

## 猫狗大战--开题报告

### 项目背景

相比较于几十年前，近年来的人工智能（AI）领域获得了长足的发展。人工智能再次的兴起，很大程度得益于图形处理单元（GPU）的运算能力以及例如 ImageNet 等公共数据集的实现。与此同时，算力和大数据也极大地突显出深度学习算法的独有魅力。通过大量标记示例和大量计算，深度学习能够获得更为准确的结果，并且很大程度上提高了 AI 算法的性能，例如图像识别，自然语言处理等。

阿里巴巴、Google 和微软等大公司正在使用 AI 解决各种各样的问题，他们通过高可用性的数据集训练机器学习模型，借助这些已有的模型和别的数据集训练出另一种高性能的模型，从而解决我们的问题。该项目采用 kaggle 上的一个竞赛题目，目的是利用 kaggle 提供的数据集训练一个模型从给定的图片中分辨是猫还是狗，这也属于计算机视觉领域的一个问题。

### 问题陈述

所选的数据集包含带标签的猫或狗的图像。由于每张图片至少包含一只动物，它属于“猫”类别或是“狗”类别，其本质是二分类问题。所以需要建立一个二分类器模型，给模型输入一张图片，模型就会预测该图片的类别。

为此，我将使用深度学习模型，并采用迁移学习的方式构建模型，模型的前半部分采用现有的卷积神经网络（如 VGG、ResNet、AlexNet 等）提取图像的特征，后半部分采用神经网络实现全连接层，从而将图像的特征映射到标签。

同时，我将在标记的数据集上进行训练和评估算法。为了使算法尽可能地具有高性能，我将调整算法体系结构、优化器（adam, adadelta 等）以及众多的参数，最终能够达到 kaggle Public Leaderboard 前 10%。

### 输入数据

输入数据来自于 kaggle 的“猫狗大战”竞赛项目，数据集包括：

- 25000 张训练图片（其中猫狗各 12500 张，文件名的前缀作为文件类别），压缩于 test.zip 文件中；
- 12500 张测试图片，无标签。

每个数据为 RGB 三通道的彩色图片，图片的质量、大小存在差异，需要根据模型的输入层将其 resize 成相同的尺寸。本项目将采用 `keras.preprocessing.image.ImageDataGenerator` 对图片进行加载与处理。

## 方案描述

在项目中将会使用深度学习算法，结合卷积神经网络（CNN）和全连接神经网络构建二分类模型。卷积神经网络在深度学习技术中极具代表性，其优点之一在于避免了对图片复杂的前期处理过程，可以直接输入原始图像。本项目模型的前半部分采用现有的卷积神经网络（如 VGG、ResNet、AlexNet 等）提取图像的特征，后半部分采用神经网络实现全连接层，从而将图像的特征映射到标签。

## 基准模型

本项目将采用 VGG、ResNet、AlexNet 等模型，通过比较各模型在测试集中的表现，选择其中性能最佳的模型。

由于该项目在 kaggle 的竞赛中共有 1314 只队伍参加比赛，而本项目的最低要求为 kaggle public leaderboard 前 10%，所以最终模型的得分不得低于第 131 名的得分（0.06127）。

## 评价指标

采用对数损失函数：

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))]$$

其中：

- n 为图片数量
- $h_{\theta}(x^{(i)})$  为模型预测是狗类别的概率
- $y_i$  为类别标签，1 代表为狗类别，0 代表为猫类别
- $\log()$  表示自然对数

损失函数的数值越小，代表模型的性能越好。本项目将使用该损失函数评估模型的性能。

## 设计大纲

本项目采用迁移学习的方式，借助 Keras 中已有的模型作为模型的前端，并采用全连接神经网络作为后端构建模型。

项目执行步骤：

1. 数据预处理
  - a. 加载 kaggle 提供的数据集
  - b. 根据训练集中的文件名，对每个数据设置标签，如 1 表示狗类别的图片，0 表示猫类别的图片
  - c. 对数据集中的每个数据进行 resize 操作，使其具有相同的尺寸

- d. 对训练数据进行图像处理，如缩放、增加噪点、偏移等，以此扩充训练集
- 2. 模型搭建
  - a. 加载 Keras 中已训练的模型（如 VGG、ResNet、AlexNet 等）
  - b. 去除模型中全连接层部分
  - c. 用前半部分 CNN 模型的输出作为输入，设置全连接层且输出为二分类
- 3. 模型训练&模型调参
  - a. 加载 CNN 模型的网络权重
  - b. 不改变 CNN 模型的权重，使用训练数据对全连接层进行训练
  - c. 尝试调整算法体系结构、优化器（adam, adadelata 等）以及众多的参数对模型进行训练
  - d. 采用不同的模型，重复 2.a 步骤
  - e. 选择性能最佳的模型
- 4. 模型评估
  - a. 使用对数损失函数对模型进行评估，并判断是否符合标准
- 5. 可视化
  - a. 可视化数据集
  - b. 可视化模型的准确率，损失函数等

## 参考文献

- [1] <https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition>
- [2] <https://www.kaggle.com/c/dogs-vs-cats/leaderboard>
- [3] <https://www.jiqizhixin.com/articles/Recent-Advances-in-Deep-Learning-An-Overview>
- [4] <http://blog.itpub.net/29829936/viewspace-2641958/>
- [5] Matthew D. Zeiler. ADADELTA: An Adaptive Learning Rate Method
- [6] Diederik P. Kingma, Jimmy. Ba.Adam: A Method for Stochastic Optimization
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun.Deep Residual Learning for Image Recognition
- [8] [https://github.com/nd009/capstone/blob/master/capstone\\_report\\_template.md](https://github.com/nd009/capstone/blob/master/capstone_report_template.md)