

# 机器学习纳米学位

猫狗大战 唐玉山

2018 年 08 月 25 日

## 1. 问题的定义

(大概 1-2 页)

### 1.1. 项目概述

在这个部分，你需要用浅显简洁的语句描述这个项目的一个总体的概念。有几个问题是需要考虑的：

需要解决的问题涉及哪个领域？做这个项目的出发点？有哪些相关的数据集或输入数据？

问题的背景信息能够让完全没接触过这个问题的人充分了解这个问题吗？

项目基于深度学习和图像识别技术，目标是通过神经网络算法训练出一个模型，使用训练的模型对图片进行识别，给出图片是狗的概率。

该项目是 Kaggle 的机器学习竞赛项目《Dogs vs. Cats Redux: Kernels Edition》，他是基于 2013 年的《Dogs vs. Cats》项目。当时深度学习还没有得到充分应用，但是现在就连种黄瓜的农民也开始使用神经网络技术提高他们的收入。现在的深度学习技术进步使得图像识别变得简便，但是如何使得图像的识别准确度得到进一步提高却是一个很大的挑战。

近年来出现了很多针对图像识别的算法，其中神经网络算法表现尤为耀眼，该算法的效果优于模式识别、支持向量机等传统

项目中的深度学习使用的是卷积神经网络(Convolutional Neural Network, CNN) 是一种前馈神经网络，他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的表现。

项目中输入的数据就是猫或者狗的图片，输出的是图片是狗的概率。模型使用的训练数据集是 Kaggle 竞赛项目《Dogs vs. Cats Redux: Kernels Edition》

的数据集，该数据集包含了 25000 张狗和猫的训练图片，12500 张测试图片。通过对训练图片进行训练最终得出在测试集上面测试表现较好的模型。

## 1.2. 问题陈述

在这个部分，你需要清楚地为你将要解决的问题下定义，这应该包括你解决问题将要使用的策略（任务的大纲）。你同时要详尽地讨论你期望的结果是怎样的。有几个问题是需要考虑的：

你是否清楚地定义了这个问题。站在读者的角度，他们能否明白你将要解决的问题是什么。

你是否详尽地阐述了你将会如何解决这个问题？

你期望什么样的结果，读者能明白你期望的这个结果吗？

项目解决的问题是如何使用深度学习识别一张图片中的是猫还是狗，给出是狗的概率。

解决方案是：首先将数据集中的猫和狗的图像分开，利用图像处理技术进行预处理。训练集切分为训练集，验证集。对图像进行缩放，归一化处理。使用神经网络技术在训练数据集上训练 InceptionV3 模型。在验证数据集上对训练模型进行验证，对模型进行调参。在测试数据集上使用训练好的模型进行测试。验证最终模型，识别图片中的狗和猫，给出图片是狗的概率。

## 1.3. 评价指标

在这里，你需要说明你将要用于评价自己的模型和结果的指标和计算方法。它们需要契合你所选问题的特点及其所在的领域，同时，你要保证他们的合理性。需要考虑的问题：

你是否清晰地定义了你所使用的指标和计算方法？

你是否论述了这些指标和计算方法的合理性？

使用 kaggle 官方的评估标准： $LogLoss$ ，与准确率作为评估指标对比， $LogLoss$  要更有说服力。因为假设有两个模型，它们准确率一样，但是一个模型对一张猫的图片做出 99%是狗的判断，另外一个模型对这张猫的图片做出 60%的判断，这样第一个模型的  $LogLoss$  会很大，第二个模型的  $LogLoss$  要小很多，这样可以根据  $LogLoss$  来判断模型好坏。

$LogLoss$ 计算公式：

$$L(Y, P(Y|X)) = -\log P(Y|X) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} \log(p_{ij})$$

变量含义：

Y：输出变量

X：输入变量

L：损失函数

N：输入样本数

M：可能的类别数

$Y_{ij}$ ：是一个二值指标，标识类别  $j$  是否是输入实例  $x_j$  的真实类别

$P_{ij}$ ：模型或分离器预测输入实例  $x_j$  属于类别  $j$  的概率

对于当前项目，只有两类可将公式化简：

$$-\frac{1}{N} \sum_{i=1}^N (y_i \log p_i + (1 - y_i) \log (1 - p_i))$$

变量含义：

$Y_i$ ：输入实例  $x_i$  的真实类别，

$P_i$ ：为预测输入实例  $x_i$  属于类别 1 的概率

对所有样本的对数损失表示对每个样本的对数损失的平均值，完美的分离器的

对数损失为 0。

## 2. 分析

(大概 2-4 页)

### 2.1. 数据的探索

在这一部分，你需要探索你将要使用的数据。数据可以是若干个数据集，或者输入数据/文件，甚至可以是一个设定环境。你需要详尽地描述数据的类型。如果可以的话，你需要展示数据的一些统计量和基本信息（例如输入的特征（**features**），输入里与定义相关的特性，或者环境的描述）。你还要说明数据中的任何需要被关注的异常或有趣的性质（例如需要做变换的特征，离群值等等）。你需要考虑：

如果你使用了数据集，你要详尽地讨论了你所使用数据集的某些特征，并且为读者呈现一个直观的样本

如果你使用了数据集，你要计算并描述了它们的统计量，并对其中与你问题相关的地方进行讨论

如果你没有使用数据集，你需要对你所使用的输入空间（**input space**）或输入数据进行讨论？

数据集或输入中存在的异常，缺陷或其他特性是否得到了处理？（例如分类变数，缺失数据，离群值等）

**Kaggle** 数据集中包含训练数据集 **Train** 文件夹和测试数据集 **Test** 文件夹，

**Train** 中的图片可以根据文件名称区分出是猫的图片还是狗的图片。

**Train** 文件夹包含了 25000 张狗和猫的图片，图片命名方式是：

猫的图片：**cat.编号.jpg**

狗的图片：**dog.编号.jpg**

**Test** 文件夹包含了 12500 张图片，图片命名方式是：

**编号.jpg**

图片数据的尺寸不一，需要进行缩放处理。例如：在 **kaggle** 数据集中的 **train**

文件夹中的最小图片 **cat.4821.jpg(60\*39)**，最大图片是

**cat\_835.jpg(1023\*768)**。

数据集中的猫狗图片除了大小和分辨率有差别外，由于是来自真实世界的照

片，猫和狗的形态多种多样，拍摄照片的场景和各不相同。这些干扰因素，增

加了识别难度。

## 2.2. 探索性可视化

在这一部分，你需要对数据的特征或特性进行概括性或提取性的可视化。这个可视化的过程应该要适应你所使用的数据。就你为何使用这个形式的可视化，以及这个可视化过程为什么是有意义的，进行一定的讨论。你需要考虑的问题：

你是否对数据中与问题有关的特性进行了可视化？

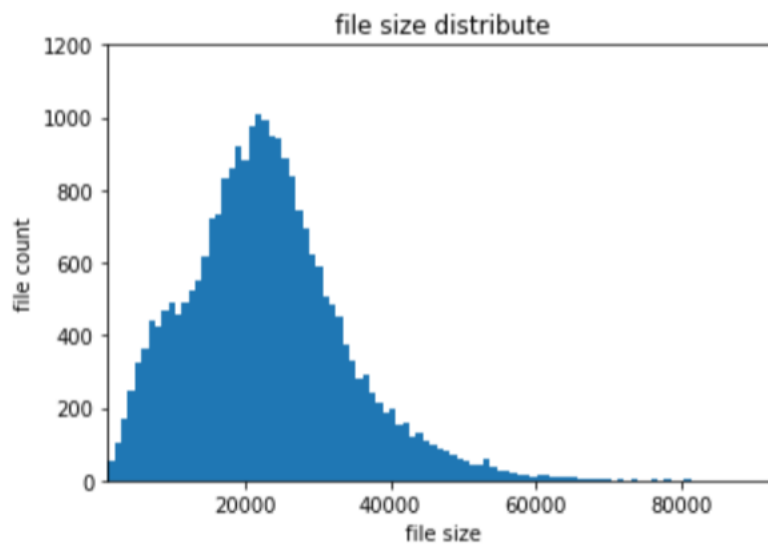
你对可视化结果进行详尽的分析和讨论了吗？

绘图的坐标轴，标题，基准面是不是清晰定义了？

对文件的大小进行了探索。

训练数据集的文件大小趋势图：

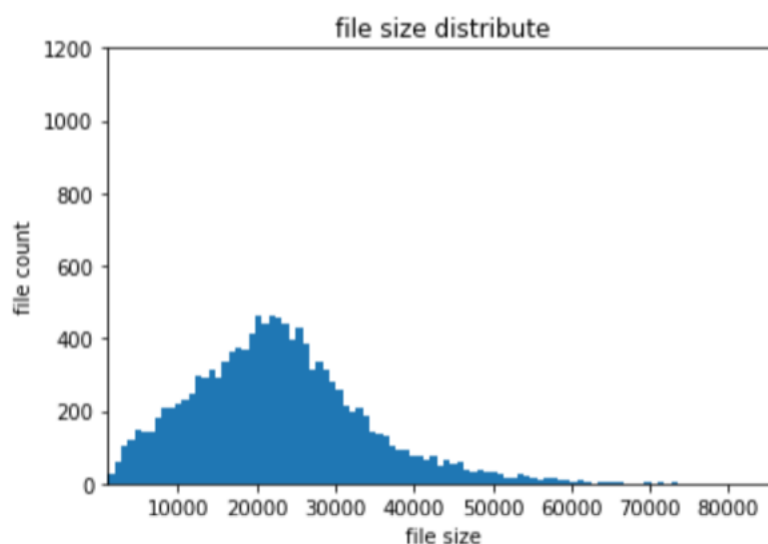
```
train 中文件的的最小值: 1106  
train 中文件的的最大值: 93323  
train 中文件的的中位数: 22165.5  
train 中文件的众数: ModeResult(mode=array([20918]), count=array([6]))
```



从图中可以看到大多数训练数据图片大小在 0-40000 字节之间

测试集中的文件大小分布图:

```
train 中文件的的最小值: 1131  
train 中文件的的最大值: 86322  
train 中文件的的中位数: 22061.0  
train 中文件的众数: ModeResult(mode=array([19960]), count=array([6]))
```



从图中可以看到大多数测试数据图片大小在 0-40000 字节之间

## 2.3. 算法和技术

在这一部分，你需要讨论你解决问题时用到的算法和技术。你需要根据问题的特性和所属领域来论述使用这些方法的合理性。你需要考虑：

你所使用的算法，包括用到的变量/参数都清晰地说明了吗？

你是否已经详尽地描述并讨论了使用这些技术的合理性？

你是否清晰地描述了这些算法和技术具体会如何处理这些数据？

### 图像处理

图像处理是指对图像进行分析、加工、处理，使其满足视觉、心里或其他要求的技术。图像处理是信号处理在图像领域上的一个应用。目前大多数的图像是以数字形式存储，因而图像处理很多情况下指图像处理。此外，基于光学理论的处理方法依然占有重要的地位。

图像处理是信号处理的子类，另外与计算机科学、人工智能等领域也有密切关系。

传统的一维信号处理的方法和概念任然可以直接应用在图像处理上，比如降噪、量化等。然而，图像属于二维信号，和一维信号相比，它有其特殊的一面，处理的方式和角度也有所不同。

由于数据集中的图像大小分辨率等存在差异，所以我们使用图像处理技术对训练数据中的图像进行预处理，例如：进行归一化操作。让图像数据方便进行进一步分析处理。

### 深度学习（deep learning）

深度学习是机器学习的分支，是一种试图使用包含复杂结构或由多重非线性变换构成的多个处理层对数据进行高层抽象的算法。深度学习是机器学习中一种对数据进行表征学习的算法。观测值（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，或者更抽象的表示成一系列边、特定形状的区域等。而使用某些特定的表示方法更容易从实例中学习任务（例如，人脸识别

或面部表情识别)。深度学习的好处是用非监督学习或半监督学习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。

表征学习的目标是寻求更好的表示方法并创建更好的模型来从大规模的未标记数据中学习这些表示方法。表示方法来自神经科学，并松散地创建在类似神经系统中的信息处理和对通信模式的理解上，如：神经编码，试图定义拉动神经元反应之间的关系及大脑中的神经元的电活动之间的关系。

至今已有的深度学习框架有：深度神经网络、卷积神经网络、深度置信网络、递归神经网络。已经被应用在计算机视觉、语音识别、自然语言处理、音频识别、生物信息学等领域，并取得了极好的效果。

### 卷积神经网络 (Convolutional Neural Network, CNN)

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。

卷积神经网络由一个或多个卷积层和顶端的全连接层（对应经典的神经网络）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络能够利用输入数据的二维结构。与其他深度学习结构相比，卷积神经网络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、前馈神经网络，卷积神经网络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。

这次项目中使用的卷积神经网络，建立一个识别猫和狗的神经网络模型

### 卷积层 (Convolutional layer)

卷积神经网络中每层卷积层由若干卷积单元组成，每个卷积单元的参数都是通过反向传播算法最佳化得到的。卷积运算的目的是提取输入的不同特征，第一

层卷积层可能只能提取一些低级的特征如边缘、线条和角等层级，更多层的网络能从低级特征中迭代提取更复杂的特征

### 反向传播 (Backpropagation BP)

反向传播是误差反向传播的简称，是一种与最优化方法（如：梯度下降算法）结合使用的，用来训练人工神经网络的常见方法。该方法对网络中所有权重计算损失函数的梯度。这个梯度会反馈给最优化方法，用来更新权值以最小化损失函数。

反向传播要求有对每个输入值想得到的已知输出，来计算损失函数梯度。它被认为是一种监督式学习方法，虽然它也用在一些无监督网络（如：自动编码器）中。

反向传播算法（BP 算法）主要由两个阶段组成：激励传播，权重更新。

三层网络算法（一个隐藏层）：

初始化网络权值（通常是小的随机值）

Do

forEach 训练样本 ex

prediction = neural-net-output(network, ex) //正向传递

Actual = teacher-output(ex)

计算输出单元的误差(prediction-actual)

计算 wh 对于所有隐藏层到输出层的权值 //反向传递

计算 wi 对于所有输入层到隐藏层的权值 //继续反向传递

更新网络权值 //输入层不会被误差估计改变

Until 所有样本正确分类或满足其他停止标准

Return 该网络



## 2.4. 基准模型

在这一部分，你需要提供一个可以用于衡量解决方案性能的基准结果/阈值。这个基准模型要能够和你的解决方案的性能进行比较。你也应该讨论你为什么使用这个基准模型。一些需要考虑的问题：

你是否提供了作为基准的结果或数值，它们能够衡量模型的性能吗？

该基准是如何得到的（是靠数据还是假设）？

本项目的最低要求是 kaggle Public Leaderboard 前 10%，也就是在 Public Leaderboard 上的 logloss 要低于 0.06127,位置是第 131 名。

## 3. 方法

(大概 3-5 页)

### 3.1. 数据预处理

在这一部分，你需要清晰记录你所有必要的的数据预处理步骤。在前一个部分所描述的数据的异常或特性在这一部分需要被更正和处理。需要考虑的问题有：

如果你选择的算法需要进行特征选取或特征变换，你对此进行记录和描述了吗？

数据的探索这一部分中提及的异常和特性是否被更正了，对此进行记录和描述了吗？

如果你认为不需要进行预处理，你解释个中原因了吗？

1.需要对训练数据进行分割：

Train 中的数据没有对猫狗进行区分，需要将猫和狗的数据分别放在不同的文件夹中。

2.图像预处理：

因为本项目中使用的是 InceptionV3 进行模型训练，需要对图片进行处理格式化为 299 x 299 的大小

### 3.2. 执行过程

在这一部分，你需要描述你所建立的模型在给定数据上执行过程。模型的执行过程，以及过程中遇到的困难的描述应该清晰明了地记录和描述。需要考虑的问题：

你所用到的算法和技术执行的方式是否清晰记录了？

在运用上面所提及的技术及指标的执行过程中是否遇到了困难，是否需要作出改动来得到想要的结果？

是否有需要记录解释的代码片段(例如复杂的函数)？

训练 InceptionV3 模型

训练步骤如下：

- 1.构建不带分类器的预训练模型
- 2.添加全局平均池化层
- 3.全连接层，可选，如果精度够用则可以不加
4. 添加一个分类器，使用 1 个神经元，sigmoid 激活函数
5. 构建我们需要训练的完整模型
- 6.首先只训练顶部的几层（随机初始化的层），锁住所有 InceptionV3d 卷积层
- 7.编译模型（一定要在锁层以后操作）
8. 在新的数据集上训练几代
- 9.现在顶层应该训练好了，开始微调 InceptionV3 的卷积层。锁住底下的几层，然后训练其余的顶层。查看每一层的名字和层号，看看应该锁多少层
- 10.我们选择训练最上面的两个 Inception block，锁住前面 249 层，然后放开之后的层
- 11.重新编译模型，使上面的修改生效，设置一个很低的学习率，使用 SGD 来微调

训练过程中由于图片预处理的时候需要大量内存和 CPU 使用，在内存不足的时候会报 `MemoryError` 的错误，这个时候两个方法：

- 1.减少数据量，会降低训练效果
- 2.分批量加载数据
- 3.增加硬件配置，如加内存，这种方法会增加训练成本，但是训练的效果比较好。
- 4.使用云服务

本项目中解决方法是第 4 种，在 `aws` 上面部署训练模型。因为本地训练的图片

数量达到了 20000 万张，所以选择了 GPU 计算实例，类型=p2.8xlarge，  
vcpu=32，内存=488G，这样可以提高训练的速度。初步估算内存使用峰值为  
80G 左右，使用内存不低于 100G，因为训练过程会消耗大量内存。

### 3.3. 完善

在这一部分，你需要描述你对原有的算法和技术完善的过程。例如调整模型的参数以达到更好的结果的过程应该有所记录。你需要记录最初和最终的模型，以及过程中有代表性意义的结果。你需要考虑的问题：

初始结果是否清晰记录了？

完善的过程是否清晰记录了，其中使用了什么技术？

完善过程中的结果以及最终结果是否清晰记录了？

训练过程中 fit 函数的 epochs 参数是训练次数，这里并不是次数越多效果越好，一定次数后的效果就有所稳定，这样设置一个合理的 epochs 值很有必要，可以节约时间。这里根据测试，使用 epochs=3，训练效果已经可以了。  
通过对训练过程中的 logloss 进行记录，当 logloss 下降到一定程度后就停止训练。

可以看到下面是首次训练的 logloss 值，最小为 0.1201

```
Train on 20000 samples, validate on 5000 samples
Epoch 1/3
20000/20000 [=====] - 2341s - loss: 0.2965 - val_loss: 0.1716
Epoch 2/3
20000/20000 [=====] - 2341s - loss: 0.2959 - val_loss: 0.1201
Epoch 3/3
20000/20000 [=====] - 2342s - loss: 0.2850 - val_loss: 0.1726
```

通过设置一个很低的学习率，使用 SGD 来微调后 Logloss 值最低降到了  
0.0382

```
Train on 20000 samples, validate on 5000 samples
Epoch 1/3
19995/20000 [=====>.] - ETA: 0s - loss: 0.0125 - acc: 0.9957Epoch 00000: val_loss improved from inf to 0.03817, saving model to saved_models/weights.best.Inception.hdf5
20000/20000 [=====] - 3050s - loss: 0.0125 - acc: 0.9957 - val_loss: 0.0382 - val_acc: 0.9888
Epoch 2/3
19995/20000 [=====>.] - ETA: 0s - loss: 0.0094 - acc: 0.9963Epoch 00001: val_loss did not improve
20000/20000 [=====] - 3049s - loss: 0.0094 - acc: 0.9963 - val_loss: 0.0404 - val_acc: 0.9882
Epoch 3/3
19995/20000 [=====>.] - ETA: 0s - loss: 0.0104 - acc: 0.9960Epoch 00002: val_loss did not improve
20000/20000 [=====] - 3052s - loss: 0.0104 - acc: 0.9960 - val_loss: 0.0436 - val_acc: 0.9888
```

## 4. 结果

(大概 2-3 页)

## 4. 1. 模型的评价与验证

在这一部分，你需要对你得出的最终模型的各种技术质量进行详尽的评价。最终模型是怎么得出来的，为什么它会被选为最佳需要清晰地描述。你也需要对模型和结果可靠性作出验证分析，譬如对输入数据或环境的一些操控是否会对结果产生影响（敏感性分析 **sensitivity analysis**）。一些需要考虑的问题：

最终的模型是否合理，跟期待的结果是否一致？最后的各种参数是否合理？

模型是否对于这个问题是否足够稳健可靠？训练数据或输入的一些微小的改变是否会极大影响结果？（鲁棒性）

这个模型得出的结果是否可信？

经过训练后的模型 Logloss 降低到 kaggle 中成绩排名的 10%以内，与期待的结果过相一致，参数经过调整后也很合理。

模型在训练数据较少的情况下表现不好，但是大数据量时，如果变动的数据量较小时对模型训练结果影响不明显。

经过在测试集上的验证，模型的效果很好，复合图片真实的情况。

## 4. 2. 合理性分析

在这个部分，你需要利用一些统计分析，把你的最终模型得到的结果与你的前面设定的基准模型进行对比。你也分析你的最终模型和结果是否确实确实解决了你在这个项目里设定的问题。你需要考虑：

最终结果对比你的基准模型表现得更好还是有所逊色？

你是否详尽地分析和讨论了最终结果？

最终结果是不是确实确实解决了问题？

对比基准模型，目前的模型表现很好。在测试集上面的表现也很稳定。

# 5. 项目结论

(大概 1-2 页)

## 5. 1. 结果可视化

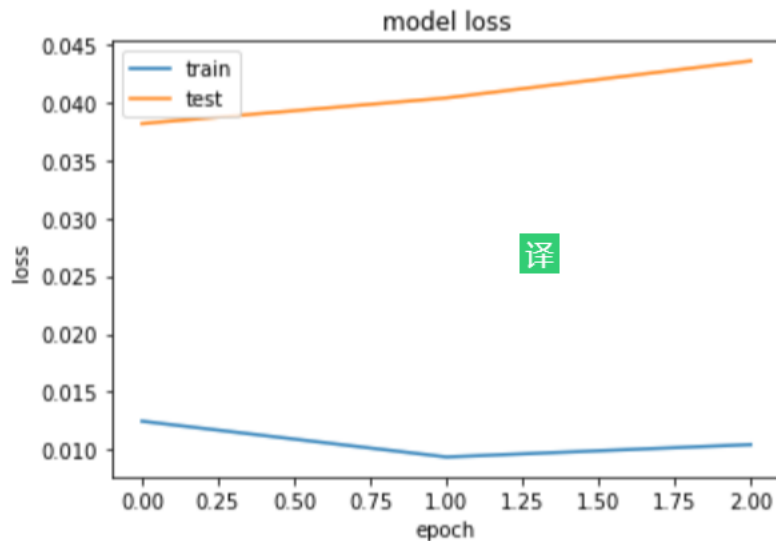
在这一部分，你需要用可视化的方式展示项目中需要强调的重要技术特性。至于什么形式，你可以自由把握，但需要表达出一个关于这个项目重要的结论和特点，并对此作出讨论。一些需要考虑的：

你是否对一个与问题，数据集，输入数据，或结果相关的，重要的技术特性进行了可视化？

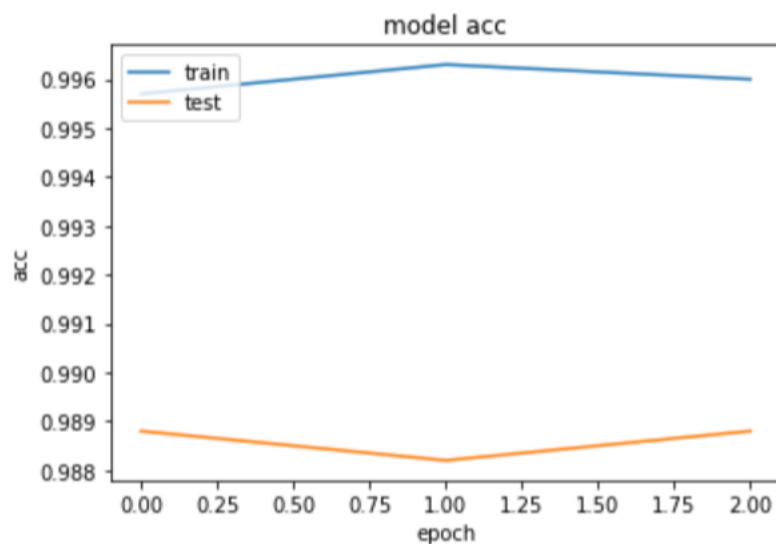
可视化结果是否详尽的分析讨论了？

绘图的坐标轴，标题，基准面是不是清晰定义了？

Logloss 的趋势图：



准确率趋势图：



## 5.2. 对项目的思考

在这一部分，你需要从头到尾总结一下整个问题的解决方案，讨论其中你认为有趣或困难的地方。从整体来反思一下整个项目，确保自己对整个流程是明确掌握的。需要考虑：

你是否详尽总结了项目的整个流程？

项目里有哪些比较有意思的地方？

项目里有哪些比较困难的地方？

最终模型和结果是否符合你对这个问题的期望？它可以在通用的场景下解决这些类型的问题吗？

项目从查看项目要求文档开始，对给出的样例进行了仔细的阅读。对项目最终

的成果有了大致的映象。

研究了 kaggle 的比赛数据，以确定基准模型。对数据集的样例的大小，图片分辨率等做了统计。最终依据模型训练的需要对图片进行预处理，归一化处理。最后进行模型训练，验证模型，进行调参。确定最终模型后在测试集上进行验证。

搜集数据，对数据进行预处理，这部分工作对于机器的硬件要求较高。需要借助 gpu 运算，初期在自己机器上面进行预处理，结果在预处理后报错显示 MemoryError，最终解决方法是使用 AWS 进行图片的预处理和后续的模式训练，使用 AWS 极大的提高了训练的速度。并且使用 AWS 的竞价方式，解约了训练成本。

使用 AWS 的时候环境配置也是一个难题，在本地运行正常的在 AWS 环境会报错，总结了一下，大多数都是环境问题，常见的文件问题：

Python 版本问题，python2.7 与 python3.6 的兼容问题，本项目使用的是 python3.6 环境。

镜像中自带库版本与训练所需的库版本冲突导致，利用 anaconda 将正常的训练环境备份，镜像使用纯净版本的操作系统，手动配置运行环境，这样会减少莫名其妙的错误。

## 5.3. 需要作出的改进

在这一部分，你需要讨论你可以怎么样去完善你执行流程中的某一方面。例如考虑一下你的操作的方法是否可以进一步推广，泛化，有没有需要作出变更的地方。你并不需要确实作出这些改进，不过你应该讨论这些改进可能对结果的影响，并与现有结果进行比较。一些需要考虑的问题：

是否可以有算法和技术层面的进一步的完善？

是否有一些你了解到，但是你还不能够实践的算法和技术？

如果你将最终模型作为新的基准，你认为还能有更好的解决方案吗？

训练过程中对机器要求比较高，

对于 InceptionV3 我们可以进一步增加网络深度，这个已经有相关算法出现如

InceptionV4，他增加了 InceptionV3 的深度。另外，Xception 是 Inception 架构的扩展，他用深度可分离的卷积代替了标准的 Inception 模块

在提交之前，问一下自己...

你所写的项目报告结构对比于这个模板而言足够清晰了没有？

每一个部分（尤其分析和方法）是否清晰，简洁，明了？有没有存在歧义的术语和用语需要进一步说明的？

你的目标读者是不是能够明白你的分析，方法和结果？

报告里面是否有语法错误或拼写错误？

报告里提到的一些外部资料及来源是不是都正确引述或引用了？

代码可读性是否良好？必要的注释是否加上了？

代码是否可以顺利运行并重现跟报告相似的结果？

## 6. 引用

[1] Dogs vs. Cats Redux: Kernels Edition, 2017.

<https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition>.

[2] Dogs vs. Cats, 2013. <https://www.kaggle.com/c/dogs-vs-cats>.

[3] Kaz Sato, How a Japanese cucumber farmer is using deep learning and TensorFlow, <https://cloud.google.com/blog/products/gcp/how-a-japanese-cucumber-farmer-is-using-deep-learning-and-tensorflow>.

[4] Log Loss, [http://wiki.fast.ai/index.php/Log\\_Loss](http://wiki.fast.ai/index.php/Log_Loss)

[5] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna, Rethinking the Inception Architecture for Computer Vision, [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/papers/Szegedy\\_Rethinking\\_the\\_Inception\\_CVPR\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.pdf)

[6] [1] Dogs vs. Cats Redux: Kernels Edition, 2017.

<https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition>.

[2] Dogs vs. Cats, 2013. <https://www.kaggle.com/c/dogs-vs-cats>.

[3] Kaz Sato, How a Japanese cucumber farmer is using deep learning and TensorFlow, <https://cloud.google.com/blog/products/gcp/how-a-japanese-cucumber-farmer-is-using-deep-learning-and-tensorflow>.

[4] Log Loss, [http://wiki.fast.ai/index.php/Log\\_Loss](http://wiki.fast.ai/index.php/Log_Loss)

[5] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon

Shlens, Zbigniew Wojna,  
Rethinking the Inception Architecture for Computer  
Vision, [https://www.cv-  
foundation.org/openaccess/content\\_cvpr\\_2016/papers/Szegedy\\_Rethinking\\_the\\_Inception\\_CVPR\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.pdf)  
[6] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet  
classification with deep convolutional neural networks  
[7] OpenCV C interface: <http://docs.opencv.org>  
[8] Lalis, Jeremias; Gerardo, Bobby; Byun, Yung-Cheol. An  
Adaptive Stopping Criterion for Backpropagation Learning  
in Feedforward Neural Network