

# 认知科学与类脑计算课程实验报告

|  |          |                 |
|--|----------|-----------------|
| 实验题目：BP 算法分析与实现  |          | 学号：201600181073 |
| 日期：2019.5.23   | 班级：智能 16 | 姓名：唐超           |
| 实验目的：<br>通过使用 BP 算法更新简单神经网络的权重，加深对 BP 算法的理解。   |          |                 |
| 实验环境：<br>Jupyter notebook & python3.7 & numpy  |          |                 |
| <p>主要算法及步骤：</p> <p>BP 算法的学习过程由正向传播过程和反向传播过程组成。</p> <p>在正向传播过程中，输入信息通过输入层经隐含层，逐层处理并传向输出层。如果在输出层得不到期望的输出值，则取输出值与期望值的误差的平方和作为目标函数，转入反向传播，逐层求出目标函数对各神经元权值的偏导数，构成目标函数对权值向量的梯度，将这个梯度乘上一个比例并取反后加到权重上。这个比例将会影响到训练过程的速度和效果，因此称为学习率。梯度的方向指明了误差扩大的方向，因此在更新权重的时候需要对其取反，从而减小权重引起的误差。网络的学习在权值修改过程中完成。误差达到所期望值时，网络学习结束。</p> <p>在这一实验构建的两层神经网络中，隐藏层有两个结点，输出层有一个结点，在每一层用 sigmoid 作为激活函数，sigmoid 函数表达式及导数如下：</p> $f(z) = \frac{1}{1 + e^{-z}}, \quad \frac{df(z)}{dz} = f(z)(1 - f(z))$ <p>用误差平方和作为 Loss：</p> $Loss = (output - real)^2$ <p>设隐藏层结点为 <math>L_2</math>，应用链式法则，则 Loss 对权重 <math>w_1, w_2</math> 的梯度为：</p> $\frac{dLoss}{dw_2} = 2(output - real) \cdot output \cdot (1 - output) \cdot L_2$ $\frac{dLoss}{dw_1} = 2(output - real) \cdot output \cdot (1 - output) \cdot w_2 \cdot L_2 \cdot (1 - L_2) \cdot input$ |          |                 |

### 程序设计：

- (1) 定义输入数据和期望输出

输入数据: `input = np.array([[0.35], [0.9]])`;

label: `real = np.array([[0.5]])`;

误差平方作为损失函数:  $Loss = (output - real)^2$

- (2) 创建神经网络

创建输入层、隐藏层、输出层节点数分别为 2, 2, 1 的神经网络, 初始化权重参数,

$w_1$  为  $2 \times 2$  矩阵,  $w_2$  为 2 维向量。

- (3) 按照网络结构, 前向传播:

```
L2 = sigmoid(np.dot(w1, input))
output = sigmoid(np.dot(w2, L2))
```

参数更新:

每次迭代对网络输出计算误差, 当 Loss 小于  $10^{-5}$  时, 网络学习结束, 否则, 按照上一栏中推导公式, 设学习率为 0.5, 对权重作以下更新:

```
w2 = w2 - e*output*(1-output)*L2.T
w1 = w1 - e*output*(1-output)*w2*L2.T*(1-L2.T)*input
```

### 调试分析：

对权重更新公式重新作了推导, 在程序中对矩阵运算作了调试。

### 测试结果及分析：

经过 76 轮迭代, 误差满足要求,  $output = 0.50309108$ 。