

## 1.CNN介绍

卷积神经网络是人工神经网络的一种，已成为当前语音分析和图像识别领域的研究热点。它的权值共享网络结构使之更类似于生物神经网络，降低了网络模型的复杂度，减少了权值的数量。该优点在网络的输入是多维图像时表现的更为明显，使图像可以直接作为网络的输入，避免了传统识别算法中复杂的特征提取和数据重建过程。卷积网络是为识别二维形状而特殊设计的一个多层感知器，这种网络结构对平移、比例缩放、倾斜或者其他形式的变形具有高度不变性。

CNNs是受早期的延时神经网络（TDNN）的影响。延时神经网络通过在时间维度上共享权值降低学习复杂度，适用于语音和时间序列信号的处理。

CNNs是第一个真正成功训练多层网络结构的学习算法。它利用空间关系减少需要学习的参数数目以提高一般前向BP算法的训练性能。CNNs作为一个深度学习架构提出是为了最小化数据的预处理要求。在CNN中，图像的一小部分（局部感受区域）作为层级结构的最低层的输入，信息再依次传输到不同的层，每层通过一个数字滤波器去获得观测数据的最显著的特征。这个方法能够获取对平移、缩放和旋转不变的观测数据的显著特征，因为图像的局部感受区域允许神经元或者处理单元可以访问到最基础的特征，例如定向边缘或者角点。

## 2.网络结构

我用的是标准的 LeNet-5 （以下卷积操作卷积核均为 $5 \times 5$ ）

- a.读入  $28 \times 28$  进行zero-padding 扩展成  $32 \times 32$       t0层
- b.进行卷积操作变成6个  $28 \times 28$ 的卷积层      c1层
- c.进行max-pooling变成 6个 $14 \times 14$  的下采样层      s2层
- d.再次进行卷积操作变成16个 $10 \times 10$ 的卷积层      c3层
- e.再次进行max-pooling变成16个 $5 \times 5$ 的下采样层      s4层
- f.第三次进行卷积才做变成120个 $1 \times 1$ 的卷积层      c5层
- g.进入bp层的hidden层得到84个 $1 \times 1$ 的bp层      f6层
- f. 最后为bp层的output层得到10个 $1 \times 1$ 的bp层      g7层

## 3.训练过程

神经网络用于模式识别的主流是有指导学习网络，无指导学习网络更多的是用于聚类分析。对于有指导的模式识别，由于任一样本的类别是已知的，样本在空间的分布不再是依据其自然分布倾向来划分，而是要根据同类样本在空间的分布及不同类样本之间的分离程度找一种适当的空间划分方法，或者找到一个分类边界，使得不同类样本分别位于不同的区域内。这就需要有一个长时间且复杂的学习过程，不断调整用以划分样本空间的分类边界的位置，使尽可能少的样本被划分到非同类区域中。

卷积网络在本质上是一种输入到输出的映射，它能够学习大量的输入与输出之间的映射关系，而不需要任何输入和输出之间的精确的数学表达式，只要用已知的模式对卷积网络加以训练，网络就具有输入输出对之间的映射能力。卷积网络执行的是有导师训练，所以其样本集是由形如：（输入向量，理想输出向量）的向量对构成的。所有这些向量对，都应该是来源于网络即将模拟的系统的实际“运行”结果。它们可以是实际运行系统中采集来的。在开始训练前，所有的权都应该用一些不同的小随机数进行初始化。“小随机数”用来保证网络不会因权值过大而进入饱和状态，从而导致训练失败；“不同”用来保证网络可以正常地学习。实际上，如果用相同的数去初始化权矩阵，则网络无能力学习。

训练算法与传统的BP算法差不多。主要包括4步，这4步被分为两个阶段：

第一阶段，向前传播阶段：

- a) 从样本集中取一个样本(X,Yp)，将X输入网络；

b) 计算相应的实际输出Op。

在此阶段，信息从输入层经过逐级的变换，传送到输出层。这个过程也是网络在完成训练后正常运行时执行的过程。在此过程中，网络执行的是计算（实际上就是输入与每层的权值矩阵相点乘，得到最后的输出结果）：

$$Op = F_n (... (F_2 (F_1 (XpW(1)) W(2)) ...) W(n))$$

第二阶段，向后传播阶段

a) 算实际输出Op与相应的理想输出Yp的差；

b) 按极小化误差的方法反向传播调整权矩阵。

#### 4.文件说明

Bp.java            封装bp层的内容  
CNN.java           包含cnn的前传、回传、更新等调用函数  
Convolution.java   包含卷积层的层和对应卷积核bias以及相应操作  
Subsampling.java   包含下采样层的层以及相应操作  
Data.java           用于存储最优参数  
Final.java          面试正式运行文件  
Test.java           系统地测试数据文件  
Test\_Bp.java        调整网络时候测试Bp是否正确的小数据运行  
Test\_CNN.java       调整网络时候测试CNN是否正确的小数据运行

#### 5.实验

\*\*默认 训练数据为1-3000(不包含1-300),测试数据为1-300 每训练3000个测试300个,所有数据训练周期为100轮(100\*3000=30万次) 每训练5次更新一次权重  
训练率为0.005

#### Bp和CNN区别

首先CNN比Bp网络层数更多，且利用权值共享和稀疏连接更有效地进行数字识别

	正确率（纯3000个）	正确率（纯70000个）	混合正确率	初始训练率
<b>Bp</b>	71.2%	96.7%	86.5%	72%
<b>CNN</b>	89.8%	99.2%	92.6%	65%

通过实验，CNN正确率明显高于Bp，不管纯数据还是混合数据，但是初始训练率低，表明其训练速率相对较慢，当然也和优化有关。

### 不同训练率（rate）

	0.01	0.005	0.001	随t变化 (0.001-0.01)
正确率	89.9%	92.6%	90.3%	91.6%
初始正确率	72.6%	68.5%	40.0%	70.2%
稳定花费训练次数	186000	93000	130000	159000

总和来看0.005正确率最高，初始正确率和训练率成反比，稳定花费训练次数无规律。

### 训练次数（trainNumber）

	30	100	300
正确率	88.6%	92.6%	92.6%
达到正确率峰值时候训练百分比	93%	85%	63%
是否稳定	否	是	是

训练100\*3000=30万次时候已经稳定，正确率也不再变化。

### 不同更新权重间隔次数（section）

	1	5	10
正确率	90.2%	92.6%	91.0%
初始正确率	67.3%	68.5%	72.9%
稳定花费训练次数	66000	93000	113000

更新间隔为5的时候正确率最高，初始正确率并没有特别明显的相关，稳定花费训练次数则呈现正相关。

每次训练不同数据量进行测试（length）

	1000	3000	5000
正确率	92.0%	92.6%	88.9%
初始正确率	43.3%	68.5%	73.0%
稳定花费训练次数	86000	93000	103000

初始正确率和稳定花费训练次数随训练量呈现正相关，而正确率则在length=3000时候取到最大值=92.6%