

2020

Machine Learning

Third Assignment

姓名	班级	学号
王成航	统计 71	2176122248
张申铎	统计 71	2176112379
王泽昊	统计 71	2176112782

基于卷积神经网络和迁移学习的图像识别模型

摘要

本文首先介绍了 CNN 的网络结构,紧接着自建了一个小型神经网络并使用小样本训练集(2000 张图片)进行了训练,在测试集(500 张图片)上得到了 73.3% 的正确率;对测试集中的数据使用了数据增强(data augmentation)后对神经网络重新进行训练,在同样的测试集上得到了 81.8% 的正确率。之后,考虑到自建的网络存在深度不够的可能性,通过从 VGG19 模型中进行迁移学习,最终在测试集上得到了 94.1% 的正确率。最后,对于小型网络的中间输出和 VGG19 模型的 Filter 进行了可视化。

关键词: CNN 图像识别 迁移学习 VGG 可视化

目录

1 前言	1
2 网络构成	1
2.1 CNN 网络	1
2.2 VGG 模型	3
2.3 网络构建	4
3 模型的训练	4
3.1 数据收集与处理	4
3.2 自建 CNN 训练	5
3.3 改进模型细节	5
3.3.1 数据增强避免过拟合	6
3.3.2 引入预训练模型 VGG19	6
3.3.3 VGG19 后端解锁	7
4 可视化	7
4.1 可视化中间输出	8
4.2 可视化过滤器	11
5 CNN 遐想	13

1 前言

卷积神经网络是一类包含卷积计算且具有深度结构的前馈神经网络, 它是深度学习的代表算法之一, 并且近年来在图像处理方面发挥着越来越大的作用.

Yann Lecun 等人于 1989 年提出基于梯度学习的卷积神经网络算法, 并且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. 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 数据集上的识别错误率 (4.94%) 第一次低于人类的预测错误率 (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.

因此, 基于 CNN 在图像识别中已取得的辉煌成就, 我们在各种书籍和课堂的启发下, 使用 Keras 实现猫狗的图像识别与分类.

2 网络构成

在设计网络结构之前, 必须要了解所需要的步骤和所要达到的目的. 而使用卷积神经网络进行图像识别, 一般需要以下四步:

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 CNN 网络

首先介绍 CNN 网络的基本结构, 对于图像分类问题, 一般来说 CNN 包括卷积层、池化层、全连接层等.

卷积层

假定一个尺寸为 6×6 的图像, 其每一个像素点都储存着图像的信息, 那就可以定义一个卷积核, 来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, 所以会通过不同的卷积核相作用得到的输出值. 一般卷积层越高, 其输出越能表现图片的特征.

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”, 如图 1. 卷积的工作原理是在图像上滑动一个 3×3 的窗口, 它在每个位置停止并提取该位置的所有像素点, 构成一个多维矩阵. 然后将其同一个权重矩阵(卷积核)做张量积, 转化为以为一维的向量. 然

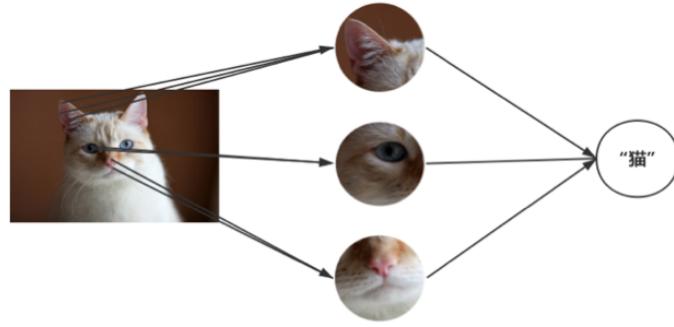


图 1: 视觉世界形成了视觉模块的空间层次结构：超局部的边缘组成局部的对象，如眼睛和耳朵，局部对象又组合成高级概念“猫”。

后将这些向量组合起来转换为多维的输出图像。详细过程如 **图 2** 所示，本文将使用 Keras 的 Conv2D 来完成这一工作。

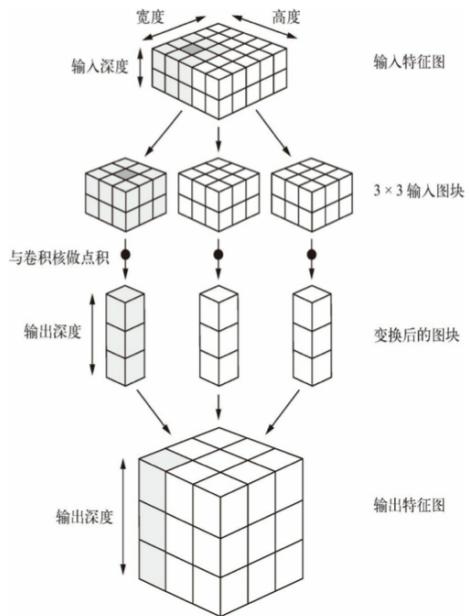


图 2: 卷积的工作原理。

池化层

池化层的输入数据就是卷积层输出的数据与相应的卷积核相作用后得到的输出矩阵。池化层的目的是

1. 减少训练参数的数量，降低卷积层输出数据的维度；
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。

本文中使用的是 MaxPooling2D 从输入的数据中对局部取最大值，然后输出。

全连接层

全连接层的每一个结点都与上一层的所有结点相连, 它起着将已经提取到的特征综合起来的作用. 而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, 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, 减少原始图像带来的参数. 在本文中, 为了生成最终的输出, 需要应用全连接层来生成分类器. 全连接层的存在大大减少特征位置对分类的影响.

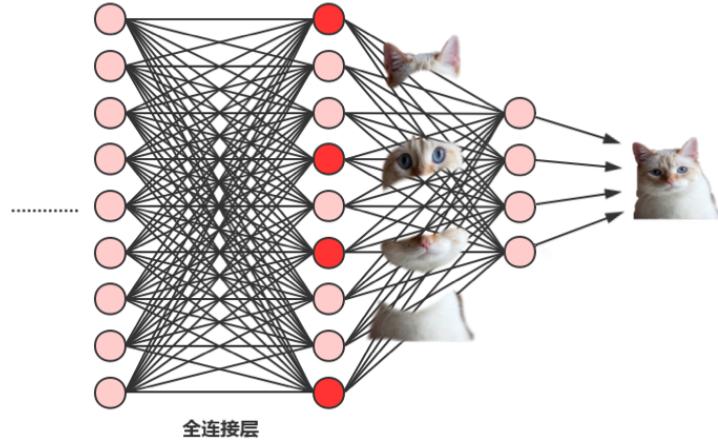


图 3: 图中正红色的神经元表示特征被激活了, 同一层的其他神经元, 要么猫的特征不明显, 要么未被发现. 当我们把这些特征组合在一起, 即为猫.

2.2 VGG 模型

在文章的第二部分, 使用了预训练的 VGG19 模型来进行迁移学习.

VGG 是由 Karen Simonyan 和 Andrew Zisserman 在 2014 年开发, ImageNet 的识别, 它是一种简单而广泛使用的卷积神经网络. VGG 目前有两种架构, 即 VGG16 与 VGG19, 它们没有本质区别, 只是网络深度不同. VGG16 相比 AlexNet 的一个改进是采用连续的几个 3×3 的卷积核代替 AlexNet 中的较大卷积核. VGG 的结构非常简洁, 整个网络都使用了同样大小的卷积核尺寸 (3×3) 和最大池化尺寸 (2×2), 图 4 是其结构的示意图.

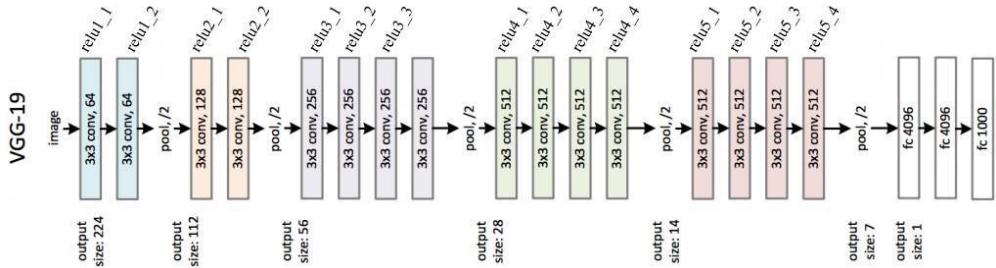


图 4: VGG 示意图.

2.3 网络构建

首先, 我们自建了一个小型的卷积神经网络, 并对其进行训练, 其网络结构如 图 5 所示.

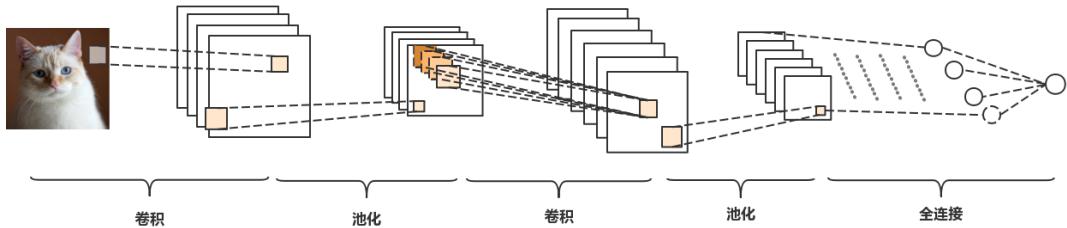


图 5: 小型 CNN 结构示意.

此 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.

其次, 我们还使用 VGG19 进行了迁移学习, 直接将数据输入 VGG19 的卷积基部分, 然后在其顶部添加 Flatten 层和 Dense 层进行输出.

3 模型的训练

在多次尝试使用配置机器失败后, 我们选用了一台配备了 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模型训练. 因为在算力上的短板, 使得我们的模型在全数据上的训练时间过长, 因此, 不得不在训练的数据量与网络大小上妥协. 但是在此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) 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. 同时, 整个数据集被分成了 20 个 batch, 每一个 batch 有 100 个样本.

3.2 自建 CNN 训练

首先, 我们对于自建的神经网络在数据及上进行了训练, 损失函数为交叉熵, 优化算法为 RM-Sprop, 学习率为 $1e-4$, 得到的结果为

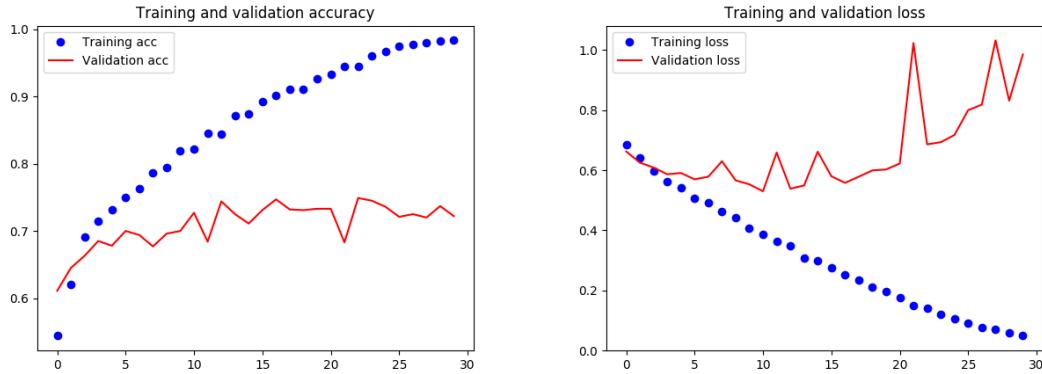


图 6: 自建网络训练结果.

此模型的准确度约为 73.3%, 并不是十分的高, 由于网络并不是很大, 训练速度也十分的快.

3.3 改进模型细节

因为算力的短板, 我们另寻他路. 希望能够尽量提高在这样的算力所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. 于是我们便引入了数据增强与预训练模型 VGG19 来进行改进. 并且我们在试验后期又加入了两块新的 GPU, 使得我们能够从容地对 VGG19 的最后段的卷积进行解锁训练, 进一步地提升了性能. 同时我们又测试了 VGG16 的性能, 发现 VGG19 比 VGG16 的表现强大约 1%.

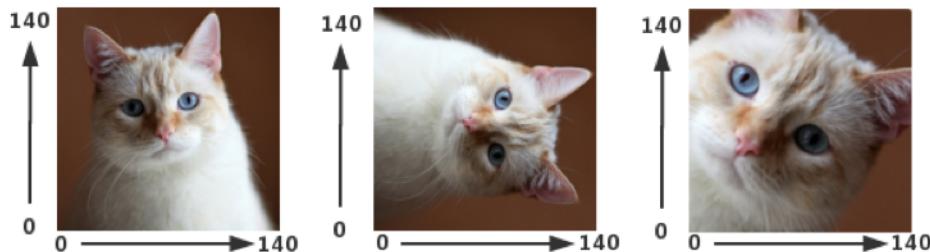


图 7: 通过随机数据增强生成的猫图像

3.3.1 数据增强避免过拟合

由于我们的限制, 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, 从而生成更多的样本达到抗过拟合的效果, 如 图 7.

其中, 数据增强的变换参数设置为, 随机 40 度的旋转角, 随机 20% 的平移, 随机 20% 的放大缩小, 随机 20% 的斜向拉伸并允许翻转的条件下, 损失函数为交叉熵, 优化算法为 RMSprop, 学习率为 1e-5, 此次改进后, 训练结果如下

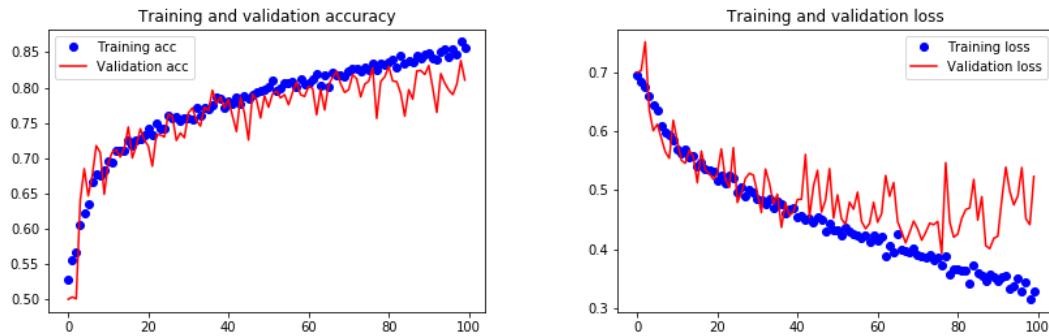


图 8: 数据增强后的训练结果.

此模型实现了在测试集上 81.8% 的准确率. 表现不错但是依然不尽如人意.

3.3.2 引入预训练模型 VGG19

由于数据集过小导致的过拟合可能是准确率不高的主要原因, 我们考虑使用 VGG19 与训练模型来进行迁移学习.

我们在 VGG19 预训练特征提取网络后面加上了两层全连接网络, 在训练集上进行了训练. 损失函数为交叉熵, 优化算法为 RMSprop, 学习率为 1e-5, 训练结果如 图 9.

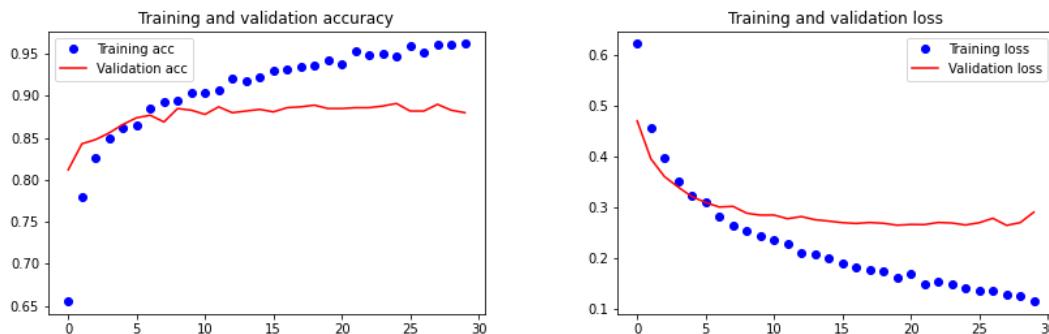


图 9: 使用 VGG19 进行简单特征后的训练结果.

此模型在测试集上的准确率达到了 88.4%, 并且还可以像上文一样使用数据增强技术来提高准确率.

3.3.3 VGG19 后端解锁

卷积神经网络的卷积部分特征提取功能从前段到后段有着逐渐抽象的特性. 前段负责物体的纹理边缘等低阶特征提取而后段负责对如位置关系这样的高阶抽象特征的提取, 因此, 可以认为这样一个阿猫阿狗数据集的低阶特征与其他数据集无异, 如猫的边缘跟车的边缘, 猫的体表纹理跟狗的体表纹理不存在质的差距.

也就是让猫是猫的那个特征并不是边缘或者斑点, 而是猫的这些边缘或者是斑点之间的关系. 所以对于在更大数据集上训练出来 VGG19 网络的前端的低阶特征提取的卷积, 我们可以认为它有着较好的泛化性. 但是对于后端的网络, 我们希望它的卷积在高阶特征上能够更多地去提取对猫狗分类这样一个问题有效的高阶特征. 于是我们解锁了后 VGG19 网络里的第五个卷积块的后四层卷积, 一层大约 2,359,808 个参数, 在我们的训练集上进行了训练. 并且为了避免过拟合, 还在网络训练过程中加入了 Dropout. 其算法性能度量与超参数设置与之前无差.

训练的结果如图 10

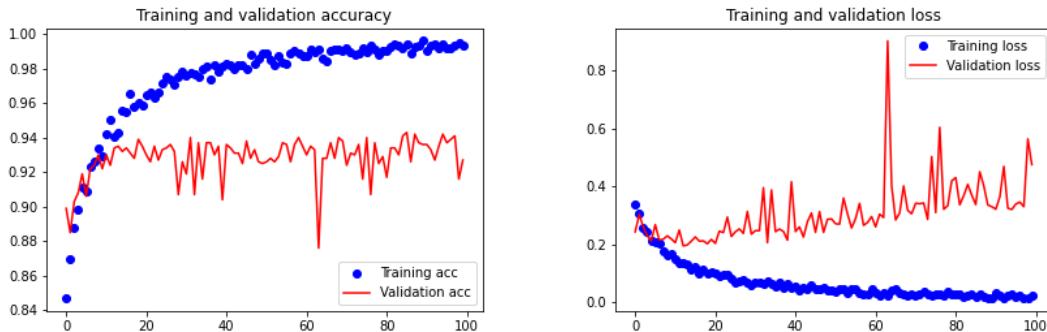


图 10: VGG 解锁后的训练结果.

从图上可以看出因为我们的训练集在有了数据增强以后依然太小, 对比训练集与验证集的训练结果可以推断出在训练 VGG19 的后端卷积层时依然出现了过拟合的现象. 不过此时此模型在测试集上已经达到了喜人的 94.1% 的准确率, 我们停止了对模型的进一步改进与调试.

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 训练完的时候输出一次可视化结果. 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程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.

在可视化 VGG19 的滤波器的过程中, 我们的 GTX1080Ti 又遇到了算力不足完成一次试验需要太久的问题, 我们不得不再租用一台双卡 Titan Xp 进行试验, 在速度上得到了超过两倍的提升使得我们很好地完成了可视化任务. 并且我们在可视化的过程中也发现了一些有趣的问题与现象.

4.1 可视化中间输出

我们对自建的简单模型所得到的卷积层进行了正向传播, 并希望在中间的每一步, 能够获得图像在经过这些滤波器作用后的输出. 我们调用了 Keras 的 Model 类来对我们的模型进行分步的实例化实现这一操作.

原图为:



得到的几个中间输出结果为

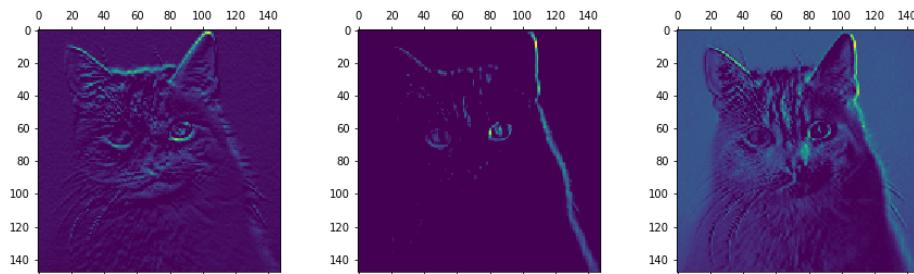
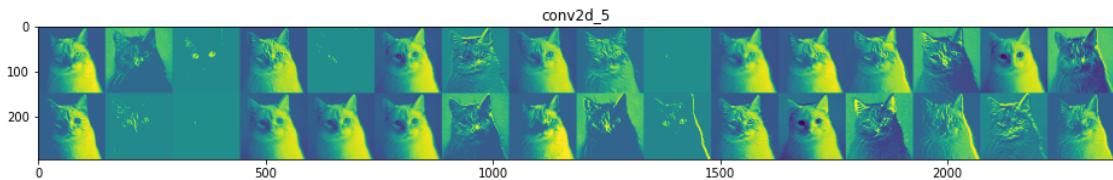
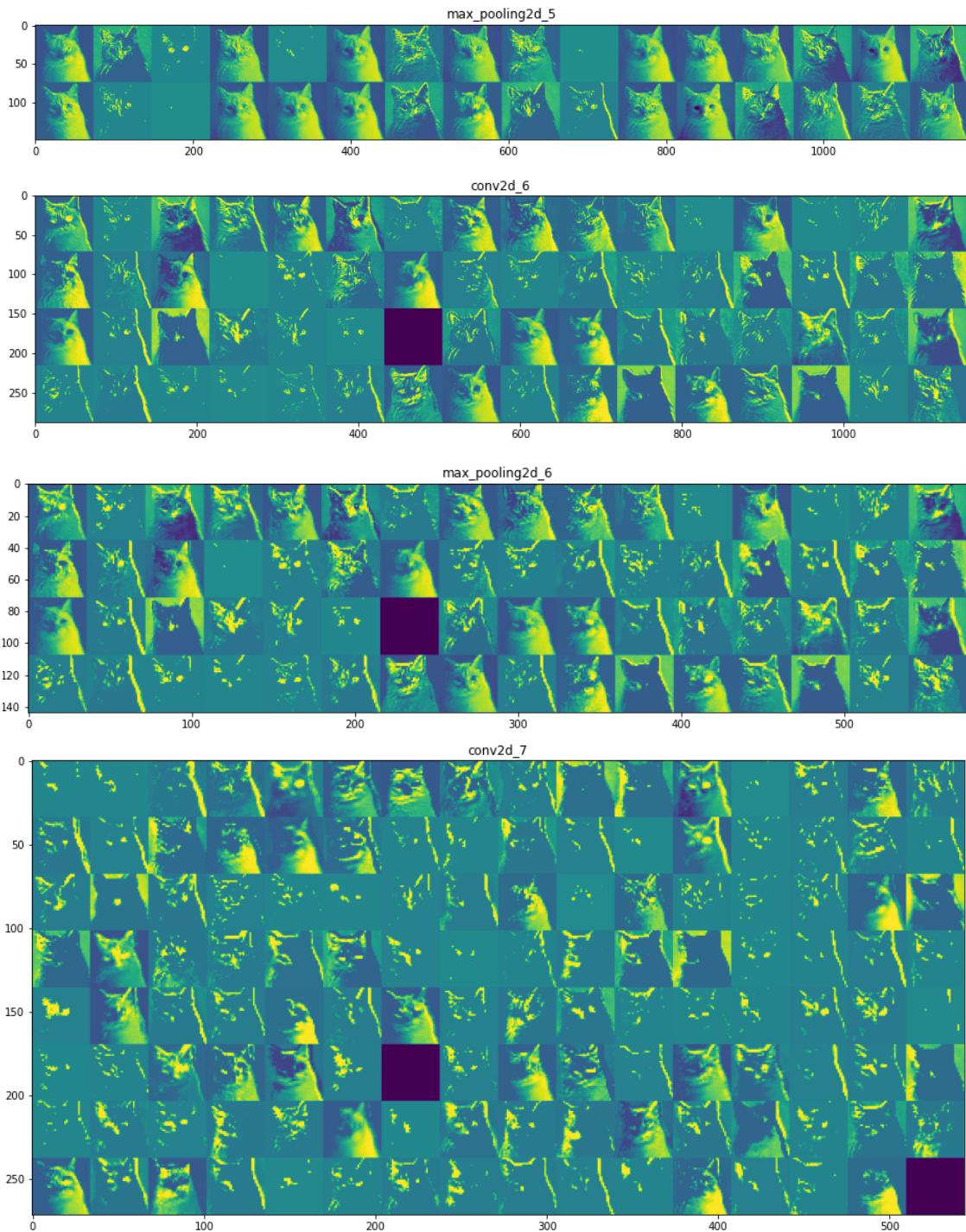
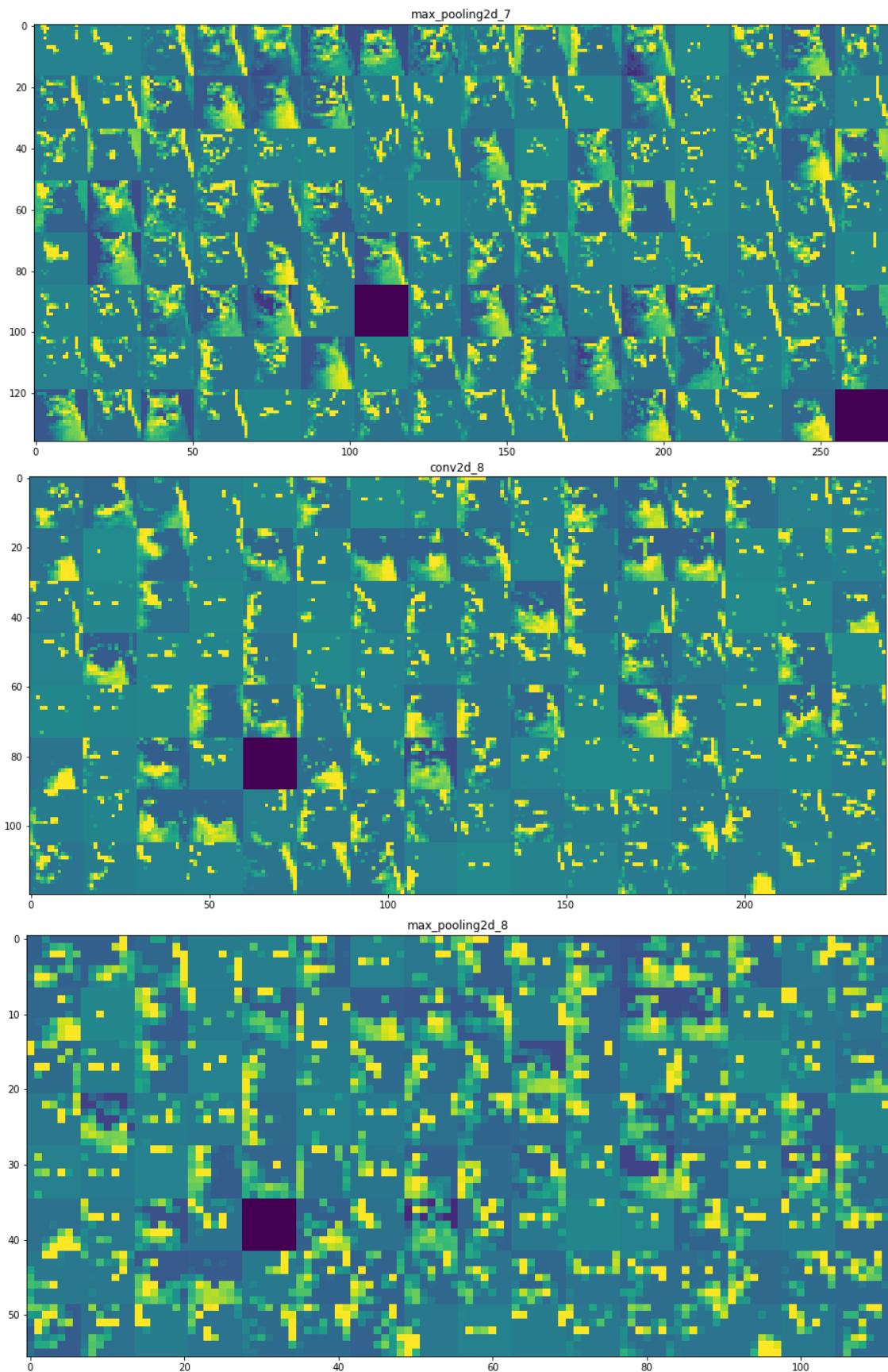


图 11: 左为第 30 个滤波器, 中为第 25 个滤波器, 右为第 15 个滤波器.

可以看到, 第 30 个滤波器似乎相应的是皮毛纹理, 而第 25 个滤波器似乎相应的是眼睛与边缘. 通过对比其这一层所有滤波器的结构我们发现所有的滤波器都应该学习到了不同的特征. 我们还对后面的所有卷积层与池化层都进行了这样的可视化, 希望能够了解机器是如何理解从低阶到高阶的特征的, 下图所示是从第五层到第八层的卷积层与池化层的前向传播可视化.







从上图中可以看出随着卷积层的深入, 滤波器所得到的激活值所可视化出来的图像的可读性在逐渐降低, 浅层的图像还是能看到猫的轮廓与明显的猫的特征, 但是随着网络的深入, 输出变得逐渐抽象, 但是大体上可以分为两类, 一类是两个明显的亮点, 可能代表猫的眼睛特征, 另一类可能是猫的外表轮廓.

4.2 可视化过滤器

我们希望可视化的是滤波器 (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滤波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滤波器 f_0 在什么样的输入 X 下给出最大的输出值 Y , 即我们需要在输入空间 X 里得到 $\arg \max_{X \in X} f_0(X)$. 那么我们如果能引入一个度量使得最大化 $f_0(X)$ 等价于最大化这个度量, 我们就可以在输入空间 X 上梯度下降来进行优化. 我们取这个度量为网络输出张量的所有坐标的均值. 从一张带有微弱噪声的灰色的 (RBG 各值在每个坐标上均为 128, 加上了一个从均匀随机的噪声 $\sigma \in [0, 20]$) 的图像开始, 调用了 Keras 后端的 Tensorflow 里的随机梯度下降函数来进行实现. 并且在对输入空间即图像更新时, 我们采用了归一化的技巧使得我们的更新能够更加稳定. 在第一次尝试可视化的过程中, 我们的训练过程因数值上溢终止, 于是我们在归一化的同时还给里面加入了一个常数 $1e-5$ 来避免这样的现象.

通过对第二个卷积模块的第一个卷积层里的第 128 个 filter 进行了 40 步的随机梯度下降, 再一次对图像张量进行归一化处理将其颜色坐标转换到 $[0, 1]$ 区间上, 通过调用 `plt.imshow()` 函数, 我们终于得到了 VGG 的 filter 的第一个可视化图像, 这个 Filter 似乎对与斑点图案的响应最大.

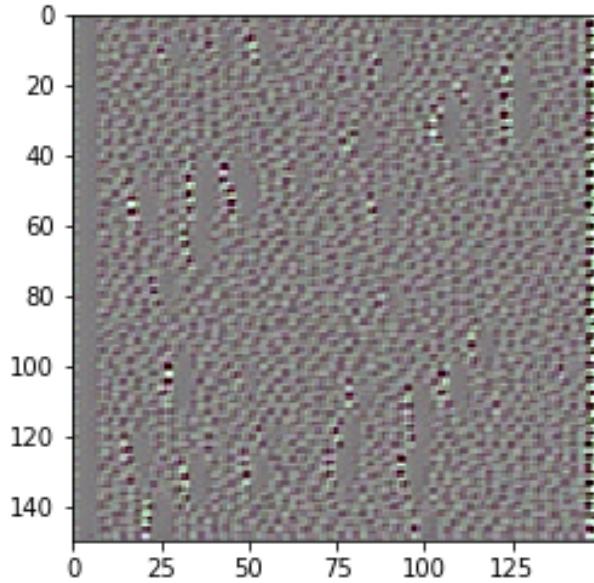


图 12: 预训练的 VGG19 网络第二个卷积模块的第一个卷积层的第 127 个滤波器所在识别的图案可视化图.

但当我们尝试将所有的滤波器图像都进行可视化的时候我们也发现了一些我们没有想到的有趣现象.

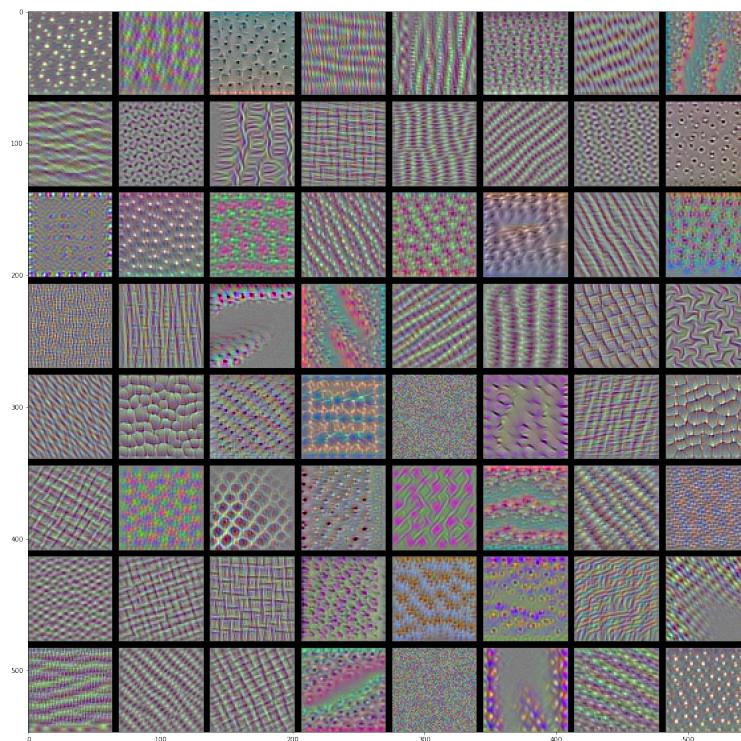


图 13: 预训练的 VGG19 网络第三个卷积模块的第一个卷积层中的 64 个滤波器识别的图案的可视化

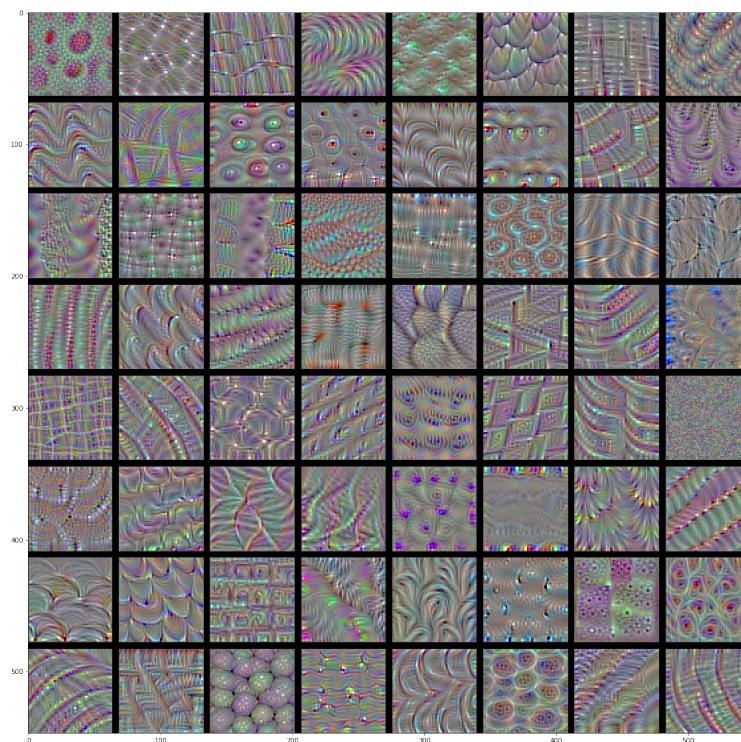


图 14: 预训练的 VGG19 网络第四个卷积模块的第一个卷积层中的 64 个滤波器所识别的图案可视化图

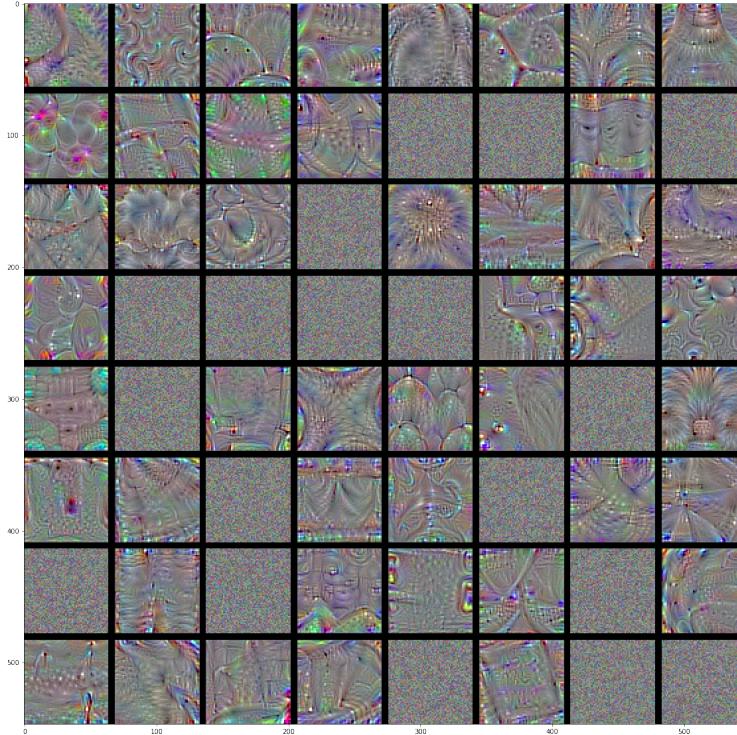


图 15: 预训练的 VGG19 网络第五个卷积模块的第一个卷积层中的 64 个滤波器识别的图案的可视化图

通过观察,我们可以发现从第三个卷积模块开始图 13,我们的滤波器所识别的图像可视化出来的结果出现了完全是噪声的图像。并且随着卷积网络的深入,这样的全部是噪声的图像的数目在不断增加图 14,到了第五个卷积模块图 15,一半的滤波器可视化出来都是与初始噪声相同的模样。换句话说有可能这些滤波器在输入空间中没有作用,这个时候随着网络的深入它所探测的特征已经超出了我们的人眼能够分辨的明显的特征,又或者是随机梯度下降无法很好的还原这些高阶的巨大的卷积块的偏好? 我们无法给出这样的现象的解释,希望老师能在这方面进行答疑。

5 CNN 遐想

在刚过去的 2019 年,技术层面上,计算机视觉的应用在整个人工智能应用领域中占比达 34.9%,已然成为各行业发展的重要支撑。而深度学习在图像识别的任务已经超过人类,可见卷积神经网络能够很好地从输入映射到隐层地特征表达,并且能够层级式地提取特征并通过最后通过内嵌的分类网络完成分类任务。

由于最新的卷积神经网络在某个程度上解决了计算机视觉领域特征表达的问题,CNN 开始在诸多研究方向如目标检测,图像分割,实例分割,图像生成,人脸识别,车辆识别,人体姿态估计等大方光彩,取得的研究成果也是远超传统算法令人振奋。深度学习或者说 CNN 通过刷脸支付,交通天眼系统,无人驾驶等商业应用正在悄悄的改变着我们的生活。并且 CNN 通过轻量化网络或者模型压缩能够在嵌入式或者移动端运行,已慢慢从实验室走向更多的商业化应用,走进寻常百姓家。

参考文献

- [1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.
- [2] LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., and Jackel L. D. Backpropagation applied to handwritten zip code recognition. 1989.
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. 2012.
- [4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2016.
- [5] 弗朗索瓦 肖莱. *Python 深度学习*. 人民邮电出版社, 2018.
- [6] François Chollet. *Deep Learning with Python*. Manning Publications, 2017.
- [7] 周志华. 机器学习. 清华大学出版社, 2016.
- [8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.
- [9] Amusi. 一文读懂 vgg 网络 - 知乎. <https://zhuanlan.zhihu.com/p/41423739>, 2018.
- [10] Cnn 全连接层. <https://blog.csdn.net/techfield/article/details/19933589>, 2018.
- [11] Cnn 卷积神经网络原理讲解和图片识别应用. <https://blog.csdn.net/kun1280437633/article/details/80817129>, 2018.