***Udacity Machine Learning Engineer Huanghui Hong***

***Capstone Project July 13, 2020***

**猫狗大战**

**1 定义**

**1.1 项目综述**

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。

**1.2 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。

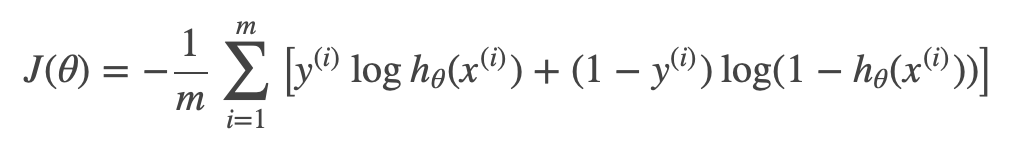
同时，我将在标记的数据集上进行训练和评估算法。为了使算法尽可能地具有高性能，我将调整算法体系结构、优化器（adam, adadelta 等）以及众多的参数，最终能够达到 kaggle Public Leaderboard 前 10%。

项目大致步骤：

1. **数据预处理**
   1. 加载kaggle提供的数据集
   2. 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
   3. 对数据集中的每个数据进行resize操作，使其具有相同的尺寸
   4. 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**
   1. 加载Keras中已训练的模型（如VGG、ResNet、AlexNet等）
   2. 去除模型中全连接层部分
   3. 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**
   1. 加载CNN模型的网络权重
   2. 不改变CNN模型的权重，使用训练数据对全连接层进行训练
   3. 尝试调整算法体系结构、优化器（adam, adadelta等）以及众多的参数对模型进行训练
   4. 采用不同的模型，重复2.a步骤
   5. 选择性能最佳的模型
4. **模型评估**
   1. 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**
   1. 可视化数据集
   2. 可视化模型的准确率，损失函数等

**1.3 指标**

采用对数损失函数：



其中：

* n为图片数量
* hθ(x(i)) 为模型预测是狗类别的概率
* yi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

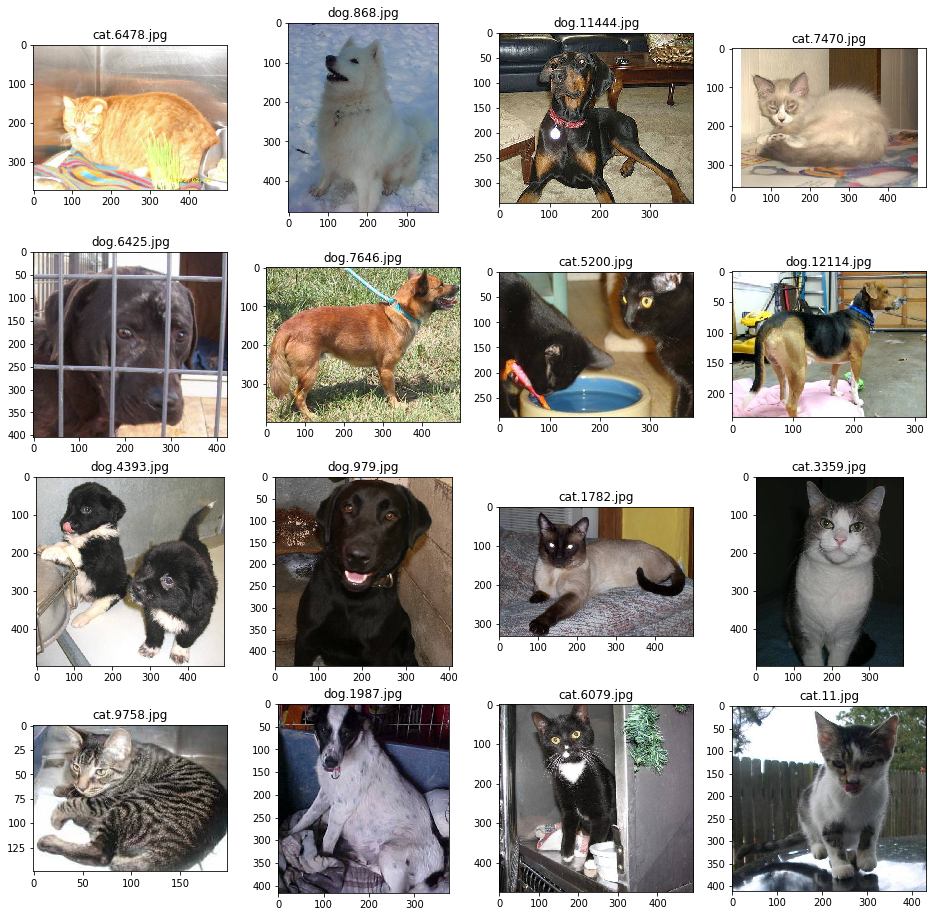
损失函数的数值越小，代表模型的性能越好。本项目将使用该损失函数评估模型的性能。

**2 分析**

**2.1 数据挖掘**

从kaggle网站下载数据集，其中包含两个压缩文件（test.zip and train.zip）和一个sample\_submission.csv。train.zip文件中含12500张猫类别图片和12500张狗类别图片，每张图片的文件名均标识了图片的类别。test.zip文件含12500张无标签的文件，用于模型的测试，并且生成类似sample\_submission.csv的文件。

下图是随机从训练集中抽取的16张图片，显示了图片的尺寸和对应的文件名。



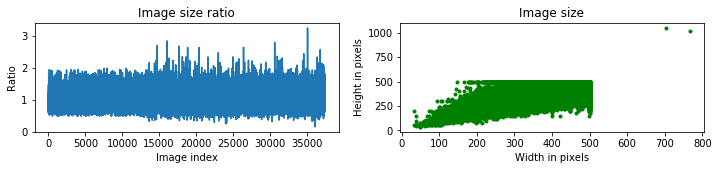
通过对实例图的观察，可得出一些初步的猜测与观点：

* 如第7、9张图所示，有些图片中包含了多个同类型的动物，这对于一些算法（如 simple softmax regression classifiers）较难进行分类。但对于CNN这类算法而言却是有助于提高其准确率。
* 图像大部分都能以动物为中心。
* 图片具有不同的尺寸。
* 图片中有大量与分类信息无关的背景，且背景的差别很大。因为我们需要避免这些噪点，有利于提升算法的泛化性。

**2.2 数据可视化**

关于我们的数据，需要注意的重要一点是图像大小的巨大差异。 实际上，我们的输入层需要具有固定的图像输入大小。 因此，我们需要知道每个图像是否可以用作输入，以及如何对其进行预处理。

如下图为我们提供每个图像的大小和宽度/高度比。



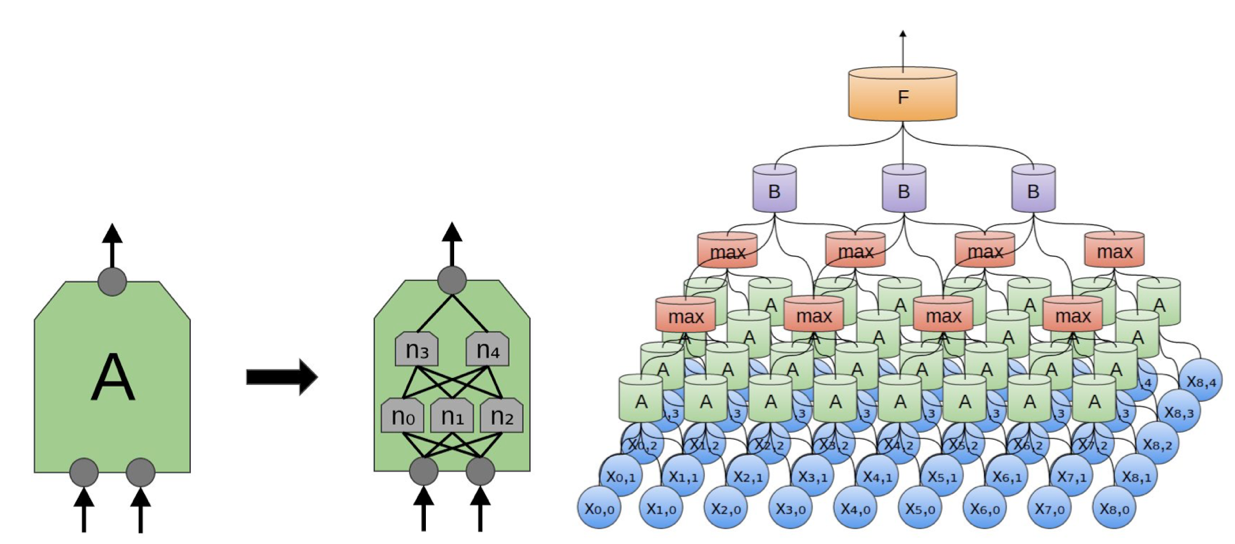
* 图片的平均比率是1.157。
* 我们可以注意到图像尺寸图中的两个离群值与其他图像数据集相比具有很高的高度和宽度。
* 在比率图上，我们还可以注意到存在离群的比率值，而其他大多数图像的比率都在2以下。
* 这些图将帮助我们确定在将图像用作算法输入之前是否需要调整图像大小或裁剪图像。

**2.3 算法与方法**

当softmax回归试图解决的问题是使用具有高可变性的数据（许多非常不同的输入实际上具有相同的标签）时，简单的softmax回归效率有限。在这种情况下：两只猫的图片可能截然不同。 动物可能处于图片中的不同位置，而具有多个动物的图像使该算法更加困难作出准确判断。

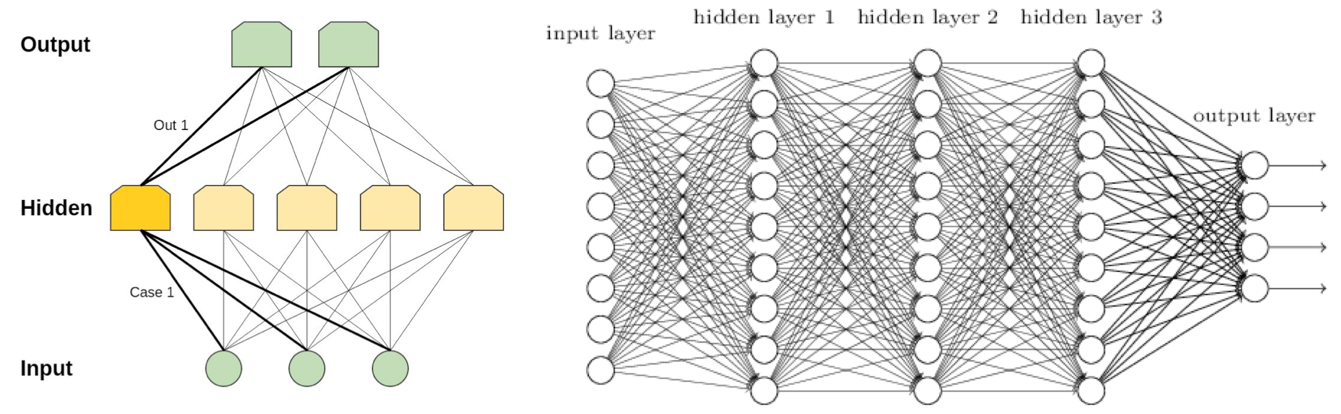
一种用于图像识别的流行工具称为卷积神经网络（CNN）。 这对于重要特征（例如猫耳的一部分）可以在任何地方的图像非常有用，在这种情况下，动物可以处于不同的位置，有时每个图像不止一个。

CNN就像一个微小的神经网络，无处不在地应用于图像的正方形部分。 它以色块（一小组像素）作为输入来分析图像的这一部分所包含的内容。 由于将相同的神经网络应用于图像的所有部分，因此无论重要特征在哪里，它都会检测出来。



最大池化层在上一层的几个封闭块的输出上获得最大的特征。 在卷积层之后添加maxpool层会“缩小”，并使随后的卷积层能够从较高的角度对图像进行处理，并具有重要的选定功能。

一个全连接层由许多神经元组成，每个神经元将所有输入组合成一个简单的计算公式，结果就是其输出。 使用一系列完全连接的层，一层的输出就是另一层的输入，可以通过大量简单的操作来创建复杂的功能。 结合提取的特征并做出诸如“嗯，我们有两个尖尖的耳朵，胡须和两个绿色的眼睛”这样的推理非常有用。 那一定是只猫。”



Drop层是有时使信息传递有时阻止它的层。 使用它进行训练（但在评估时让一切顺利）可以迫使算法在脑海中多次写入相同的推理，而减少“记忆”数据。 它迫使它学习方法论，而不是记住一切。 当算法倾向于过度拟合数据时，非常适合使用。

因此，我们将在算法中使用一些CNN提取重要特征，然后使用多个完全连接的层以有用的方式组合这些特征，以确定图像是关于猫还是狗。

训练包括将图像作为输入，使我们的网络输出每个类别的概率得分（例如：我们的模型可能会说“我60％确信这是一只猫，而40％这是一只狗”），并且由于图像被标记（假设它是一只猫），并且我们知道每个神经元权重的影响，我们会自动调整这些权重以使我们的模型更接近100％的信任度。 该调整称为“反向传播”。 为此，我们计算损失函数，并使用优化器找到其最小值，以找到这些权重的最佳组合。

具有卷积层的算法需要训练大量数据。 由于我们的数据集仅包含25000张图像，这些图像将分为训练，验证和测试集，因此我们将使用图像处理技术来生成更多图像。

通过调整以下参数，我们将能够改善算法：

* 输入图像的通道数（1为灰度，3为彩色图像）。
* 输入图像的高度和宽度。高分辨率图像可提供更好的图像细节，但需要更多的计算量和更多的训练图像。
* 优化器的学习率。高学习率使优化器可以快速找到其优化任务的结果，这也可能意味着它找不到最佳结果，从而导致最终的算法不正确。高对于很少的训练数据有好处，低对于拥有大量数据的情况更好。
* 神经元的初始权重和偏见。
* 架构的层。
* 图层类型（卷积，完全连接，maxpool，Dropout层）。
* 图层参数（例如：每层神经元的数量，CNN的输入和输出长度，Dropout的参数）。
* 图像生成算法及其参数（水平图像翻转，亮度和对比度变化）。
* 每批的示例数。越高，精度越低，但优化器速度越快。
* 用于训练的批次数。

**2.4 Benchmark**

根据这个项目的考核标准，需要获得kaggle比赛前10%的成绩。在kaggle竞赛中，截止目前提交成绩的人数为1314名。前10%则意味着分数需要超过131名，而在排行榜中可以看到第131名的得分为0.06127。故基准模型指标得分需要小于0.06127。

**3 方法**

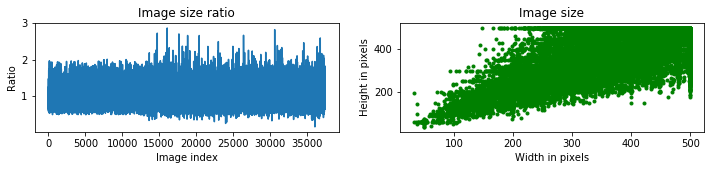
**3.1 数据预处理**

为了有效学习，算法需要一些高质量的输入，但是静态图像的大小不能太大，以致于无法处理有限的信息。要在大量信息中找到模式，我们需要大型模型，由于我们的计算能力有限，所以不可能是大型模型。此外，如果要过滤的信息较少，则更容易理解图像中的重要内容。因此主要是通过调整大小或裁剪来获得较低分辨率的图像。

从数据集中删除了上面提到的宽度和高度最大的两个离群值图像。

由于使用TensorFlow函数存在错误，已从该过程中删除了用于生成其他训练图像的图像裁剪操作：裁剪后的图像由随机的彩色像素组成，因此无法用于训练。因此，调整图像大小是可选择的操作，并且有必要检查图像是否因这种转换而失真太多。从对宽度/图像比率的研究中发现了一个特殊的异常值。由于类别之间的对称性（和一点完美主义），还删除了第二个不太明显的离群值：它的图像尺寸比为3.8，高于其他图像。可能没有必要删除第二个图像，因为狗和猫类别之间的一个图像差异不会对算法造成太大的影响。通过绘制调整大小的图像并在数据集中保留尽可能多的图像。

移除这三个离群值后的图像统计数据显示出更加一致的新数据集：



由于我们有24997张图像（12499的猫类型和12498的狗类型）和12500张测试图片，它们分为训练，验证和测试集。 在训练期间，我们可能需要多次使用相同的图像，但是每次都随机更改其对比度，亮度和随机翻转。 每个新图像都非常类似于其原始图像，但仍有助于算法泛化性的提升。

训练，验证是通过拆分原始标记的数据集而创建的，并保持每组的猫/狗比例相等。

带有标签的图像在集合中重新分区：

|  |  |  |
| --- | --- | --- |
|  | Cats Number | Dogs Number |
| Training Set | 10499 | 10498 |
| Validation Set | 2000 | 2000 |

图像信息存储在单个Pandas DataFrame中，可方便地使用数据填充训练算法。 该算法将使用计算机中的文件位置，图像标签（0表示猫，1表示狗）和设置名称（训练，验证）。

数据集DataFrame末尾的部分数据：

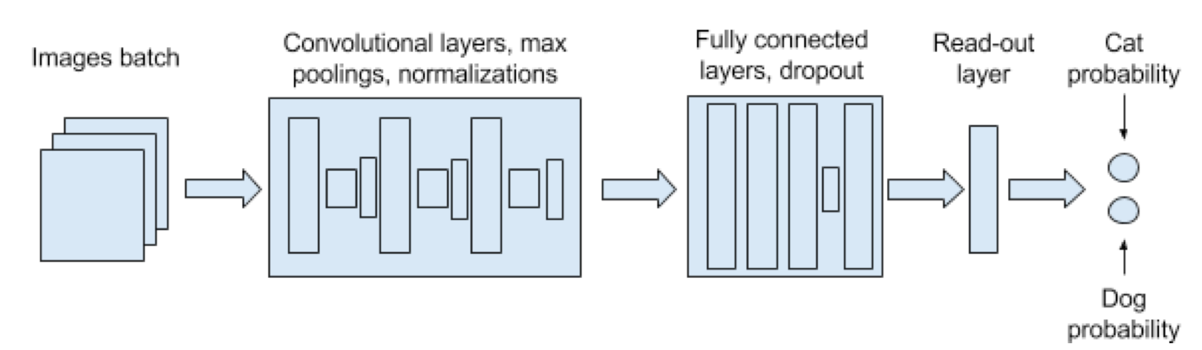


**3.2 实现**

使用接口ImageDataGenerator对每个用于训练或测试的数据做进一步处理，例如归一化图片的每个像素值，并为下面所建立的模型提供迭代器。

图

如下图所示，结合迁移学习的思想，我们需要载入已经在ImageNet训练过的ResNet50/InceptionV3/Xception模型。这些模型已通过大量数据的训练，卷积层能否提炼出对识别图像有用的特征，并通过全连接层将这些特征映射到标签。



所以载入模型后，我们需要冻结卷积层的权值，并自定义全连接层，卷积层的输出则为全连接层的输入。再通过训练得到全连接层的权值，并且得到最终的分类模型。

**具体实施步骤：**

1. 导入没有top层的ResNet50模型
2. 使用ResNet50模型对数据进行预测，保存特征向量。（ResNet50模型的权值是固定的，所以其输出的特征向量是确定的。对于全连接层。这些特征向量则是其数据集。在进行训练时，就不需要每一代都重复巨大的卷积层的计算，大大降低运算量。）
3. 构建全连接层，将步骤2的特征向量作为其输入，并设置二分类的输出层。
4. 编译和训练模型
5. 保存模型
6. 使用模型进行数据预测

**选用参数：**

* 使用keras.application.resnet50作为预训练模型。
* 使用adadelta, Adam, sgd作为优化器分别对模型进行训练。
* 使用dropout=0.5防止模型过拟合。

**3.3 改进**

对于adadelta, Adam, sgd三种优化器，模型的分数都在0.05-0.065之间波动。虽然效果已经很好，但是依然还有改进的空间。

通过训练InceptionV3和Xception模型并重复以上的训练过程，可以得到对应于InceptionV3和Xception模型的特征向量。每个模型所对应的特征向量shape为(25000, 2048)，合并三个模型的特征向量后，得到shape为(25000, 6144)的特征向量。每个(1, 6144)向量对应一张图片，也是全连接层的一次输入数据。

相对于一个模型输出的特征向量，经过三个特征向量组合训练后的效果有明显的提升，loss基本在（0.01-0.023）之间波动。

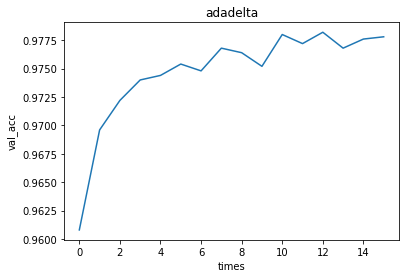
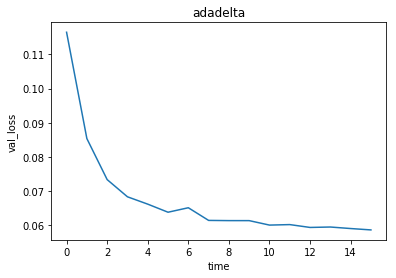
**4 结果**

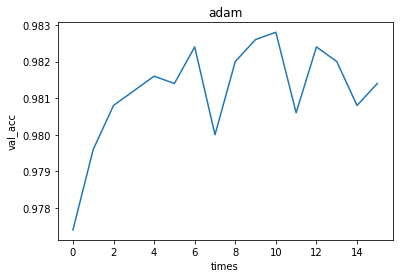
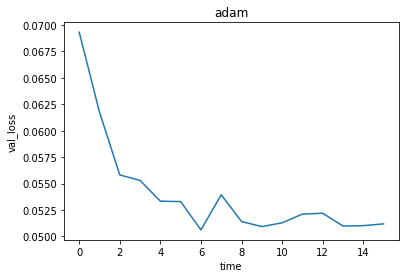
**4.1 模型评估与验证**

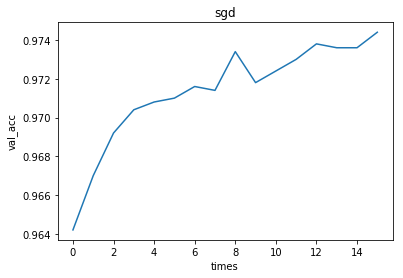
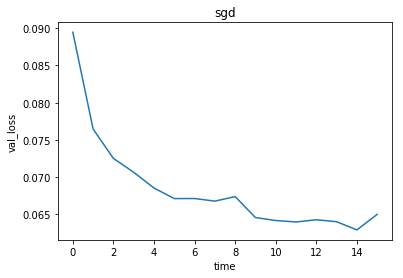
通过模型训练之后所返回train\_history数据,可描绘loss和accuracy随时间的变化趋势。

**优化之前**

只用resnet50模型得到的特征向量进行训练，设置参数batch\_size=128, nb\_epoch=16, validation\_split=0.2。其损失函数和精度的变化趋势：



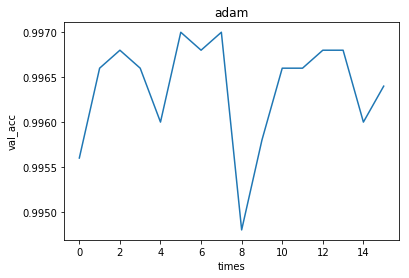
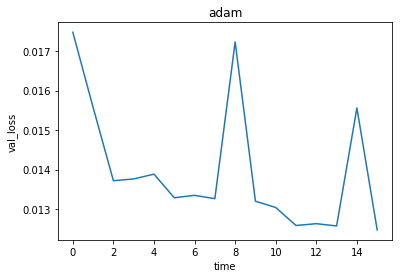
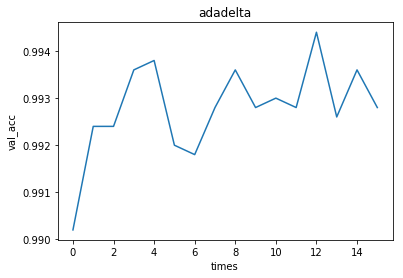
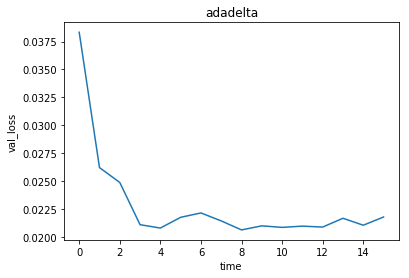


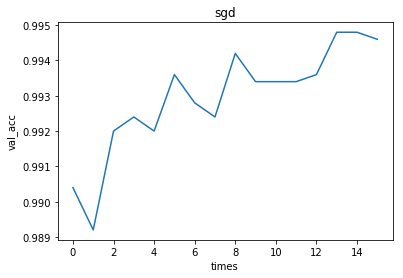
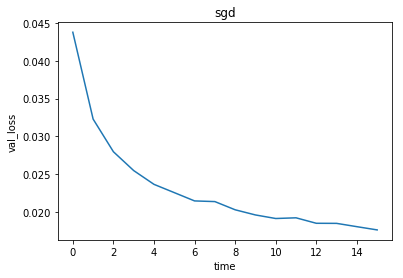


如上图可知，三个优化器训练的准确率在0.0974-0.0983之间波动。

**优化之后**

使用ResNet50/InceptionV3/Xception模型预测同一个数据集，得到三个(25000, 2048)特征向量数据集。行数不变将其进行合并，得到一个(25000, 6144)特征向量数据集。设置参数batch\_size=128, nb\_epoch=16, validation\_split=0.2，其损失函数和精度的变化趋势：





如上图可知，使用联合迁移学习模型的方法的精度有一定程度的提高。分别使用三种优化器进行训练的准确率都在99.2以上。

将预测生成的文件提交到kaggle，且得分为：

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

三个优化器的结果都能满足项目的最低要求（前10%在kaggle leadboard），且训练循环次数仅为16次。证明联合模型的方法能提供有效的图像特征，使得全连接层快速地收敛。

**5 结论**

**6 参考文献**