机器学习纳米学位毕业项目：猫狗图片识别

徐早立

1. 项目定义

给一张猫或狗的图片，能够识别出图片中的动物是猫还是狗。对于人类来说这近乎于本能，可对于人工智能来说，这已经算得上是一种挑战了。在很长一段时间里，人工智能仍然显得有些视觉识别困难，常常指鹿为马，黑白不分。直到深度学习和卷积神经网络的广泛应用，人工智能的视觉识别能力才迎来了突破。

本项目将实践深度学习和卷积神经网络（CNN）的各项技术，实现一个具有猫狗图片识别功能的人工智能。本项目也同时对应数据网站 Kaggle 的一个猫狗识别项目 [1]。这一项目属于计算机视觉（Computer Vision）领域里的图片分类问题。具体地讲，本项目属于机器学习中的分类问题。同时，本项目分类的类别总数为 2，因此是一个二分类问题。

评价这一人工智能的视觉识别能力需要有一个量化的标准。本项目将采用 Kaggle 上这一项目的评价标准 logloss 值，定义为：

$$\mathrm{J} = \frac{1}{m}\sum_{i=1}^{m}[y_i ln\hat{y}_i + (1-y_i)ln(1-\hat{y}_i)]$$

其中 y 表示实际图片中的动物是猫还是狗。如果实际图片中是猫，则定义 y=0；如果实际图片中是狗，则 y=1。而 $\hat{y}$表示人工智能判定一张图片中的动物是狗的概率。m 为测试图片总数。logloss 值越低，则意味着所构建的人工智能猫狗图片识别能力越强。

本项目的目标设定为 logloss 值小于 0.05629，这一目标对应于进入 Kaggle 公开排行榜前 100 名。

2. 项目分析

2.1 数据搜集

本项目的人工智能需要“看过”大量的猫狗图片，才能学到识别猫狗的本领。因此，本项目需要大量的猫狗图片来训练人工智能，同时还需要一些猫狗图片来检验人工智能的识别能力。这些图片都来自 Kaggle 的猫狗识别项目。因此，本项目的数据搜集工作相对简单，直接在 Kaggle 项目主页上下载即可。

Kaggle提供的数据正好分为训练集和测试集。训练集用来训练人工智能，而测试集则用来评判这一人工智能的识别能力。

2.2 探索性数据分析

在开始利用图片训练人工智能之前，有必要先大致了解一下所拥有的素材——数据集。经统计，训练集共有 25000 张图片，标记为猫和标记为狗的图片各有 12500 张，如图 1 所示。每张图片的图片名中含有该图片的分类和序号信息。比如，命名为“cat.0.jpg”意味着这张图片中的动物是猫，图片编号是 0。测试集包含 12500 张猫或狗的照片。测试集的图片只有序号，没有图片的分类信息。

训练集中猫和狗的图片示例如图 2 所示（图片已进行过缩放调整）。初步来看，训练集的数据特征有：

1）猫狗照片数量均等，没有类别不平衡的问题。
2）图片的尺寸大小各异。如图 2(a) 中猫的照片尺寸为 417×299，而图 2(b) 中狗的图片尺寸为 251×359。
3）有的图片包含多只猫或者多只狗，如图 2(c)所示。但是粗略来看，没有发现哪张图片又有猫又有狗。
4）数据集中含有一些异常图片（既没有猫又没有狗），如图 2(d)所示。因为异常图片的数量比较少且暂时没有找到合适的方法来筛选这些异常图片，因此本项目采用的策略是不对异常图片进行剔除。

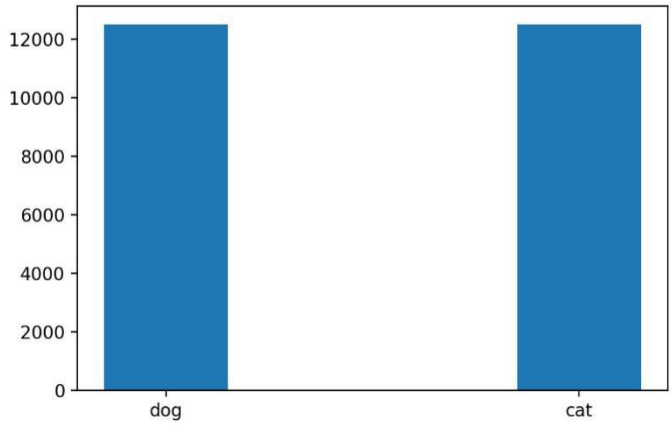

图 1 训练集中猫/狗照片数量柱状图

(a)

(b)

(c)

(d)

图 2 训练集图片示例。 (a) 猫的图片示例； (b) 狗的图片示例；(c) 两只猫的图片； (d) 异常图片示例。

2.3 迁移学习

有了训练数据之后，下一步就是搭建和训练神经网络。最初本项目的训练方案，是参照一些经典的神经网络结构搭建一个相同或者类似的神经网络结构，然后输入本项目的训练图片去训练这个神经网络。所谓的“训练”，就是指通过优化前面定义的 logloss 值来更新神经网络里的参数值。本项目一开始尝试这种从头训练神经网络的方式，可是训练的结果却始终不令人满意（参见后文）。

本项目最终选用的是第二种方案，使用迁移学习技术。迁移学习就是直接利用已经在其他项目上训练过的神经网络，再以此为基础来构建本项目的图片识别器。利用已经训练过的神经网络也有两种方式：一种是把其当做初始网络，然后以此为起点开始更新神经网络的参数。另一种是把其当做特征提取器，作为一种数据预处理的手段，输出本项目图片识别器所需要的图片特征编码。本项目采用的是后者，下面进行简略说明。

本项目迁移学习利用的是在 ImageNet [2] 数据集上已经训练过的 VGG-16 网络 [3]。VGG-16 [4] 的大致结构如图 3 所示。Input 代表图片输入，Conv 表示卷积层，Pool 表示最大池化层，FC 表示全连接层，Softmax 为 Softmax 输出层。作为特征提取器，就是把本项目的图片输入这个已经训练过的 VGG-16 网络，然后获得第一个全连接层的 4096 个单元输出值。这些输出值就可以看做是经过这个已经训练过的 VGG-16 网络提取出来的每张图

片的特征编码。这些编码可作为本项目猫狗识别器的输入，再来训练一个简单的猫狗识别器。更多基于卷积神经网络的迁移学习介绍参见 [5]。

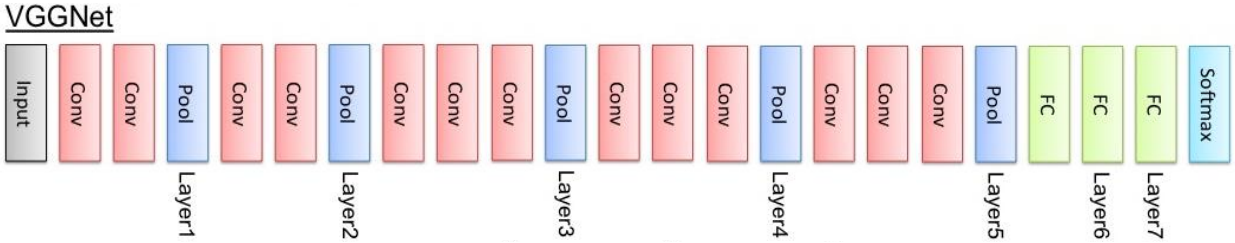

图 3 VGG-16 网络结构

2.4 其他数据预处理

除了使用迁移学习对图片进行特征提取编码以外，还需要做一些其他的数据预处理工作：

1）划分训练集和验证集。本项目采用“留出法”将训练集划分为互斥的训练集和验证集，其划分比例为 0.96/0.04（训练/验证），因此验证集共有 1000 张图片。如果两个不同的图片识别器的识别准确率相差 0.1%，那么在验证集上可以体现出来（0.1%×1000=1 张图片）。训练集用来训练神经网络，验证集用来检验不同神经网络模型的识别能力差异。验证集的 logloss 值可以用来作为对测试集 logloss 值的一种估计。

2）图片标签独热编码。本项目有猫狗两类图片， [0,1] 表示猫，[1, 0] 表示狗。

2.5 构建图片识别器

使用迁移学习完成图片预处理编码后，只需要再搭建一个简易的神经网络结构当做图片识别器，就可以达到很好的识别能力。本项目所搭建的神经网络结构仍然参照 VGG-16 的后段，由两个全连接层和一个 softmax 层组成。第一个全连接层的输出数为 64，而第二个全连接层的输出数为 2，且不经过 RELU 激活函数转换。最后的 softmax 层完成预测概率的归一化。

2.6 神经网络结构说明

下面说明为什么本项目最终使用的是 VGG-16+迁移学习这样的实现方式。表 1 记录了本项目所使用过的卷积神经网络以及其对应的训练结果。

最初项目直接使用了在图像分类项目 [6] 中搭建的简单卷积神经网络。结果显示，这一模型在训练集和验证集上的 logloss 值都太高。

接着项目参照 AlexNet [7] 搭建网络，其训练集和验证集的 logloss 值相对于第一个网络结构有所降低，但仍然还是太高。

随后项目参照 VGG-16 搭建网络，其训练集和验证集的 logloss 值进一步降低。验证集的准确率超过了 90%，达到 94%。采用这一模型对训练集的图片进行预测，然后将结果上传到 Kaggle，得到测试集上的 logloss 值为 0.22924，依然没有达到项目设定的目标。

卷积神经网络	**训练集 logloss**	**验证集 logloss**	**验证集准确率**	**Kaggle logloss**
图片识别项目 CNN	0.3	0.5	76%	NA
AlexNet	0.2	0.45	80%	NA
VGG-16	0.12	0.18	94%	0.22924
VGG-16+迁移学习	0.02	0.044	98.3%	0.05331

表 1 本项目使用过的卷积神经网络（CNN）结构及其对应的训练集 logloss 值，验证集 logloss 值，验证集准确率和提交到 Kaggle 后得到的测试集 logloss 值。

接下来可能的尝试是进一步训练一个“更大”的卷积神经网络，或者使用其他“更好”的卷积神经网络架构，例如 ResNet [8]。但是本项目并没有选择这两种做法，基于以下三点原因：

1) “更大”的卷积神经网络并不一定能得到更好地结果。事实上，无论是 AlexNet 或者 VGG-16，其实都已经足够“复杂”。这是因为只要不加 dropout 层，这些神经网络都能把训练集的 logloss 值训练到非常小，准确率达到近乎 100%。当然此时的验证集准确率会比较低，模型呈现严重过拟合的状况。“更大”的网络只会加深模型的过拟合程度。
2) “更好”的网络架构如 ResNet 是有可能得到更好的结果的。没有选择的原因是训练的时间开销会很大。本项目训练一次 VGG-16 大致需要 3.5 个小时（使用的 GPU 为 GeForce GTX 1060），而 ResNet 只会更慢 [9]。
3) 本项目采用了 CS231n [10]课程中关于如何使用卷积神经网络的建议：“don’t be a hero”，即尽量不要尝试去搭建一个新的卷积神经网络结构。最好的策略是直接使用一个已经在 ImageNet 数据集上得到很好结果的网络，然后下载这个已经训练好的模型对项目图片进行迁移学习，最后再对模型进行微调。

因此，本项目最后使用的策略是经典神经网络结构+迁移学习，而 VGG-16 是这一策略下的第一种尝试，其结果就已经达到项目目标了。其他网络结构+迁移学习是有可能获得更好地结果的，这有待于之后进行尝试。

2.7 模型调参和其他

影响本项目结果的神经网络超参数有：

1）第一个全连接层的输出数。这一数值取得过大，则模型容易过拟合。这一数据过小，则容易欠拟合。最终模型选择的输出数选定为 64。

2）第一个全连接层输出单元的留存率（keep_prob）。第一个全连接层后接了一个 dropout 层，用来降低过拟合程度。留存率最终选定为 0.1。

3）训练轮数（epoch）。神经网络模型采用梯度下降算法来进行训练，优化网络参数。只要模型设置得当，随着训练轮数越多，训练集的 loss 值基本上总是降低的，然而验证集上的的 loss 值有时反而会上升。这意味着模型学到了只在训练集中存在的特征，而这些特征对于区分验证集甚至测试集的图片并没有帮助。因此训练轮数不能过多。当然训练轮数也不能过小，这样有可能会造成欠拟合，也就是说优化还没有完成训练就结束了。最终训练轮数选定的是 20 次。

本项目的图片特征编码输入和第一个全连接层之间，还使用了 batch normalization [11] 的技术，这使得优化的收敛速度加快。Batch normalization 和对输入特征进行归一化的作用类似。归一化使得各个特征都处于同一数量级，这样当模型通过梯度下降进行收敛时，不会出现在数量级过大的特征方向上收敛速度缓慢，而在另外一些数量级相对过小的特征方向上收敛速度过快而来回震荡，使得总体收敛速度缓慢。因此 batch normalization 可以使得收敛速度在各个方向都保持比较平稳的态势。

结果与讨论

首先展示的是训练集和验证集的 loss 值随着训练轮数的变化情况，如图 4 所示。可以看到，logloss 值随着训练轮数的增加而减少，并且在训练的尾声基本到达稳定状态，不再下降。训练集的 logloss 值大约为 0.02，而验证集的 logloss 值大约为 0.04。并且训练集的 logloss 值要一直略低于验证集的 loss 值，这说明模型仍然有轻微的过拟合现象。

类似地，图 4 也展示了训练集和验证集的预测准确率随着训练轮数的变化情况。可以看到，预测准确率随着训练轮数的增加而一直增加，且验证集的准确率很快就达到基本稳定的状态，而训练集的预测准确率仍随着训练轮数增加而缓慢增加。同时也可以看到，训练集的准确率也要一直略高于验证率。到训练尾声训练集的准确率基本到达 99%~100%，而验证集的预测准确率达到 98.3%，同样显示出略微过拟合的情况。

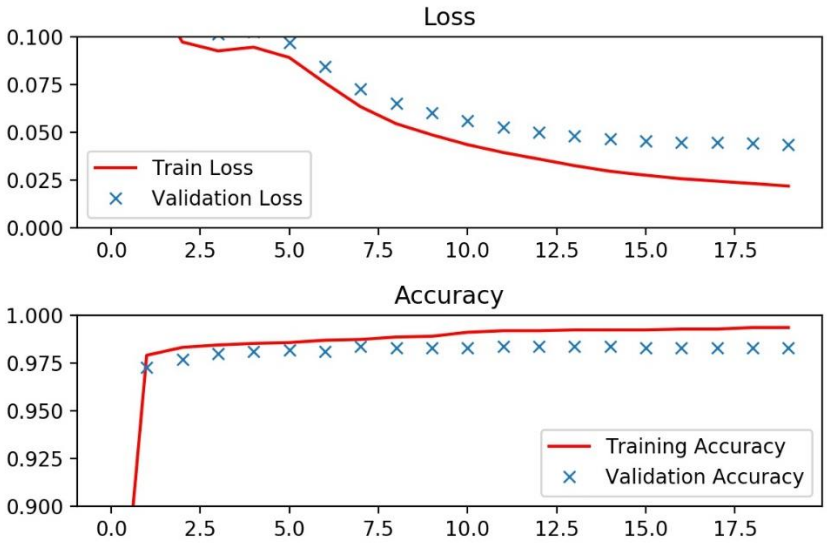

图 4: 训练集和验证集的 loss 值和预测准确率与训练轮数的关系

训练完这一神经网络后，把测试集的图片经过同样的迁移学习学到图片特征编码后输入神经网络进行预测，得到每张图片预测为狗的概率值。之后生成一个 csv 文档，文档中包含两列，第一列为测试集图片的 id，第二列为对应图片预测为狗的概率值。然后把这一 csv 文档提交到 Kaggle，得到测试集的 loss 值为 0.05331（图 5），小于 0.05629，满足项目设定的目标。

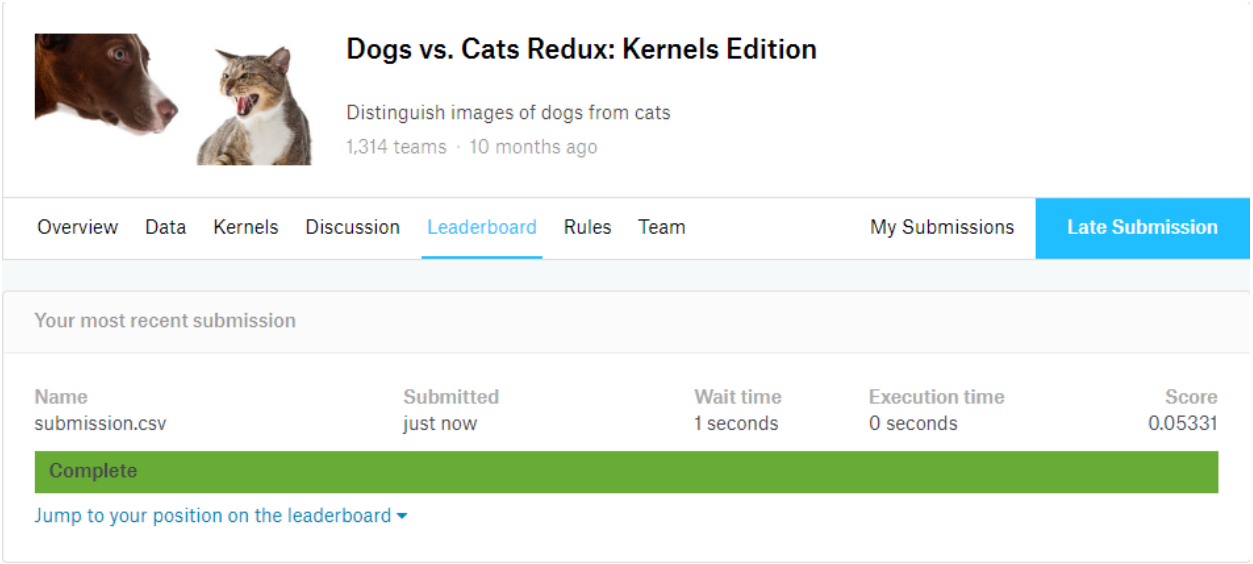

图 5 对测试集做完预测后提交到 Kaggle 得到测试集 loss 值的截图

最后，对测试集的数据做了简单的可视化展示。从测试集中随意选取三张图片，然后从模型的预测中得到预测这三张图片中的动物为狗的概率和为猫的概率，如图 6 和图 7 所示。可以看到，模型的预测准确率以及把握（预测正确的概率）都很高。

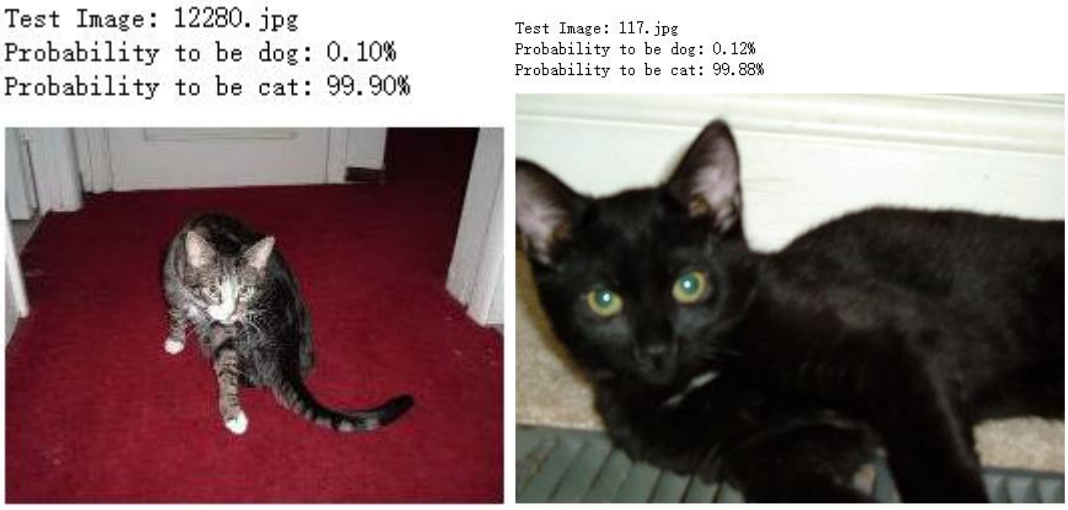

图 6 测试集的预测示例 1 和 2

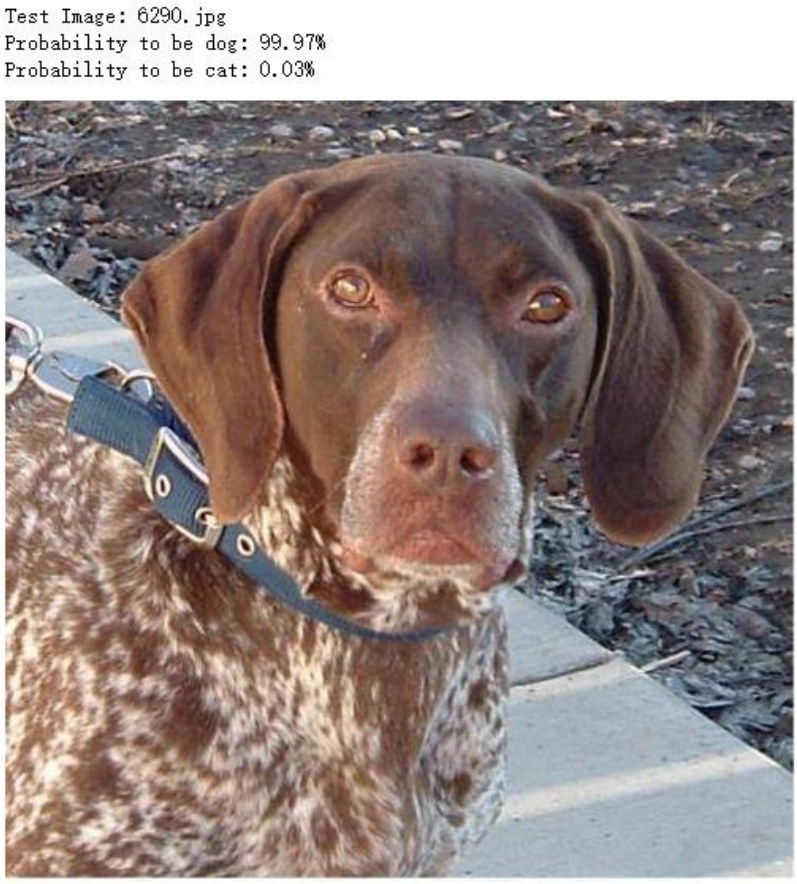

图 7 测试集的预测示例 3

后续为解决轻微过拟合的问题，可以采取的措施是使用更多的训练图片。方法之一是联合使用本项目的数据集和另一个包含猫狗图片的数据集——The Oxford-IIIT Pet 数据集 [12]。方法之二是使用 data augmentation 技术 [13] 通过对现有图片进行变形来获取更多的合成图片。

总结

本项目实现了一个具有猫狗图片识别功能的人工智能，它的 logloss 值为 0.05331，小于设定的 0.05629，满足要求。训练这一人工智能主要使用了迁移学习的技术，利用了在 ImageNet 数据集上已经训练过的 VGG-16 网络作为图片特征提取器，然后再以这些提取到的图片特征编码作为输入，来训练一个简单的用神经网络搭建的猫狗图片识别器。

参考文献

[1] https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition，Kaggle 猫狗识别项目

[2] http://www.image-net.org, ImageNet 数据集

[3] https://github.com/machrisaa/tensorflow-vgg，已经在 ImageNet 上训练过的 VGGNet 网络

[4] https://arxiv.org/pdf/1409.1556.pdf, 包含 VGG-16 的 VGGNet 论文

[5] http://cs231n.github.io/transfer-learning/#tf, cs231n 课程对迁移学习的讲解

[6] https://github.com/udacity/cn-deep-learning/tree/master/image-classification，图像分类项目

[7] https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf， AlexNet 网络

[8] https://arxiv.org/pdf/1512.03385.pdf，ResNet 网络

[9] https://github.com/mrgloom/kaggle-dogs-vs-cats-solution，各种卷积神经网络训练时间对比

[10] http://cs231n.github.io/convolutional-networks/#architectures，CS231n 课程中关于如何使用卷积神经网络的建议

[11] https://arxiv.org/abs/1502.03167，batch normalization 技术

[12] http://www.robots.ox.ac.uk/~vgg/data/pets/，The Oxford-IIIT Pet 数据集

[13] https://www.coursera.org/learn/convolutional-neural-networks/lecture/AYzbX/data-augmentation，deeplearning.ai 课程对 data augmentation 的讲解