

# 多层感知器

维基百科，自由的百科全书

**多层感知器**（Multilayer Perceptron,缩写MLP）是一种前向结构的人工神经网络，映射一组输入向量到一组输出向量。MLP可以被看作是一个有向图，由多个的节点层所组成，每一层都全连接到下一层。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（或称处理单元）。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。<sup>[1][2]</sup> 多层感知器遵循人類神經系統原理，學習並進行數據預測。它首先學習，然後使用權重存儲數據，並使用算法來調整權重並減少訓練過程中的偏差，即實際值和預測值之間的誤差。主要優勢在於其快速解決複雜問題的能力。多層感知的基本結構由三層組成：第一輸入層，中間隱藏層和最後輸出層，輸入元素和權重的乘積被饋給具有神經元偏差的求和結點,主要優勢在於其快速解決複雜問題的能力。<sup>[3]</sup> 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。<sup>[4]</sup>

## 目录

### 理论

[激活函数](#)

[层](#)

[学习](#)

### 术语

### 应用

### 文献

## 理论

### 激活函数

若每个神经元的激活函数都是线性函数，那么，任意层数的MLP都可被约简成一个等价的单层感知器。<sup>[5]</sup>

实际上，MLP本身可以使用任何形式的激活函数，譬如阶梯函数逻辑Sigmoid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。由于具有良好可微性，很多S函数，尤其是双曲正切函数（Hyperbolic tangent）及逻辑函数，被采用为激活函数。

在深度学习的最新发展中，线性整流(ReLU)更频繁地被用来克服与S函数相关的数值问题。

两个历史上常见的激活函数都是 S函数，形式是

$y(v_i) = \tanh(v_i)$  和  $y(v_i) = (1 + e^{-v_i})^{-1}$ 。

第一个是个双曲正切函数，值域为 -1 到 1；第二个是个逻辑函数，形状很相似但是值域为 0 到 1。令  $y_i$  为第  $i$  个节点（神经元）的输出，而  $v_i$  是输入连接的加权和。也有其他的激活函数，例如线性整流函数，径向基函数（用于径向基函数网络，另一种监督神经网络模型）。

## 层

MLP由三层或更多层非线性激活节点组成(一个输入层和一个具有一个或多个隐藏层的输出层)。由于多层互连是完全连接的，所以一层中的每个节点都以一定的权重  $w_{ij}$  连接到下一层的每个节点。

## 学习

MLP 在感知器中进行学习，通过每次处理数据后改变连接权重，降低输出与预测结果的误差量。这是有监督学习的一个例子，通过反向传播来实现，反向传播是线性感知器中最小均方算法的推广。

我们可以将输出节点  $j$  的第  $n$  个数据点的误差表示为  $e_j(n) = d_j(n) - y_j(n)$ ，其中  $d$  是目标值， $y$  是由感知器预测的值。调整节点权重的方式是，尝试通过修正节点权重最小化输出的整体误差

$$\mathcal{E}(n) = \frac{1}{2} \sum_j e_j^2(n).$$

使用梯度下降，每个权重的修正量为

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n)$$

其中  $y_i$  是前一个神经元的输出， $\eta$  是学习率。 $\eta$  需要精心挑选，保证权重可以快速收敛而不发生震荡。

式中的导数取决于局部场  $v_j$ 。场是变化的。很容易证明输出节点的导数可以简化为

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = e_j(n) \phi'(v_j(n))$$

其中  $\phi'$  是激活函数的导数。 $\phi'$  是不变的。对于隐藏节点的权重变化，分析更加困难，但是可以看出相关的导数是

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = \phi'(v_j(n)) \sum_k -\frac{\partial \mathcal{E}(n)}{\partial v_k(n)} w_{kj}(n).$$

代表输出层的第  $k$  个节点的权重变化会影响这个导数。因此，为了改变隐藏层权重，输出层权重根据激活函数的导数而改变，因此该算法代表激活函数的反向传播<sup>[6]</sup>。

## 术语

术语“多层感知器”不是指具有多层的单感知器，每一层由多个感知器组成。另一种说法是“多层感知器网络”。此外，MLP的“感知器”不是最严格意义上的感知器。真正的感知器在形式上是人工神经元的一个特例，它使用一个阈值激活函数，如阶跃函数。MLP感知器可以使用任意激活函

数。一个真正的感知器执行二进制分类(或者这个或者那个)，一个MLP神经元可以自由地执行分类或者回归，这取决于它的激活函数。

后来应用术语“多层感知器”时，没有考虑节点/层的性质，节点/层可以由任意定义的人工神经元组成，而不是具体的感知器。这种解释避免了将“感知器”的定义放宽到一般意义上的人工神经元。

## 应用

---

常被MLP用来进行学习的反向传播算法，在模式识别的领域中算是标准监督学习算法，并在计算神经学及并行分布式处理领域中，持续成为被研究的课题。MLP已被证明是一种通用的函数近似方法，可以被用来拟合复杂的函数，或解决分类问题。

MLP在80年代的时候曾是相当流行的机器学习方法，拥有广泛的应用场景，譬如语音识别、图像识别、机器翻译等等，但自90年代以来，MLP遇到来自更为简单的支持向量机的强劲竞争。近来，由于深度学习的成功，MLP又重新得到了关注。

## 文献

---

1. Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961
2. Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams.“Learning Internal Representations by Error Propagation”. David E. Rumelhart, James L. McClelland, and the PDP research group. (editors) , Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations. MIT Press, 1986.
3. Sustainable Construction Safety Knowledge Sharing: A Partial Least Square-Structural Equation Modeling and A Feedforward Neural Network Approach, Sustainability 2019, 11(20), 5831; <https://doi.org/10.3390/su11205831>
4. Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function *Mathematics of Control, Signals, and Systems*, 2 (4) , 303–314.
5. Neural Networks for pattern recognition 第一版. Oxford University Press. 1995. ISBN 0198538642.
6. Haykin, Simon. Neural Networks: A Comprehensive Foundation 2. Prentice Hall. 1998. ISBN 0-13-273350-1.

---

取自“<https://zh.wikipedia.org/w/index.php?title=多层感知器&oldid=71675660>”

---

本页面最后修订于2022年5月17日 (星期二) 05:51。

本站的全部文字在知识共享 署名-相同方式共享 3.0协议之条款下提供，附加条款亦可能应用。（请参阅使用条款）  
Wikipedia®和维基百科标志是维基媒体基金会的注册商标；维基™是维基媒体基金会的商标。  
维基媒体基金会是按美国国内稅收法501(c)(3)登记的非营利慈善机构。