



北京航空航天大学
BEIHANG UNIVERSITY

模式识别与机器学习大作业

神经网络与反向传播

院(系)名称	高等理工学院
学 号	16231235
姓 名	李谨杰
指 导 教 师	秦曾昌

2019 年 6 月

目录

一、引言.....	1
二、BP 神经网络原理讲解.....	1
2.1 神经网络.....	1
2.2 反向传播.....	1
2.3 Iris 数据集简介.....	2
2.4 自编码器（AutoEncoder）.....	2
三、神经网络算法详解.....	3
四、实验结果及讨论.....	3
Stage1.....	3
Stage2.....	4
五、收获、体会及建议.....	7
六、参考文献.....	7

一、引言

人工神经网络 (Ann) (或神经网络 (NN)) 的学习模式是一种受到生物学神经系统启发的方法, 这种系统在无需借助明确的定量操作的情况下, 在真实世界上进行复杂的计算。这个技术最初的灵感来自于对大脑中生物电网络的检查, 这些网络由神经元和突触组成。在一个神经网络模型中, 简单的节点 (称为“神经”或“单位”) 是相互连接的, 形成一个节点网络, 因此我们将这种网络成为“神经网络”。

每个节点都有一组输入, 这些输入类似于生物神经元中的输入突触。每个节点也有一个“激活功能”, 告诉节点什么时候开火, 类似于一个生物神经元。在最简单的形式中, 如果求和输入大于某个值, 则此激活函数只能生成“1”, 否则生成“0”。然而, 激活功能并不一定要这么简单——事实上, 要创建可以做有用事情的神经网络, 它们几乎总是必须更加复杂, 至少对于网络中的一些节点来说。通常, 前馈神经网络至少有三层——输入层、隐藏层和输出层。输入层不需要处理, 它只是把数据输入网络的地方。数据然后从输入层进入隐藏层。而隐藏层依次馈送到输出层。网络中的实际处理发生在隐藏层和输出层的节点中。

二、BP 神经网络原理讲解

2.1 神经网络

任何有监督学习算法的目标都是找到一个最能将一组输入映射到其正确输出的函数。例如一个简单的分类任务, 其中输入的是动物的图像, 正确的输出是动物的名称。对于一个直观的例子, 神经网络的第一层可能负责使用图像中单个像素的输入来学习图像中线条的方向。第二层可以结合在第一层中学习的特征, 学习识别简单的形状, 如圆。每一个更高层次都会学习到越来越多的抽象特征, 比如上面提到的那些可以用来对图像进行分类的特征。每一层都在其下面的层中找到模式, 并且正是这种能力创建了独立于外部输入的内部表示, 从而为多层网络提供了强大的功能。开发反向传播算法的目的是找到一种训练多层神经网络的方法, 使其能够学习适当的内部表示, 从而使其能够学习输入到输出的任意映射。

在数学上, 神经元的网络函数 $f(x)$ 被定义为其他函数 $g_i(x)$ 的组合。这可以方便地表示为网络结构, 箭头表示变量之间的依赖关系。广泛使用的组成类型是非线性加权和, 其中:

$$f(x) = K \left(\sum_i w_i g_i(x) \right)$$

公式中的 K (一般代指激活函数) 是一些预定义的函数, 例如双曲正切函数。

下面将函数 g_i 的集合称为向量 $g = (g_1, g_2, \dots, g_n)$, 这很方便。

2.2 反向传播

反向传播算法需要为每个输入值提供一个已知的、所需的输出, 以便计算损失函数梯度。因此, 它通常被认为是一种有监督的学习方法。

平方误差函数为:

$$E = \frac{1}{2}(t - y)^2$$

其中 E 是平方误差, t 是训练样本的目标输出, y 是输出神经元的实际输出。对于每个神经元 j , 其输出 o_j 定义为

$$o_j = \varphi(\text{net}_j) = \varphi\left(\sum_{k=1}^n w_{kj} x_k\right)$$

神经元的输入网络是以前神经元输出 O_K 的加权和。如果神经元位于输入层之后的第一层, 则输入层的 O_K 等同于网络的输入 x_k 。神经元的输入单位数为 n 。变量 w_{ij} 表示神经元 i 和 j 之间的权重。

激活函数 ϕ 一般是分线性和可微的。一种常用的激活函数是 logistic 函数, 例如:

$$\varphi(z) = \frac{1}{1 + e^{-z}}$$

这种函数有一种很好的微分形式:

$$\frac{\partial \varphi}{\partial z} = \varphi(1 - \varphi)$$

两次使用链式法则计算误差对权重 w_{ij} 的偏导数。

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} \frac{\partial \text{net}_j}{\partial w_{ij}}$$

最终我们可以得到:

$$\frac{\partial E}{\partial w_{ij}} = \delta_j x_i$$

以及:

$$\delta_j = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j} = \begin{cases} (o_j - t_j) \varphi(\text{net}_j) (1 - \varphi(\text{net}_j)) & \text{if } j \text{ is an output neuron} \\ (\sum_{l \in L} \delta_l w_{jl}) \varphi(\text{net}_j) (1 - \varphi(\text{net}_j)) & \text{if } j \text{ is an inner neuron} \end{cases}$$

2.3 Iris 数据集简介

Iris 数据集是常用的分类实验数据集, 由 Fisher, 1936 收集整理。Iris 也称鸢尾花卉数据集, 是一类多重变量分析的数据集。数据集包含 150 个数据集, 分为 3 类, 每类 50 个数据, 每个数据包含 4 个属性。可通过花萼长度, 花萼宽度, 花瓣长度, 花瓣宽度 4 个属性预测鸢尾花卉属于 (Setosa, Versicolour, Virginica) 三个种类中的哪一类。

2.4 自编码器 (AutoEncoder)

自动编码器是一种数据的压缩算法, 其中数据的压缩和解压缩函数是数据相关的、有损的、从样本中自动学习的。在大部分提到自动编码器的场合, 压缩和解压缩的函数是通过神经网络实现的。

自编码器是一个自监督的算法, 并不是一个无监督算法。自监督学习是监督学习的一个实例, 其标签产生自输入数据。目前自编码器的应用主要有两个方面, 第一是数据去噪, 第二是为进行可视化而降维。配合适当的维度和稀疏约束, 自

编码器可以学习到比 PCA 等技术更有意思的数据投影。

三、神经网络算法详解

如果调用已有的程序包，只需要先训练神经网络，后预测数据。误差逆传播算法细节如下：

首先输入训练集和学习率。

- (1) 在 0-1 范围内随机初始化网络中所有连接权和阈值
- (2) 根据当前参数计算样本输出 y
- (3) 按链式法则计算输出层神经元的梯度项 g
- (4) 按链式法则计算隐层神经元的梯度项 e
- (5) 更新连接权与阈值
- (6) 循环，直至达到终止条件

四、实验结果及讨论

Stage1

1. 我选定 IRIS 数据集进行分类，构建单隐含层神经网络进行训练。我训练的网络结构如图 5.1。随机选用 105 个数据训练，剩下 45 个数据检测。

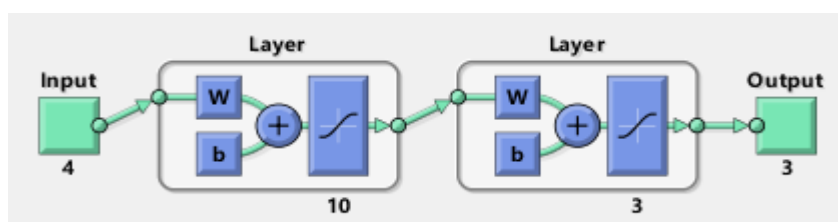
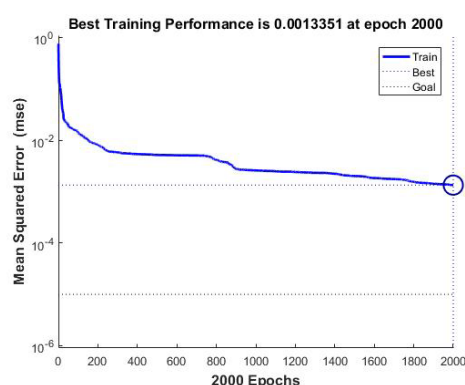


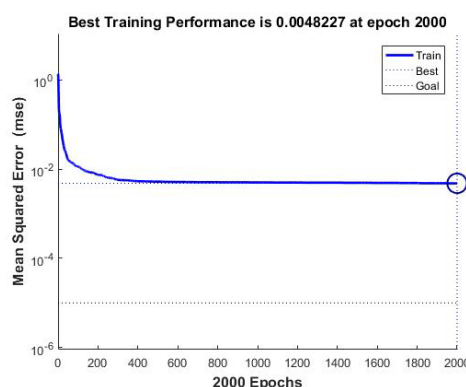
图 5.1 神经网络结构

输入层有 4 个节点，隐含层有 10 个节点，输出层有 3 个节点，改变学习率，迭代 2000 次，结果如下：

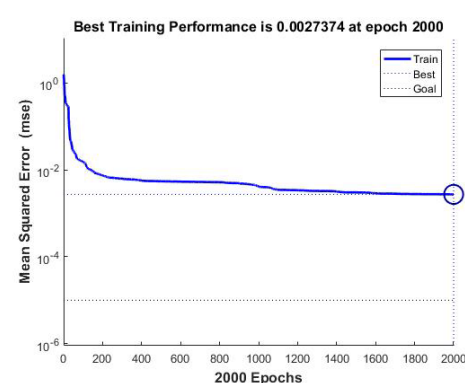
α	0.05	0.1	0.2	0.3	0.4	0.5
正确率	97.78%	95.56%	95.56%	97.78%	95.56%	93.33%



$\alpha = 0.2$



$\alpha = 0.3$



$\alpha = 0.4$

图 5.2 不同学习率下的误差训练曲线

根据上述实验可以得出如下结论：

- (1) 学习率对最终结果产生一定的影响：学习率较小时，需要较长的学习时间才能达到更大学习率相同的结果；学习率过大，会造成过拟合，最终分类正确率反而有所下降。
- (2) 由于我模型的初始权值是随机赋值的，每一次的训练结果不尽相同，分类正确率在 93%~98%间摆动，因而神经网络的训练具有一定的随机性。
- (3) 有时模型的均方误差可以下降到很小的值，但有时又会停滞在 0.005 左右，这说明此时神经网络陷入了局部最小值，难以得到全局最优解。

2. 与感知机模型进行对比：

由于单神经元的感知机只能解决二分类问题，我用三个神经元构建感知机与三层神经网络进行对比，训练过程如图 5.3。

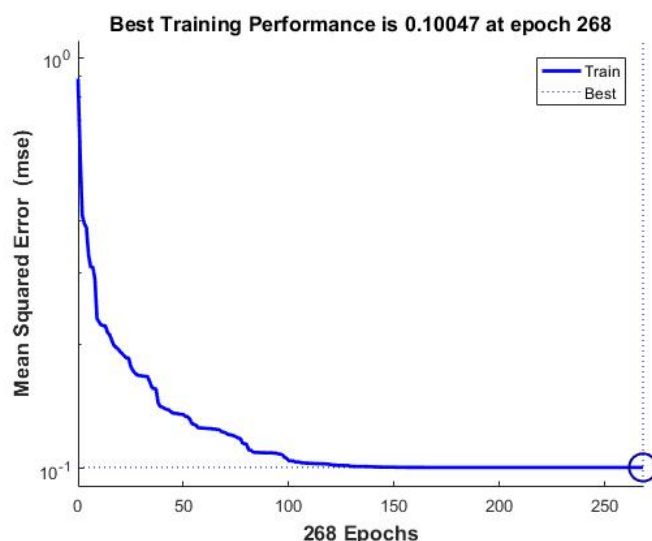


图 5.3 感知机训练过程

设置学习率为 0.1，迭代 2000 次，训练，分类相同的数据，性能列表如下：

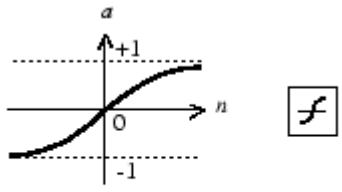
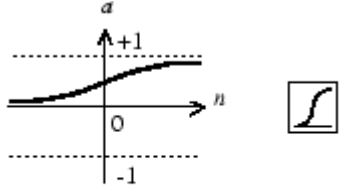
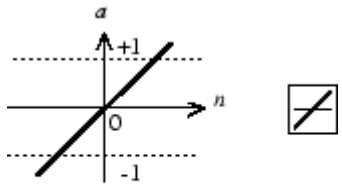
	分类正确率	MSE	运行时间/s	达到最佳结果的周期
感知器	82.22%	0.1005	2.1400	403
BP 神经网络	95.56%	0.0051	12.7095	2000

综上，对于 IRIS 数据集这样的非常简单的分类问题，神经网络的分类正确率略优于感知器，但感知器的计算速度要比 BP 神经网络快许多。当分类问题由 IRIS 变为图像这样的复杂问题之后，也许神经网络的性能会大大超越多层感知器。

Stage2

1. Matlab 的激活函数有 TANSIG, LOGSIG, 和 PURELIN 三种。其基本特性列表如下：

缩写	TANSIG	LOGSIG	PURELIN
全称	双曲正切 S 型(Tan-Sigmoid)传输函数	对数 S 型(Log-Sigmoid)传输函数	线性激活函数
表达式	$a = \frac{2}{1 + \exp(-2 * n)} - 1$	$a = \frac{1}{1 + \exp(-n)}$	$a = n$

图像	 $a = \text{tansig}(n)$ Tan-Sigmoid Transfer Function	 $a = \text{logsig}(n)$ Log-Sigmoid Transfer Function	 $a = \text{purelin}(n)$ Linear Transfer Function
----	--	---	--

选择三层 BP 网络，输入层有 4 个节点，隐含层有 10 个节点，输出层有 3 个节点，同时改变隐含层和输出层的激活函数，学习率设为 0.1，迭代 2000 次，结果如下：

激活函数	TANSIG	LOGSIG	PURELIN
正确率	95.56%	91.11%	82.22%
停止次数	2000	1062	284
MSE	0.0051	1.74e-06	0.0935

其中，logsig 函数的训练过程如下：

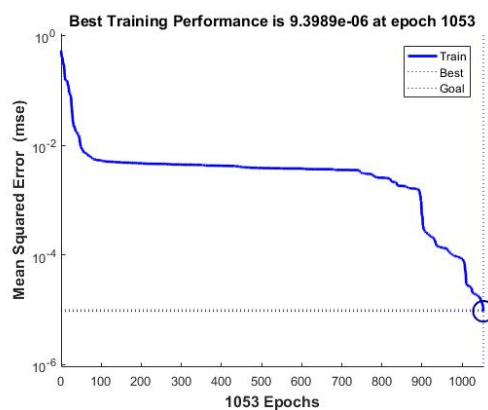


图 5.4 logsig 激活函数训练过程

如图可知发生了过拟合。将训练次数改为 500 次，则分类正确率上升到 93.33%。

综上，不同的激活函数对效率和准确性产生影响。训练效率 PURELIN>LOGSIG>TANSIG，训练准确性 TANSIG>LOGSIG>PURELIN，但 LOGSIG 可以达到更优的拟合结果。因此，激活函数常用 TANSIG 或者 LOGSIG，配以合适的学习率提升训练效率。

将隐含层改为两层，每层十个节点，训练正确率为 95.56%，MSE 为 0.0022，训练过程图如图 5.5：

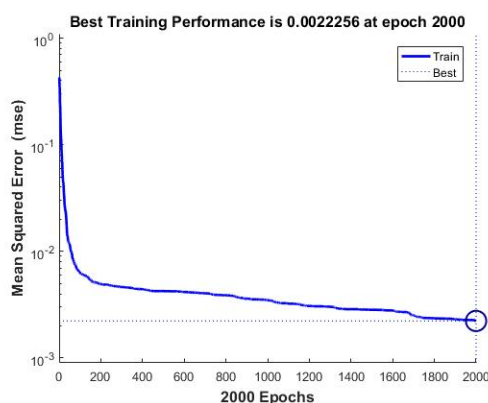


图 5.5 两层隐含层训练过程

将隐含层改为三层，每层十个节点，训练正确率为 91.11%，MSE 为 4.88e-07，训练过程图如图 5.6:

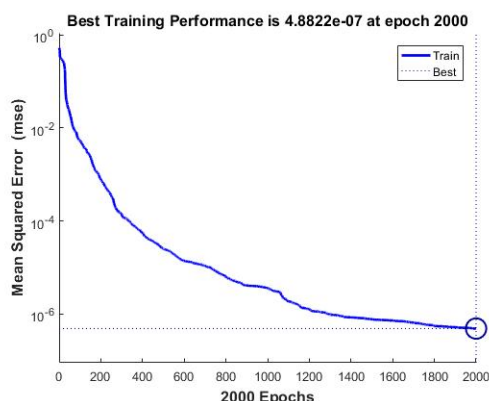


图 5.6 三层隐含层训练过程

从如上实验可以看出，其他条件相同，增加节点相同的隐含层，会增大学习速率，使得神经网络更容易拟合复杂的映射关系，但也更容易陷入过拟合的问题中，造成准确率的下降。因此，要合理选择隐含层的层数与训练的次数，确保得到最优解。

2. 我认为生物神经网络的工作方式与神经网络模型接近，但还有诸多不一致的地方，后期神经网络模型的发展可能与生物神经网络差距较大。我观看了 Hinton 教授 15 年 6 月在剑桥的一个讲座，结合网友对该讲座的讨论，得到以下结论：认知这两者的异同，需要回答生物神经学界提出的四大否定理论：

1. 生物神经元如何进行数值计算？（Cortical neurons do not communicate real-valued activities）

这点被认为是通过神经元发出的电信号的频率、强度模拟数字计算体系。

2. 生物神经元如何求导？（How do neurons know $\frac{dy}{dx}$ (Gradient)）

生物神经元有一种通过简单观测，来修正参数的方法，这种方法与梯度下降算法原理不同，效果相近。

$$\Delta \text{Neuron}(t) = y + t(\text{Target} - y) \text{ where } t = 1, 2, 3, \dots, \text{Time}$$

3. 神经元是否需要发射两种信号，包括前向传播和反向传播？（The neurons need to send two different types of signal）

生物神经元可能存在反向逆权值，通过这个逆权值，建立循环连接，而不是硬件式地直接跳回去，这也是一个生物与计算机神经网络的区别。

4. 生物神经网络元没有同值的互连接（Neurons do not have point-wise reciprocal connections with the same weight in both directions.）。

由上述的几个问题及讨论可知，神经网络模型受生物启发而诞生，但由于计算机与生物性质的众多不同，在许多方面还存在很大差异。后期的深度学习模型如 RNN、GAN 等也与人脑机制相差较大。

（3）编码器

我们将训练最简单的自动编码器，它是一个前馈、非循环神经网络，有一个输入层、一个输出层和一个或多个连接它们的隐藏层。不同之处在于，对于自动

编码器，输出层具有与输入层相同的节点，不是训练它预测给定输入 x 的目标值 y ，而是训练自动编码器重建自己的输入 x 。因此，一般自动编码器中间神经元节点数要少于输入节点数。

我修改网络结构如图 5.7:

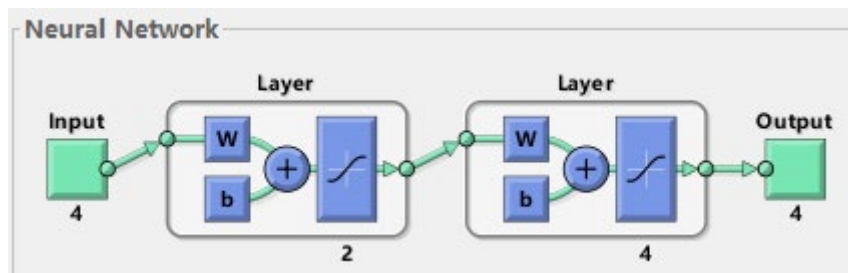


图 5.7 一层隐含层的自动编码器网络

自动编码器的作用是，先从输入值提取特征，再由提取的特征反向重建输入，与真实输入对比，以此来判断特征提取的优劣。调整学习率为 0.01，训练次数为 2000 次，激活函数选择 `tansig` 函数，更改隐层的节点数，得到实验结果如下：

隐层节点数	1	2	3	4	5
MSE	9.9698	9.9216	9.9216	9.9698	9.9215

可以看出隐藏层中节点的数量对模型的性能影响不大。

保持隐层每层均有 2 个节点，改变隐层层数，得到实验结果如下：

隐层层数	1	2	3	4
MSE	9.9216	9.9215	9.9698	9.9698

可以看出隐含层的层数对于模型性能影响也不大。

五、收获、体会及建议

经过本次实验，我深入理解了 BP 神经网络的原理，并将其性能与感知器进行对比，还学习了自动编码器。我将课上讲的知识学以致用，锻炼了编程能力，收获很大。机器学习的算法有很多，只有付诸实践才能真正领会这些算法的精神。

六、参考文献

- [1] 机器学习课件 秦曾昌
- [2] 周志华. 机器学习. 清华大学出版社
- [3] 百度百科: IRIS (IRIS 数据集) <https://baike.baidu.com/item/IRIS/4061453>
- [4] 实验指导书: Guides to a Series of Experiments
- [5] MATLAB 文档
- [6] 神经网络模型算法与生物神经网络的最新联系 <https://www.cnblogs.com/neopenx/p/4756246.html>
- [7] 深度学习之自编码器 AutoEncoder <https://blog.csdn.net/marsjhao/article/details/73480859>