分类号：_____ 单位代码：_____

学　　号：_____

浙江大学

博士学位论文读书报告

中文论文题目：卷积神经网络

英文论文题目：**Convolutional neural networks**

姓名：罗浩

导师：姜伟

专业：控制科学与工程

学号：11532034

学院：控制学院

报告日期　2016年1月

摘　　要

卷积神经网络(Convolution neural networks, CNN)是一种由传统的神经网络(Neural networks, NN)发展而来的深度学习方法。传统的神经网络随着网络层数的增加，参数量与计算量会急剧增加，由于计算机计算能力的限制制约了传统神经网络的发展。更重要的是随着层数的增加，神经网络在反向传播是会出现梯度消失(Gradient vanish)的现象，导致网络无法训练。

卷积神经网络通过局部连接、权值共享和池化采样三个步骤，解决了传统神经网络参数量巨大，无法训练的问题。这也使得卷积神经网络可以从原始数据中直接提取特征进行模式识别任务，取代了传统的人工提取特征加上训练分类器的模式。本篇读书报告将从卷积神经网络的这三个特性切入，分析卷积神经网络作为新一代机器学习技术的有效性。

关键词： 卷积神经网络，神经网络，局部连接，权值共享，池化采样

目　　次

1 传统神经网络的瓶颈

卷积神经网络的前身是神经网络(Neural networks, NN)，为了介绍卷积神经网络，我们先介绍神经网络的数学原理并以此引出神经网络的缺陷瓶颈。

1.1 神经网络的数学原理

神经网络(Neural networks, NN)是一种典型的机器学习方法，是现代卷积神经网路的基础前身。神经网络的基础单元为神经元，其为仿造人脑的神经元细胞所设计，在学术界也称作感知机，结构图如图1所示。每一个神经元有若干个输入，用$X=[x_1,x_2,x_3,...,x_i]^T$表示，对于输入$x_i$有一个权重系数$w_i$，表示为$W=[w_1,w_2,w_3,...,w_i]$，另外加一个常数偏置b，之后通过一个非线性的激活函数f，最后的输出写作：

$$y=f(\sum w_ix_i+b)=f(WX+b) \tag{1-1}$$

若干个这样的神经元全连接起来，便可以得到一个多层感知机(Multi-layer Perceptron, MLP)，也叫做多层神经网络。在多层神经网络中，第一层叫做输入层(input layer)，最后一层叫做输出层(output layer)，中间的都叫做隐层(hidden layer)，图2是一个单隐层神经网络的例子示意。从输入到输出的过程是把数值从低层传向高层，这个过程叫做前馈传播。两层神经元之间都有一条线连接，这条线代表着这两个神经元之间的权重系数。单隐层单输出的神经网络拓展到更一般的形式，有权重系数$w_{ij}^{(l)}$，其中l代表第l到$l+1$层，i代表第$l+1$的第i个神经元，j代表第l的第j个神经元，另外$b^{(l)}$表示第l到$l+1$层的偏置。最后前馈传播的公式可以表示为：

$$a_i^{(l+1)}=f(z_i^{(l+1)})=f(\sum w_{ij}^{(l)}a_j^{(l)}+b^{(l)}) \tag{1-2}$$

利用这个传播公式，神经网络就可以从输入x得到输出的值，之后利用反向传播算法(back propagation, BP)训练得到最优的参数及(W,b)。BP算法是一种基于梯度下降的优化方法，基本原理有点像下山，我们的目标是找到目标函数的最小值。目标函数的分布曲面就好像一个山脉，我们想要去山脉的最低点。最简单的做法就是沿着下山的方向不停走，梯度下

降法就是基于这种原理，函数的梯度方向的反方向就是下山最快的方向，所以只要求出函数的梯度，我们就可以渐渐向函数的最小值逼近。假设给神经网络输入一个x，便可以得到一个预测值$h_{W,b}(x)$，我们定义一个损失函数:

$$J(W,b,x,y) = \frac{1}{2}||y - h_{W,b}(x)||^2 = \frac{1}{2}||y - a^{(l+1)}||^2 = \frac{1}{2}||y - f(z_i^{(l+1)})||^2 \tag{1-3}$$

其中

$$f(z_i^{(l+1)}) = f(\sum w_{ij}^{(l)} a_j^{(l)} + b^{(l)}) \tag{1-4}$$

假设神经网络的输出层看作网络的第$l+1$层，y是输入样本x的真实标签，$f()$是激活函数。之后我们便可以损失函数J求第l层到输出层的参数集$(W^{(l)}, b^{(l)})$的梯度:

$$\frac{\partial J}{\partial w_{ij}^{(l)}} = \frac{\partial J}{\partial z_i^{(l+1)}} \frac{\partial z_i^{(l+1)}}{\partial w_{ij}^{(l)}} = \delta^{(l+1)} \frac{\partial z_i^{(l+1)}}{\partial w_{ij}^{(l)}} \tag{1-5}$$

$$= \delta^{(l+1)} \frac{\partial \sum w_{ij}^{(l)} a_j^{(l)} + b^{(l)}}{\partial w_{ij}^{(l)}} \tag{1-6}$$

$$= \delta^{(l+1)} a_j^{(l)} \tag{1-7}$$

同理有:

$$\delta_i^{(l+1)} = \frac{\partial J}{\partial z_i^{(l+1)}} \tag{1-8}$$

$$= \frac{\partial \frac{1}{2}||y - f(z_i^{(l+1)})||^2}{\partial z_i^{(l+1)}} \tag{1-9}$$

$$= -(y - f(z_i^{(l+1)}))f'(z_i^{(l+1)}) \tag{1-10}$$

如果$f()$是sigmoid激活函数，那么有:

$$f'(z_i^{(l+1)}) = f(z_i^{(l+1)}[1 - f(z_i^{(l+1)}] \tag{1-11}$$

根据导数的链式法则，我们可以得到递推公式:

$$\frac{\partial J}{\partial w_{ij}^{(l-1)}} = \frac{\partial J}{\partial z_i^{(l+1)}} \frac{\partial z_i^{(l+1)}}{\partial z_k^{(l)}} \frac{\partial z_k^{(l)}}{\partial w_{kj}^{(l-1)}} \tag{1-12}$$

$$= \delta_i^{(l+1)} \frac{\partial z_i^{(l+1)}}{\partial z_k^{(l)}} \frac{\partial z_k^{(l)}}{\partial w_{kj}^{(l-1)}} \tag{1-13}$$

$$= \delta_i^{(l+1)} \frac{\partial \sum w_{ij}^{(l)} a_j^{(l)} + b^{(l)}}{\partial z_k^{(l)}} \frac{\partial z_k^{(l)}}{\partial w_{kj}^{(l-1)}} \tag{1-14}$$

$$= \delta_i^{(l+1)} \frac{\partial \sum w_{ik}^{(l)} f(z_k^{(l)}) + b^{(l)}}{\partial z_k^{(l)}} \frac{\partial z_k^{(l)}}{\partial w_{kj}^{(l-1)}} \tag{1-15}$$

$$= (\sum w_{ik}^{(l)} \delta_i^{(l+1)}) f'(z_k^{(l)}) \frac{\partial z_k^{(l)}}{\partial w_{kj}^{(l-1)}} \tag{1-16}$$

$$= (\sum w_{ik}^{(l)} \delta_i^{(l+1)}) f'(z_k^{(l)}) a_j^{(l-1)} \tag{1-17}$$

$$= \delta_i^{(l-1)} a_j^{(l-1)} \tag{1-18}$$

用这个公式一路递推过去便可以求得每一层的梯度，之后利用更新公式便可以不停地更新参数:

$$w_{ij}^{(l)} = w_{ij}^{(l)} - \alpha \frac{\partial J}{\partial w_{ij}^{(l)}} \tag{1-19}$$

$$b^{(l)} = w_{ij}^{(l)} - \alpha \frac{\partial J}{\partial b^{(l)}} \tag{1-20}$$

其中α表示学习率，属于人工设定的参数，来控制学习的步长。通过多次迭代训练，网络将会收敛到一个最优值，这就是神经网络的数学原理。

1.2 神经网络的瓶颈

基于神经网络的数学原理，其存在一些固有的缺点。

（1）梯度越来越稀疏。上一节已经介绍了神经网络的BP算法，根据链式法则可以逐层推导出每一层神经网络的梯度。前一层的网络梯度是后一层的网络梯度乘以当前层的梯度，这就造成了梯度值变得越来越稀疏，最后出现梯度消失的问题。更加糟糕的是如果每一层的梯度大于1，那么将会使得梯度变得越来越大，产生梯度爆炸的问题。梯度爆炸的网络是不收敛的，而一个可以训练的网络通常都存在梯度消失的问题。梯度消失也限制了神经网络的层数，从而限制了神经网络能够表达的泛化能力。

（2）参数量与计算量巨大。传统的神经网络采用全连接的方法。假设两层分别有m和n个神经元，在不考虑偏置参数b的情况下，共需要$m \times n$个权重参数w。而且这种增

长随着层数的增加急剧增加，同时计算量也类似特性。计算机能过实现的参数量和计算量是有限，这也使得神经网络的层数收到了限制，即网络的泛化能力收到了约束。

（3）通常需要手动提取特征。因为传统神经网络的层数和参数量受到了限制，所以在使用神经网络的时候，通常我们不能直接将未处理的原始数据作为网络的输入。因此我们需要手动的对数据进行特征提取，把提取的特征向量作为网络的输入，来降低输入数据的维度。而手动提取特征是十分繁琐而不通用的，需要针对于具体任务设计特殊的特征提取方法。

当然神经网络还存在一些其他缺点，而卷积神经网络主要是解决了神经网络的缺点（2）、（3）。而（2）和（3）的解决同时也顺便减轻了（1）带来的影响。因此本文主要介绍以上所阐述的缺点。

2 卷积神经网络

卷积神经网络是深度学习的标志性成果，其前身是神经网络。2012年，Hinton团队首次利用卷积神经网络Alexnet获得ImageNet挑战赛的冠军，并大幅提高识别准确度。卷积神经网络主要是针对神经网络的缺点做了改进，总的概括起来为三个特性——局部连接、参数共享、池化采样。

2.1 局部连接

在传统的神经网络的图像分类问题中，如果我们要直接用原图像作为网络输入进行训练，那么每一个像素都要为之分配一个神经元。也就是说一个1000×1000像素的单通道灰度图像在输入层我们就需要10^6。如果下一层有100个神经元输出，那么参数量又要扩大一百倍。这样的网络如果最终要达到能够应用的程度，将会有巨大的参数量。

根据视觉神经相关研究的表明，我们的视觉神经元是有层次感。低层的视觉神经元更加关注具体的局部细节（例如边缘，纹理等），而高层视觉神经元更加关注高层特征等（例如轮廓、空间关系等）。低层神经元的实现就是通过局部连接的思想实现，因为低层的视觉特征只需要关注很小的一个区域（patch）的图像，而不需要关注整幅图像。这个被关注的区域就称为感受野，而实现方式就是通过局部连接。例如我们只关心一个10×10的区域，只需要100个参数就可以得到下一层神经元的输出。单独拿出来看，这就是一个10×10卷积核对图像中的这个patch做了一次卷积操作。我们可以看到，输出的这个神经元的值只和这个patch有关，并没有用到整幅图像的值，这就是局部连接。

2.2 参数共享

2.3 池化采样

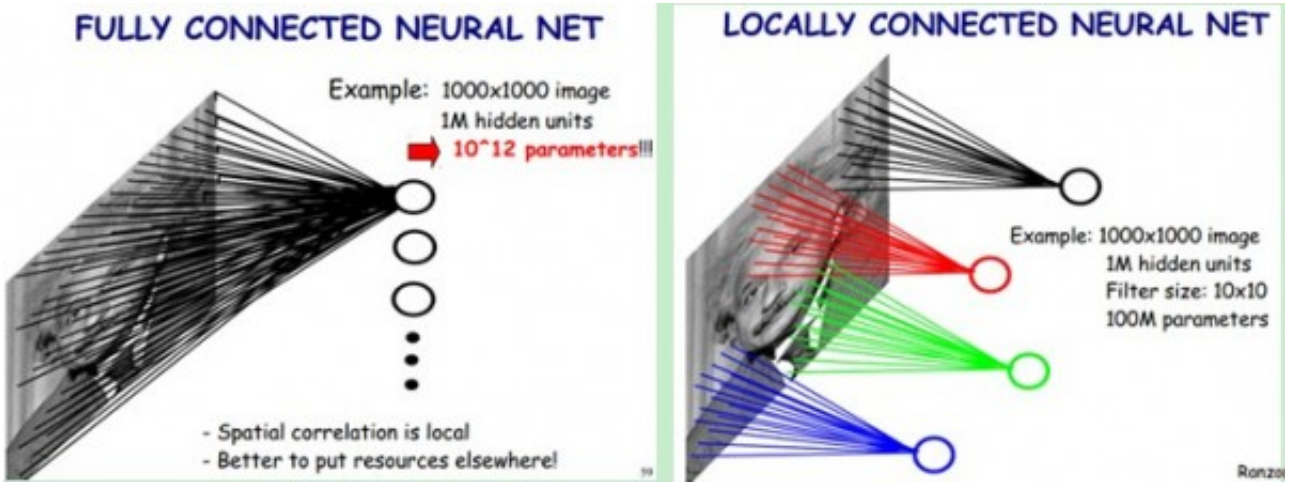

图 2-1　卷积神经网络的局部连接

参考文献