

目录

1	介绍	1
2	CNN 网络结构	1
2.1	卷积层	1
2.2	池化层	2
2.3	全链接层	2
2.4	网络构建	2
3	实验过程与结果	4
3.1	数据收集与处理	4
3.2	第一次试验结果	4
3.3	改进实验细节	4
3.3.1	引入预训练 VGG16 与 VGG19	4
3.3.2	数据增强避免过拟合	4
4	可视化	5
4.1	可视化中间输出	5
4.2	可视化过滤器	5

1 介绍

深度学习其实是机器学习的一部分，机器学习经历了从浅层机器学习到深度学习两次浪潮。深度学习模型与浅层机器学习模型之间存在重要区别。浅层机器学习模型不使用分布式表示 (distributed representations)，而且需要人为提取特征。模型本身只是根据特征进行分类或预测，因此人为提取的特征好坏很大程度上决定了整个系统的好坏。特征提取及特征工程不仅需要专业的领域知识，而且需要花费大量人力物力。深度学习模型是一种表示学习 (representation learning)，能够学到数据更高层次的抽象表示，能够自动从数据中提取特征。另外，深度学习的模型能力会随着深度的增加而呈指数增长。Yann Lecun 等人在 1989 年提出基于梯度学习的卷积神经网络算法 [1]，并成功地将其应用在手写数字字符识别，并在当时的技术和硬件条件就能取得低于 1% 的错误率。2012 年，在计算机视觉“世界杯”之称的 ImageNet 图像分类竞赛中，Geoffery E.Hinton 等人凭借卷积神经网络 Alex-Net 以超过第二名近 12% 的准确率一举夺得该竞赛冠军，霎时间学界业界纷纷惊愕哗然。自此边揭开了卷积神经网络在计算机视觉领域逐渐称霸的序幕，此后每年的 ImageNet 竞赛的冠军非卷积神经网络莫属。直到 2015 年，卷积神经网络在 ImageNet 数据集上的性能 (4.94%) 第一次超过了人类预测错误率 (5.1%)。近年来，随着卷积神经网络相关领域研究人员的增多，技术的日新月异，卷积神经网络也变得愈来愈复杂。从最初的 5 层，16 层，到诸如 MSRA 提出的 152 层 ResNet 甚至上千层网络已被广大研究者和工程实践人员司空见惯。基于 CNN 在图像识别中已取得的斐然成就，我们将也在各种书籍和课堂的启发下利用现学得的知识搭建一个 CNN 网络。借助 Keras、TensorFlow 实现猫狗的图像识别与分类。

2 CNN 网络结构

在设计网络结构之前，我们必须先要了解我们需要的步骤和所要达到的目的，针对图像识别，一般我们需要以下四步：

1. 卷积层初步提取特征.
2. 池化层提取主要特征.
3. 全连接层将各部分特征汇总.
4. 产生分类器，进行预测识别.

2.1 卷积层

我们知道假定一个尺寸为 6×6 的图像，每一个像素点都储存着图像的信息，那我们可以定义一个卷积核，从图片中提取一定的特征。但机器一开始是无法确定要识别的部分具有哪些特征，所以是通过不同的卷积核相作用得到的输出值，通过比较，可以发现，卷积层输出值越高，越说明匹配程度越高，越能表现给图片的特征。

以要分辨的猫举例，第一层卷积层能学习较小的局部模式（比如猫耳的边缘、瞳孔等），第二层卷积层由第一层特征组成更大的模式（耳朵、眼睛、鼻子），以此类推，形成最终的抽象概念“猫”。

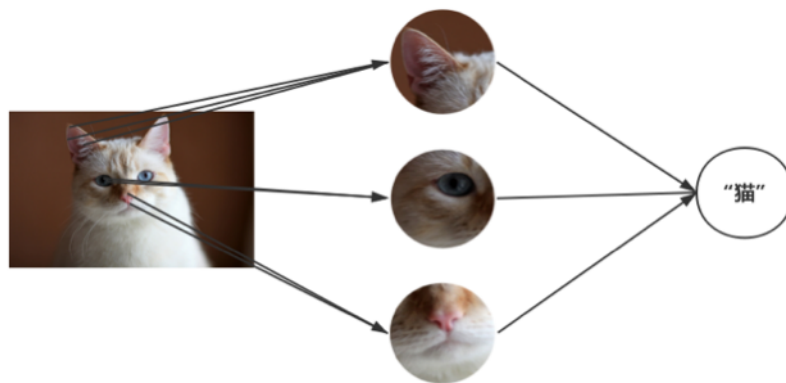


图 1: 视觉世界形成了视觉模块的空间层次结构：比如猫的超局部的边缘组成局部的对象，比如眼镜或耳朵，这些局部对象又组合成高级概念，比如“猫”

卷积的工作原理是在 3D 特征图上滑动这些 3×3 的窗口，在每个可能的位置停止并提取周围特征的 3D 图块。然后每个 3D 图块学到的同一个权重矩阵（卷积核）做张量积，转化为 1D 向量。对这些向量再进行空间重组，转换为 3D 输出特征图。详细过程见下**图 2**，本文将使用 Keras 的 Conv2D 层。

2.2 池化层

池化层的输入就是卷积层输出的原数据与相应的卷积核相乘后的输出矩阵。池化层的目的：

1. 为了减少训练参数的数量，降低卷积层输出的特征向量的维度；
2. 减小过拟合现象，只保留最有用的图片信息，减少噪声的传递；本文将使用 MaxPooling2D 从输入特征图中提取窗口，并输入每个通道的最大值。

2.3 全链接层

全连接层和卷积层的根本区别在于，全连接层从输入特征空间中学到的是全局模式，而卷积层学到的是局部模式。卷积层和池化层的工作就是提取特征，并减少原始图像带来的参数。然而，为了生成最终的输出，我们需要应用全连接层来生成一个等于我们需要的类的数量的分类器。全连接层的存在大大减少特征位置对分类的影响。

2.4 网络构建

基于以上的讨论，我们的 CNN 由 Conv2D（使用 relu 激活）和 MaxPooling2D 层交替堆叠构成。这里我们使用 4 个 Conv2D+MaxPooling 的组合来增大网络容量，也进一步减小特征图的尺寸，使其在连接层 Flatten 层时尺寸不会太大。由于我们面对的是一个二分类问题，所以网络的最后一层是使用 sigmoid 激活的单一单元，使用二元交叉熵作为损失函数。

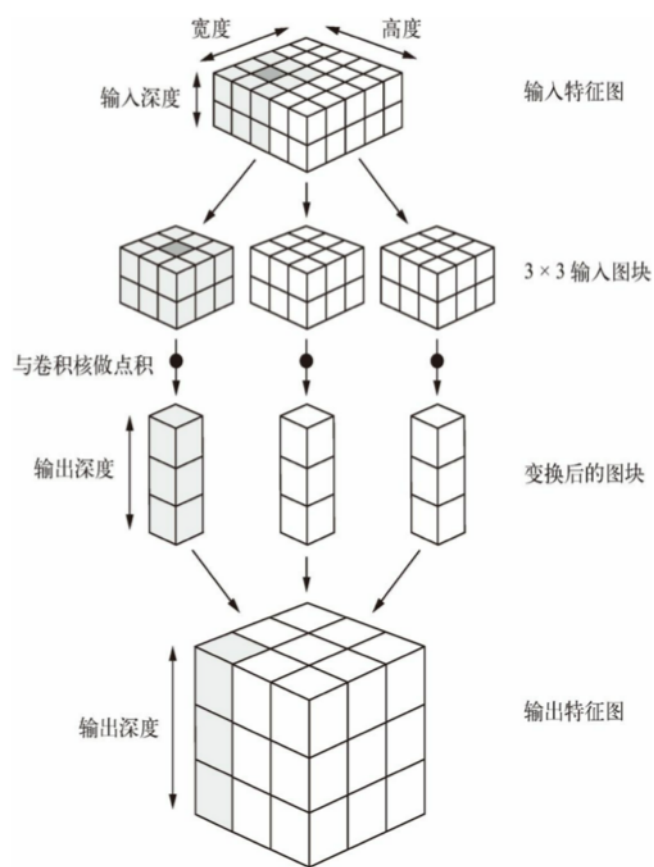


图 2: 卷积工作原理.

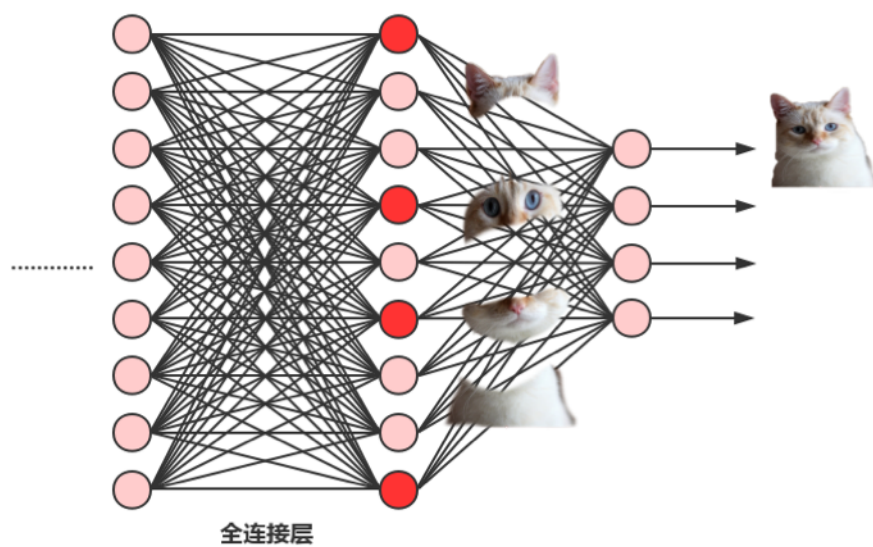


图 3: 图中正红色的神经元表示特征被激活了，同一层的其他神经元，要么猫的特征不明显，要么未被发现。当我们把这些特征组合在一起，即为猫。

3 实验过程与结果

我们在多次尝试配置一台高端机器失败后,最后选用了一台装备了 i7-8700K 与单卡 GTX1080Ti 的机器 (keras 2.2.4, tensorflow 1.4.1) 上进行了我们简单模型的实验. 因为我们在算力上的短板,使得我们的模型在全数据上的训练时间过长,我们不得不在实验的数据量与网络大小上妥协. 但是在这个基础上我们对模型进行了的几次改进,依旧取得了不错的成果.

3.1 数据收集与处理

本文使用 Kaggle 上的猫狗分类数据集, 这个数据集 (training 的部分) 包含 25000 张猫狗图像 (两个类别都有 12500 张). 我们将其两类分别随机分出了 1000 张作为训练集, 各 500 张作为验证集 500 张作为测试集数目.

数据预处理的步骤大致如下:

1. 读取图像文件.
2. 将 JPEG 文件解码为 RGB 像素网格 (150×150)
3. 将这些像素网格转换为浮点数张量
4. 将像素值 (0 255 范围内) 缩放到 [0,1] 区间

我们调用了 Keras 的 `preprocess.image` 类里的 `ImageDataGenerator` 来完成这项工作.

整个数据集被分成了 20 个 batch, 每一个 batch 有 100 个样本.

3.2 第一次试验结果

3.3 改进实验细节

因为我们的算力短板, 我们另寻他路. 希望能够尽量提高在这样的算力能够允许自由实验的前提下达到最好的结果. 我们从第一次试验的结果里可以发现我们的问题主要是算力能够驱动的数据量太小导致了过拟合. 于是我们便引入了预训练模型 VGG16 与 19 来进行改进.

3.3.1 引入预训练 VGG16 与 VGG19

3.3.2 数据增强避免过拟合

由于我们的算力缘故, 能够调用的学习样本并不算多, 可能会出现过拟合的情况, 所以我们采用数据增强的方法, 利用多种能够生成可信图像的随机变换来增加样本, 增强泛化能力. 在 Keras 中, 可以利用 `ImageDataGenerator` 读取的图像进行多次随机变化, 其中的随机变换由多个参数控制, 如角度、缩放的范围、平移范围等, 从而达到抗过拟合的效果.

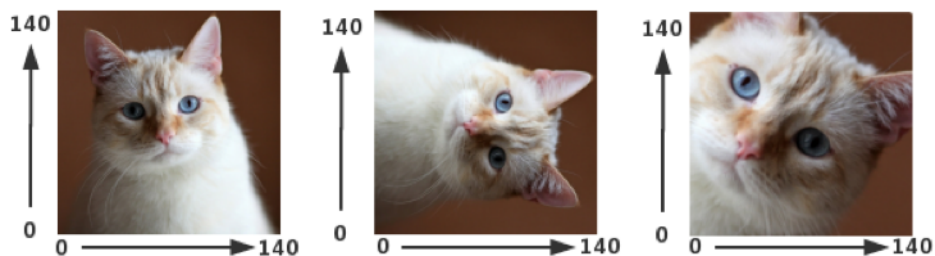


图 4: 通过随机数据增强生成的猫图像

4 可视化

我们最开始的打算是希望能够可视化整个模型训练过程中的特征图与滤波器的选择的变化, 让 keras 在自动化训练的每一个 epoch 的时候输出一张可视化结果. 但是因为这样的操作涉及更改 Keras 源码, 需要重写 keras 的训练函数, 在查看了 Keras 源码以后我们发现这个工作的工作量远远超出了我们的预计. 于是我们选择了一种相对简单但是也相当直观的可视化方式. 在模型训练完毕了以后对于模型的每一层的特征图与滤波器的滤波现象进行可视化.

4.1 可视化中间输出

4.2 可视化过滤器

我们希望可视化的是过滤器 (卷积部分 + ReLu) 部分的功能, 换句话说, 我们希望可视化的是这样的过滤器提取出来的是什么特征. 为了做到这一点, 我们需要知道这样的过滤器在什么样的输入下给出最大的输出值, 即我们需要在输入空间中找到能够使得

参考文献

- [1] 周志华. 机器学习. 清华大学出版社, 2016.
- [2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.
- [3] Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19:1–24, 2018.
- [4] Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197), 2017.
- [5] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. *arXiv preprint arXiv:1710.09668*, 2017.
- [6] Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925, 2019.
- [7] Matthew.yy. 关于 2019ncov 新冠肺炎的建模. <https://blog.csdn.net/Zengmeng1998/article/details/104231869>.
- [8] Sirs 传染病模型求解及 matlab 实现. https://blog.csdn.net/qq_37694821/article/details/86578645.
- [9] 玩转 seir 传染病模型. <https://blog.csdn.net/jinking01/article/details/104145509>.
- [10] Sir 及 seir 建模的简单示例. <https://blog.csdn.net/arcers/article/details/104238668>.
- [11] Hans Nesse. Seir model. <http://www.public.asu.edu/~hnesse/classes/seir.html>.
- [12] 5 分钟入门 google nlp 模型: bert. <https://www.jianshu.com/p/d110d0c13063>.
- [13] Stanford University. The stanford question answering dataset. <https://rajpurkar.github.io/SQuAD-explorer/>.
- [14] 揭开 albert 的神秘面纱. <https://blog.csdn.net/u012526436/article/details/101924049>.
- [15] 李如. 【nlp】google bert 详解. <https://zhuanlan.zhihu.com/p/46652512>.
- [16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Google AI Language Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019.
- [17] 西安交通大学国家天元数学西北中心. 新型冠状病毒疫情传播风险预测再分析. <http://xiammt.xjtu.edu.cn/info/1004/2097.htm>.
- [18] 中华人民共和国卫生健康委员会卫生应急办公室. 截至 2 月 28 日 24 时新型肺炎最新情况. <http://www.nhc.gov.cn/xcs/yqfkdt/202002/4ef8b5221b4d4740bda3145ac37e68ed.shtml>.
- [19] 百度. 实时更新: 新型冠状病毒肺炎疫情地图. https://voice.baidu.com/act/newpneumonia/newpneumonia/?from=osari_pc_3.
- [20] 巫师财经. 疫情对经济的影响: 专题报告. <http://cn.hk.uy/b3y>.
- [21] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>.
- [22] 陈兴. 个人介绍-陈兴的学术主页. <http://www.escience.cn/people/xingchen/index.html>.