

认知科学与类脑计算课程实验报告

实验题目：Hebb 学习		学号：201600181073
日期：2019. 5. 23	班级： 智能 16	姓名： 唐超
实验目的： 根据 Hebb 学习模型的相关知识，使用 Python 语言实现带噪声的数字点阵识别模型。		
实验环境： Jupyter notebook & python3.6 & numpy		
主要算法及步骤： <p>Hebb 算法最简单可以描述为:如果一个处理单元从另一处理单元接收输入激励信号，而且如果两者都处于高激励电平，那么处理单元之间的加权就应当增强。用数学来表示，就是两节点的连接权将根据两节点的激励电平的乘积来改变，即</p> $\Delta w_{ij}=w_{ij}(n+1)-w_{ij}(n)=\eta y_i x_j$ <p>其中 $w_{ij}(n)$ 表示第 $(n+1)$ 次调解前，从节点 j 到节点 i 的连接权值；$w_{ij}(n+1)$ 是第 $(n+1)$ 次调解后，从节点 j 到节点 i 的连接权值；η 为学习速率参考；x_j 为节点 j 的输出，并输入到节点 i；y_i 为节点 i 的输出。</p> <p>对于 Hebb 学习规则，学习信号简单地等于神经元的输出</p> $r = f(W_i^T X)$ <p>权向量的增量变成</p> $\Delta W_i = \eta f(W_i^T X) X$ <p>这个学习规则在学习之前要求在 $W_i = 0$ 附近的小随机值上对权重进行初始化。这个规则说明了如果输出和输入的积是正的，则权增加，否则减小。</p>		
程序设计： <ol style="list-style-type: none">设计 6*5 数字点阵。有数字部分用 1 表示，空白部分用-1 表示，将数字 0-2 的矩阵设计好存储到列表中。创建网络。 定义 Hebb 类，包括函数 train 和 predict，初始化权重 W 为形状为(30,30)零矩阵。 函数 train 包括待训练的一组数字点阵，迭代次数，学习率三个参数，对每一次迭代，输入的		

每一个数字点阵，对 W 做以下更新：

```
self.W += eta * np.dot(Num.T, Num)
```

函数 predict 用权重 W 对待预测的数字点阵做点积：

```
pre = np.dot(self.W, np.asarray(inputNum))
```

对 pre 中大于 0 的值赋 1，小于 0 的赋-1

3. 数字识别测试。

定义如实验一中的增加噪声函数：

```
def addNoise(originNum, noiseSize):
```

对原始数字点阵加入随机噪声，输入创建好的网络中，对结果用 “*” 代替 1，用 “ ” 代替-1 并输出。

调试分析：

在做矩阵运算时，对矩阵维度做了一些调试。

测试结果及分析：

训练时设置 600 次迭代，学习率为 0.2。对于 “half_zero” 的测试结果如下：

half_zero测试：

```
*****
```

```
*   *
```

```
* * *
```

```
*   *
```

```
***
```

```
*   *
```

```
*   *
```

```
*   *
```

```
*   *
```

```
***
```

对于 “one”，“two” 加入 3 个随机噪声，测试结果如下：

one随机噪声测试:	two随机噪声测试:
**	* *
*	*
*	**
*	**
*	* *
* *	** **
**	* *
*	*
*	*
*	**
*	*
*	**
*	*
*	*****

由于数字点阵较少，这一任务较为简单，对于 10 左右的随机噪声，仍能够有效识别：

one随机噪声测试:	two随机噪声测试:
*	* **
* *	***
* *	*
**	*
**	**
**	*****
**	* *
*	*
*	*
*	**
*	*
*	*****