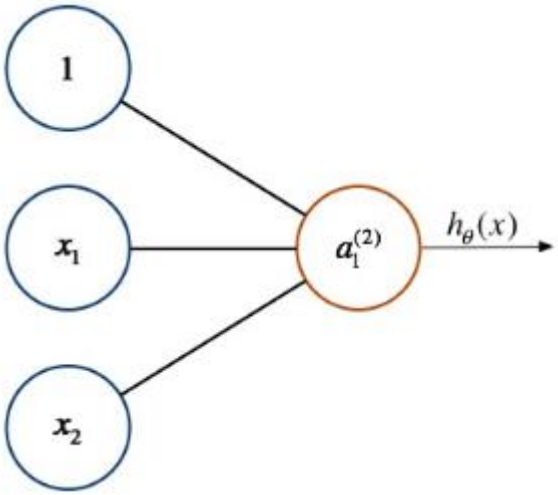
一、 MLP 算法简介

MLP 全称是 Multi-layer Perceptron，也就是多层感知器，是一种前馈人工神经网络，映射一组输入向量到一组输出向量。MLP 可以被看做是一个有向图，由多个节点层组成，每一层全连接到下一层。除了输入节点，每个节点都是一个带有非线性激活函数的神经元（处理单元 unit）。一种被称为反向传播算法的监督学习方法常被用来训练 MLP。MLP 是感知器的推广，克服了感知器不能对线性不可分数据进行识别的弱点。

那么我们首先来看一下什么是感知器。感知机(perceptron)是二类分类的线性分类模型，其输入为实例的特征向量，输出为实例的类别，取值为+1 和-1 两个值，对应于输入空间中将实例划分为正负两类的分离超平面，属于判别模型。感知机学习旨在求出将训练数据进行线性划分的分离超平面，为此，导入基于误分类的损失函数，利用梯度下降法对损失函数进行极小化，求得感知机模型。感知机学习算法具有简单，并且易于实现的特点，分为原始形式和对偶形式。感知机预测是用学习得到的感知机模型对新的输入实例进行分类。它是神经网络与支持向量机的基础。感知器是最简单的人工神经网络，只有一个神经元，不包括隐藏层，结构如下图所示。



二、 感知机模型

假设输入 X 在 n 维特征空间中，输出 Y 的取值为{ +1,1 }，假设小写的 x 表示一个实例的特征向量，对应于输入特征空间中的点，函数值 f(x)表示输出实例的类别，把由输入特征空间到输出空间的如下函数 f(x)称为感知机：

$$f(x) = sign(w \cdot x + b)$$

其中 w 和 b 为感知机的参数，由于 x 是 n 维的，$w \cdot x$是内积，w 当然也是 n 维的 ,和输入特征空间的维数相对应 ,把 w 经常称为权值向量(weight vector)或权值(weight)，$w \cdot x$是内积得到的是一个实数值，b 是一个实数值，通常把 b 叫作偏置(bias)。sign 是符号函数，对应于两个类别值+1,1。

假设训练数据集是线性可分的，感知机学习的目标是能够求得一个能够将训练集正实例点和负实例点完全正确分开的分离超平面，也就是能够求解出正确的 w 和 b 的值。我们知道在监督学习问题中是将预测值 f(x)和真实值 Y 的不一致或者错误程度定义了一个损失函数，通过最小化损失函数来求解 f(x)的参数，这里对于二分类的感知机而言，感知机的损失函数就是误分类的点到超平面的总距离。

首先，我们知道对于线性方程$w \cdot x + b = 0$而言，n 维输入特征空间中的一点 x0 到超平面的距离为：

$$\frac{1}{\|w\|}|w \cdot x_0 + b|$$

对于误分类的样本来说有$-y_i(w \cdot x + b) > 0$,如果有 M 个误分类的点，那么误分类的点到超平面的距离总和是：

$$-\frac{1}{\|w\|}\sum_{x_i \in M} y_i(w \cdot x + b)$$

由此可以得到感知机的损失函数为：

$$L(w\ ,\ b) = -\sum_{x_i \in M} y_i(w \cdot x + b)$$

显然，误分类点越少，且误分类点离超平面越近，损失函数值就越小，一般采用随机梯度下降法来求解使损失函数最小的 w 和 b。

首先任意取一个w_0，b_0，用梯度下降法不断极小化目标函数，极小化过程不是一次就使 M 中所有误分类的点的梯度下降，而是一次随机选取一个误分类点使其梯度下降。

假设误分类点集合 M 是固定的，那么损失函数$L(w\ ,\ b)$的梯度就由以下条件给出：

$$\nabla_w L\left(w\ ,\ b\right) = -\sum_{x_i \in M} y_i x_i$$

$$\nabla_b L\left(w\ ,\ b\right) = -\sum_{x_i \in M} y_i$$

随机选取一个误分类点（x_i，y_i）对 w 和 b 进行更新，就可以得到

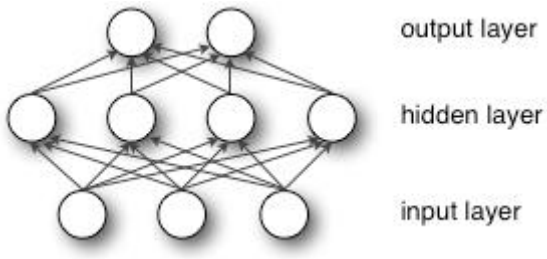
$$w < -w + \eta y_i x_i$$

$$b < -b + \ \eta y_i$$

其中的η是步长，又叫做学习率。通过迭代就可以使期待损失函数不断减小，直到为 0.

三、 多层感知机

多层感知机（MLP，Multilayer Perceptron）也叫人工神经网络（ANN，Artificial Neural Network），除了输入输出层，它中间可以有多个隐层，最简单的 MLP 只含一个隐层，即三层的结构，如下图：



从上图可以看到，多层感知机层与层之间是全连接的（全连接的意思就是：上一层的任何一个神经元与下一层的所有神经元都有连接）。多层感知机最底层是输入层，中间是隐藏层，最后是输出层。

输入层中，如果输入一个 n 维向量，那么就有 n 个神经元；

隐藏层和输入层是全连接的，如果输入了向量 X，那么隐藏层的输出就是 $f(w_1x+b_1)$，其中w_1是权重，b_1是偏置，函数 f 可以是常见的 sigmoid 函数等。

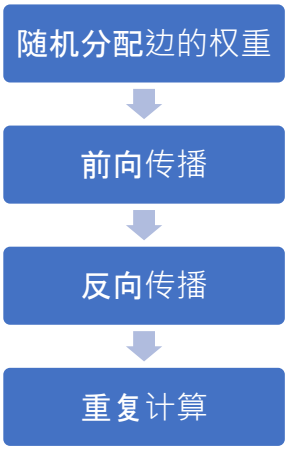
隐藏层到输出层可以看做是一个多类别的逻辑回归，也就是 softmax 回归，因此输出层的输出实际就是 softmax（$w_2x_1+b_2$），x_1表示隐藏层的输出 $f(w_1x+b_1)$。

以上三个层总结起来就是公式

$$f(x)=G(b^{(2)}+W^{(2)}\left(s\left(b^{(1)}+W^{(1)}x\right)\right))$$

关于参数仍然是采用梯度下降法，和感知机中的方法一样。

四、 MLP 算法的流程



1. 所有边的权重随机分配；

2. 前向传播：利用训练集中所有样本的输入特征，作为输入层，对于所有训练数据集中的输入，人工神经网络都被激活，然后经过前向传播，得到输出值。

3. 反向传播：利用输出值和样本值计算总误差，再利用反向传播来更新权重。

4. 重复 2~3，直到输出误差低于制定的标准。

五、 MLP 算法的优缺点

优点

- 高度的并行处理；
- 高度的非线性全局作用；
- 良好的容错性；
- 具有联想记忆功能；

非常强的自适应、自学习功能。

缺点

- 网络的隐含节点个数选取非常难；
- 计算成本高，极其耗时；
- 学习速度慢；
- 容易陷入局部极值；
- 学习可能会不够充分。

六、 总结：

MLP 多层感知器是一种前向结构的 ANN 人工神经网络，它能够处理非线性可分离的问题，值得深入研究。为了实现 MLP 多层感知器，会用到 BP 反向传播算法。MLP 可使用任何形式的激活函数，但为了使用反向传播算法进行有效学习，激活函数必须限制为可微函数。MLP 算法应用范围较广，扩展性也强，可应用于语音识别、图像识别、机器翻译等领域。