Machine Learning Engineer Nanodegree

Capstone Proposal

顾飞 优达学城
2019年03月20日

Proposal

- Transfer Learning
- Amazon Web Service
- Domain Background
- Problem Statement
- Datasets and Inputs
- Datasets Analysis
- Solution Statement
- Benchmark Model
- Evaluation Metrics
- Project Design

Transfer Learning

迁移学习是计算机视觉中一种备受欢迎的学习方法，通过迁移学习，我们可以在省时的基础上建立准确的模型。进行迁移学习时，你可以从解决不同问题时遇到的已知的模式开始学习，而不需要从头开始。这样你可以利用以前学到的知识，避免从头开始。迁移学习可视作沙特尔所说的“站在巨人的肩膀上”的深度学习版本。

在计算机视觉中，迁移学习通常通过使用预训练模型来实现。预训练模型指在大型基准数据集上进行问题解决训练的模型，而用于训练的问题与我们实际想要解决的问题类似。由于训练这类模型的计算量与计算成本低，更常见的应用是在已出版的文献中导入和使用这类模型。

从深度学习的角度来看，图像分类问题可以通过迁移学习来解决。实际上，图像分类中几项最新的研究成果都是基于迁移学习方案得出的。迁移学习，就是把在某个领域获取的知识迁移到对另一个领域的学习中。传统的机器学习，尤其是有监督学习，对数据的样本数量、数据分布的统一性、标签的完整性等都有着严苛的要求。而迁移学习解决的正是在机器学习任务中面对样本不足、标签不全等情况，如何借助外部其他分布的数据来有效地学习这一问题。

在图像识别问题中，是将训练好的模型通过简单调整来解决新的问题。从图像中提取特征，不一定需要算力强大的GPU，训练上百层的神经网络。

卷积神经网络中卷积层和池化层可以抽取图片的几何特征，比如浅层的卷积用于抽取出一些直线，角点等简单的抽象信息，深层的卷积层用于抽取人脸等复杂的抽象信息，最后的全连接层是对图片分类的处理。因此，我们可以使用网络的前N-1层提取特征。

例如，利用在 ImageNet 数据集上训练好的 ResNet50 模型来解决一个自定义的图像分类问题：保留训练好的 ResNet50 模型中卷积层的参数，只去掉最后一个全连接层，将新图像输入训练好的神经网络，利用前N-1层的输出作为图片的特征，将 ResNet50 模型作为图片特征提取器，提取得到的特征向量作为输入训练新的单层全连接网络来处理新的分类问题，或者将这些特征代入SVM，LR等其它机器学习模型进行训练和预测。

在数据量足够的情况下，迁移学习的效果往往不如完全重新训练，但是迁移学习所需要的训练时间和训练样本要远远小于训练完整的模型。

Amazon Web Service

由于此项目要求的计算量较大，也许你能够在本地 CPU 上训练。训练过程可能持续几个小时，取决于迭代次数、神经网络的大小和其他因素。

一种更快速的方法是在 GPU (图形处理器)上进行训练。虽然 CPU 可以比 GPU 计算更复杂的运算，但是吞吐量较小。因此在训练神经网络时，GPU 比 CPU 的速度要快很多。GPU 在同一时间会执行更多的计算，当我们处理大数据时这很重要。

你也可以自己购买 NVIDIA GPU，但是并非像将 GPU 插入机器那么简单。即使你知道如何安装 GPU，但是正确地安装 cuDNN 还是有一定的挑战。因此，我们不建议这么使用 GPU。

我们建议使用 Amazon EC2 实例，使用 Amazon p3.2xlarge 云服务器来完成该项目。

PS：如果你不想自己从头开始训练模型，重复造轮子。也可以使用预训练模型，提高训练准确度。

Domain Background

Dogs vs. Cats Redux: Kernels Edition

项目来源于 kaggle 在 2013 年组织的一场比赛，它使用25000张（约543M）猫狗图片作为训练集，12500张(约271M)图片作为测试集，数据都是分辨率400x400左右的小图片，目标是识别测试集中的图片是猫还是狗。赛题网址：https://www.kaggle.com/c/dogs-vs-cats。

Problem Statement

深度学习中最突出的问题之一是图像分类。图像分类的目的是根据潜在的类别对特定的图像进行分类。图像分类的一个经典示例是在一组图像中识别猫和狗。

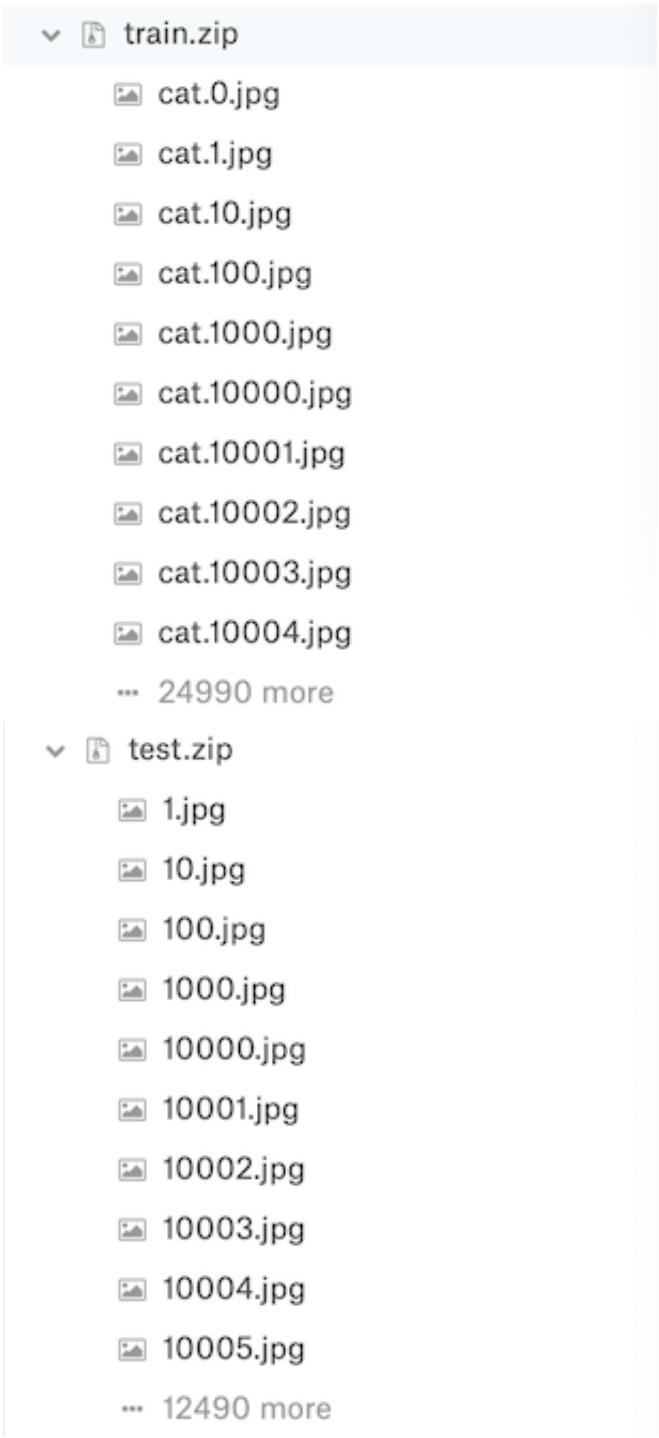
本文将介绍如何在图像分类问题中实施迁移学习解决方案。主要是使用"监督学习"实现一个图像分类器，来识别一张图片是猫还是狗。

对于图像识别，在数据量足量大的情况下，一般使用深度学习中的卷积神经网络（Convolutional Neural Networks, CNN），而本文将从迁移学习的角度，看看如何应用现有的深度学习模型（ResNet50、InceptionV3 和 Xception），从图片中提取特征，供分类器使用。使用此方法，即无需大量学习和训练模型的时间成本，又能解决图片识别相关的大多数问题。

本项目需要对测试样本进行分类，然后基于 CNN 的模型进行训练，并将结果上传至 kaggle 进行评分。

Datasets and Inputs

此数据集可以从 kaggle 上下载。Dogs vs. Cats Redux: Kernels Edition



下载 kaggle 猫狗数据集解压后分为 3 个文件 train.zip、 test.zip 和 sample_submission.csv。

train 训练集包含了 25000 张猫狗的图片，猫狗各一半，每张图片包含图片本身和图片名。命名规则根据 “type.num.jpg” 方式命名。

test 测试集包含了 12500 张猫狗的图片，没有标定是猫还是狗，每张图片命名规则根据 “num.jpg”，需要注意的是测试集编号从 1 开始，而训练集的编号从 0 开始。

sample_submission.csv 需要将最终测试集的测试结果写入.csv 文件中，上传至 kaggle 进行打分。

Datasets Analysis

在做计算机视觉的任务时，第一步很重要的事情是看看要做的是什么样的数据集，是非常干净的？还是存在各种遮挡的？猫和狗是大是小？图片清晰度一般怎么样？会不会数据集中有标注错误的数据，比例是多少？

- train 训练集包含了 25000 张猫狗的图片，平均宽=404px，平均高=360px，最小的宽=42px，最大宽=1050px，最小高=32px，最大高=768px；
- test 测试集包含了 12500 张猫狗的图片，平均宽=404px，平均高=359px，最小的宽=37px，最大宽=500px，最小高=44px，最大高=500px；
- 还可以通过opencv，找出模糊图片，原理就是使用了cv2.Laplacian()这个方法，代码如下。图片越模糊，imageVar的值越小，图像越模糊。 参考代码：来源于：https://blog.csdn.net/u014642834/article/details/78532798
- 为了提高模型的识别精确度，有必要去除异常图片，如分辨率低于(100 * 100) 的图片资源

Solution Statement

为了提高模型的表现，本项目决定使用预训练网络，最终选择了ResNet50, Xception, Inception V3 这三个模型，由于在笔记本上跑的，三个模型导出的时间耗了一天时间，时常有中途下载失败的情况。 这三个模型都是在 ImageNet 上面预训练过的，由此我们实际的预测训练会带来极高的初始精度。我们可以将多个不同的网络输出的特征向量先保存下来，后续即使是在普通笔记本上也能轻松训练。

有了三个特征向量后，我们需要将这三条向量进行合并成一条特征向量。

为了尽量利用我们有限的训练数据，我们将通过一系列随机变换堆数据进行提升，这样我们的模型将看不到任何两张完全相同的图片，这有利于我们抑制过拟合，使得模型的泛化能力更好。 在Keras中，这个步骤可以通过keras.preprocessing.image.ImageGenerator来实现，这个类使你可以：在训练过程中，设置要实行的随机变换，通过.flow或.flow_from_directory(directory)方法实例化一个针对图像batch的生成器，这些生成器可以被用作keras模型相关方法的输入，如fit_generator、evaluate_generator、predict_generator。

Benchmark Model

目前 Leaderboard 上展示了 1314 支队伍的成绩，排名第一的 score 是 0.03302，Top2% 的成绩是 0.04357。本项目的最低要求是 kaggle Public Leaderboard 前 10%，即 0.06149。

关于模型选型有以下三个方向：

- 1、普通的 CNN方法直接训练一个"基准模型"，依次建立卷积层、全连接层、sigmoid输出分类、通过keras.preprocessing.image.ImageGenerator 进行一系列随机变换堆数据进行提升，抑制过拟合。 有关 CNN 的知识，推荐：Understanding of Convolutional Neural Network

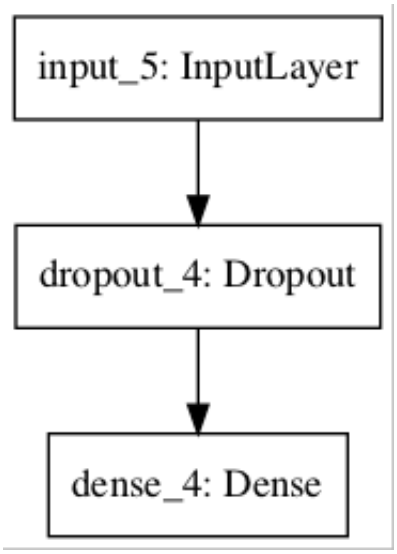
```
model.add(Dropout(0.5))
model.add(Convolution2D(4, 5, 5,input_shape=(224, 224,3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
model.compile(loss='binary_crossentropy',
              optimizer='rmsprop',
              metrics=['accuracy'])
```

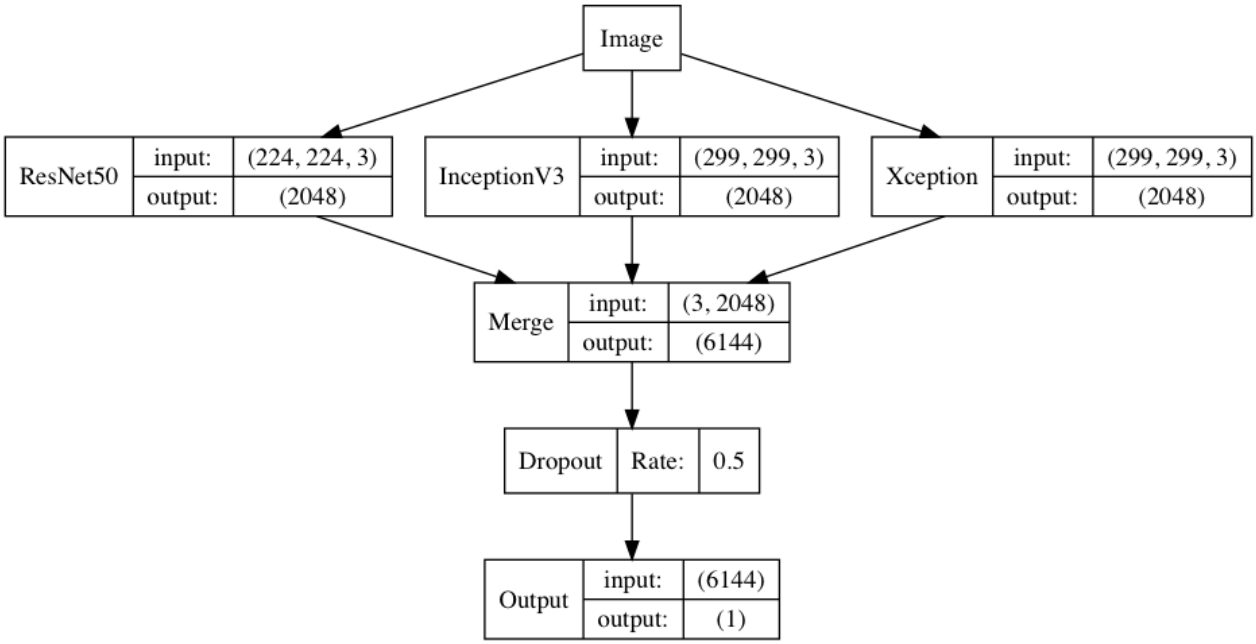
- 2、站在巨人的肩膀，使用预训练好的 ResNet50 方法进行训练，融合ResNet50, Xception, InceptionV3 三大模型，训练速度快，准确率高。 推荐 ResNet 相关知识的好文章：
  - Understanding and Coding a ResNet in Keras
  - ImageNet: VGGNet, ResNet, Inception, and Xception with Keras

```
inputs = Input(X_train.shape[1:])
x = Dropout(0.25)(inputs)
x = Dense(1, activation='sigmoid')(x)
model = Model(inputs, x)

model.compile(optimizer='adadelta',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```



图一



图二

- 3、使用TensorFlow迁移学习，可选 VGG16 模型 有关VGG16的知识，推荐：VGG16 –

Convolutional Network for Classification and Detection

Evaluation Metrics

对数损失（Log loss）亦被称为逻辑回归损失（Logistic regression loss）或交叉熵损失（Cross-entropy loss）。 交叉熵是常用的评价方式之一，它实际上刻画的是两个概率分布之间的距离，是分类问题中使用广泛的一种损失函数。 本文实际上是二分类问题， 因此可以采用 logloss 损失函数作为评价指标， 计算公式如下：

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

其中：

- n 是测试集中图片数量
- $\hat{y}$ 是图片预测为狗的概率
- y_i如果图像是狗，则为1，如果是猫，则为0
- $\log()$ 是自然（ 基数^e）对数

采用交叉熵作为损失函数可以有效的解决梯度消失和梯度爆炸的问题。 交叉熵损失越小，代表模型的性能越好。上述评估指标可用于评估该项目的解决方案以及基准模型。

Project Design

- 第一步先进行数据集分析，异常数据清洗
- 第二步准备数据集，数据集归类
- 第三步模型特征选择，在ImageNet 的预训练模型基础上，融合ResNet50, Xception, InceptionV3三大模型，把三个模型合并在一起，每个图片就有2048 * 3个权重值
- 我们基于这些权重值建立一个全连接，训练深度神经网络的时候，总是会遇到两大缺点： （1）容易过拟合 （2）费时 Dropout可以比较有效的缓解过拟合的发生，在一定程度上达到正则化的效果
- 接下来进行模型训练，我们选取validation_split=0.2，表示总样本的20%用来进行验证
- 优化器选择，optimizer='adadelta'，Adadelta是对Adagrad的扩展，最初方案依然是对学习率进行自适应约束，但是进行了计算上的简化。特点是，训练初中期，加速效果不错，很快；训练后期，反复在局部最小值附近抖动
- 最后进行预测数据集