

东东的专栏

存储和存储编码

目录视图

摘要视图

RSS 订阅

个人资料



东东是个胆小鬼

访问：10266次

积分：224

等级：BLOG > 2

排名：千里之外

原创：11篇 转载：0篇

译文：0篇 评论：0条

文章搜索

文章分类

linux (1)

其他 (0)

编程 (0)

分布式 (0)

存储系统 (0)

计算机基础 (0)

硬件 (0)

文章存档

2016年11月 (1)

2016年09月 (2)

2016年08月 (1)

2016年07月 (2)

2016年06月 (1)

展开

阅读排行

一起学shell (一) : 合并 (5514)

Python 和字符编码 (1409)

一个监督的赫布学习 (Hebb Learning) 的例子 (1020)

异步赠书：9月重磅新书升级，本本经典 CSDN新版博客内测群495397560期待你加入 程序员9月书讯 每周荐书：ES6、虚拟现实、物联网（评论送书）

一个监督的赫布学习 (Hebb Learning) 的例子

2016-09-03 22:47

1020人阅读

评论(0)

收藏

举报

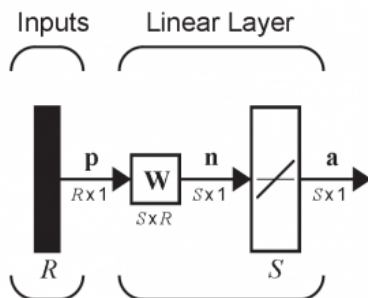
版权声明：本文为博主原创文章，未经博主允许不得转载。

赫布学习 (Hebb Learning) 基于赫布规则 (Hebb Rule)：

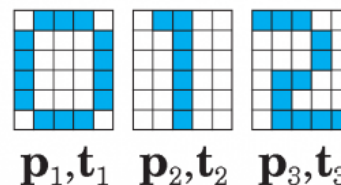
When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of cells firing B, is increase.

赫布规则大致说的是如果神经细胞刺激不断加强，两者联系加强。

首先看看一个简单的神经网络的结构（以识别为例）：



左边P (R×1的向量) 是输入，表示待识别物体的R个特征。W是权重矩阵，通过计算特征和权重矩阵的乘法，用于形成S个结果，S是判别函数。最终形成a (S×1向量) 的结果。下面以位矩阵的数字识别为例：



问题：有6×5大小的像素矩阵用于表示数字0,1,2，如下图所示

每个数字矩阵用一个一维的特征向量表示，比如0对应的特征向量为p1：

$$p1 = [-1 \ 1 \ 1 \ 1 \ 1 \ -1 \ 1, -1 \ -1 \ 1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1]^T$$

STM32F10x 单片机移植 (463)

计算二维空间某点的最近 (353)

Android 开发笔记 (SDK (352)

Ubuntu14.04 上安装pyg (327)

Android 开发笔记 (准备 (271)

MongoDB 性能测试 (183)

Delicate/Elaborate/Exqui (118)

评论排行

一起学shell (一) : 合并 (0)

STM32F10x 单片机移植 (0)

一个监督的赫布学习 (He (0)

无题 (0)

Python 和字符编码 (0)

MongoDB 性能测试 (0)

计算二维空间某点的最近 (0)

Android 开发笔记 (SDK (0)

Ubuntu14.04 上安装pyg (0)

Android 开发笔记 (准备 (0)

推荐文章

* CSDN新版博客feed流内测用户征集令

* Android检查更新下载安装

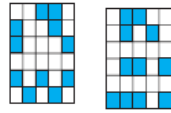
* 动手打造史上最简单的Recycleview 侧滑菜单

* TCP网络通讯如何解决分包粘包问题

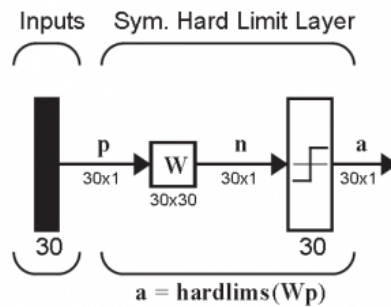
* SDCC 2017之大数据技术实战线上峰会

* 快速集成一个视频直播功能

其中-1代表这个像素不上色, 1反之, t1-t3分表代表结果是0,1,2。那我们的问题是如果识别带有误差, 或者只有部分像素的例子。如下面图中应该识别为多少呢?



分析: 使用如下的神经网络,



权重矩阵W通过下面等式计算:

$$W = p_1 \cdot p_1^T + p_2 \cdot p_2^T + p_3 \cdot p_3^T$$

在我们这个例子里, 权重函数如下

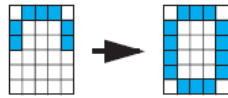
关闭

```

3 -1 -1 -1 1 -1 1 -1 3 -1 -1 1 -1 3 -1 -1 3 1 1 -1 -1 3 -1 1 -1 1 1 -1 1 3
-1 3 3 -1 -3 -1 -3 -1 -1 -1 -1 -3 -1 -1 -1 -1 1 -3 -1 -1 -1 -1 -3 -1 -3 1 3 1 -1
-1 3 3 -1 -3 -1 -3 -1 -1 -1 -1 -3 -1 -1 -1 -1 1 -3 -1 -1 -1 -1 -3 -1 -3 1 3 1 -1
-1 -1 -1 3 1 3 1 -1 -1 3 3 1 -1 -1 3 3 -1 -3 1 3 3 -1 -1 1 3 1 1 -1 1 -1
1 -3 -3 1 3 1 3 1 1 1 1 3 1 1 1 1 1 -1 3 1 1 1 1 3 1 3 -1 -3 -1 1
-1 -1 -1 3 1 3 1 -1 -1 3 3 1 -1 -1 3 3 -1 -3 1 3 3 -1 -1 1 3 1 1 -1 1 -1
1 -3 -3 1 3 1 3 1 1 1 1 3 1 1 1 1 1 -1 3 1 1 1 1 3 1 3 -1 -3 -1 1
-1 -1 -1 3 1 3 1 1 1 3 3 1 -1 -1 3 3 -1 -3 1 3 3 -1 -1 1 3 1 1 -1 1 -1
-1 -1 -1 3 1 3 1 -1 -1 3 3 1 -1 -1 3 3 -1 -3 1 3 3 -1 -1 1 3 1 1 -1 1 -1
1 -3 -3 1 3 1 3 1 1 1 1 3 1 1 1 1 1 -1 3 1 1 1 1 3 1 3 -1 -3 -1 1
-1 -1 -1 -1 1 -1 1 3 -1 -1 -1 1 3 -1 -1 -1 -1 1 1 -1 -1 -1 3 1 -1 1 -3 -1 -3 -1
3 -1 -1 -1 1 -1 1 -1 3 -1 -1 1 -1 3 -1 -1 3 1 1 -1 -1 3 -1 1 -1 1 1 -1 1 3
-1 -1 -1 3 1 3 1 -1 -1 3 3 1 -1 -1 3 3 -1 -3 1 3 3 -1 -1 1 3 1 1 -1 1 -1
-1 -1 -1 3 1 3 1 -1 -1 3 3 1 -1 -1 3 3 -1 -3 1 3 3 -1 -1 1 3 1 1 -1 1 -1
3 -1 -1 -1 1 -1 1 -1 3 -1 -1 1 -1 3 -1 -1 3 1 1 -1 -1 3 -1 1 -1 1 1 -1 1 3
1 1 1 -3 -1 -3 -1 1 1 -3 -3 -1 1 1 -3 -3 1 3 -1 -3 -3 1 1 -1 -3 -1
1 -3 -3 1 3 1 3 1 1 1 1 3 1 1 1 1 1 -1 3 1 1 1 1 3 1 1
-1 -1 -1 3 1 3 1 -1 -1 3 3 1 -1 -1 3 3 -1 -3 1 3 3 -1 -1 1 3 1
-1 -1 -1 3 1 3 1 -1 -1 3 3 1 -1 -1 3 3 -1 -3 1 3 3 -1 -1 1 3 1
3 -1 -1 -1 1 -1 1 -1 3 -1 -1 1 -1 3 -1 -1 3 1 1 -1 -1 3 -1 1 -1
-1 -1 -1 -1 1 -1 1 3 -1 -1 -1 1 3 -1 -1 -1 -1 1 1 -1 -1 -1 3 1 -1
1 -3 -3 1 3 1 3 1 1 1 1 3 1 1 1 1 1 -1 3 1 1 1 1 3 1 3
-1 -1 -1 3 1 3 1 -1 -1 3 3 1 -1 -1 3 3 -1 -3 1 3 3 -1 -1 1 3 1
1 -3 -3 1 3 1 3 1 1 1 1 3 1 1 1 1 1 -1 3 1 1 1 1 3 1 3
1 1 1 1 -1 1 -1 -3 1 1 1 -1 -3 1 1 1 1 -1 -1 1 1 1 -3 -1 1 -1
1 3 3 -1 -3 -1 -3 -1 -1 -1 -1 -3 -1 -1 -1 -1 1 -3 -1 -1 -1 -3 -1 -3
1 1 1 1 -1 1 -1 -3 1 1 1 -1 -3 1 1 1 1 -1 -1 1 1 1 -3 -1 1 -1
3 -1 -1 -1 1 -1 1 -1 3 -1 -1 1 -1 3 -1 -1 3 1 1 -1 -1 3 -1 1 -1 1 1 -1 1 3

```

S判别函数我们使用hardlims，当输入大于0则结果为1，当小于0 则结果为-1. 针对一个特定识别过程（如下图）：



下面是实现这个过程的Python 代码，使用到numpy 库。

```

#_*_coding:utf-8_*_
import os
import sys
import numpy as np
mat0 = np.matrix([-1,1,1,1,-1,\
1,-1,-1,-1,1,\
1,-1,-1,-1,1,\
1,-1,-1,-1,1,\
1,-1,-1,-1,1,\
-1,1,1,1,-1])
mat1 = np.matrix([-1,1,1,-1,-1,\
-1,-1,1,-1,-1,\
-1,-1,1,-1,-1,\
-1,-1,1,-1,-1,\
-1,-1,1,-1,-1,\
-1,-1,1,-1,-1])

```

关闭

```
mat2 = np.matrix([1,1,1,-1,-1,\
-1,-1,-1,1,-1,\
-1,-1,-1,1,-1,\
-1,1,1,-1,-1,\
-1,1,-1,-1,-1,\
-1,1,1,1,1])
mat0t = mat0.getT()
mat0p = mat0t.dot(mat0)
mat1t = mat1.getT()
mat1p = mat1t.dot(mat1)
mat2t = mat2.getT()
mat2p = mat2t.dot(mat2)
print "=====matrix 0======"
print(mat0p)
print "=====matrix 1======"
print(mat1p)
print "=====matrix 2======"
print(mat2p)
matw = mat0p+mat1p+mat2p
print "=====matrix sum======"
print matw
testa0 = np.matrix([-1,1,1,1,-1,\
1,-1,-1,-1,1,\
1,-1,-1,-1,1,\
-1,-1,-1,-1,-1,\
-1,-1,-1,-1,-1,\
-1,-1,-1,-1,-1])
mata0 = matw.dot(testa0.getT())
print "===== raw mata0 ====="
print mata0
for ii in xrange(mata0.size):
    if mata0[ii] > 0:
        mata0[ii] = 1
    else:
        mata0[ii] = -1
print "===== After testa0 ====="
print mata0
```

备注：这是Neural Network Design 的一个例子，作者用python 代码实现

关闭

查看原文：<http://blog.fooool.net/2016/09/%e4%b8%80%e4%b8%aa%e7%9b%91%e7%9d%a3%e7%9a%84%e8%b5%ab%e5%b8%83%e5%ad%a6%e4%b9%a0%ef%bc%88hebb-learning%ef%bc%89%e7%9a%84%e4%be%8b%e5%ad%90/>

顶 踩
0 0

上一篇 无题

下一篇 STM32F10x 单片机移植Modbus

相关文章推荐

- 监督学习 - 一个典型的工作流程
- 用户画像系统应用与技术解析--汪剑
- 深度学习的过往
- 实时流计算平台Blink在阿里集团的应用实践--陈守元
- 一个关于NTFS权限/DTS/ASP.Net的问题解决过程
- Java 9新特性解读
- Hebb和Delta学习规则
- Cocos2d-x 实战演练基础篇
- Qt Creator: Unknown debugger type "No engine"
- Unity3D移动端实战经验分享
- 解决qt调试时Unknown debugger type "No Engine"
- 程序员如何转型AI工程师--蒋涛
- 一起学shell（一）：合并文件夹下所有文件
- QT - Unknown debugger type "No engine"
- Hebb学习规则 以及 Hebb网络
- 半监督学习(书)Introduction to Semi-Supervised Learning



免费云服务器



电路板抄板



呼叫中心系统



笔记本租赁



澳洲移民条件



车牌识别系统



婚庆摄影

查看评论

暂无评论

您还没有登录,请[登录](#)或[注册](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved



关闭