

# BP 神经网络算法实验报告

完成手写 0-9 数字识别

## 摘要:

构造一个三层的 BP 神经网络，完成手写 0-9 数字的识别。设计网络的结构，比如层数，每层的神经元数，单个神经元的输入输出函数；根据数字识别的任务，设计网络的输入和输出；实现 BP 网络的错误反传算法，完成神经网络的训练和测试，最终识别率达到 70% 以上；

## 1. 引言

### 问题描述:

现代生活中，很多场景下需要进行手写字体识别。构造 BP 神经网络并用手写数字的数据进行训练，使其具有识别手写数字的能力。

### 方法描述:

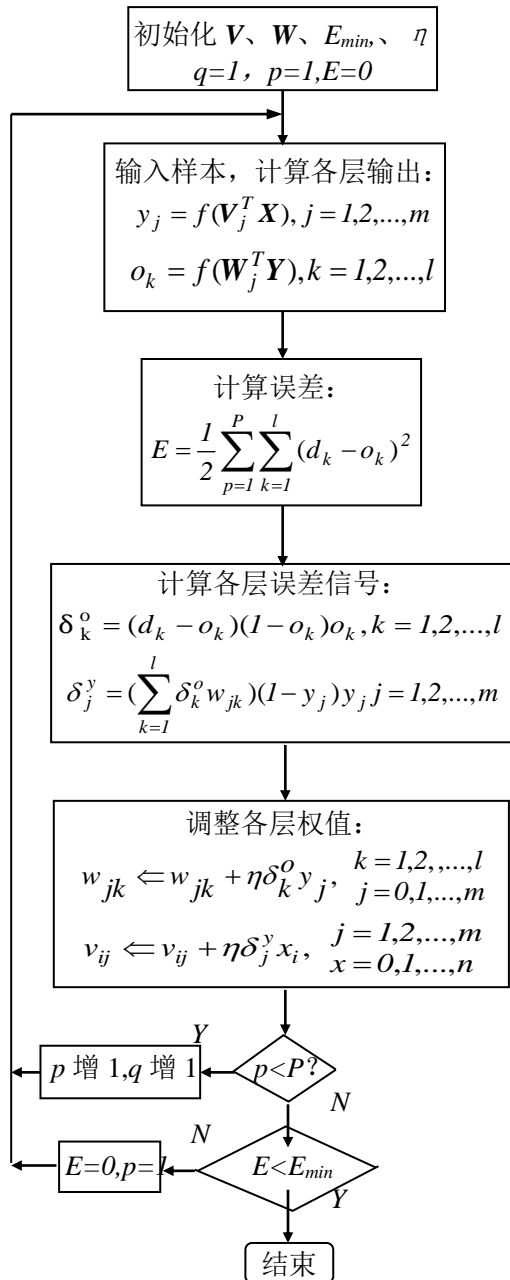
BP (Error Back Proragation, 误差反传) 算法是一种包含隐层的多层前馈网络。它的学习规则是使用误差的梯度下降法, 通过误差反向传播来不断调整网络的权值和阈值, 使网络的误差最小。

- 非线性映射能力: BP 神经网络能学习和存贮大量的输入-输出模式映射关系, 而无需事前了解描述这种映射关系的数学方程, 只要能提供足够多的样本模式对供 BP 网络进行学习训练, 它便能完成由  $n$  维输入空间到  $m$  维输出空间的非线性映射。
- 泛化能力: 当向网络输入训练时未曾见过的非样本数据时, 网络也能完成由输入空间向输出空间的正确映射。
- 容错能力: 输入样本中带有较大的误差甚至个别错误对网络的输入输出规律影响很小。

## 2. 实验过程

### BP 算法思想流程:

- 1) 初始化
- 2) 输入训练样本, 计算各层输出;
- 3) 计算网络输出误差;
- 4) 计算各层误差信号;
- 5) 调整各层权值;
- 6) 对所有样本完成一次轮训。



设计 BP 神经网络，实现手写数字识别

- 1) 使用 mnist 手写数字数据库提供的训练数据集和测试数据集，训练数据集有 60000 个样本，测试数据集有 10000 个样本，每个样本是一个 784 维的向量，是一张手写 0-9 数字的图片按从上到下，从左到右进行向量化的结果。
- 2) 采用三层 BP 神经网络——输入层，隐层和输出层。输入层的神经元数根据使用的数字识别训练集或测试集中输入向量的维数决定，为 784；隐层的神经元数可调整以提高识别率，经过多次测试取 45；输出层的神经元数取 10，分别代表数字 0-9；学习率为 0.2。
- 3) 网络的输入：一个 784 维的输入向量。网络的输出：输出层有 10 个神经元，分别代表数字 0-9，每个神经元的输出值表示识别出的数字为该神经元代表的数字的可能性，值越大则可能性越高，最终识别出的数字为 10 个神经元

中输出值最大的神经元所代表的数字。

### 3. 结果分析

调整隐层神经元数和学习率，使用相同的训练和测试数据集，得出的识别率会发生变化，经过多次调整，最终取神经元数为 45，学习率为 0.2.

结果：运行 10 次，识别率均在 93%~95%之间，满足要求。

一次运行：

```
有 60000 个训练样本
训练中...
训练完成
有 10000 个测试样本
测试中...
测试完成
正确识别的个数: 9397.0
总识别率: 93.97%
各个数字的样本个数: [ 980. 1135. 1032. 1010. 982. 892. 958. 1028. 974. 1009.]
各个数字的正确识别个数: [ 961. 1123. 935. 955. 929. 792. 927. 943. 881. 951.]
各个数字的识别率: [0.98061224 0.98942731 0.90600775 0.94554455 0.94602851 0.88789238
0.96764092 0.91731518 0.90451745 0.94251734]
```

分析与改进：可以看到不同数字的识别率存在差异，这与数字的识别难度有一定关系，也跟训练样本的数量有关。因此，要使识别更加精准，可能需要更多的训练样本，或者找到更好的权值调整方法。