

《神经网络与深度学习》第六章知识点

张翼鹏

November 6, 2019

目录

1	卷积神经网络的介绍	2
1.1	局部感受野	2
1.2	共享权重和偏置	2
1.3	混合层	3
1.4	综合	3
2	卷积神经网络的实际应用	3
2.1	训练过程	3
2.2	复盘	5
3	代码部分注意事项	5
4	其他深度学习的模型	5
4.1	递归神经网络 RNN	5
4.2	长短期网络 $LSTM$	6
4.3	深度信念网络 DBN	6
4.4	强化学习	6

摘要

在上一章，我们知道深度网络有着比浅层网络更强大的潜力，若是可以用合适的方法训练出理想的深度网络，那么它能获得更强大的能力。但是上一章也告诉我们训练深度网络具有重重困难，本章，我们将给出可以克服这些困难的技术，并在实战中应用它们，训练出令人满意的深度神经网络。

1 卷积神经网络的介绍

对于手写数字识别的例子，我们之前一直使用的是全连接层的神经网络，即每个神经元与其相邻层上的所有神经元相连。输入层是784个像素点的灰度值，即我们把一个二维的图像（ $28 * 28$ ）抻成了一个一维的向量（ $784 * 1$ ）进行输入。这样的一个网络架构没有考虑图像的空间结构，而是在以完全相同的方式去对待相距很远和相距很近的输入元素。这种情况下神经网络想获得图像的空间结构信息只能从训练数据中去逐渐推断。

而卷积神经网络在初始架构时就会利用图像的空间结构，所以它非常适用于图像分类问题。卷积神经网络采用了以下三种基本概念，依然以手写数字识别为例展开介绍。

1.1 局部感受野

在卷积神经网络中，输入层是由 $28 * 28$ 的方形排列的神经元组成，第一个隐藏层的神经元不会与所有输入相连接，而是分别只连接输入层的一个小区域，比如 $5 * 5$ 的区域，即25个输入神经元。这个区域被称为隐藏神经元的局部感受野。

与普通神经网络类似，隐藏神经元与局部感受野内的每个输入神经元的连接上有一个权重，隐藏神经元自身有一个偏置，可以认为隐藏神经元学习分析的只是它的局部感受野。

我们在输入层移动这个 $5 * 5$ 的小区域，每在某个方向上移动1个像素（移动的距离称为跨距，跨距在不同网络中可调），便是一个不同的局部感受野，对应一个不同的隐藏神经元。在本例中，第一个隐藏层拥有 $24 * 24$ 个神经元（隐藏层依然保留了图像的二维结构）。

1.2 共享权重和偏置

对于第一个隐藏层中每一个神经元，我们使用相同的权重和偏置。这意味着第一个隐藏层的所有神经元检测完全相同的特征，只是检测的位置不同。这种设定能很好地适应图像的平移不变性。

我们把输入层到第一个隐藏层的映射称为一个特征映射，把定义特征映射的权重和

偏置称为共享权重和共享偏置，二者共同称为一个卷积核或滤波器。一个特征映射只能检测一种局部特征，为了完成图像识别，我们需要多个（记为 k ）不同的特征映射，它们共同构成了一个卷积层。

共享权重和偏置的一个优点是大大减少了卷积网络的参数，加快了训练速度，有助于我们最终建立卷积深度网络。

1.3 混合层

除了上述的卷积层，卷积神经网络中还包括混合层。混合层通常紧接着卷积层使用，功能是简化卷积层输出的信息。

混合层的每个神经元在卷积层的每一个特征映射上对应一小块互不重叠的区域（比如 2×2 ），它以某种方式将这一小块区域的所有激活值混合。比如最大值混合，即输出区域内最大的激活值；或者 $L2$ 混合，即输出区域内所有激活值的平方和的平方根。

1.4 综合

我们将上述所有概念综合在一起，可以得到卷积神经网络的基本结构：

28×28 输入层 $\rightarrow k \times 24 \times 24$ 卷积层 $\rightarrow k \times 12 \times 12$ 混合层 $\rightarrow 10 \times 1$ 全连接输出层

我们依然可以用梯度下降法和反向传播法训练网络中的权重和偏置。

2 卷积神经网络的实际应用

本节我们将编写程序，去测试卷积神经网络在手写数字识别上的效果。

2.1 训练过程

浅层神经网络：

首先构建一个只包含一个由100个神经元构成的隐藏层的浅层神经网络，学习速率 $\eta = 0.1$ ，小批量数据大小为10，无规范化。训练60迭代期后，分类准确率达到97.80%。（虽然该网络使用了柔性最大值和对数似然代价函数，且没有规范化，但它们此时对分类准确率的影响并不大，我们暂时不考虑它们。）

卷积-混合层：

在输入层和第一个隐藏层之间插入一个卷积层和一个最大值混合层（本程序将卷积层和混合层看作一层），使用20个特征映射， 5×5 的局部感受野， 2×2 的混合窗口。训练60 迭代期后，分类准确率达到98.78%，有了一个明显的提升。

第二个卷积-混合层：

在第一个卷积-混合层后插入第二个卷积-混合层，其他超参数不变，分类准确率达到99.06%。

此处有两个问题：

1、应用第二个卷积-混合层意味着什么？

第二个卷积-混合层的输入是 $12 * 12$ 的“图像”，每个“像素”代表原始图像中某个局部特征的存在与否。可以认为这一层的输入是原始图像的一个经过抽象和凝缩的版本，但依然含有大量空间结构，所以使用第二个卷积-混合层是有意义的。

2、第一个卷积-混合层含有20个特征映射，所以会给第二个卷积-混合层输入20幅 $12 * 12$ 的“图像”，那么第二个卷积-混合层该如何处理这些多重图像呢？

我们将允许该层的神经元访问它的局部感受野内的所有（ $20 * 5 * 5$ ）输入神经元。

使用 \tanh 激活函数：

分类准确率基本不变，但训练速度快了一些。

使用修正线性单元激活函数：

分类准确率达到99.23%。

扩展训练数据：

将50000幅训练样本扩展到25000幅，分类准确率达到99.37%。

额外的全连接层：

我们试着增加全连接层神经元的数量，或者增加一个额外的全连接层，分类准确率达到99.48%。

弃权：

对两个全连接层使用弃权技术减少过拟合，迭代期由60变为40，两个全连接层均包含1000个神经元，分类准确率达到99.60%。

为什么不对卷积-混合层使用弃权？

卷积层先天具有强大的对过拟合的抵抗。因为共享权重意味着卷积滤波器被强制从整个图像中学习，使得它们不太可能去选择训练数据中的局部特质。

组合的神经网络：

创建多个神经网络，让它们投票来决定最合适的分类，因为不同的神经网络由于不同的随机初始化可能产生不同的错误。我们训练了5个神经网络，最终，分类准确率达

到99.67%。

2.2 复盘

上一章讲了深度网络学习的很多障碍，特别是梯度不稳定现象，而我们本章对神经网络做了诸多改进，帮助我们继续前行：

1. 使用卷积层极大减少参数数量，使学习更容易。
2. 使用弃权和卷积层等技术减少过拟合。
3. 使用修正线性单元加速训练。
4. 使用GPU并进行长时间训练。
5. 扩展训练数据
6. 使用正确的代价函数和权重初始化。

3 代码部分注意事项

安装Theano：直接pip安装，但是需要前置安装numpy、scipy，或者Anaconda。

安装后可以用CPU运行程序，但是非常慢，1000次迭代大概需要5分钟。

如果想用GPU需要配置环境，过程较为复杂，已经发群里。需要注意的是，只有NVIDIA的显卡才可调用GPU，AMD显卡不行。

Theano的一些编程特点、network3代码的主要结构。

4 其他深度学习的模型

4.1 递归神经网络RNN

在卷积神经网络（CNN）中，各样本之间是独立的，且输入层的输入值完全决定了最后的输出。

而RNN是某种体现出随时间动态变化的特性的神经网络，所以RNN尤其适用于语音识别和自然语言处理等需要处理时序数据或过程的领域中。RNN中每个输出不仅受本次输入影响，还与网络之前的输出有关，即每一刻的输出都是带着之前输出值加权之后的结果。

但是，RNN对于短期记忆的模型效果很好，却无法进行长期记忆的输出，因为权重累加过于庞大，可能导致结果失真、运算效率低下。所以LSTM应运而生。

4.2 长短期网络LSTM

LSTM中的神经元可以判断当前哪些值需要被遗忘，哪些值需要被长期记忆。事实上，LSTM就是在RNN的基础上，增加了对过去状态的过滤，从而可以选择哪些状态对当前更有影响，而不是简单地选择最近的状态。

4.3 深度信念网络DBN

DBN是一种生成式模型，即在DBN中，我们有时可以指定某些特征神经元的值，然后反向运行，生成输入值。举例来说，DBN在手写数字图像上的训练同样可以用来生成类似的手写数字图像，换句话说DBN可以学习写数字的能力。

从概率分布的角度考虑，每个样本均有特征 x_i 对应分类标记 y_i ，我们可以将所有分类模型分为两种：

生成式模型：学习得到联合概率分布 $P(x, y)$ ，然后求条件概率分布。此类模型能够学习到数据生成的机制。当样本数足够多时，算法能够更快地收敛，但计算量较大。

判别式模型：学习得到条件概率分布 $P(y|x)$ 。此类模型比较节省计算资源，由于我们不需要输入的分布来计算条件概率，所以我们可以对输入进行抽象（如降维），从而简化学习问题。

DBN的另一个特点是可以进行无监督或半监督的学习。例如在学习图像数据时，即使训练数据是无标签的，DBN也可以从中学习有用的特征用来理解其他图像。

4.4 强化学习

强化学习模型由两部分组成，一个是智能体（agent），另一个是环境（environment）。

智能体指的是强化学习算法，环境则表示智能体执行动作时所处的场景。环境首先向智能体发送一个状态，然后智能体基于其知识采取动作来响应该状态。之后，环境发送下一个状态，并把奖励返回给智能体。智能体用环境所返回的奖励来更新其知识，对上一个动作进行评估，然后响应下一个状态。如果智能体的某个行为策略导致环境正的奖赏(强化信号)，那么它以后产生这个行为策略的趋势便会加强。智能体的目标是在每个离散状态发现最优策略以使期望的奖赏和最大。