

基于神经网络的交通路牌分类

王子悦
北京交通大学

摘要

本文介绍了使用神经网络进行交通路牌分类的一种办法。传统的机器学习算法在处理实际问题时局限性较大，随着硬件与软件的不断进步，神经网络处理分类问题的能力越来越强，本文中的研究是在 Tensorflow 框架下的分类问题。

1 简介

神经网络，或称作连接模型，是一种模仿动物神经网络行为特征，进行分布式信息处理的算法模型。这种网络依靠系统的复杂程度，通过学习调整内部节点之间相互连接的关系，达到处理信息、解决问题的目的。神经网络最早是由心理学家 W·McCulloch 和数理逻辑学家 W·Pitts 在 1943 年提出的，后在工程实践中被逐步应用。神经网络是机器学习算法中的重要一种，随着 GPU 硬件性能的不断进步以及 Pytorch、Tensorflow 等深度学习框架的不断完善，使用神经网络处理一些典型的问题越来越受到人们的欢迎。其中分类问题是机器学习领域的经典问题，我们采用哈佛大学 2020 年 cs50 课程提供的交通路牌数据集进行分类问题的学习和研究，本次课题分为三大部分：安装代码环境；学习 Python 以及 Tensorflow 的使用；编写代码。本人的基础较为薄弱，仅在高 中计算机课上学习过几讲 Python 编程，所以需要花一些时间重新温习 Python 的使用，并为我的笔记本装好编程环境。本次课题中遇到的主要问题在环境安装和实际工程代码编写上，这会在后面的几部分中分别阐述。最终，在尝试了多种神经网络构成后，该路牌分类问题在 10 个 Epoch 内就可以得到 98% 的分类正确率，平均训练时间 1 秒每个 Epoch。分析认为，能

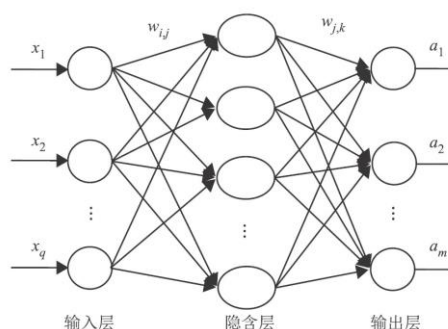


图 1: 神经网络结构

够得到如此高准确率的原因很大可能是因为该问题比较简单，图片也很小，只有 36 乘 36，对于更复杂的分类问题，本神经网络模型不一定能够表现得很好。这一点有待于后续研究。

2 安装代码环境

经过网络查询学习，我将本课题需要用到的软件理清并明晰了其相互关系。首先，我们需要用到 Anaconda，这是一个环境管理器，可以非常方便的切换环境和管理依赖包。我使用的是 Anaconda3，这款软件的安装很简单，没有过多问题，也没有复杂的版本要求。安装了 Anaconda 之后就不需要再单独安装 Python 环境了，只需要在 Anaconda 自己的命令行里通过 `conda create XX`、`conda activate` 等命令，即可创建环境和管理环境。

接着，我们需要 Pycharm，这是一个功能很强大的 IDE，可以方便编写代码时进行纠错、管理依赖包等。本课题使用的是 Pycharm 2020 社区版，是开源且免费的。

最后，安装相关依赖包和 Tensorflow，查阅资料可知，安装依赖包可以在 conda 的命令 行中进行，也可以在 Pycharm 的控制台进行，

57 使用 `pip install` 进行安装, `pip show` 进行查看。
58 `Pip` 是一个管理依赖包的包, 提供一些好用的
59 命令。`Tensorflow` 可以通过 `pip` 直接安装, 这
60 里就涉及到 CPU 版本和 GPU 版本的问题。深
61 度学习运算可以在 CPU 或 GPU 上运行, 但
62 GPU 是并行计算, 对于图像的处理速度是远
63 远快于 CPU 的, 但本课题使用的图像过小,
64 二者差别不大, 甚至 CPU 的计算速度可能快
65 于 GPU, 故本课题采用 CPU 版本的 `Tensorflow`。

66 如果需要使用 GPU 版本的 `Tensorflow`, 则
67 需要几点要求: 首先, 电脑要有英伟达公司的
68 显卡, 其次, 要安装 GPU 驱动程序 `CUDA` 和
69 加速程序 `CUDAnn`。这里不做详细说明, 读者
70 可以自行了解。

71 至此, 代码环境安装完毕。整个安装过程
72 问题很多, 往往是安了这个卸了那个, 根本的
73 原因在于没有搞清各个软件都是做什么的, 彼
74 此有什么关系。这给了我们一个提示, 在遇到
75 复杂问题时, 应该下定决心搞清楚各个部分的
76 作用, 不能仅仅按照教程去安装, 期盼程序能
77 够成功, 即使现在成功了, 未来也还是会遇到
78 很多问题, 不如一次理清。

79 3 学习 Python 以及 Tensorflow

80 3.1 学习 Python

81 `Python` 是一种面向对象的编程语言, 其语
82 法简单, 结构要求不像 `C`、`Java` 那样严格, 在
83 `Pycharm` 的帮助下有很好的编码体验。我首先
84 学习了一些 `Python` 的基础知识, 使用的材料
85 是北京交通大学陈一帅先生所提供的 `Python`
86 入门教程。有了 `C` 语言等其他编程语言的基础,
87 `Python` 的入门很顺利, 主要学习了一些数据格
88 式如数组、元组、列表等格式的相关语法。

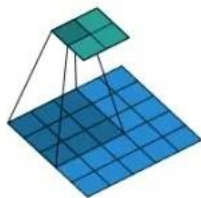


图 2: 卷积层示意图

90 3.2 学习 Tensorflow

91 `Tensorflow` 是一种主流的深度学习框架,
92 其提供了简单、清晰的神经网络编程方式, 复
93 杂的神经网络计算被 `Tensorflow` 打包抽象成一
94 些简单的函数供编程者使用。正是这样的深度
95 学习框架使得机器学习从业者和广大爱好者在
96 编程时能大大提升效率, 促进了神经网络研究
97 的迅速发展。

98 在学习 `Tensorflow` 之前, 我观看了哈佛大
99 学 2020CS50 课程中关于神经网络的讲座, 了
100 解了神经网络的运作原理和基本应用。神经网
101 络由一个个神经元组成, 相互之间有不同的连
102 接方式每一个连接相当于一个函数, 用数量众
103 多的神经元函数通过复杂的方式组合, 就可以
104 拟合出极其精巧的一个函数, 用来解决实际问
105 题, 比如本课题中的分类问题。神经网络图示
106 见图 1。

107 了解了神经网络后, 我主要学习了
108 `Tensorflow` 在建模方面的几个有用的函数, 如
109 `Sequential`、`Conv2d`、各类 `Pooling` 和 `Dense` 等。
110 在神经网络中有不同的层, 如卷积层、池化层、
111 全连接层等等。卷积层是通过设定一个卷积核,
112 对原有图片进行逐个像素的卷积相乘操作, 卷
113 积层可以学习到一些图像的特征, 如边缘、形
114 状等等, 多个卷积层的叠加可以提取出更复杂
115 的特征。卷积层示意图见图 2。池化层可以减
116 小图像特征的大小, 降低复杂度和计算量, 因
117 此应当在网络中周期性的插入池化层。全连接
118 层则是与前一层每个神经元都相连接的一个单
119 元, 同样可以提取特征。

120 4 编写代码

121 4.1 导入数据与预处理

122 本课题使用的数据集是哈佛大学 `cs50` 课
123 程提供的数据集 `GTSRB-德国交通标志数据集`。
124 数据分为 43 个种类的交通标志, 每个类别有
125 数百个数据不等。首先我利用一个循环函数导
126 入所有数据, 这里遇到一个问题就是路径的确
127 定, 在 `Python` 中, 写路径时如果用“\”做分隔
128 符, 很可能产生转义字符的问题, `Python` 将
129 “\0”等内容识别成其他语义, 于是报错。所
130 以要使用 “//” 来代替 “\”。这里我使用了
131 `glob` 函数, 查询文件夹内的所有 `.ppm` 文件也就
132 是数据集的文件格式, 并将其装入一个列表中。

133 接着进行数据预处理，一开始我并没有进
134 行这一步，但在后续神经网络编写完成后发现
135 会报错：某个输入和预想的不一样，经过查询
136 发现，输入神经网络的图片应该为同一大小。
137 于是在预处理阶段使用 `cv2.resize` 全部切割成
138 36 乘 36 的大小。

139 4.2 编写网络模型

140 网络模型的结构如下：一个卷积核为 3 的
141 卷积层、一个最大池化层、一个 `flatten` 层，一
142 个 128 单元的全连接层、一个 43 单元的全连
143 接层。

144 使用 `Sequential` 函数按顺序写完网络结构
145 后，将模型进行编译。使用 `adam` 优化、
146 `sparsecategorical_crossentropy` 做损失函数，以分
147 类正确率为验证标准。这里的损失函数遇到一
148 些问题，在数据预处理部分，我最开始将
149 `lable` 也就是标签转化成了 `one-hot` 编码形式，
150 `one-hot` 编码形式是以 1 的位置存储表示标签的
151 数值。在运用 `sparse` 交叉熵损失函数时，出现
152 了维度不匹配的报错，于是通过查询资料，明
153 晰了编码和相应损失函数的关系，将标签仍然
154 使用数字编码形式，错误解除。最后将数据输
155 入模型中，进行训练并评估。

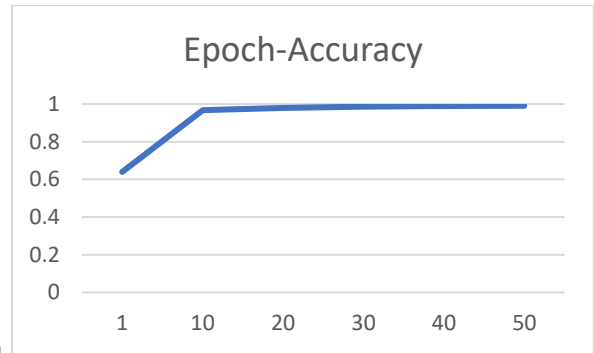
156 5 模型评估

157 在进行 50 个 `epoch` 训练后，我们发现 10
158 个 `Epoch` 内就能够达到 98% 的测试准确率，50
159 个 `Epoch` 后即可达到 99% 的正确率，此后，正
160 确率不再有显著变化。折线图见表 1。分析认
161 为，能够得到如此高准确率的原因很大可能是
162 因为该问题比较简单，图片也很小，只有 36
163 乘 36，对于更复杂的分类问题，本神经网络
164 模型不一定能够表现得很好。

165 6 总结与未来的工作

166 未来的研究主要有两个方面：一是处理更
167 大更复杂的图片，二是建立更好更快的模型。
168 目前深度学习发展速度很快，模型不断朝着高
169 精度、轻量化的方向演进，希望在未来的研究
170 中能够为业界提供创意与创新。

171 总的来看，本次课题收获良多，在深度学
172 习方面，我的基础比较薄弱，仅有一两节课的
173 `Python` 基础。在本次课题中，我熟悉了 `Python`



174 表 1: 训练轮数与正确率折线图

177 `Python` 基础。在本次课题中，我熟悉了 `Python`
178 的使用，学会了深度学习环境的安装，并且用
179 `Tensorflow` 编写了第一个神经网络。在不够了
180 解 `AI` 时感觉 `AI` 很神秘，通过这次课题以及北
181 京交通大学信息网络专题课的学习，我对 `AI`
182 有了更深入的了解，`AI` 也是一种算法，一种
183 特殊的基于学习的算法。通过这次课题，我对
184 `AI` 产生了浓厚的兴趣，希望在将来的研究中
185 能在 `AI` 领域有所建树。

186 参考文献

- 187 [1] 邱锡鹏. 《神经网络与深度学习》[J]. 中文信息学
188 报, 2020, 34(07):4.
189 [2] 王妃. 基于卷积神经网络的路牌检测和识别[D]. 浙
190 江师范大学, 2018.
191 [3] 陈超, 齐峰. 卷积神经网络的发展及其在计算机视觉
192 领域中的应用综述 [J]. 计算机科
193 学, 2019, 46(03):63-73.