读LeNet的第二部分

CONVOLUTIONAL NEURAL NETWORKS FOR ISOLAED CHARACTER RECOGNITION

通过梯度下降训练的多层网络可以从大量例子中学得复杂，高维，非线性映射的映射，这使得他足以胜任图像识别工作。在传统的模式识别模型中，一个手工设计的特征提取器从输入收集相关信息并剔除无关信息。生成的特征向量，再经过传统的分类器做分类。在这种方案下，标准、全连接的多层网络也可以用来分类器。

而一个可能更有趣的方案是，让特征提取器他自己学习去。

在字母识别中，基本可以直接把原数据（只把图片的尺寸统一一下）丢给特征提取器。虽然可以通过普通的全连接，前向传递网络完成，并在一些任务比如字母识别任务中表现不错，但是这里有两个问题。

1. 一般图片很大，像素点有几百个，而通过全连接（比如第一层有100个单元），那直接第一层就涉及到一万个权重。这就要求有更强大的系统，并且需要更多的训练数据*（否则会欠拟合）*。而且这么多参数，可能以现在硬件水平根本存不下。还有啊，其实非结构化网络一个主要的缺点是对图片和语音应用来说，当输入有平移或者扭曲的变化时，他们不会自动适应。*（比如一个图片扭一扭拉一拉，其实改变是很小的，对系统不该是颠覆性的，前边还认识，拉了一下不认识了这种）*。在字母图片，或者其他2d, 1d信号，在被输入之前，必须得先大概尺寸上规范下，然后放到中间。可惜，没有程序能把这个归一化做的完美，因为手写体通常在单词级别进行归一化，导致单独字母的尺寸，斜体，和位置都不一样。（*这里我不太理解.....是不是意思是，外国人写字时候，比如都是字母a， 在dad里，和 apple里，同一个人写出来的这个a还不一样？*）。再结合不同的书写风格，导致了特征的位置不固定。理论上说，一个够大的全连接网络对这些不同的字母”变体“，通过学习最终可以给出正确的结果。然而，这样的学习可能导致很多处于不同位置的单元的权重却很接近，以便应对特征可能出现在图片的不同位置。学习这些权重需要很多的训练数据来覆盖每种”变体“。而在后边说的卷积网络中，通过对不同空间复制权重，就能直接自动实现以不变应”变体“的效果。lol
2. 全连接结构的一个缺点是，输入的数据的拓扑属性完全被忽略了！（*查了下拓扑学，是研究几何图形或空间在连续改变形状后还能保持不变的一些性质的学科*） 还有，他以任意顺序输入变量都不会影响训练的结果。但相反的，图片（或者语音）

有很强的2d局部结构，：在空间或时间上有很强的相关性。（*这个我在学cnn时候也想到了*）。由于局部相关，识别空间和时间对象之前，先提取整合局部特征会很有帮助，配置邻近的变量们时，可以看做在归纳为一些小的类别（比如边，角...）。卷积网络通过把隐藏单元的”感受野“限制在局部来促进局部特征的提取。（*这个感受野是专有名词吗.. 总之这个表达的就是卷积的过程，检查单元看做老师，输入看做学生，全连接那种是老师一个个的检查学生，现在是一个区域一个区域的检查，团伙打击lol*）

A. Convolutional Networks

卷积网络组合了3个结构理念，来保证可以应对一定程度的偏移、缩放和扭曲：局部感受野、共享权重、空间或时间的二次采样。一个解决字母识别的典型的卷积网络叫做LeNet-5（图1所示）。输入层接收到一个经过简单的尺寸同一，字母居中处理后的图片。在这个模型中，一层中每个单元的输入来自上一层相接近的一组单元。

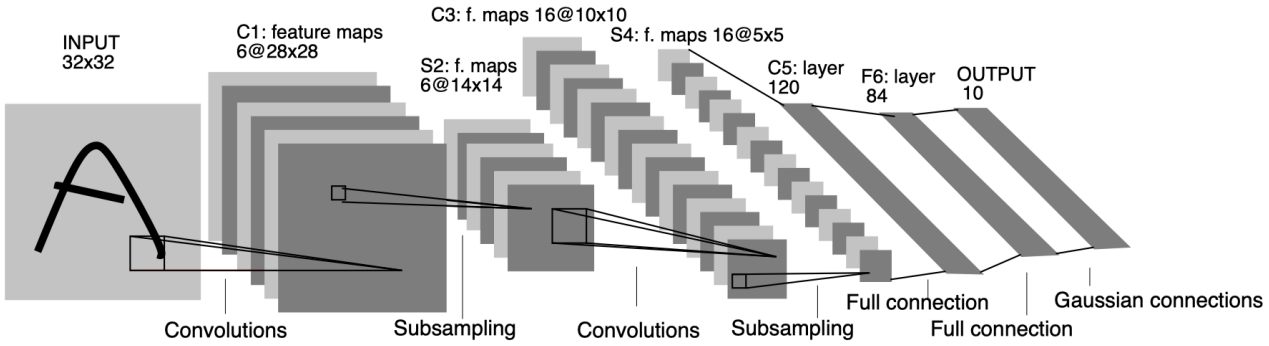

Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

图1

连接单元到输入的局部感受野的主意要追溯到60年代的感知器，还有差不多同时被Hubel和Wiesel在猫视觉系统里发现的局部敏感，方向选择。局部感受野在视觉学习神经网络中已经被用过很多次了。通过局部感受野，神经元你可以提取触及视觉特征，比如方向边，重点，角（或者类似的特征，在语音频谱中）。然后这些特征被随后的几层用来检测更高级的特征。像前边说的，输入的扭曲或者偏移会导致特征的位置的改变。而且，初级特征检测器在图像的局部有用，那么可能在整个图片上也有用。（*这里有点不太懂，应该是那个猫的项目得出的结论？*）这个知识点我get了，所以我通过让一组单元（他们的感受野对应图片的不同局部位置）使用同样的权重向量。一层中的单元组织在一个平面里，他们共享使用同样的一组权重。这一组单元的输出叫做特征图。特征图里的所有单元对图的不同区域做着同样的操作。一个完整的卷积层由不同的特征图组成（每个特征图用不同的权重向量），所以不同的特征可以被提取到不同的特征图里。举个具体的例子，比如图1里，LeNet-5的第一层。第一个

隐藏层的所有单元组织在6个平面里，每个平面是一个特征图。一个特征图里的一个单元对应有25个输入，位于输入的一个5*5区域里，这就叫做该单元的感受野。每个大院有25个输出，因此有25个可训练的系数，再加一个偏差b，一共26个。一个特征图里临近单元的感受野集中在上一层对应的的邻近单元。因此邻近单元的感受野会重叠。比如LeNet-5第一层隐藏层中，水平相邻的单元的感受野会重叠4列5行。像之前说的，一个特征图中所有的单元共享同样的25个权重，偏差b也是，所以他们在输入的所有位置检测的是相同类别的特征。该层另一个特征图用另一组不同的权重和偏差所以可以提取不同的局部特征。在LeNet-5中，输入中的每个位置会被分别位于6个特征图例的单元来提取6个不同的特征。特征图的一列实现会用一个单元的感受野扫描输入的图片，然后在特征图中对应的位置存储这个单元的状态。这个操作等价于 一个卷积后加上偏差再压扁，因此模型名字就叫做卷积网络。卷积的核心设定是将连接的权重被特征图里的所有单元们共享。一个有趣的性质是如果输入的图片偏移了，特征图的输出也会偏移同样的距离，而其他部分不会变。这个属性是卷积网络对输入的偏移和扭曲的鲁棒性的基础。

当一个特征被提取后，他的绝对位置信息就不是很重要了。只有他相对于其他特征的相对位置是重要的。比如，当我们知道输入图片有一个水平线端点在右上角角和左上角，还有一个垂直线的端点在图片的下部，我们可以分辨出输入图片是7。特征精确的位置不只与识别无关，甚至可能有害因为字母的位置信息可能会变。我们可以通过一种简单的方法来使的被编码在特征图里特征位置信息变得模糊，那就是减少特征图的空间分辨率。这可以通过一个东西完成，他叫做取样层 — 他负责做局部平均和取样，降低特征图的分辨率，降低输出对偏移和扭曲的敏感度。LeNet-5的第二层就是抽样层。这层压缩成了6个特征图，每个对应上一层的相应特征图。每个单元的感受野是上一层的对应特征图例2*2的一个区域。每个单元计算4个输入的平均值，乘一个可以训练的系数，加一个偏差，然后结果传递到sigmoid。邻近的单元的感受野不会重叠。所以，一个抽样层的特征图的宽和高只有只有上一层的一半。可训练的系数和偏差控制着sigmoid函数的效果。如果系数小，那抽样层的单元约为线性，如果系数大，根据偏差的值可以看做and或or操作。*（这里系数大的情况不是很明白，我没理解错的话，如果average < -b 那就是0， > -b 那就是1吧。 怎么跟与或操作符对应呢。）*卷积层和取样层一般是交替的，形成双金字塔形状：在每层中，随着空间的缩小，特征图的数量增加。（*意思我明白，可双金字塔是怎么个摆出来的*），还是图1中，第三个隐藏层中每个单元的输入来自上一层的多个特征图。卷积层和取样层的结合的灵感来自Hubel和Wiesel的notions of “simple” and “complex” cells，was implemented in Fukushima’s Neocognitron，尽管那时没有出现全局的监督程序，例如反向传播。通过这种降低空间分辨率，再由表达的多样化（很多特征图）来补偿的方法，可以对输入的几何变换保持很大程度的不变性。（*我这翻译的好拗口*）

因为所有的权重是通过反向传播学习的，卷积网络可以被看做他们合成了自己的特征提取器。（*来了，首位呼应，一开始提到了传统的手动设定的特征提取器*），分享权重的技术又一个有趣的副作用，降低了可变参数的数量，因此降低了机器的需

求，降低了测试错误和训练错误的差距。图1中的网络，包括340908个链接，但因为权重共享所以只有6万个可训练的参数。

固定尺寸的卷积网络已经应用于很多领域，包括手写体识别，打印字符识别，在线手写体提识别，以及人脸识别。（*固定尺寸是什么意思。。。*）固定尺寸的卷积网络在各个时间点之间分享权重，被称作Time-Delay Neural Networks TDNNs. TDNNs 被用于音素识别（没有采样），balabala

B. LeNet-5

这一章描述LeNet-5结构和使用的更多细节。LeNet-5包含7层，不包括输入，每层都有可训练的参数。输入是32*32像素的图片。*这比数据库中最大的字母要大多了（*字母内容基本分布在28*28中心的20*20的区域）。这样的原因是希望可以让笔触端点或者角落也能够出现在感受野的中心，这样有利于检测到一些潜在的特征。在LeNet-5中，最后一层卷积层（图里的C3）的感受野的中心形成了一个位于32*32中心的20*20的区域。（*不懂，C3不是10*10吗。。。。*）输入像素的值被归一化了，所以背景层（白色）对应-0.1，前景（黑色）对应1.175。这对于输入的平均值大约为0，方差约为1，有利于加速学习。（*是和分布有关吗？谁能指导一下我*）

接下来的卷积层都叫Cx，取样层是Sx，全连接层是Fx，x是层数。

C1是一个卷积层，有6个特征图。特征图里每个单元连接着输入的5*5的区域。特征图尺寸是28*28，避免连接输入时越界。C1包括156个参数，和122304个连接。（*122304=28*28*156， 156=6*(5*5+1)*）

S2是一个取样层，有6个14*14的特征图，特征图里每个单元连接着输入的2*2的区域。这4个输入加起来乘以一个可训练的系数，加一个偏差，结果传到sigmoid。相邻感受野不重叠，因此S2的特征图的长宽是前一层C1的一半。S2有12个可以训练的参数和5880个链接。（*12=6*(1+1), 5880=6*14*14*(4+1) 最后那个1是sigmoid应该）*

C3是一个卷积层，有16个特征图。特征图里每个单元连接着S2的5*5的区域。表1展示了C3特征图与S2特征图的结合。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE I

EACH COLUMN INDICATES WHICH FEATURE MAP IN S2 ARE COMBINED BY THE UNITS IN A PARTICULAR FEATURE MAP OF C3.

表1

这里为什么不把S2和C3的每个特征图都连起来呢？有两层原因。首先是不完全的链接模式可以保持连接数保持在合理范围内。更重要的是，他打破了对称性。不同的特征图由于获得不同的输入，导致会提取出不同的特征（希望是互补的最好了）。（*emmm.... 我怎么觉得有些牵强，而且那之前s1和c2为什么不这么做...，哦原来没有s1，那没事了*）表中连接模式背后的关系如下：前6个C3特征图取S2中3个邻近的特征图作为输入。后边6个取邻近的4个。再后边3个取不相邻的4个。（*到现在我才发现邻近可能应该翻译为相邻= =。。。。。不过也差不多吧，最邻近的几个他们之间一定是相邻的。。*）最终最后一个特征图取S2所有的作为输入。C3一共1516个可训练参数，一共151600个连接。

S4是一个取样层，16个特征图，每个5*5。每个单元连接C3的2*2的相邻区域，就像之前的C1和S2一样。S4有32个训练参数和2000个链接。

C5是卷积层，120特征图，每个单元连接S4里所有16个特征图的5*5邻域。这里全部链接是因为，S4也是5*5，C5的特征图是1*1：这相当于S4和C5的全连接。C5标记为卷积层，而不是标成全连接层，是因为如果LeNet-5的输入变大，其他不变，那结果就再是1*1了。第7章描述了动态增加卷积网络大小的过程。C5有48120个可训练的链接。

F6，84个单元，（这个数字的原因来自于输出层的设计，后边会介绍。）和C5全连接，有10164个可训练参数。

就像传统神经网络那样，F6的计算就是Wx+b然后传到激活函数（*原文没用这个词，不过我只知道这个词= = 应该差不多吧*）。

激活函数用的是

$$f(a) = A \tanh(Sa) \qquad (6)$$

这里A是振幅，S决定斜率，这个f是奇函数，有水平渐近线+A和-A。

常量A一般选为1.7159。选择这个函数数的基本原理是在附录A。（*哦*）

最终，输出层是Rdial Basis Function（RBF），每个类有84个输入，每个输出通过yi = sum_j(xj - wij) ^2 （*这个RBF部分我略过了。因为我现在只认识softmax，后边还有一小部分是loss function的，也略了*）