机器学习（进阶）纳米学位毕业项目

猫狗大战

杨晨

2017 年 12 月 25 日

目 录

1 定义

1.1 项目概述

“猫狗大战”是Kaggle上最为著名的娱乐型竞赛项目之一,至今已经举办过多次。本项目是2017年3月举办的“Dogs vs. Cats Redux: Kernels Edition”。该项目的目标是在测试数据集中分辨出猫和狗的图片。

这是一个经典的卷积神经网络（Convolutional Neural Network，CNN）图像分类项目。CNN是机器学习领域的重要方法之一，也是目前最前沿的技术之一。其名称中“卷积”表示这种方法主要运用卷积运算，“神经网络”表示这种方法属于监督学习中的神经网络方法。

CNN 的特点之一是利用卷积运算大幅度减少训练神经网络所需的参数数量，使得训练更深的神经网络成为可能,这也正是“深度学习”这一名称的来源之一。

深度学习是机器学习中一种基于对数据进行表征学习的算法。观测值（例如一幅图像）可以使用多种方式来表示，如每个像素强度值的向量，或者更抽象地表示成一系列边、特定形状的区域等。深度学习的好处是用非监督式或半监督式的特征学习和分层特征提取高效算法来替代手工获取特征[1]。

本项目的目标就是运用CNN技术，对12,500张测试图片判断图中动物是狗的概率（0是猫，1是狗）。

1.2 问题陈述

Kaggle上的“猫狗大战”项目有不止一个版本。本项目的要求是标记出“图片中是狗的几率”（ 0是猫,1是狗),以此来进行图像分类。该项目提供了25,000张训练图片，12,500张测试图片。

这是一个经典的CNN图图像类项目。目前对深度学习的研究在CNN图像分类领域已经有很多成果，有很多著名的数据集和有效的模型。其中最著名的数据集之一就是ImageNet，成熟的模型包括VGG、ResNet、Inception等。

在这个项目中，我将选择一些成熟模型分别进行训练，最后合并所有模型进行组合训练，并将每个模型的结果与组合模型的结果进行对比。

本项目将使用迁移学习的方法提高模型效率和简化训练工作量。具体来说，

将使用 Keras 的内置模型以及成熟的 ImageNet 数据集权重进行预训练，在此基础上将模型的后两层换成自己的全连接层和输出层。

1.3 评价指标

在 Kaggle 上，竞赛中使用对数损失（LogLoss）作为评价指标，本项目也采用同样的指标。

损失函数$\mathrm{L}(y,\hat{y})$用于评价预测值$\hat{y}$与真实值y之间的差异程度。损失函数的值是梯度下降法的核心，模型训练的过程在数学上就是（寻找参数）令损失函数最小化的过程。对数损失函数是最常见的用于分类问题的损失函数，其公式为：

$$\mathrm{L}(y,\hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$$

对数损失函数指标是一个连续值，与正确率（正确样本数/总样本数）指标相比，对判断错误的样本会给予更严重的惩罚，因此对模型性能有着更高的要求。从直观上理解，该指标不仅要求模型能够“正确地”分类，而且要求能够“可靠地”分类，避免“蒙对”的情况。使用这一指标，使得我们对模型可以进行更细致的对比和评价。在当前主流模型普遍具有较强图像分类能力的环境下，使用对数损失是一个非常恰当的指标。

2 分析

2.1 数据可视化

该项目的数据来自对应的 Kaggle 竞赛数据集。完整的数据集中包括 25,000 张已标记的图片——其中猫和狗各 12,500 张，和 12,500 张未标记的测试图片。

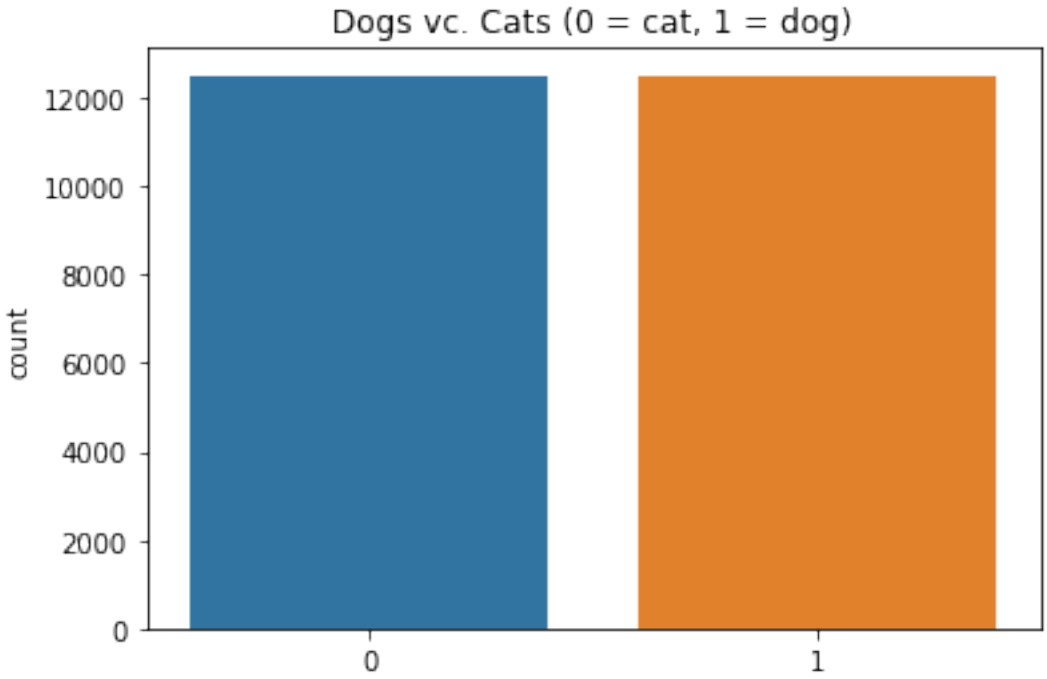
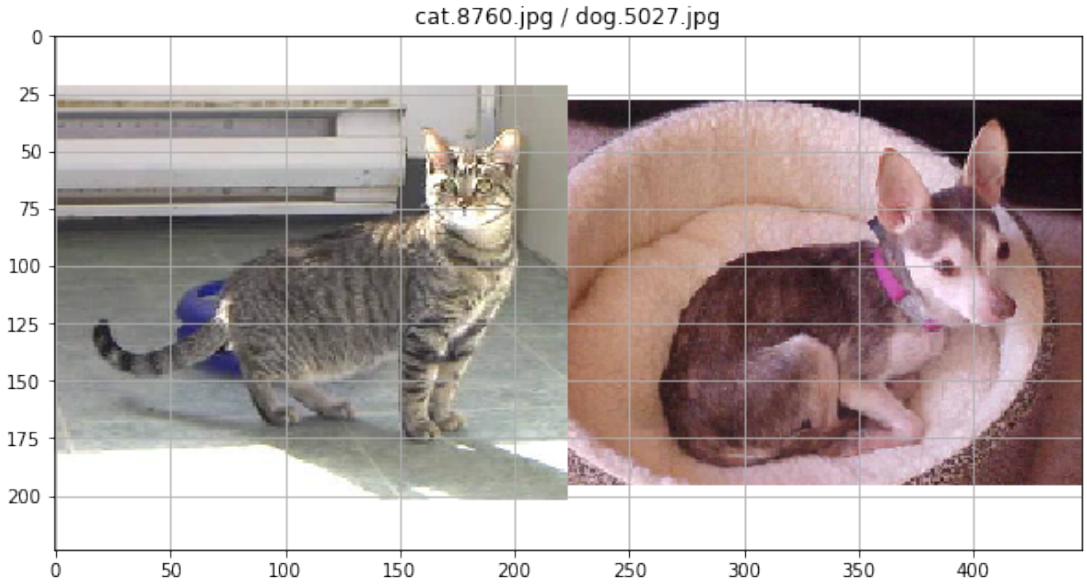
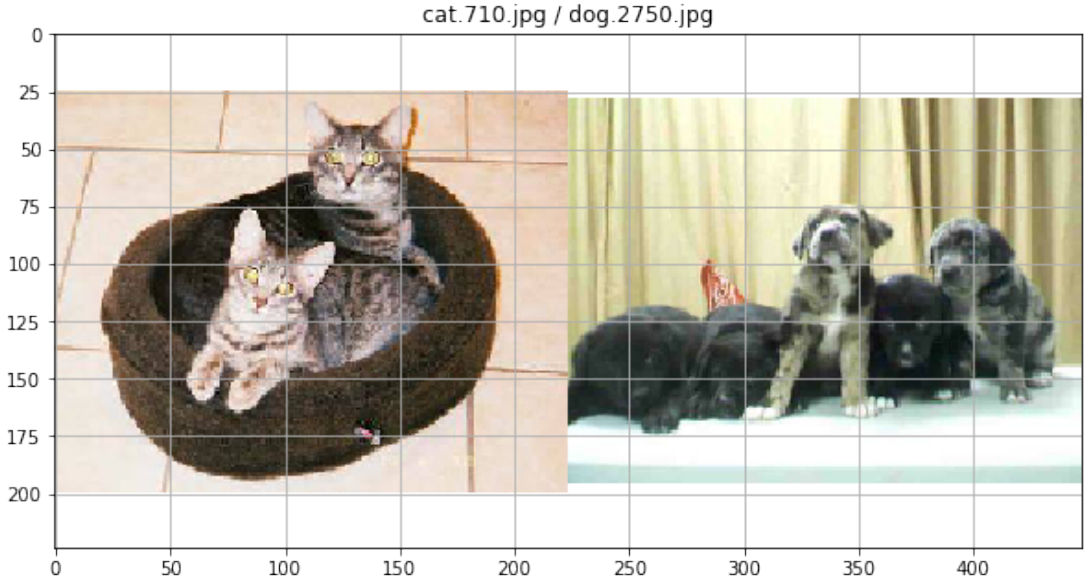


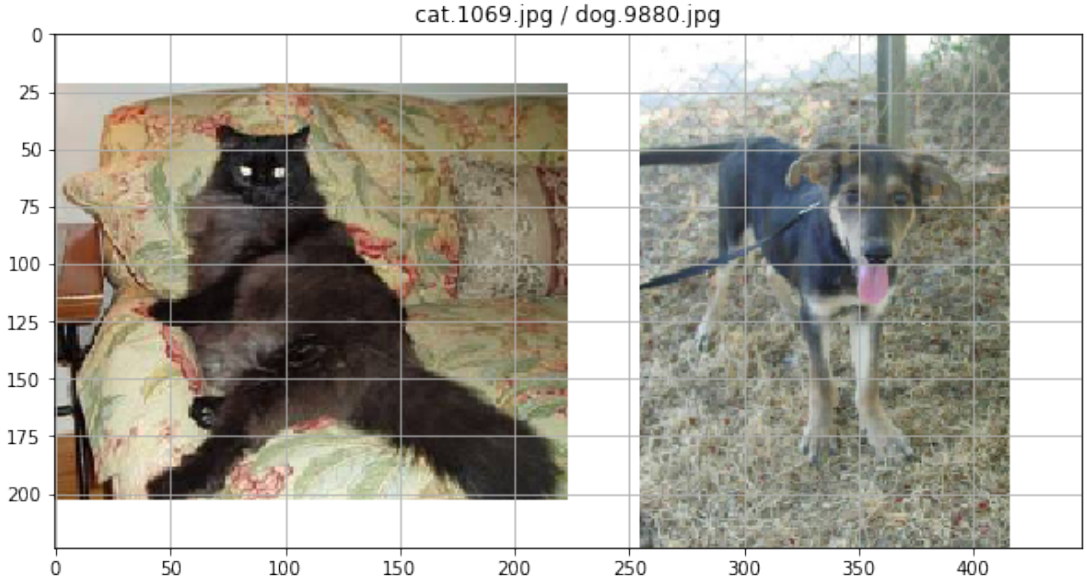
图 1：通过对文件分类的统计可以看到，训练数据集中共有 25,000 张图片，其中猫狗图片各 12,500 张，呈平均分布。

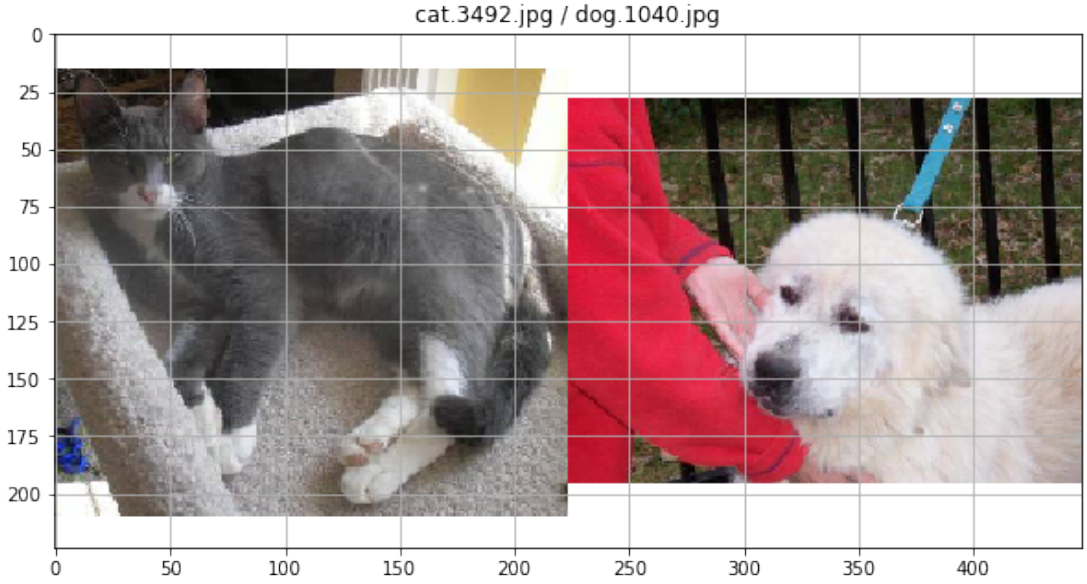
所有图片均来自日常拍摄的猫或狗的照片，其中包括一些像素质量较差或经过特殊处理的照片，还有一些照片中有不止一只动物（但猫和狗不会出现在同一张照片中），或有人等干扰性内容。训练集中的图片用文件名进行了标记。猫的图片文件名为“cat.<number>.jpg”，狗的图片文件名为“dog.<number>.jpg”。测试集中的图片以数字序列命名，没有分类标记。

可以看到数据集中的图片有着不同的分辨率和长宽比，在输入模型前需要进行预处理，将其调整为统一的尺寸。Keras 库内置了图片处理函数，可用于完成这一工作。具体预处理方法将根据模型需求确定。


cat.8760.jpg / dog.5027.jpg
0
25
50
75
100
125
150
175
200
0
50
100
150
200
250
300
350
400


cat.710.jpg / dog.2750.jpg
0
25
50
75
100
125
150
175
200
0
50
100
150
200
250
300
350
400


cat.1069.jpg / dog.9880.jpg
0
25
50
75
100
125
150
175
200
0
50
100
150
200
250
300
350
400


cat.3492.jpg / dog.1040.jpg
0
25
50
75
100
125
150
175
200
0
50
100
150
200
250
300
350
400

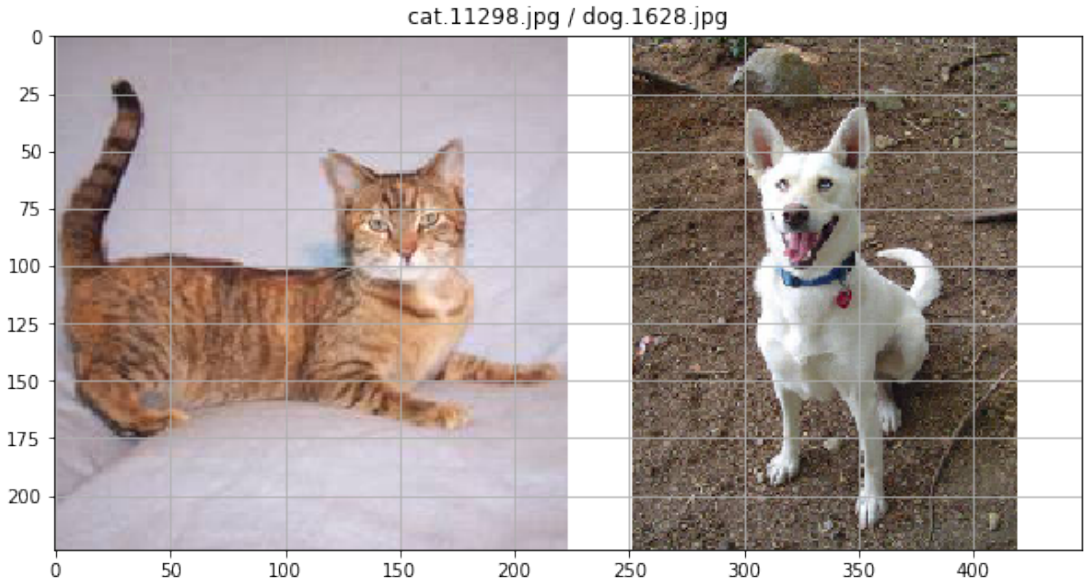


图 2：这些图片都来自日常拍摄的照片。照片质量参差不齐，尺寸各异。有些照片中有不止一只动物（但猫和狗不会出现在同一张照片中），有些还有干扰性内容——例如人类。由于模型要求输入统一的照片尺寸，因此在训练前需要对图片进行预处理，包括填充成相同比例以及缩放尺寸等。

2.2 算法和技术

2.2.1 神经网络

在机器学习和认知科学领域，人工神经网络（artificial neural network，ANN），简称神经网络（neural network，NN）或类神经网络，是一种模仿生物神经网络的结构和功能的数学模型或计算模型，用于对函数进行估计或近似。神经网络由大量的人工神经元联结进行计算。大多数情况下人工神经网络能在外界信息的基础上改变内部结构，是一种自适应系统。[来源请求]现代神经网络是一种非线性统计性数据建模工具。典型的神经网络具有以下三个部分：

结构。结构指定了网络中的变量和它们的拓扑关系。例如，神经网络中的变量可以是神经元连接的权重（weights）和神经元的激活值（activities of the neurons）。

激活函数。大部分神经网络模型具有一个短时间尺度的动力学规则，来定义神经元如何根据其他神经元的活动来改变自己的激励值。一般激励函数依赖于网络中的权重（即该网络的参数）。

学习规则。学习规则指定了网络中的权重如何随着时间推进而调整。这一般被看做是一种长时间尺度的动力学规则。一般情况下，学习规则依赖于神经元的激活值。它也可能依赖于监督者提供的目标值和当前权重的值。例如，用于手写识别的一个神经网络，有一组输入神经元。输入神经元会被输入图像的数据所激发。在激活值被加权并通过一个函数（由网络的设计者确定）后，这些神经元的激活值被传递到其他神经元。这个过程不断重复，直到输出神经元被激发。最后，输出神经元的激励值决定了识别出来的是哪个字母。

和其他机器学习方法一样，神经网络已经被用于解决各种各样的问题，例如机器视觉和语音识别。这些问题都是很难被传统基于规则的编程所解决的[2]。

2.2.2 深度学习

深度学习的基础是机器学习中的分散表示。分散表示假定观测值是由不同因子相互作用生成。在此基础上，深度学习进一步假定这一相互作用的过程可分为多个层次，代表对观测值的多层抽象。不同的层数和层的规模可用于不同程度的抽象。

深度学习运用了这分层次抽象的思想，更高层次的概念从低层次的概念学习得到。这一分层结构常常使用贪婪算法逐层构建而成，并从中选取有助于机器学习的更有效的特征。

深度神经网络是一种具备至少一个隐层的神经网络。与浅层神经网络类似，深度神经网络也能够为复杂非线性系统提供建模，但多出的层次为模型提供了更高的抽象层次，因而提高了模型的能力。深度神经网络通常都是前馈神经网络，但也有语言建模等方面的研究将其拓展到递归神经网络。卷积深度神经网络在计算机视觉领域得到了成功的应用。此后，卷积神经网络也作为听觉模型被使用在自动语音识别领域，较以往的方法获得了更优的结果[1]。

深度神经网络（Deep Neural Networks，DNN）是一种判别模型，可以使用反向传播算法进行训练。权重更新可以使用下式进行随机梯度下降法求解：

$$\Delta w_{ij}(t+1) = \Delta w_{ij}(t) + \eta \frac{\partial C}{\partial w_{ij}}$$

其中，η为学习率，∂C为损失函数。这一函数的选择与学习的类型（例如监

督学习、无监督学习、增强学习）以及激活函数相关。例如，为了在一个多分类问题上进行监督学习，通常的选择是使用 ReLU 作为激活函数，而使用交叉熵作为代价函数。

深度神经网络很容易产生过拟合现象，因为增加的抽象层使得模型能够对训练数据中较为罕见的依赖关系进行建模。对此，权重递减或者稀疏等方法可以利用在训练过程中以减小过拟合现象。另一种较晚用于深度神经网络训练的正规化方法是 Dropout，即在训练中随机丢弃一部分隐层单元来避免对较为罕见的依赖进行建模[1]。

2.2.3 卷积神经网络

卷积神经网络是一种前馈神经网络，它的人工神经元可以响应一部分覆盖范围内的周围单元，对于大型图像处理有出色表现。

卷积神经网路由一个或多个卷积层和顶端的全连通层（对应经典的神经网路）组成，同时也包括关联权重和池化层（pooling layer）。这一结构使得卷积神经网路能够利用输入数据的二维结构。与其他深度学习结构相比，卷积神经网路在图像和语音识别方面能够给出更好的结果。这一模型也可以使用反向传播算法进行训练。相比较其他深度、前馈神经网路，卷积神经网路需要考量的参数更少，使之成为一种颇具吸引力的深度学习结构[3]。

这里以一张照片为例，视觉化观察卷积神经网络的工作过程：

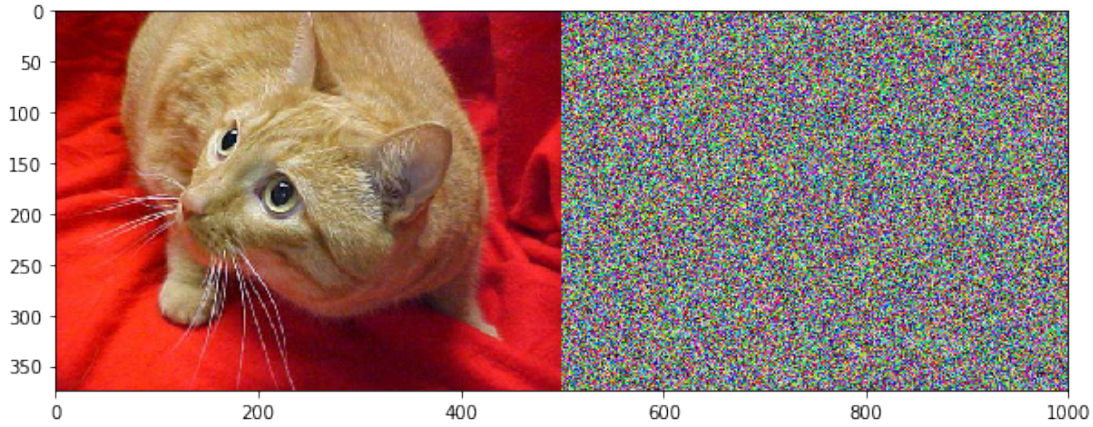


图 3：左边是原始照片；右边是经过一次卷积计算的结果，使用了 3 个 3×3 的卷积核。可以看到输出基本全是噪点，基本看不到有用的信息。部分原因是因为 3 个输出层堆叠在一起，人眼难以分辨。

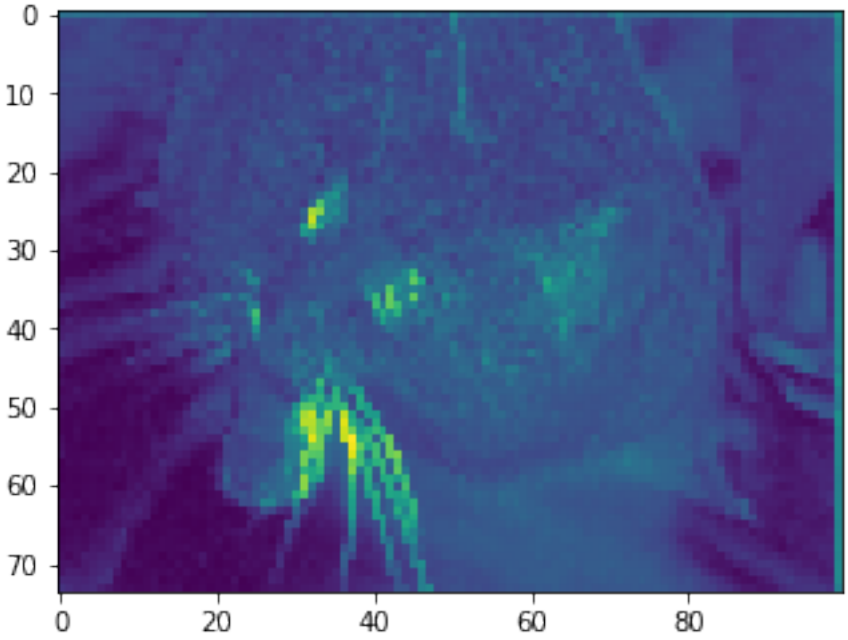


图 4：这是采用同样的卷积层，但卷积核减少为一个（只有一个输出层），并增加了一个激活层后的结果。可以看到神经网络已经识别出猫的主体特征，尤其是对眼睛和胡须格外敏感。

2.3 基准指标

本项目来源于 Kaggle 的 Dogs vs. Cats Redux: Kernels Edition 竞赛，该竞赛使用 LogLoss 作为评估标准。在公开的结果排名中，TOP10%的 LogLoss 值在 0.05629 以下。本项目的目标就是模型的 LogLoss 值低于 0.05629。

3 具体方法

3.1 选择模型

本项目使用 TensorFlow 和 Keras 库。TensorFlow 是一个用于数值计算的开源软件库。其灵活的架构可以在多种平台上展开计算。TensorFlow 由 Google 开发出来，用于机器学习和深度神经网络方面的研究[4]。Keras 是一个高层神经网络 API，支持灵活地使用模块——网络层、损失函数、优化器、初始化策略、激活函数等——自由组合配置，搭建各种功能模型[5]。

Keras 内置了一组常用模型，包括 Xception、VGG16、VGG19、ResNet50、InceptionV3、InceptionResNetV2、MobileNet 等。本项目选择 ResNet50、

Xception 和 InceptionV3 三个模型作为基准和对比模型，最后组合三个模型的特征训练一个最终模型。

3.2 数据预处理

本项目选择的三个模型都对输入数据有特定的要求,其共同点是都要求输入尺寸一致的图片；不同点是：

1. ResNet50 默认输入图片尺寸是 224×224；InceptionV3 和 Xception 默认输入图片尺寸是 299×299。
2. ResNet50 将图片 RGB 每个通道减去 ImageNet 相应通道的平均值，并将通道顺序由 RGB 调整为 BGR；InceptionV3 和 Xception 将图片像素输入从(0, 255)缩放到(-1, 1)区间。

针对以上特点，本项目采用如下数据预处理方式；

1. 在读取图片式，对长宽比例不符合 1:1 的图片进行填充，以白色底边将图片填充为 1:1 比例。
2. 读取图片时进行尺寸缩放,其中针对 ResNet50 的图片缩放至 224×224；针对 InceptionV3 和 Xception 的图片缩放至 299×299。
3. 对图片应用 Keras 对应模型内置的预处理函数。

3.3 迁移学习和 Fine-tune

本项目将使用迁移学习和 fine-tune 的思路，对于基础模型，将最后两层（通常是全连接层和输出层）替换为自己的模型，利用基础模型有效提取数据特征的能力，加入自己的全连接层和输出层，以更好地适应具体数据集的需要。这一做法尤其适合小数据集的训练。本项目对三个基础模型做了统一处理：在基础模型后加入一个保留率 0.5 的 Dropout 层和一个采用 Sigmoid 激活函数的输出层。

同时，基础模型采用了针对 ImageNet 数据集训练后的结果作为初始权重。这种方法可以充分利用 ImageNet 大数据集的训练结果，并减少本地训练的计算量。

最后，由于针对基础模型的训练是“冻结”的，没有必要每次都跑完整个模

型，所以先用基础模型输出特征向量，再使用自己的最后两层针对导出的特征向量进行多代训练。该方法可大大减少训练所需的计算资源和时间[6]。

3.4 实现

因图片处理任务较多，为避免内存不足的问题，在训练时将使用 Keras 提供的 ImageDataGenerator 类进行分批处理。原始的 ImageDataGenerator 没有提供填充图片的方法，因此自行实现了一个 MyImageDataGenerator 类，继承了 ImageDataGenerator 并改写了相应方法，作用是在缩放图片前增加填充图片功能，其他功能和使用方法不变。为实现 MyImageDataGenerator 类，同时实现了 MyDirectoryIterator 类（继承自 DirectoryIterator 类）。

对图片的预处理采用各模型提供的 preprocess input 函数,模型的 include top 参数设置为 False（不实用全连接层和输出层），其他保持默认。

从图 5，图 6，图 7，图 8 可以看到，针对基础模型导出的特征向量进行训练，三个模型的训练效率和结果都非常好，差异很小，最终的 LogLoss 都满足我们的目标，准确度也达到了 99%左右，已经很接近贝叶斯最优水平。我们还发现 ResNet50 和组合模型都出现了一定的过拟合嫌疑,InceptionV3 表现出更好的泛化能力，而 Xception 则在训练集和验证集上有同样优秀的表现。

图 9 是将四个模型的预测值提交到 Kaggle 的结果，可以看到四个模型都很优秀，而组合模型虽然有过拟合倾向，但预测结果仍然是最好的。

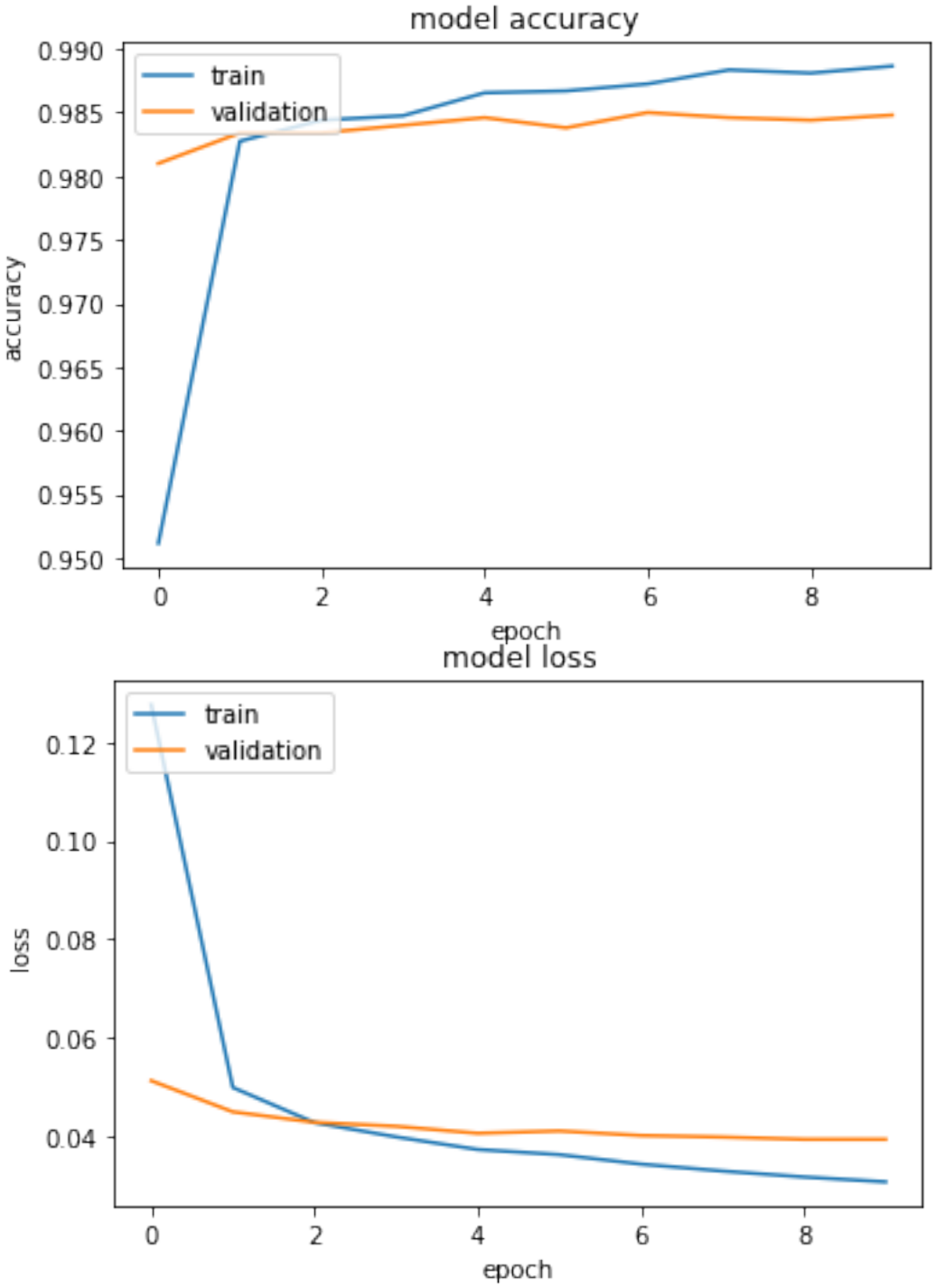


图 5：ResNet50 的 LogLoss 和 Accuracy 曲线，在第一代就达到非常好的结果，训练效率很高。图中可以看到训练集与验证集的曲线出现了一定分离，训练集的表现更好，针对训练集过拟合的倾向。

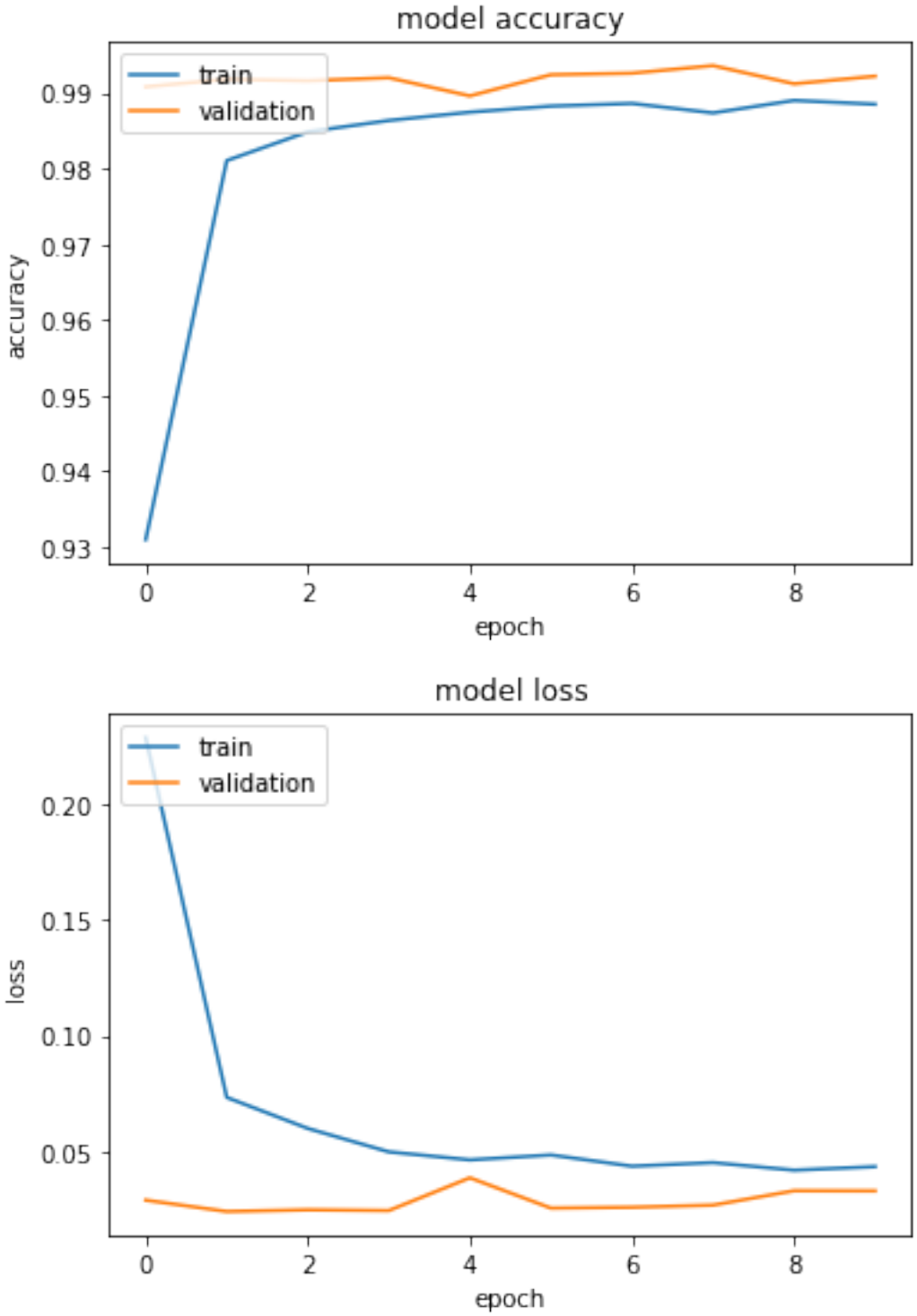


图 6：InceptionV3 的 LogLoss 和 Accuracy 曲线，同样表现出了很高的训练效果和结果。注意到训练集曲线和验证集曲线出现了与 ResNet50 相反的分离，验证集表现更好，在一定程度上表明该模型具有更好的泛化能力。

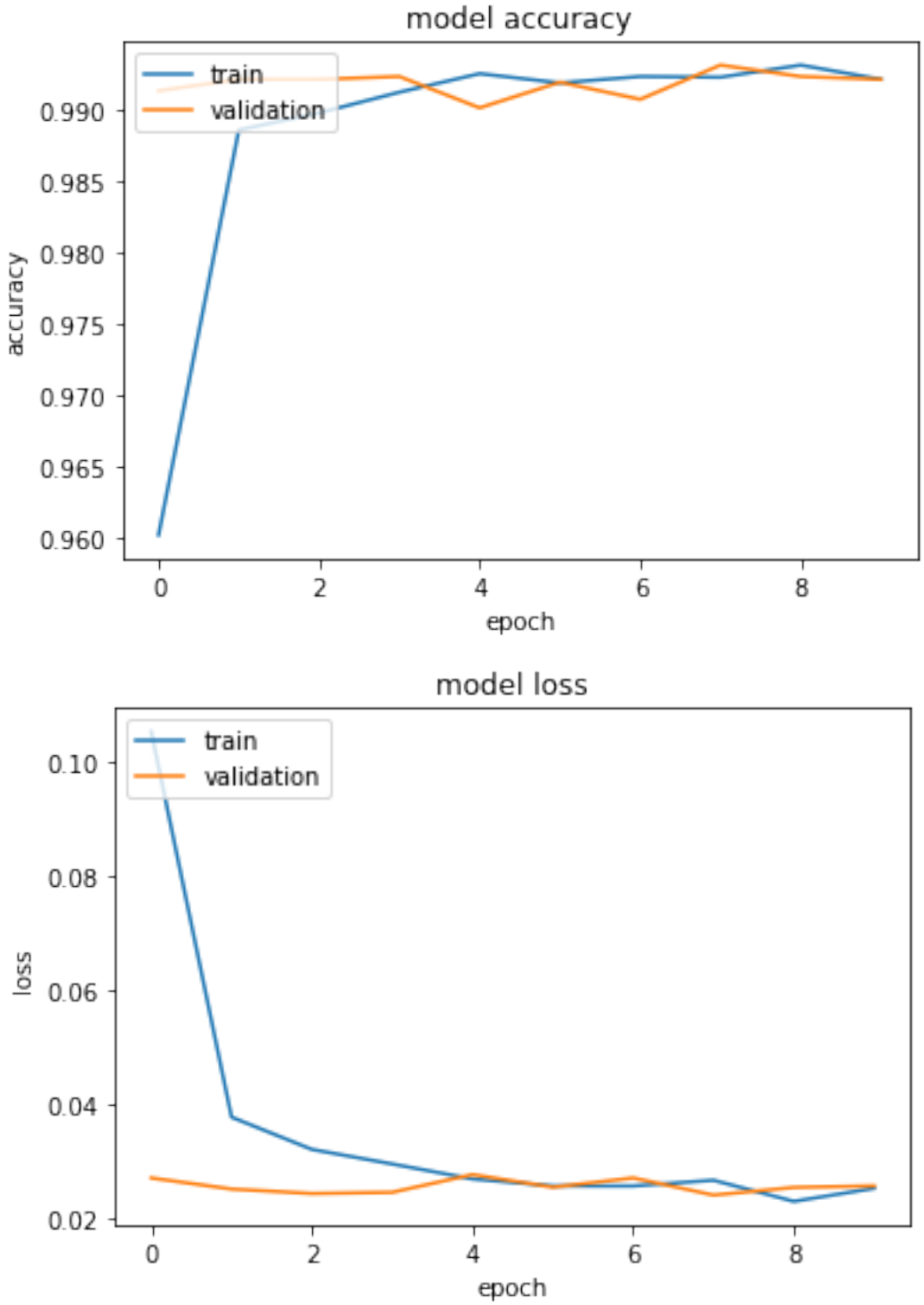


图 7：Xception 的 LogLoss 和 Accuracy 曲线，除了同样表现出很高的训练效率和结果外，训练集和验证集曲线几乎没有分离，显示出高度的一致性，结果令人振奋。

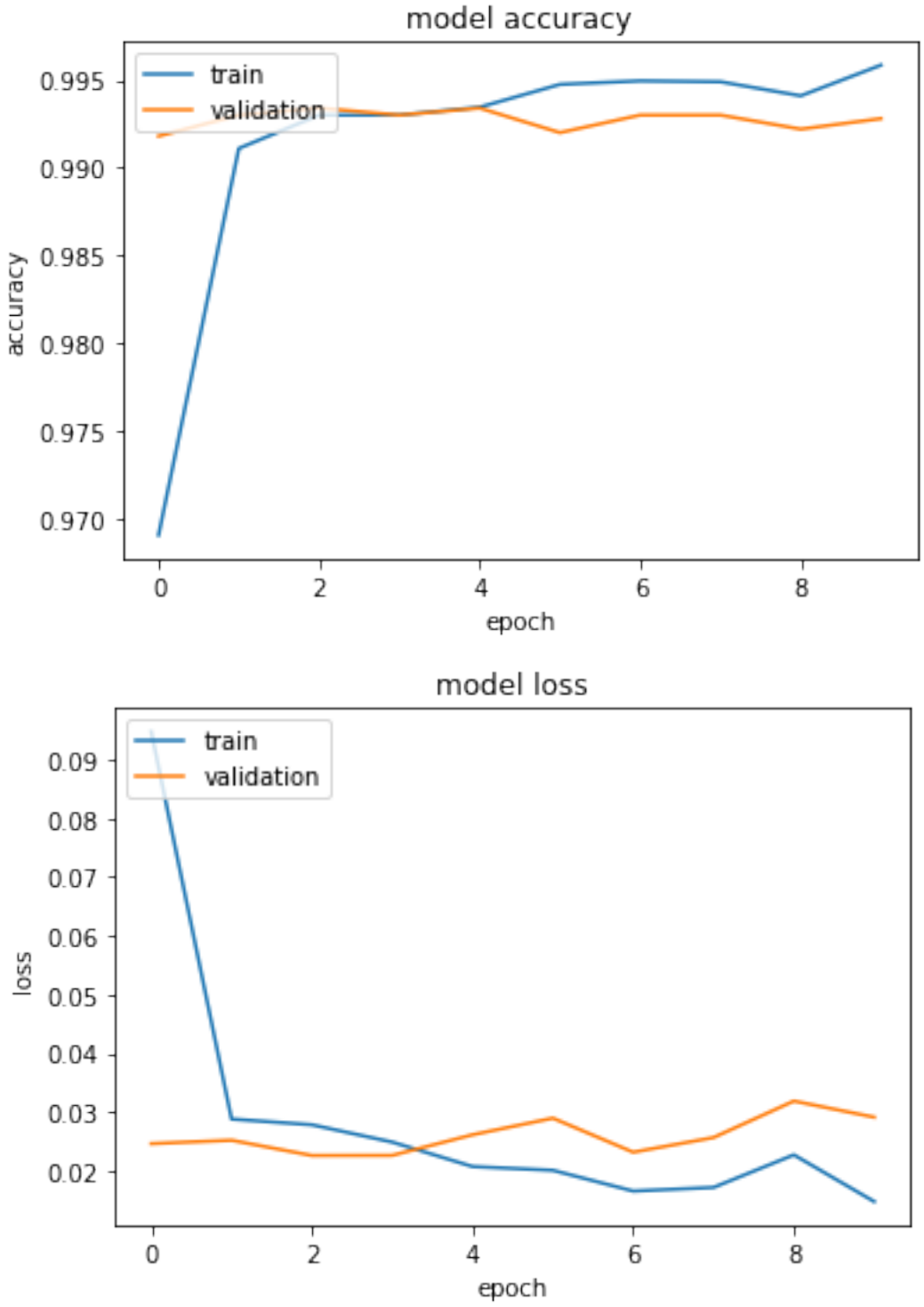


图 8：结合三个基础模型特征向量后的模型的 LogLoss 和 Accuracy 曲线。该曲线向 ResNet50 的曲线一样表现出过拟合的倾向，不过最终的结果还需要将预测值提交到 Kaggle 进行评估。

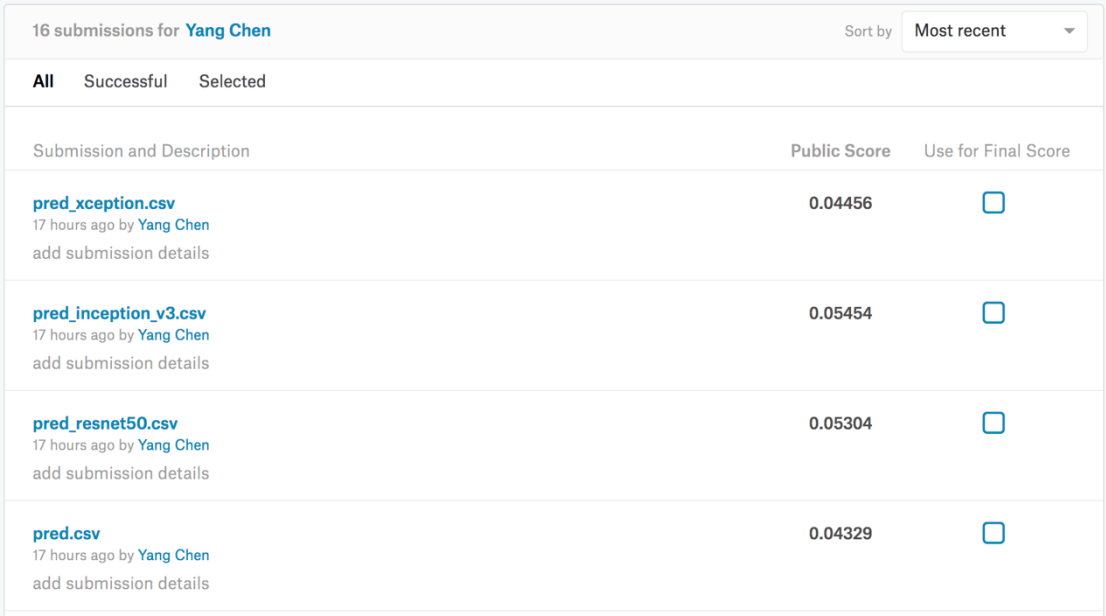


图 9：将四个模型的预测值提交到 Kaggle 后的评估结果，图中的 Public Score 即为 LogLoss 值。可以看到四个模型的表现同样优秀。组合模型结果最好，其次是 Xception，ResNet50 和 Inception 的表现更接近，总体上比组合模型和 Xception 略差。

4 结果

4.1 模型评价与验证

	val_logloss	val_accuracy	Test_logloss
ResNet50	0.0393	0.9848	0.05304
InceptionV3	0.0327	0.9922	0.05454
Xception	0.0255	0.9922	0.04456
组合模型	0.0291	0.9928	0.04329

对比四个模型，在验证集的 LogLoss 上 Xception 有比较明显的优势，在 Accuracy 上除了 ResNet50 外三个模型旗鼓相当。在最终的测试集，综合综合模型胜出，Xception 以微弱差距紧随其他，ResNet50 和 InceptionV3 的表现更加接近，整体上比前二者差一些。

4.2 结果分析

本项目选择的基础模型，ResNet50，InceptionV3 和 Xception 都是在图像

识别领域性能非常优秀的模型，可以说足以胜任本项目的需要。尽管如此，将三个模型组合使用，仍然可以进一步提高结果。

值得注意的是，在评估单模型表现时，可以看到三个模型在训练集和验证集上表现出不同的特点，如何在组合上对这些不同特点加以利用，例如降低过拟合风险，提高表现更优秀的基础模型的权重，则有待进一步研究。

5 结论

“猫狗大战”是 Kaggle 非常流行的竞赛之一，也是一个非常有趣的项目。在网络上，针对该竞赛的研究和讨论非常多。由于 CNN 模型具有很强的灵活性，网上能看到的解决方案和模型选择也各有特点，提供了丰富的参考和对比资料。最终通过利用 ImageNet 数据集的训练结果，以及选择擅长图像分类的基础模型和对组合模型的利用，本项目的尝试得到了不错的结果。

本项目后续还可以选择增加训练集的方式加以改进，具体来说就是对已有训练集中的图片进行一定程度的裁剪、旋转、水平镜像、改变颜色等处理，产生新的训练图片，这也是针对小数据集常用的一种操作。

图像分类是人类非常擅长的任务，准确度可以无限接近 100%，基本相当于贝叶斯最优水平，而 CNN 目前仍然无法达到这一水平。以本项目为例，图 10 是预测值中最接近 0.5 的 5 张图片，代表了模型最难确认的一组照片。这些照片的问题有主体占画面比例较小、动物没有露出正脸、动物是全黑的等，的确增加了识别难度，然而人类还是能轻而易举地辨认出这些图片内容。在这方面，CNN 离人类的平均水平尚有较大差距。

不过，这个例子恰好是人类最擅长的图像识别任务之一，而在某些领域的图像识别上，CNN 的表现并不输于人类。事实上，由于 CNN 的极限是贝叶斯最优水平，并非“人类水平”，所以即使 CNN 的表现最终超过人类，也是完全可能的。

通过这个项目，完整地进行了一次从问题分析、数据观察，到模型选择、调整，再到最终结果分析的过程，不仅提高了对 CNN 的理解和运用，也对如何思考和解决问题有了更深刻的理解，这就是本项目最大的收获。

5731.jpg

5864.jpg

6787.jpg

10310.jpg

10601.jpg

图 10：模型最难辨认的 5 张照片，第一张照片中的猫咪占整体画面比例较少，第二张的两只猫咪都没有露出正脸，而后三张照片中的猫和狗都是全黑的，一定程度上增加了粪便难度。然而即使如此，人类还是能轻而易举地分辨出这些照片。

6 参考资料

[1] 深度学习，维基百科

[2] 人工神经网络，维基百科

[3] 卷积神经网络，维基百科

[4] TensorFlow 中文社区

[5] Keras Documentation

[6] 猫狗大战，杨培文