容易被忽视的基础知识） - 知乎 (zhihu.com)]

[损失函数（Loss Function） - 知乎 (zhihu.com)]

本文地址：[TLearning (caodong0225.github.io)]