

2023年秋季学期人工神经网络第一次作业



© 说明：完成作业可以使用你所熟悉的编程语言^Q和平台，比如 C, C++、MATLAB、Python等。作业链接：2023年秋季学期人工神经网络第一次作业：<https://zhuoqing.blog.csdn.net/article/details/133764230>。

01 学习算法

一、题目内容

1、背景介绍

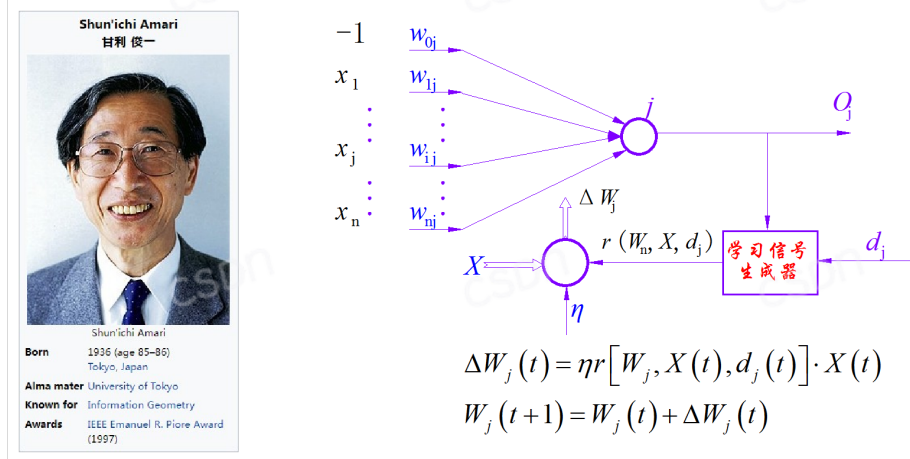
在第一章介绍了日本学者“**甘利俊一**”提出的统一公式，把对神经元输入连接权系数的修正 ΔW 分成了三个独立成分的乘积：学习速率 η ，学习信号 $r(W, x, d)$ 以及输入向量 X 。

内容来源：[csdn.net](https://zhuoqing.blog.csdn.net/article/details/133764230)

作者昵称：卓晴

原文链接：<https://zhuoqing.blog.csdn.net/article/details/133764230>

作者主页：<https://zhuoqing.blog.csdn.net>



▲ 图1.1 神经网络中神经元学习的统一公式

当学习信号 $r(W, x, d)$ 取不同形式，可以得到神经元的三大类不同修正方式（无监督、有监督、死记忆）：

【表1-1 不同的神经元学习算法】

学习规则	权值调整	学习信号	初始值	学习方式	转移函数
Hebbian	$\Delta W = \eta f(W^T X) X$	$r = f(W^T X)$	随机	无监督	任意
Percetron	$\Delta W = \eta [d - \text{sgn}(W^T X)] X$	$r = d - f(W^T X)$	任意	有监督	二值函数
Delta	$\Delta W = \eta [d - f(W^T X)] f'(W^T X) X$	$r = [d - f(W^T X)] f'(W^T X)$	任意	监督	连续可导
Widrow-Hoff LMS	$\Delta W = \eta (d - W^T X) X$	$r = d - W^T X$	任意	监督	连续
Correlation 相关,外积	$\Delta W = \eta d X$	$r = d$	0	监督死记忆	任意

下面给出神经元模型和训练样本数据，请通过编程实现上述表格中的五种算法并给出计算结果。通过这个作业练习，帮助大家熟悉神经元的各种学习算法。

作者昵称：卓晴

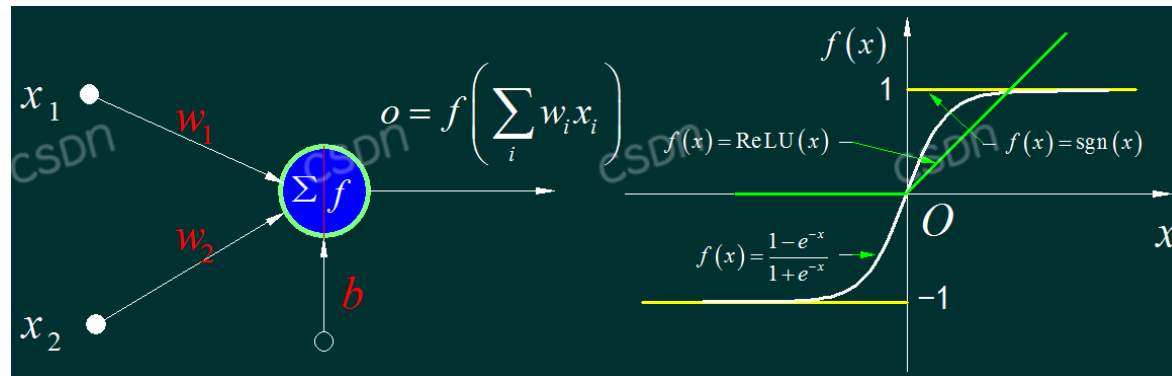
原文链接：<https://zhuoqing.blog.csdn.net/article/details/133764230>

作者主页：<https://zhuoqing.blog.csdn.net>

2、神经元模型

下面给出神经元模型，根据不同的算法要求：

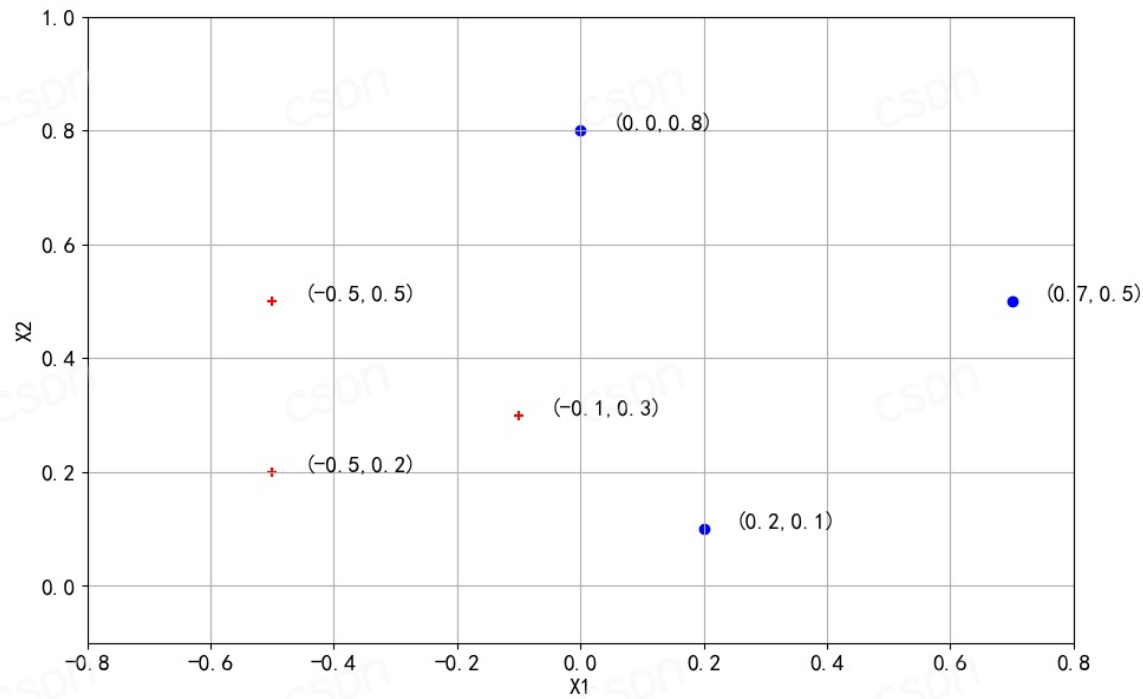
- 选择相应的传递函数种类（离散二值函数、双曲正切函数、ReLU函数）：除了Perceptron算法选择二值函数外，其它都选择双曲正切函数，
- 神经元权系数（ w_1, w_2, b ）都初始化成 0。



▲ 图1.1.2 神经元及其传递函数

3、样本数据

训练样本包括 6 个数据，它们的分布如下图所示：



▲ 神经元训练数据分布

【表1-2 样本数据】

序列	X1	X2	类别
1	-0.1	0.3	-1
2	0.7	0.5	1
3	-0.5	0.2	-1
4	-0.5	0.5	-1
5	0.2	0.1	1
6	0.0	0.8	1

```
1 import sys,os,math,time
2 import matplotlib.pyplot as plt
3 from numpy import *
```

内容来源：csdn.net
作者昵称：卓晴
原文链接：https://zhuoqing.blog.csdn.net/article/details/133764230
作者主页：https://zhuoqing.blog.csdn.net

```

4
5 xdim = [(-0.1,0.3), (0.7,0.5), (-0.5,0.2),(-0.5,0.5),(0.2,0.1),(0,0.8)]
6 ldim = [-1,1,-1,-1,1,1]
7
8 print("序列", "X1", "X2", "类别")
9
10 count = 0
11 for x,l in zip(xdim, ldim):
12
13     count += 1
14     print("%d %3.1f %3.1f %d"%(count, x[0], x[1], l))

```

二、作业要求

1、必做内容

1. 给出每个学习算法核心代码；
 2. 给出经过一轮样本学习之后神经元的权系数数值结果 (w1,w2,b) ；
- * 权系数初始化为 0 ；
- * 学习速率 $\eta = 1$ ；
- * 训练样本按照 表格1-2 的顺序对神经元进行训练 ；

2、选做内容

1. 在坐标系中绘制出经过一轮训练之后，权系数 (w1,w2) 所在的空间位置 ；
2. 简单讨论一下不同算法对于神经元权系数的影响 ；

```

1 import sys,os,math,time
2 import matplotlib.pyplot as plt
3 from numpy import *
4
5 xdim = [(-0.1,0.3), (0.5,0.7), (-0.5,0.2),(-0.7,0.3),(0.7,0.1),(0,0.5)]
6 ddim = [1,-1,1,1,-1,1]
7
8 def sigmoid(x):
9     return 1/(1+exp(-x))
10
11 def hebbian(w,x,d):
12     x1 = [1,x[0],x[1]]
13     net = sum([ww*xx for ww,xx in zip(w, x1)])
14     o = sigmoid(net)

```

内容来源：csdn.net

作者昵称：卓晴

原文链接：https://zhuoqing.blog.csdn.net/article/details/133764230

作者主页：https://zhuoqing.blog.csdn.net

02 感知机

一、感知机算法求解分类问题

1、样本数据

利用单个神经元，使用感知机算法求解样本分类问题。样本数据就采用第一道大题中的六个样本数据。
见**【表1-2 样本数据】**。

2、作业要求

1. 绘制出网络结构图，并给出算法核心代码；
2. 对比不同学习速率对于训练收敛的影响；

二、感知机识别字母

1、样本数据

如下是三个字母 **A,B,C,D,E,J,K** 的 7×9 的点阵图，共有三种字体。将它们转换成由 $(-1,1)$ 组成的63维向量。

内容来源：csdn.net

作者昵称：卓晴

原文链接：<https://zhuoqing.blog.csdn.net/article/details/133764230>

作者主页：<https://zhuoqing.blog.csdn.net>

Input from
Font 2

D

Input from
Font 3

D

A

# #	# #	# #	# #	# #
# #	.	.	.	# #
# #	# #	.	.	.
.	.	# #	.	.
# #	# #	.	.	.
# #	# #	.	.	# #
# #	# #	# #	# #	# #
.	.	E	.	.

A
 #.#.#.#.
 #.#.#.#.
 #.#.#.#.
 #.#.#.#.
 #.#.#.#.
 #.#.#.#.
E

A
 # # # # # # #
 . # # . # # #
 . # . # # .
 . # # # # .
 . # # # # .
 . # # # # .
 # # # # # # #
 E .

. B

. . # # # #

. . # # # #

. . # # # #

. . # # # #

. . # # # #

#

. # # #

. . . J

[illegible]

. B
 . . . # # #
 . . . # # #
 . . . # # #
 . . . # # #
 . . . # # #
 . # # # #
 . . . J

. . C
 # # # . # #
 # # # . #
 . # # # .
 . # # # .
 # # # #
 # # # #
 # # # #
 # # # #
 # # # #
 # # # #
 K

C

[illegible]

▲ 图2.2.1 A,B,C,D,E,J,K三种字体点阵

2、作业基本要求

1. 建立由七个神经元组成的简单感知机网络，完成上述七个字母的识别训练；
2. 测试训练之后的网络在带有一个噪声点的数据集合上的识别效果。给图片增加一个噪声点就是随机在样本中选取一个像素，将其数值进行改变（从-1改变成1，或者从1改变成-1）。

3、选做内容

1. 测试上述感知机网络在两个噪声点的数据集合上的识别效果；
2. 对比以下两种情况训练的感知机的性能。
 - 第一种情况：只使用没有噪声的七个字母进行训练；
 - 第二种情况：使用没有噪声和有一个噪声点的样本进行训练；
3. 对比不同的学习速率对于训练过程的影响。

三、感知机算法收敛特性

☐ 这是选做题目

请证明对于线性可分的两类数据集合，使用感知机算法进行分类。感知机算法收敛步骤数量的上限与学习速率无关。

03 Adaline 网络

☐ 这是选做题目。

一、题目内容

1、背景介绍

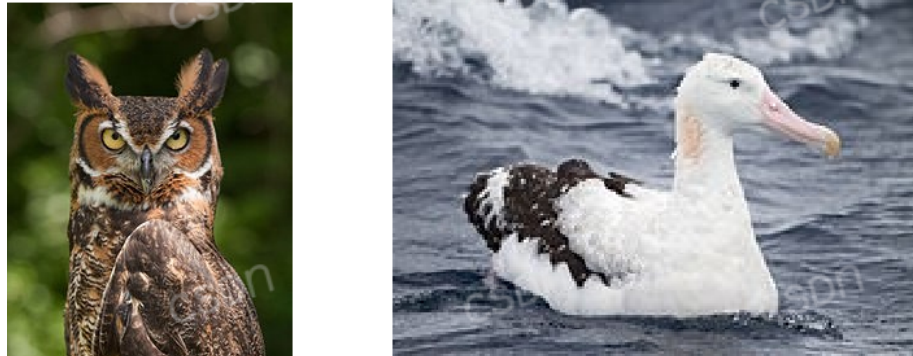
自适应线性神经元 ADALINE (Adaptive Linear Neuron) 是由 Bernard Widrow 与 Ted Hoff 在 1959 年提出的算法。关于他们提出算法前后的故事，大家可以参照网文：[The ADALINE - Theory and Implementation of the First Neural Network Trained With Gradient Descent](https://zhaoqing.blog.csdn.net/article/details/133764230) 进行了解。

内容来源：csdn.net

下面也是根据上述网文中所介绍的两种鸟类（猫头鹰与信天翁）数据集合，产生相应的分类数据集合。大家使用 ADALINE 算法完成它们的分类器算法。

原文链接：https://zhaoqing.blog.csdn.net/article/details/133764230

作者主页：https://zhaoqing.blog.csdn.net



▲ 图3.1.1 猫头鹰与信天翁

2、样本数据

(1) 数据参数

根据Wikipedia 中关于 信天翁 [Wandering albatross](#) 和 猫头鹰 ([Great horned owl](#)) 的相关数据，这两种鸟类的题中和翼展长度如下表所示。

【表1-3 两种鸟类的体型数据】

种类	体重(kg)	翼展(m)
信天翁	9	3
猫头鹰	1.2	1.2

使用计算机产生两个鸟类体型随机数据数据，下表给出了每一类数据产生的参数：

【表1-4 两类鸟类数据产生参数】

鸟类	体重平均值	体重方差	翼展平均值	翼展方差	个数	分类
信天翁	9000	800	300	20	100	1
猫头鹰	1000	200	100	15	100	-1

内容来源：csdn.net

作者昵称：卓晴

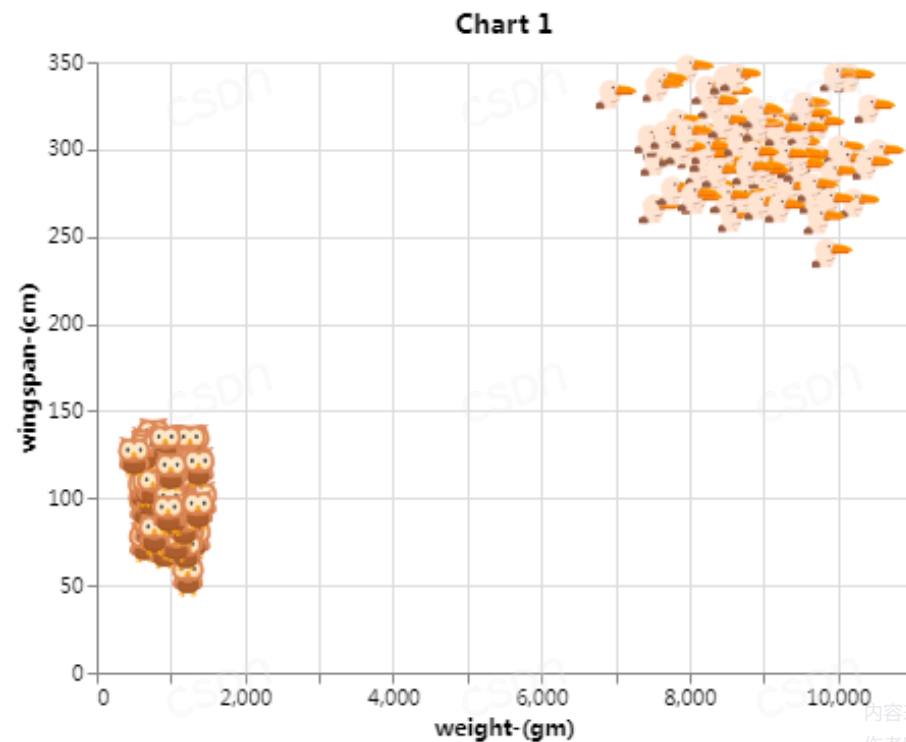
原文链接：<https://zhuoqing.blog.csdn.net/article/details/133764230>

作者主页：<https://zhuoqing.blog.csdn.net>

(2) Python示例代码

下面给出了产生随机样本数据的 Python 示例代码。大家可以参照这些代码，使用自己熟悉的 编程语言来实现。

```
1 def species_generator(mu1, sigma1, mu2, sigma2, n_samples, target, seed):
2     '''creates [n_samples, 2] array
3
4     Parameters
5     -----
6     mu1, sigma1: int, shape = [n_samples, 2]
7         mean feature-1, standar-dev feature-1
8     mu2, sigma2: int, shape = [n_samples, 2]
9         mean feature-2, standar-dev feature-2
10    n_samples: int, shape= [n_samples, 1]
11        number of sample cases
12    target: int, shape = [1]
13        target value
14    seed: int
15        random seed for reproducibility
```



▲ 图3.1.2 产生两类数据的分布

内容来源：csdn.net

作者昵称：卓晴

原文链接：<https://zhuoqing.blog.csdn.net/article/details/133764230>

作者主页：<https://zhuoqing.blog.csdn.net>

二、作业要求

1. 构造一个 ADALINE 神经元，完成上述两类鸟类的分类。由于需要进行分类，在对 ADALINE 的输出在经过一个符号函数（sgn）便可以完成结果的分类；
2. 利用上述数据对 ADALINE 进行训练。观察记录训练误差变化的曲线。
3. 讨论不同的学习速率对于训练结果的影响，看是否存在一个数值，当学习速率超过这个数值之后，神经元训练过程不再收敛。

■ 相关文献链接：

- [The ADALINE - Theory and Implementation of the First Neural Network Trained With Gradient Descent](#)
- [Wandering albatross](#)
- [Great horned owl](#)

● 相关图表链接：

- [图1.1 神经网络中神经元学习的统一公式](#)
- [表1-1 不同的神经元学习算法](#)
- [图1.1.4 神经元及其传递函数](#)
- [图1.1.5 神经元训练数据](#)
- [表1-2 样本数据](#)
- [图2.1.1 分类数据及其在三维坐标中的位置](#)
- [图2.2.2 C,H,L字母的点阵图](#)
- [图2.2.3 与三个字母Hamming距离为1的噪声样本](#)
- [图3.1.1 猫头鹰与信天翁](#)
- [表1-3 两种鸟类的体型数据](#)
- [表1-4 两类鸟类数据产生参数](#)
- [图3.1.2 产生两类数据的分布](#)

内容来源：csdn.net

作者昵称：卓晴

原文链接：<https://zhuoqing.blog.csdn.net/article/details/133764230>

作者主页：<https://zhuoqing.blog.csdn.net>