Dogs vs. Cats Project

mnlife

mnlife@foxmail.com

Abstract

2013 年,kaggle 上举办了一个娱乐比赛: Dogs vs. Cats。而在去年,Machine learning 领域发生了很大变化, Deeplearning 取得了巨大的突破,其中一个典型的例子就是在 IMAGENET 的挑战赛上, Classification error 由 2011 年的 26%, 降至 2012 年 16%,而其创新性的使用了 deep convolutional neural network。之后, Classification error 逐年降低,说明了 Deep convolutional neural network 优越性。

而这个项目,使用 Inception-ResNet V2, 该网络由 Fei wang, Mengqing Jiang, Chen qian, shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, Xiaoou Tang 提出. 这个网络是一个高度模块化的图像分类网络架构.将这个网络迁移至该项目, 希望可以取得良好的效果。

1. Introduction

项目所要解决的为一个典型的二分类问题:给定一张图像,识别其是狗或猫。在图像中,它们有着各种不同的姿态, 这个识别带来了很大的困难, 尤其是在 2012 年 Deeplearning 出现之前, 这几乎是一个不可能完成的任务。 不过, 随着近几年深度学习不断出现新的结构, 各种学习模型的不断优化, 使得该问题可以更好的被解决。

尤其是采用了 Inception-ResNet V2 之后, 对狗和猫进行识别, 只是用来验证该网络的优越性而已, 而以上的问题应该可以被很好地解决。

不过, 由于本次的训练数据集只有 25000, 所以很难使用 Inception-ResNet V2 模型进行恰当的拟合, 对模型的容量选择也需要进行不断的尝试。

2. Solution Statement

采用了该网络后, 模型容量增大后导致的 underfitting 问题会变得非常显著, 而模型容量过小又不能很好的对这个分类问题进行表达, 这就成为了一个比较棘手的问题。

可以先从理论方面来分析一下: 本次模型的容量不是问题, 为了让模型泛华的更好的办法就是使用更多的数据进行训练。 但在本次任务中拥有的数据量很有限, 解决这一问题的一个方法是创建假数据并添加到训练集当中。 对于一些机器学习任务, 创建假的数据相当的简单。

数据集增强对一个具体的分类问题是特别有效的。图像是高维的并包括各种巨大的变化因素,其中有许多可以轻易的模拟。 即使模型已使用卷积和池化技术对部分平移保持不变, 但沿训练图像每个方向平移几个像素的操作通常可以大大改善泛华。 有许多操作已被证明非常有效: 如平移, 旋转或缩放图像。

在神经网络的输入层注入噪声也是数据增强的一种方式。 神经网络被证明对噪声不是非常健壮(Tang and Eliasmith, 2000)。 改善神经网络健壮性的一种方法是将随机噪声添加到输入再进行训练。 向隐藏单元施加噪声也可以,这可以看做在抽象层上对数据集进行增强。 Poole et al.(2014)最近表明, 噪声的幅度被细心调整后该方法是非常有效的。不过对于正则化策略 Dropout, 可以看做是通过与噪声相乘构建新输入的过程。

通过上面的理论分析, 人工设计的数据集增强方案可以大大减少机器学习技术的泛化误差, 可以通过使用上面的方法, 很好的解决训练数据不足引起的欠拟合问题, 大大改善模型的性能, 不过需要注意的是, 应该讲训练集, 扩充数据集, 测试集, 数据增强方式获得的数据集进行随机后, 再从中分割出 train set, test set, validation set。

进行数据增强后, 训练数据集还是稍显不足, 可以对 Inception-ResNet V2 采用迁移学习, freeze 浅层的网络, 只对深层的网络进行训练, 选用 binary crossentropy, 采用 Adam 学习算法（该算法实际上为 Momentum 与 RMSProp 的结合), 并在学习的过程中进行一些超参数的调整, 如 batch_size, epochs 等等。

3. Evaluation Metrics

如下交叉熵损失公式为这次的评估指标:

$$\text{LogLoss} = -\frac{1}{n}\sum_{i=0}^{n} 1\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]$$

where:

- $\hat{y}_i$ 是将一个 feature 预测为狗的可能性

• y_i 为 1，则该图片为狗，否则为猫
• ln()是一个自然对数 (base e)
log loss 越小越好。

这次项目的任务为预测二值型变量 y 的值，具有两个类的分类问题可以归结为这种形式。

此时最大似然的方法是定义 y 在 x 条件下的 Bernoulli 分布。

Bernoulli 分布仅需单个参数来定义。神经网络只需要预测 $P(y = 1 \mid x)$ 即可。为了使这个数是有效的概率，它必须处在区间[0, 1]中。

为满足该约束条件需要一些细致的设计工作。假设我们打算使用线性单元，并且通过阈值来限制它成为一个有效的概率：

$$P(y = 1 \mid x) = max\{0, min\{1, w^T h + b\}\}$$

这样定义的条件概率分布，我们无法使用梯度下降来高效训练它。当$\mathbf{w^T + b}$处于[0, 1]区间外时，模型的输出对其参数的梯度都将是 0。梯度是 0 通常是有问题的，因为学习算法对于如何改善相应的参数不再具有指导意义，因此这样来预测二值型输出是有问题的。

相反，使用 sigmoid 输出单元结合最大似然来实现，那么可以保证无论模型何时给出了错误的答案时，总能有一个较大的梯度。()

sigmoid 输出单元定义为：

$$\hat{y}_i = \sigma(w^T + b)$$

我们可以认为 sigmoid 输出单元具有两个部分。首先，它使用一个线性层来计算$z = w^T h + b$。其次，它使用 sigmoid 激活函数将 z 转化成概率。

我们暂时忽略对 X 的依赖性，只讨论如何改变 z 的值来定义 y 的概率分布。Sigmoid 可以通过构造一个非归一化（和不为 1）的概率分布$\tilde{P}(y)$来得到。我们可以随后除以一个合适的常数来得到有效的概率分布。然后对它取指并归一化，可以发现这服从 Bernoulli 分布，该分布受 z 的 sigmoid 变换控制：

$$\log \tilde{P}(y) = yz$$
$$\tilde{P}(y) = \exp(yz)$$
$$P(y) = \frac{\exp(yz)}{\sum_{y'=0}^{1} \exp(y'z)}$$
$$P(y) = \sigma((2y - 1)z)$$

基于指数和归一化的概率分布在统计建模的文献中很常见。用于定义这种二值型变量分布的变量 z 被称为分对数（logit）。

而 sigmoid 最适合使用最大似然学习，因为代价函数中的 log 抵消了 sigmoid 中的 exp。如果没有这个效果，sigmoid 的饱和性会阻止基于梯度的学习做出好的改进。我们使用最大似然来学习一个由 sigmoid 参数化的 Bernoulli 分布，它的损失函数为：

$$J(\theta) = -\log P(y|x)$$
$$= -\log \sigma((2y - 1)z)$$
$$= \zeta((1 - 2y)z)$$

上述公式中，将损失函数写成 softplus 函数的形式，我们可以看到它仅仅在$(1 - 2y)z$取绝对值非常大的负值时才会饱和。因此饱和只会出现在模型已经得到正确答案时：当 y = 1 且 z 取非常大的正值时，或者 y = 0 且 z 取非常小的负值时。当 z 的符号错误时，softplus 函数的变量$(1 - 2y)z$ 可以简化为|z|。当|z|变得很大并且 z 的符号错误时，softplus 函数渐进的趋向于它的变量|z|.对 z 求导则渐近的趋向于 sign（z），所以对于极限情况下极度不正确的 z，softplus 函数不会收缩梯度。这个性质很有用，因为它意味着基于梯度的学习可以很快的改正错误 z。

当我们使用其他的损失函数，如均方误差等，损失函数就会在$\sigma(z)$饱和时饱和。Sigmoid 激活函数在 z 取非常小的负值时会饱和到 0，当 z 去非常大的正值时会饱和到 1。这种情况一旦发生，无论此时模型给出正确或错误的答案，梯度就会变得非常小而难以去学习。因此，最大似然几乎总是训练 sigmoid 输出单元的优选方法。

理论上来说，sigmoid 的对数总是确定和有限的，因为 sigmoid 的返回值总是被限制在开区间(0, 1)上，而不是使用整个闭区间[0, 1]的有效概率。在软件实现时，为了避免数值问题，最好将负的对数似然写作 z 的函数，而不是$\hat{y} = \sigma(z)$的函数。如果 sigmoid 函数下溢到 0，那么之后对$\hat{y}$取对数就会得到负无穷。

从上述理论分析可以看出，sigmoid 使用最大似然来学习恰好可以抵消它易饱和的特性。因此这次的设计的迁移学习模型采用 sigmoid 输出单元结合最大似然来实现。

4. Datasets and Inputs

在 kaggle 的 Dogs vs Cats 竞赛中，公开了如下的数据集，该数据集包含 train set 与 test set,图像为猫或者狗，任何一个图像都有与之对应的 labels。其中，train set 包含 25000 张图像,test set 包含 12500 张。 对于其中的图像,其中的 labels 都有特别的定义: dog = 1， cat = 0。这样定义后，可以使用 sigmoid 函数来表示网络的输出级，并对相应的 feature 进行预测。

图像中包含各种各样姿态的猫与狗：有站着的，躺着的，坐着的，有正脸，侧脸。 半身照，全身照，这给识别带来了相当大的麻烦。不过,现在的深度学习模型的容量也越来越大，而且本次使用 Inception-ResNet V2 网络，它可以构建很深的深度学习模型而不必担心 Gradient explosion 与 gradient vanishing。在采用了该网

络后，理论上模型容量可以做得很大，所以对于这些复杂姿势的拟合也不会有什么问题。

不过,由于 train set 太小了，当模型的容量太大时，很容易导致 under fitting，所以此时对于模型容量的选择也是一个需要 trade off 的方面。

此外，如果用以上数据集对网络进行训练,很大程度上可能会由于训练集太小而使网络 Underfitting，所以对数据集进行进行扩充是很有必要的。还有一个很好的数据集可供选择: The Oxford-IIIT Pet Dataset，使用该数据集可以作为扩充数据集对网络进行训练。

对于数据预处理，当输入数据的尺寸不一致时，可以通过改变 kernel 的大小来变形为相同的向量；之后对训练集进行划分时，有必要将训练集拆分为 train set 与 validation set，以便于更好进行模型的优化。

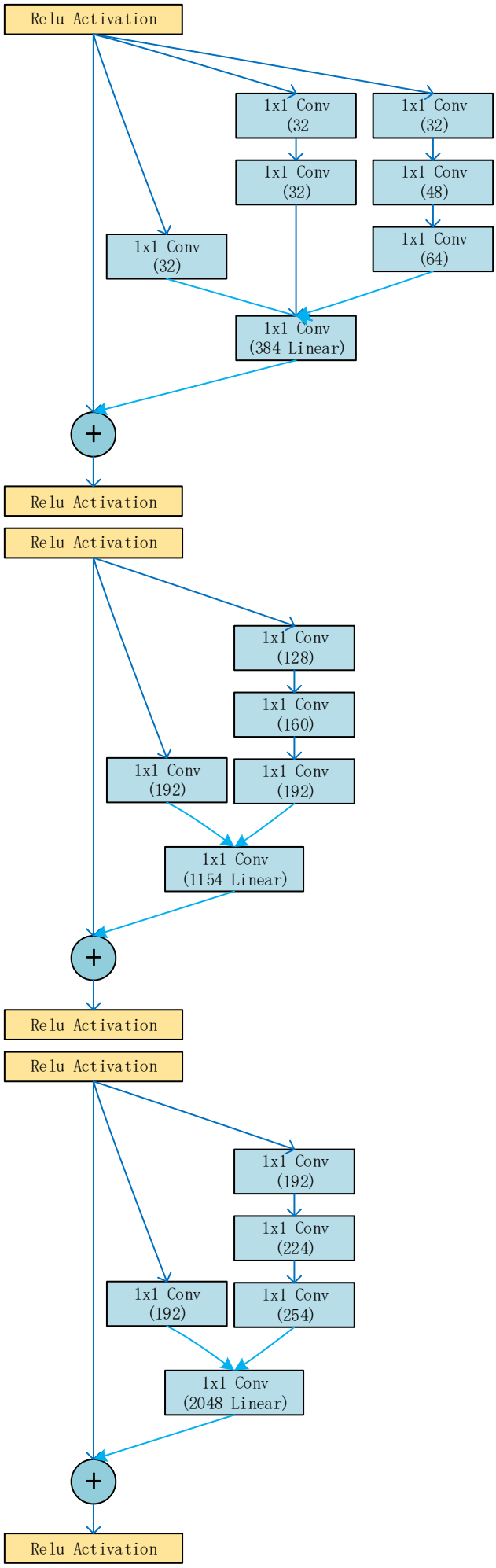
5.Algorithm

该项目由于数据集小，在采用容量较大的模型时就很难进行拟合，所以采用迁移学习是一个比较好的解决办法。模型采用了在 imgnet 上预训练的几个模型：InceptionResNetV2，DenseNet201，Xception，ResNet50，VGG19，InceptionV3。

其中InceptionV3一个最重要的改进是分解，将7x7分解成两个一维的卷积（1x7,7x1），3x3 也是一样（1x3,3x1），这样的好处，既可以加速计算（多余的计算能力可以用来加深网络），又可以将 1 个 conv 拆成 2 个 conv，使得网络深度进一步增加，增加了网络的非线性，还有值得注意的地方是网络输入从 224x224 变为了 299x299，更加精细设计了 35x35/17x17/8x8 的模块

由于这三个模型的预训练模型都是在 imagenet 数据集上得到的,所以需要对它们进行相应的调整,将输出由 softmax 改为 sigmoid,因为该损失函数非常适合二分类问题,这一点在上面也有提到过.

用到了下面三个经典的模块：



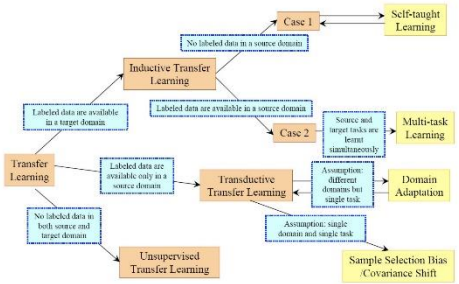
由于用到了多个模型，所以采用模型融合是必不可少的。

为了加快训练的速度，所以本项目没有对整个模型进行训练，而是将训练集在每个预训练模型上进行预测,将该特征提取出来,并与 labels 进行配对并存储。后续训练过程中直接将该特征作为输入直接进行训练即可。

数据增强也有使用，对图像进行翻转，平移等生成更多的训练数据。

•迁移学习

Transfer Learning Settings	Related Areas	Source Domain Labels	Target Domain Labels	Tasks
Inductive Transfer Learning	Multi-task Learning	Available	Available	Regression, Classification
	Self-taught Learning	Unavailable	Available	Regression, Classification
Transductive Transfer Learning	Domain Adaptation, Sample Selection Bias, Co-variate Shift	Available	Unavailable	Regression, Classification
Unsupervised Transfer Learning		Unavailable	Unavailable	Clustering, Dimensionality Reduction



数学定义如下所示：

假设有一个相似领域的解决方案 $\boldsymbol{D_s} = \{\boldsymbol{X_s}, \boldsymbol{f_s(X)}\}$和学习任务 Ts,一个相似的待解决问题 $\boldsymbol{D_T} = \{\boldsymbol{X_T}, \boldsymbol{f_T(X)}\}$ 与学习任务 $\boldsymbol{T_T}$ ，迁移学习用来帮助实现预测函数$\boldsymbol{f_T}(.)$ 与 $\boldsymbol{D_T}$ ，这一过程需要使用到 $\boldsymbol{D_s}$ 与 $\boldsymbol{T_s}$，其中$\boldsymbol{D_s} \neq \boldsymbol{D_T}$，或者 $\boldsymbol{T_s} \neq \boldsymbol{T_T}$。

迁移学习的核心问题是，找到原问题与新问题之间的相似性，才可以顺利的实现知识的迁移。

迁移学习，是指利用数据，任务，或模型之间的相似性，将在旧领域学习过的模型，应用于新领域的一种过程。它主要解决三个方面的问题：

（1）数据的特征空间不同（就是原任务与目标任务的特征是不同的）；

（2）数据的分布是不同的（这种情况一般特征空间是相同的）；

（3）标注标签的花费很昂贵以至于很难标注或者几乎不可能标注。

不过在应用迁移学习时，当源域与目标域不相关时，暴力迁移可能是不成功的。在更糟的情况下，它可能伤害在目标域中的学习表现，这种情况称为负迁移。

•模型融合

在机器学习任务中，模型融合是一个非常强大的提高准确性的方法。

•数据增强

让机器学习模型泛华的更好的办法是使用更多的数据进行训练。不过在实际中拥有的数据是有限的。解决这一问题的一种办法是创建假数据并添加到训练集中。

数据集增强对于一个具体的分类问题来说是特别有效的方法：对象识别。图像是高维的并包括各种巨大的变化因素，其中有许多可以轻易的模拟。即使模型已经使用了卷积与池化技术对部分平移保持不变，沿训练图像每个方向平移几个像素的操作通常可以大大改善泛华。许多其他的操作如旋转图像或者缩放图像也已被证明非常有效。

不过在应用数据增强时要小心，不能使用改变类别的转换。例如在识别“b”和“d”，“6”和“9”时，要注意不能进行图像的翻转。

在神经网络的输入层注入噪声(Sietsma and Dow, 1991)也可以看做数据增强的一种方式。对于许多分类甚至一些回归任务而言，即使小的随机噪声被加到输入，任务仍应该是能被解决的。然而，神经网络被证明对噪声不是非常健壮的(Tang and Eliasmith. 2010)。改善神经网络健壮性的方法之一是简单地将随机噪声添加到输入在进行训练。输入噪声注入是一些如监督学习算法的一部分，如去噪自编码器(Vincent et al. 2008a)。向隐藏单元施加噪声也是可行的，这可以看作是在多个抽象层上进行的数据集增强。Poole at al.(2014)最近表明，噪声的幅度被细心调整后，该方法是非常高效的。

在比较机器学习的基准测试结果时，考虑其采取的数据集增强是很重要的。通常情况下，人工设计的数据集增强方案可以大大减少机器学习技术的泛化误差。将一个机器学习算法的性能与另一个进行对比时，对照试验是必要的。在比较机器学习算法 A 和机器学习算法 B 时，应该确保这两个算法使用同一人工设计的数据集增强方案。假设算法 A 在没有数据集增强时表现不佳，而 B 结合大量人工转换的数据后表现良好。在这样的情况下，很可能是合成转化引起了性能改进，而不是机器学习算法 B 比算法 A 更好。有时候，确定实验是否已经适当控制需要主观判断。例如，向输入注入噪声的机器学习算法是执行数据集增强的一种形式。通常，普适操作（例如，向输入添加高斯噪声）被认为是机器学习算法的一部分，而特定领域（如随机的裁剪图像）的操作被认为是独立的预处理步骤。

6. Benchmark Model

该问题所要参考的基准模型为 kaggle Dogs vs. Cats 的 leaderboard 中的 Public Leaderboard 的排行榜前 3%，该分数为 log loss。该分数的详细计算方式参照 Evaluate Metrics 中的详细介绍。

7.Data preprocess

在本次的项目中，将图像的尺寸处理成一致大小，并进行了简单缩放等处理。同时将训练数据分为训练集与验证集,测试集保持不变.下面对这些技术做一些简要的描述：

数据预处理中，标准的第一步是数据归一化。虽然这里有一系列可行的方法，但是这一步通常是根据数据的具体情况而明确选择的。特征归一化常用的方法包含如下几种：

•简单缩放

•逐样本均值消减(也称为移除直流分量)

•特征标准化(使数据集中所有特征都具有零均值和单位方差)

简单缩放

在简单缩放中，我们的目的是通过对数据的每一个维度的值进行重新调节（这些维度可能是相互独立的），使得最终的数据向量落在 [0,1]或[－1,1] 的区间内（根据数据情况而定）。这对后续的处理十分重要，因为很多默认参数（如 PCA-白化中的 epsilon）都假定数据已被缩放到合理区间。

例子:在处理猫狗分类问题时，我们获得的像素值在 [0,255] 区间中，常用的处理是将这些像素值除以 255，使它们缩放到 [0,1] 中.

逐样本均值消减

如果你的数据是平稳的（即数据每一个维度的统计都服从相同分布），那么你可以考虑在每个样本上减去数据的统计平均值(逐样本计算)。

例子：对于猫狗分类这个问题，这种归一化可以移除图像的平均亮度值 (intensity)。很多情况下我们对图像的照度并不感兴趣，而更多地关注其内容，这时对每个数据点移除像素的均值是有意义的。注意：虽然该方法广泛地应用于图像，但在处理彩色图像时需要格外小心，具体来说，是因为不同色彩通道中的像素并不都存在平稳特性。

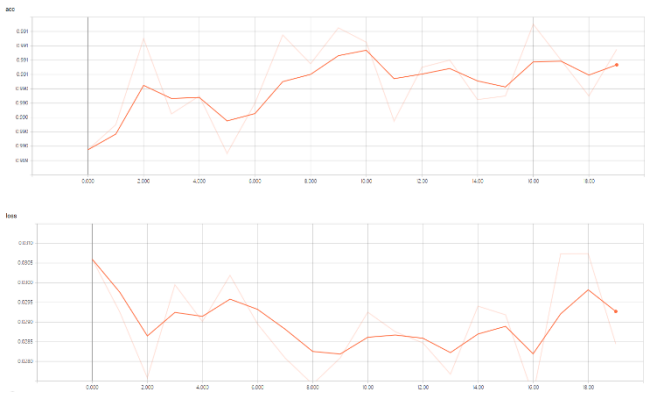
特征标准化

特征标准化指的是（独立地）使得数据的每一个维度具有零均值和单位方差。这是归一化中最常见的方法并被广泛地使用(例如,在使用支持向量机(SVM)时，特征标准化常被建议用作预处理的一部分)。在实际应用中，特征标准化的具体做法是：首先计算每一个维度上数据的均值（使用全体数据计算），之后在每一个维度上都减去该均值。下一步便是在数据的每一维度上除以该维度上数据的标准差。

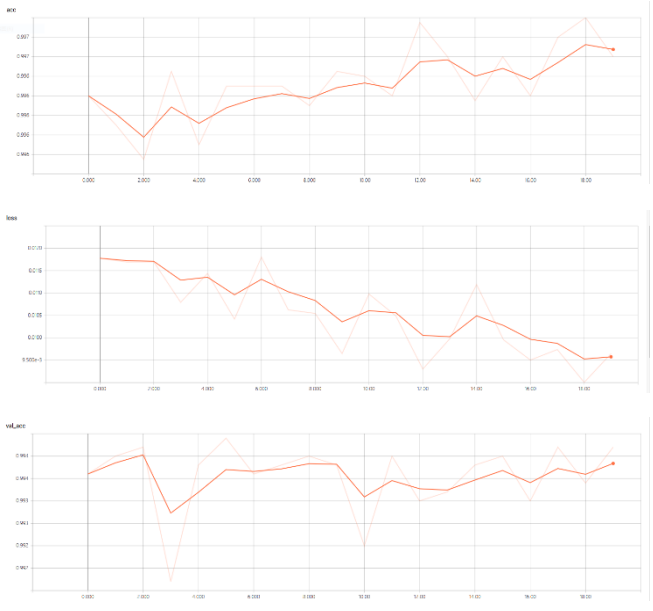
例子:在 InceptionResNetV2 等网络中，大量的使用了 Batch Normilization，对特征的每个分量独立地使用标准化处理。

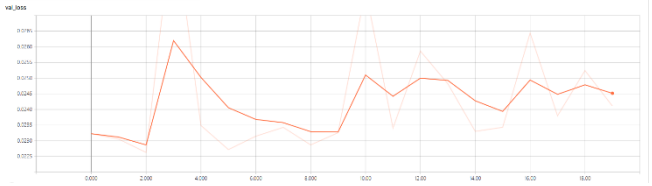
8.Implementation

最初是使用单个 InceptionResNetV2 进行迁移学习，全连接网络直接由模型输出加一个 sigmoid 单元,没有使用 Dropout.并且在学习的过程中，由于没有进行特征提取，整个模型都需要训练，训练时间有将近一小时左右，训练效果如下图所示：



在这次的训练过程中训练的 accuracy 和 loss 都还不错，不过测试集在 kaggle 上进行测试，测试集的 loss 为 0.41。之后查阅了相关资料，发现 logloss 这个函数对于预测错误的样本，它的损失非常大。所以在提交测试之前，预测这里用到了一个小技巧，将每个预测值限制到了 [0.005, 0.995] 个区间内。因为对于 logloss 来说，预测正确的样本，0.995 和 1 相差无几，但是对于预测错误的样本，0 和 0.005 的差距非常大，是 15 和 2 的差别。之后再次用到这个技巧后提交，loss 为 0.05775。这样的一个结果，已经满足了本项目的最低要求： kaggle Public Leaderboard 前 10%。此次的训练波形如下图所示：





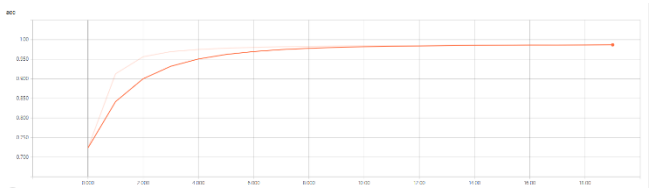
其中，前两张图为训练集的 accuracy 与 loss,后两张图片为测试集上的 accuracy 与 loss。一点小小的改动效果相当明显，这说明有时不仅需要对模型进行改进，在其他一些方面也要做出努力，可能会取得意想不到的效果。

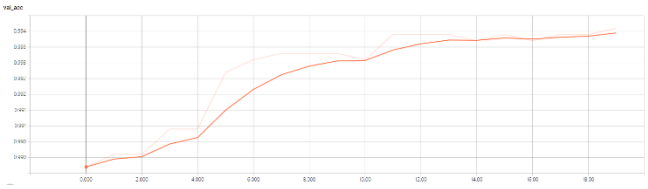
在这个训练结果中，train accuracy 没有降低，反而升高了，并且 val accuracy 与 val loss 都没有降低。这主要是由于最后的全连接层参数太少，所以说在这次训练中，最后的 10epoch 训练，对这个融合模型基本没什么影响。主要是迁移学习的源网络的预训练权重对分类的准确度产生了很大的影响，之后的训练集只能说是对输出的全连接网络的权重进行了微调。从这一点上又可以看出，在小数据集上进行训练，使用迁移学习的效果会远远好于对一个模型进行重新训练。

因为数据集较小，重新开始进行训练的话，模型容量的选择是一个很难抉择的问题。模型容量太小，难以对该问题进行合理的表达；模型容量太大的话，又没有那么多的数据进行训练，如果强行用大容量的模型进行训练的话，在训练集上很容易过拟合。从而测试集的表现也会非常差。

总之，迁移学习在小数据集上不失为一个很好的方法。不需要多少训练，进行特征提取后训练的速度也会非常快。而且训练的网络表现也不差。

之后再次对模型进行改进，使用 InceptionResNetV2，Xception， InceptionV3 进行特征的提取，将提取的特征融合后，作为输入进行训练。加入了 2 层全连接层(融合模型输出->1024->1)与 Dropout 层(Dropout 为 0.5)，在训练过程中发现 train accuracy 一直在震荡，并且 loss 也不再下降，之后采用固定学习率的 Adadelta，lr = 0.005。改进后，模型的 train accuracy 不断趋于 1，模型的 loss 也有明显的减小，如下图所示：





最后提交到 kaggle 的分数为 0.3722，取得了第七名的成绩。

最后进行了预测结果的提交，最优的 loss 为 0.03722。

9. Clonclusion

在这次项目中，有些算法还没有用到，如数据增强，模型微调，检测异常值,进行更细致的数据处理等等。使用了这些算法后，我相信 loss 还会有进一步的降低。另外,还可以使用一些最新提出的效果更好的模型,如 DenseNet,CVPR2017BestPaper, https://github.com/liuzhuang13/, DenseNetSENet,ImageNet2017 冠军模型 https://github.com/hujie-frank/SENet/. 使用这些模型融合后,我相信我的模型会更加接近 top1%.

References

[1] Dogs vs. Cats Redux: Kernels Edition: www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition

[2] Stanford: Deep learning and computer vision Class. http://study.163.com/course/courseMain.htm?courseId=1003223001

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. https://arxiv.org/abs/1512.03385, 2015. 12,10

[4] S. Xie, R. Girshick, P. Dollar, Z. Tu, Kaiming He Aggregated Residual Transformations for Deep Neural Network. https://arxiv.org/abs/1611.05431. 2017. 04,11

[5] Ian Goodfellow, Yoshua Bengio, Aaron Courville: Deeplearning Book. www.deeplearningbook.org

[6] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, Xiaoou Tang: Residual Attention Network for Image Classification. https://arxiv.org/abs/1704.06904. 2017. 04,23

[7]Andrew Ng: Deepleanring.ai. https://mooc.study.163.com/smartSpec/detail/1001319001.htm

[8] https://github.com/ypwhs/dogs_vs_cats

[9]Model Ensembling:
https://mlwave.com/kaggle-ensembling-guide/
[10]Sinno Jialin Pan, Qiang Yang :Transfer Learning.
https://link.zhihu.com/?target=https%3A//www.cse.ust.hk/~qyang/Docs/2009/tkde_transfer_learning.pdf
[11]ShiKun Liu: Transfer Learning.
https://www.zhihu.com/question/41979241
[12]迁移学习手册:
http://jd92.wang/assets/files/transfer_learning_tutorial_wjd.pdf
[13]数据预处理：
http://ufldl.stanford.edu/wiki/index.php/数据预处理#.E5.9F.BA.E4.BA.8E.E6.AD.A3.E4.BA.A4.E5.8C.96ICA.E7.9A.84.E6.A8.A1.E5.9E.8B
[14]keras 预训练模型：https://keras.io/applications/