



北京航空航天大学
BEIHANG UNIVERSITY

机器学习实验报告

实验二、人工神经网络

院（系）名称	自动化科学与电气工程学院
专业名称	自动化
学生学号	15031117
学生姓名	柳天宇
指导教师	秦曾昌

2018 年 6 月 19

人工神经网络是通过模拟生物神经网络工作过程对于完成复杂分类问题的一种有效方法。在之前的实验中我们使用感知器实现了简单的二分类。作为神经网络最基本的单元，单个感知器的学习能力非常有限，无法解决线性不可分问题。然而通过多个感知器构成一定结构的网络，并通过加入非线性的激活函数，我们可以构建出具有更强学习能力的神经网络模型，完成更加复杂的分类问题。

1.理论准备：神经网络模型的构建

（1）根据分类对象初步确定模型

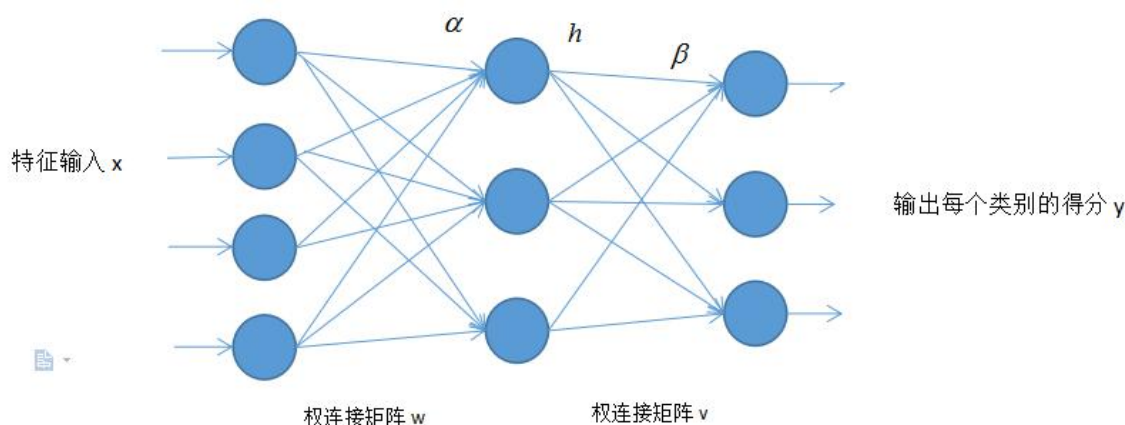
根据输入样本的特征维数和输出结果的类别数可确定输入层和输出层神经元的个数，此次实验中使用的数据集为 Iris 数据集，从表中可以看出数据集共采集了鸢尾花的 4 个属性：sepal length（花萼长度）、sepal width（花萼宽度）、petal length（花瓣长度）和 petal width（花瓣宽度）（单位是 cm）。而鸢尾花的种类有 Iris setosa、Iris versicolor 和 Iris virginica 三种，即分类器的任务是完成三种类别的分类。

特征维度 4

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	type
1	5.1	3.5	1.4	0.2	iris setosa
2	4.9	3.0	1.4	0.2	iris setosa
3	4.7	3.2	1.3	0.2	iris setosa
4	4.6	3.1	1.5	0.2	iris setosa
5	5.0	3.6	1.4	0.2	iris setosa
...					
51	7.0	3.2	4.7	1.4	iris versicolor
52	6.4	3.2	4.5	1.5	iris versicolor
53	6.9	3.1	4.9	1.5	iris versicolor
54	5.5	2.3	4.0	1.3	iris versicolor
55	6.5	2.8	4.6	1.5	iris versicolor
...					
101	6.3	3.3	6.0	2.5	iris virginica
102	5.8	2.7	5.1	1.9	iris virginica
103	7.1	3.0	5.9	2.1	iris virginica
104	6.3	2.9	5.6	1.8	iris virginica
105	6.5	3.0	5.8	2.2	iris virginica

3 种类别

可以确定神经网络模型中 Input-layer 共有 4 个神经元，output-layer 神经元数目为 3。隐层的层数以及各隐层神经元的数目未知，初步拟定设置 1 个隐层，hidden-layer 中神经元数目为 3。



(2) 前向传播计算输出及误差:

$$\alpha = w^T x + \theta; \text{隐层输出 } h = \text{sigmoid}(\alpha)$$

$$\beta = v^T h + \gamma; \text{输出 } \hat{y} = \text{sigmoid}(h)$$

$$\text{均方误差的计算: } E_k = \frac{1}{2} \sum_{j=1}^3 (y_j^k - \hat{y}_j^k)^2$$

(3) 反向传播训练网络, 计算权值和阈值的梯度:

$$\text{隐层至输出层 } v \text{ 矩阵梯度计算: } \frac{\partial E_k}{\partial v_{hj}} = \frac{\partial E_k}{\partial y_j} \frac{\partial y_j}{\partial \beta_j} \frac{\partial \beta_j}{\partial v_{hj}} = \hat{y}_j^k (1 - \hat{y}_j^k) (y_j^k - \hat{y}_j^k) h_h;$$

$$\text{输入层至隐层 } w \text{ 矩阵梯度计算: } \frac{\partial E_k}{\partial w_{hi}} = \sum_{j=1}^3 \frac{\partial E_k}{\partial \beta_j} \frac{\partial \beta_j}{\partial h_j} \frac{\partial h_j}{\partial w_{hi}} = e_h x_i$$

$$\text{其中, } g_j = \hat{y}_j^k (1 - \hat{y}_j^k) (y_j^k - \hat{y}_j^k); \quad e_h = h_h (1 - h_h) \sum_{j=1}^3 v_{hj} g_j$$

$$\text{按梯度下降法更新权值: } w = w - \eta \frac{\partial E}{\partial w}; \quad v = v - \eta \frac{\partial E}{\partial v}; \quad \theta_j = \theta_j - \eta g_j; \quad \gamma_h = \gamma_h - \eta e_h$$

当训练集上的累积误差 $E = \frac{1}{m} \sum_{k=1}^m E_k$ 达到最小 (或小于目标值) 则训练完成。

3.实验内容及结果分析

3.1 选取数据集, 建立多层神经网络完成对数据集的分类

数据集中每个类别的数据样本分别为 50 个, 从每个类别的样本中各选取 10 个样本组成训练集, 保证训练样本的平衡性, 训练集共 30 个样本。

由于激活函数各层的激活函数均使用了 **sigmoid**, 故输出输出层 3 神经元的

输出均在 0~1。为了度量误差，进行反向传播，也为了便于从神经网络的输出结果判断类别，将鸢尾花的三个类别进行编码：Iris setosa、Iris versicolor 和 Iris virginica 各编码为 100，010，001。

从输出 y 向量与 (1, 0, 0)，(0, 1, 0)，(0, 0, 1) 的接近程度来做出分类决策。

3.1.1 训练集表现：

- (1) 使用梯度下降法，每次迭代计算一个样本的误差并反向传播更新权值。
- (2) 设定学习率为 0.2
- (3) 初始权值矩阵和偏置矩阵均设为 0

1500 次迭代后对训练集的分类结果如下（1 到 10 代表样本序号，每行的数值代表该样本在每个类别上的得分）

第一类样本：

	1	2	3	4	5	6	7	8	9	10
1	0.8608	0.8590	0.8829	0.8692	0.8855	0.8767	0.8832	0.8640	0.8759	0.8573
2	0.2672	0.2590	0.2429	0.2494	0.2408	0.2465	0.2412	0.2561	0.2441	0.2590
3	0.0174	0.0172	0.0150	0.0161	0.0147	0.0155	0.0149	0.0167	0.0155	0.0173

第二类样本：

	11	12	13	14	15	16	17	18	19	20
1173	0.0531	0.0591	0.0627	0.0612	0.0575	0.0447	0.1136	0.0695	0.0468	0.0689
2190	0.4089	0.4109	0.4120	0.4200	0.4132	0.3894	0.4716	0.4383	0.3922	0.4318
3173	0.6871	0.6479	0.6237	0.6339	0.6568	0.7457	0.3822	0.5821	0.7316	0.5836

第三类样本：

	20	21	22	23	24	25	26	27	28	29	30
1	0.0689	0.0328	0.0352	0.0346	0.0360	0.0330	0.0376	0.0350	0.0347	0.0390	0.0392
2	0.4318	0.3633	0.3673	0.3665	0.3686	0.3636	0.3728	0.3675	0.3671	0.3760	0.3765
3	0.5836	0.8234	0.8100	0.8142	0.8053	0.8223	0.7952	0.8115	0.8132	0.7858	0.7845

1500 次迭代后分类器在第一类和第三类样本输入时，能够在其所在类别上给出最高得分，做出正确的预测。而对于第二类样本则无法给出正确的预测。分类的效果并不好。说明该网络具有对于样本分类的能力，但 1500 次迭代对于该网络的训练是不够的。

随着迭代次数增加，均方误差逐渐减小，分类效果提升，

3000 次迭代后在训练集的表现如下：

第一类样本：

1	2	3	4	5	6	7	8	9	10
0.9150	0.9141	0.9234	0.9180	0.9243	0.9211	0.9234	0.9163	0.9205	0.9134
0.1506	0.1469	0.1376	0.1411	0.1367	0.1395	0.1370	0.1450	0.1385	0.1469
0.0046	0.0046	0.0043	0.0045	0.0043	0.0044	0.0043	0.0046	0.0044	0.0047

第二类样本：

	11	12	13	14	15	16	17	18	19	20
1	0.0378	0.0390	0.0441	0.0450	0.0426	0.0240	0.1091	0.0625	0.0251	0.0579
2	0.5179	0.5229	0.5637	0.5901	0.5630	0.3153	0.7801	0.7030	0.3343	0.6742
3	0.4364	0.4138	0.3453	0.3242	0.3632	0.7365	0.0861	0.1848	0.7082	0.2117

第三类样本：

21	22	23	24	25	26	27	28	29	30
0.0129	0.0145	0.0140	0.0147	0.0131	0.0163	0.0143	0.0143	0.0174	0.0175
0.1664	0.1755	0.1724	0.1782	0.1670	0.1949	0.1748	0.1744	0.2105	0.2120
0.9112	0.9018	0.9050	0.8990	0.9105	0.8823	0.9026	0.9030	0.8657	0.8641

可见此时网络可以对于绝大部分样本正确分类，，第二类的部分样本出现错分。增加迭代次数到 15000 后可对所有训练样本正确分类，且在正确类别上的的得分远高于在其他类别的得分。

15000 次迭代后：

第一类样本：

1	2	3	4	5	6	7	8	9	10
0.9579	0.9578	0.9610	0.9594	0.9612	0.9602	0.9609	0.9586	0.9602	0.9579
0.0582	0.0556	0.0526	0.0537	0.0522	0.0531	0.0523	0.0548	0.0527	0.0557
4.6025e-04	4.5769e-04	4.3403e-04	4.4506e-04	4.3225e-04	4.3936e-04	4.3381e-04	4.5174e-04	4.3939e-04	4.5708e-04

第二类样本：

11	12	13	14	15	16	17	18	19	20
0.0185	0.0151	0.0206	0.0173	0.0220	0.0107	0.0542	0.0287	0.0105	0.0397
0.9549	0.9368	0.9516	0.9527	0.9588	0.8331	0.9724	0.9680	0.8220	0.9718
0.0237	0.0352	0.0238	0.0248	0.0204	0.1228	0.0092	0.0145	0.1326	0.0116

第三类样本：

21	22	23	24	25	26	27	28	29	30
0.0032	0.0045	0.0037	0.0042	0.0034	0.0050	0.0041	0.0045	0.0055	0.0058
0.0309	0.0378	0.0336	0.0363	0.0317	0.0522	0.0361	0.0391	0.0774	0.0938
0.9826	0.9768	0.9804	0.9781	0.9819	0.9655	0.9783	0.9758	0.9444	0.9297

3.1.2 分类器性能检验

使用训练后的网络对测试集进行分类，检验该网络在数据泛化后的分类性能，输入测试集的 120 个样本，根据每个类别上的得分做出类别的预测，即将得分最高的类别数值置为 1，其他两个类别的数值置为 0，分类结果如下：

测试集第一类样本分类结果：

	1	2	3
4	1	0	0
5	1	0	0
6	1	0	0
7	1	0	0
8	1	0	0
9	1	0	0
10	1	0	0
11	1	0	0
12	1	0	0
13	1	0	0
14	1	0	0
15	1	0	0
16	1	0	0

测试集第二类样本分类结果：

	1	2	3
58	0	1	0
59	0	1	0
60	0	1	0
61	0	1	0
62	0	1	0
63	0	1	0
64	0	1	0
65	0	1	0
66	0	1	0
67	0	1	0
68	0	1	0
69	0	0	1
70	0	1	0

测试集第三类样本分类结果：

	1	2	3
106	0	0	1
107	0	0	1
108	0	0	1
109	0	0	1
110	0	0	1
111	0	0	1
112	0	0	1
113	0	0	1
114	0	0	1
115	0	0	1
116	0	0	1
117	0	0	1
118	0	0	1

分类准确率的统计

```

0 - correct=0;%记录正确分类个数
1 - for i=1:120
2 -     if (output(i,:)==B(i,:))%预测结果与真实类别一致
3 -         correct=correct+1;
4 -     end
5 - end
6 - accuracy=correct/120 %准确率

```

命令窗口

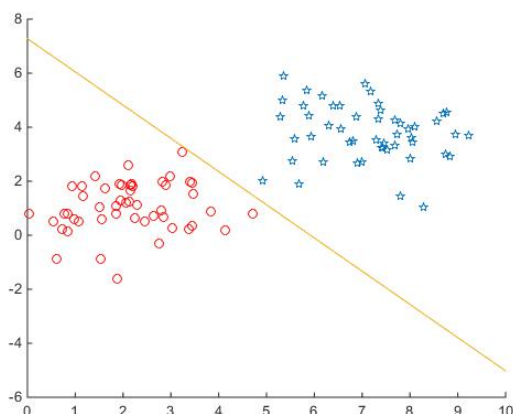
accuracy =

0.9667

分类准确率为 96.67%，可见该神经网络对于鸢尾花数据集具有很好的分类性能，第二类中的个别样本分类器产生了错分，第一类和第三类样本没有错分。

3.2 神经网络与感知器的比较

对于二维特征的简单的线性可分的二分类问题感知器模型取得了很好的效果，在如下的样本测试中，感知器和神经网络（输入层、隐层、输出层神经元个数分别为 2，2，2）均能完成对于所有样本的正确分类，但感知器在速度方面明显更快。



但感知器用于分类问题存在很大局限性，对于非线性可分的分类问题，感知器无法完成正确分类，而神经网络在非线性可分的问题中仍然具有很好的性能。

对于感知器模型，输出始终是输入的线性组合，因此只能解决线性可分的分类问题。

与感知器模型相比，神经网络具有更复杂的结构，训练需要更多时间。但神经网络具有更强的学习能力；同时神经网络每个神经元的输入输出之间采用了非线性的激活函数，使得整个网络的输入和输出之间不再局限于简单的线性关系，即神经网络可以逼近的函数种类更为广泛。

3.3 模型参数对于神经网络性能的影响

在实验中，改变迭代次数与学习率的大小，分类性能均会发生变化，得到如下结果：

迭代次数 1500 时学习率与准确率的关系如下：

学习率（lr）	0.1	0.2	0.4
准确率	70%	86%	84%

迭代次数 6000 时学习率与准确率的关系如下：

学习率（lr）	0.1	0.2	0.4
准确率	87%	93%	94%

迭代次数 15000 时学习率与准确率的关系如下：

学习率（lr）	0.1	0.2	0.4
准确率	96.67%	96.67%	96%

可见，随着迭代次数的增加，不同学习率下，模型的性能均有所提高。

在迭代次数较少的情况下，大学习率的表现较好，这是由于刚开始时每个权值距离最优解的距离很远，在大的学习率下，模型能够更快的接近最优解；逐渐增加迭代次数，权值逐渐靠近最优解，在最优解的附近，在大的学习率下使用梯度下降，使权值在最优解的附近来回振荡而不再进一步接近最优解，进一步增加迭代次数模型精度也不再提高；此时较小的学习率对于最优解的逼近更好，迭代次数增至 15000 时，较小学习率下准确率更高。

综合不同学习率的特点，更好的方法是在训练刚开始阶段采用较大学习率，使各个权值较快地靠近最优解，之后逐渐减小学习率，实现对于最优解的进一步逼近。

3.4 激活函数和隐层的数目对性能的影响

3.4.1 激活函数对分类性能的影响

考虑将隐层激活函数更换为双极型 sigmoid 函数，输出层更换为 softmax 函数，分类器性能有所提升，在 $lr=0.2$ ，迭代次数 15000 时，分类准确率为 97.3%，好于使用 sigmoid 函数。

在该模型中，将输出层激活函数改为线性函数，可以加快训练速度，在测试集上也有较好的表现。

若将输入层到隐层、隐层到输出层均采用线性函数，分类准确度显著降低，此时神经网络模型等同于多层的感知器。

可见使用非线性函数作为激活函数提高了神经网络的学习能力，可以使其对于任意函数实现逼近，同时在选择输出层的激活函数时应该特别注意和输出结果的匹配，例如输出结果为 $-1 \sim +1$ 时若在输出层选择 sigmoid 则无论怎样训练也无法达到训练目标。

3.4.2 隐层数目对分类性能的影响

在该数据集的分类中，使用单隐层的神经网络可以达到很好的分类效果，增加隐层数目使得每次迭代所需时间变长，但增加隐层层数会在训练过程中降低均方误差。

单隐层结构训练集分类结果：

1	2	3	4	5	6	7	8	9	10
0.9900	0.9906	0.9906	0.9907	0.9906	0.9907	0.9907	0.9907	0.9907	0.9907
0.0244	-0.0014	-8.1931e-04	-0.0049	1.5175e-04	-0.0026	-0.0026	-0.0021	-0.0052	-0.0037
-0.0011	1.7835e-04	-2.8641e-04	2.7883e-04	-3.1921e-04	-1.3501e-04	-1.9750e-04	1.0977e-04	2.2289e-04	3.3646e-04

2 个隐层结构在训练集上的分类结果：

	1	2	3	4	5	6	7	8	9	10
1	0.9977	0.9980	0.9981	0.9983	0.9980	0.9981	0.9981	0.9981	0.9982	0.9982
2	0.0280	0.0050	-0.0029	-0.0113	0.0064	-2.2241e-04	-0.0013	0.0042	-0.0108	-0.0077
3	0.0296	0.0060	-0.0024	-0.0122	0.0078	5.8974e-04	-6.0421e-04	0.0053	-0.0115	-0.0081

可见增加一个隐层后，输出结果在正确类别上的得分与目标输出 1 更加接近，就是说增加一个隐层后，网络对于训练集的输入输出关系的拟合效果更好。但将训练结果用于测试集后，增加一个隐层对于该数据集在分类的准确度上并没有提升，在迭代次数较少的情况下，单隐层结构的表现甚至更好。表明增加隐层数目会使模型在训练集上的精度提高，但同时需要计算的参数增多，训练时间增长。此外，对训练集更好的逼近效果并不一定会提高测试集的表现。

3.5 自编码网络

自编码网络的作用是将输入样本的特征压缩到隐藏层，再在输出层重构样本，就是说自编码网络的输出应该尽可能地接近输入，而隐层则是那些可以代表输入的重要因素，利用中间隐层实现了对输入的压缩表达。自编码网络的过程实际上类似于特征提取，通过反向传播找到那些对于表达原始数据最重要的特征，输入层至隐层实现了将原始数据的高维特征压缩至低维，而隐层到输出层则是通过低维特征重构原始数据。

(1) 自编码网络的构建：因为要求输出等于输入，故输出层神经元个数与输入层相等，同时隐层输出是对于输入数据的压缩表达，故为使网络起到压缩信息的作用，应该使隐层神经元数目小于输入输出层。

(2) 测试结果

输入 12 维向量如下：

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0	1	-1	0	1	-1	0	-1	0	1	0

隐层神经元数目为 8 时输出如下：

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.9746	3.9164e-04	0.9873	-0.9790	3.1544e-05	0.9722	-0.9706	1.6416e-04	-0.9714	-2.3607e-04	0.9697	1.0000

隐层神经元数目为 6 时输出如下：

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.9462	3.9629e-04	0.9631	-0.9478	0.0024	0.9486	-0.9521	0.0010	-0.9992	-0.0021	0.9994	-0.0037

隐层神经元数目为 2 时输出如下：

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.7089	1.0000	-1.0000	-0.9925	-0.4014	0.9974	-0.9873	-1.0000	-1.0000	0.4849	-1.0000	1.0000

可见隐藏层神经元数目的改变对于特征的重构效果有着很大影响，压缩后保留的特征较多时重构的效果与原特征较为接近，如果压缩后维数过低，会导致信息的丢失，很难重构出原特征。

4.问题讨论：生物神经网络与人工神经网络的差异

人工神经网络在结构和训练过程方面与生物神经网络很类似，但二者也有一些明显的差别。

人脑在成长学习的过程中，除了各个神经元之间连接的强度发生变化外，结构也在发生变化，例如新的知识学习会使人脑建立新的突触。但是在人工神经网络的训练过程中，神经元之间的连接方式是固定的，仅仅是连接的强度发生变化。此外对于较简单的神经网络，例如 BP 神经网络，实际上是一种分层的结构，相邻层之间有连接，而在不相邻层之间没有连接，这一点与生物神经网络真正的网状结构还有所差别，就是说人工神经网络实际上模仿了生物神经网络的部分过程，但目前来看在复杂度上还不及生物神经网络。因此人类的很多活动人工神经网络还无法完成。

5.总结

本次实验通过 BP 神经网络的构建实现了对于鸢尾花数据集的分类任务，加深了对于神经网络工作过程、反向传播原理的理解。巩固了梯度下降法的使用。神经网络相比于感知器具有更加复杂的结构，影响其分类性能的因素也更加复杂：不同的网络的结构构建方式、参数设置、最优解的求解方式均会对神经网络的性能产生影响。我们要根据不同的数据集恰当选取这些参数，以提高神经网络精度和效率。同时实验过程中我也认识到神经网络的作用不仅仅在于分类，通过自编码网络还可以完成数据的压缩、特征选择等功能。

Github 网址：<https://github.com/ECHOLIuty/PR-report>