数字图像处理 - 第二次大作业

运行环境

Python3，tensorflow, sklearn, metric-learn

1. 问题重述

一般情况下，我们在使用神经网络进行对图像分类问题的训练时，都有充足的训练集。当我们面对一类特殊的情况：我们掌握了大量已知类别的训练集，并且已经针对这些数据训练出了神经网络。如果我们遇到一些新的只有少量训练集的新类别，我们已经不能用原来的方法进行神经网络的训练了。那么我们应该如何处理这样的情况？实际上，这种情形更加接近于人类认知事物的过程。

一般来说，我们可以提取出新类别的特征，再根据已知类别中的包含这些特征的近似类别进行权重分析，通过已有类别来对新类别进行分类；另一方面，我们也可以学习特征空间的某种度量，使得在这种度量下，每种类别之间的距离都很接近，从而可以进行这种度量下的聚类。

2. 完成情况

在原实验建议的范围内，我们完成了：

- 跑通 AlexNet 已有代码，并提取 AlexNet 中 fc7 层的基类特征
- 使用 Logistic Regression，SVM, KNN, 决策树, 贝叶斯, MLP 等方法对 fc7 层的特征进行训练

在原有要求以外，我们完成了：

- Vaner 模型的学习和实现
- Prototypical 模型的学习和实现
- 基于度量学习的PrototypicalNetwork
- 基于线性空间正交投影方法的Network Embedded Regression
- 基于限制特征分布的VANER

3. 主流思路的调研与分析

3.1. 基本思路

从本质上来看，Alexnet = 提取feature（fc7的输出) + 分类器（fc8的输出）

当新增加类别时，feature还可以一样地提，但是问题来了——分类器没办法分出新的类。怎么办？

有两种基本的思路：

1. **学习一种对feature的度量**，这种度量可以对同类型feature得到较相近的结果，而对不同类型的feature得到较远的结果。从而，对于一种新的类别的feature，利用这种度量进行区分即可。
2. 对feature的分布做再优化，即**控制feature的分布情况**，使得这种分布可以被已知的度量函数进行度量。

3.2. PrototypicalNetwork

3.2.1. 直观思路理解

由于一开始训练的feature依赖于分类器，分类器可能做了聚类，因此无法保证一开始得到的feature是"聚类"的，因此需要手动保证聚类——在feature上再去embedding。

因此，学一个映射，从feature到embedding vector，保证同一类得到的vector是接近的。

- 此处，对于距离函数的定义为欧几里得函数。

从而，对于新的图片，我也假设经过这样的网络后它的vector也是聚类的。

- 这种方法假定了一开始学习出来的embedding是对所有图片都通用的，但这并不一定正确。因此，添加新图片后，可能不满足这一假设。

3.2.2. 数学本质

本质上是学一个embedding，使得经过这种embedding后的feature服从指数分布族。

有定理表明（见论文中的引用），对于一个指数分布族，可以被某个**Bregman divergence**距离函数刻画其分布。而对于k个服从指数分布族的分布，某一个新的变量z属于其中某个分布的概率，只和**它与这种分布的期望点的距离**有关。

因此，只要我们保证某种embedding使得同一类的feature服从指数分布族，则只需要学出：

1. 这一类分布的期望点（均值点）-> 可以用样本均值刻画。（大数定律）
2. 某种满足Bregman divergence条件的距离函数。
3. 保证样本所有变量满足自身服从当前分布的概率最大，即在这种距离函数度量下，与均值最接近。

就可以学出一种分类器来。

而对于prototypicalNetwork，反了过来，学的是参数分布：

- 如果我们固定某种满足Bregman divergence条件的距离函数，只要学出样本均值，就可以刻画出一种指数分布族。然后，我们要求网络输出的embedding尽可能保证自身服从当前分布的概率最大，即限制了每一种分布的方差。但这里**假定了feature服从由这种距离函数刻画的指数分布族**。

3.2.3. 可能改进的点

1. **修改人为度量函数**：更换不同的Bregman divergence距离函数去学习feature embedding，看哪种效果更好。

   原论文用的欧式距离时，分类器是线性分类器，但还可以换成别的比如马氏距离。

2. **学习度量函数**：学一种Bregman divergence距离函数。即：

   输入两个向量，输出它们的距离。并且这个输出还需要额外保证样本均值与各个样本之间的距离最小（因为Bregman divergence的定义）。（loss是样本均值和各个样本之间的距离）

3.3. Relation Network

3.3.1. 直观思路理解

样本个数很小时，提取出feature后，无法用来很好地定义一个分类器。为此，定义一个"比较器"，单纯比较两个feature是否相近。

此处"比较器"本质上即为一种新的度量方式（可以类比欧几里得距离）。

因此，相对于PrototypicalNetwork的“先embedding再求欧氏距离”来保证“同一类接近、不同类分离”，此篇论文把这一做法通用化，直接学习了一种度量方式，使得不同feature经过这种度量后之间直接得到了符合“同一类接近，不同类分离”的特性。

3.3.2. 可能改进方向

然而，尽管这种方法是通用的学习，但由于卷积网络的局限性，很可能学不到一些很高级的度量。这也加大了对网络结构的依赖，而且全是未知的。

因此，这种方法的瓶颈在于对“分类器”网络的结构的调整。

对于我们作业中的问题，无法仿照这篇论文的思路进行，因为fc7.npy是全连接层的输出，而原论文中是卷积层网络的输出。全连接层的输出无法再接卷积层，从而我们问题中的分类器网络不可以使用卷积层，极大地限制了网络的功能。

3.4. VANER

3.4.1. 直观思路理解

由于人类对新事物的认识是先联想已经知道的东西，后加独特特征，因此对新图片的feature提取可以分为2步：

1. 根据先验知识，推断在先验知识空间下它的feature
2. 根据已有的feature提取器，提取它本身独特的特征

然后将这两个特征混合起来，作为它的总和特征。

更抽象地，我们可以把人学习的过程看做：

1. 提取feature，F
2. 将F投影到自己曾经见过的feature域，分量为$F_{//}$
3. 在这个分量上，找到它应该对应的分类器参数
4. 再将这个分类器参数进行微调。

我们假定神经网络提取出的feature已经较为准确（即可以通过某种分类器很好地进行分类）。因此，可以假定：**在feature域中，不同种类的图片的feature可以张成一个feature域的子空间。**即：神经网络将高维图片降维到了feature域，并且每种图片都在feature域的子空间。

基于上述假设，我们可以假定，存在一种映射f，可以将feature空间映射到分类器参数空间，即假定feature已经提取得十分合理。

事实上，对于fc8层而言，本质是与fc7的特征进行矩阵相乘。对于同一类而言，它们的乘积会尽可能大；而对于不同类而言，它们乘积的结果会尽可能地小；从而从某种意义上，fc8这层分类器恰好是fc7特征的对偶空间。

3.4.2. 数学本质

根据1000类中每一类，提取feature的平均值，作为这一类的平均feature向量。这样，得到了1000个点。

（需要注意的是，这里对feature提取平均值时并没有假定“同一类接近，不同类分离”，因此可能有问题）

接着，对于这1000个高维空间的点，我们**假定这些点局限在高维空间的一个低维子空间中。**直观意义上看，我们假定的是，base class里的点并不能充满整个feature空间，新图属于的空间无法被base class张成。（这是非常合理的，因为feature域有4096维，而base class只有1000类，它们的秩最多只有1000）

换句话说，由于feature空间内还存在别的新class，因此光靠base class无法表示整个空间，因此base class的feature张成的空间必定是feature空间的一个真子空间。

我们构造一个从feature空间到子空间的投影V，假定V保持子空间内积，而不管子空间的补空间。

而当增加新class时，再去限制：增加新class后的子空间仍然在这个映射下保持内积，从而求出v_{new}，即新class在子空间中的映射。

本质上，可以将新class的feature在子空间和它的补空间进行分解，新class的feature在子空间上的分量才满足保持内积的特性，因此论文中求出的v_{new}本质上是新class的feature在子空间的投影。

3.4.3. 改进

3.4.3.1. 改进1

由于论文中通过机器学习的方法学了一种投影V，而在线性空间中的正交投影能满足论文中对V的所有限制，因此我们可以直接用数值计算的方法，构造一个线性空间的正交投影V。对于T，只需要用最小二乘法求得方程VT=W的解即可。

具体地，我们定义一个正交投影V，使得feature空间投影到base class的feature张成的子空间中。从数学角度而言，本质上要保证：

1. base class中的feature之间的内积在这个投影下恒定不变（论文已保证，即$||A - VV^T||$要尽可能小）
2. 新class的feature向量中，与base class正交的部分，在这个投影下投影成了0.（论文未保证）

接着，再利用最小二乘法定义一个映射T，保证在T的作用下，这个子空间内的点可以被投影到分类器参数空间内对应的点。即最小化$||VT - W||$。

其实，本质上而言，如果想定义一个这样的正交投影映射，可以这样定义：将原feature空间的所有列向量进行施密特正交化，得到一个标准正交向量组，再找它们的正交补空间，从而直接可以构造对应的映射向量，并且满足上面两条性质。

至此，我们得到了正交投影V，以及最小二乘法得到的T。

而对于一张新的图片而言，我们可以首先得到它的feature向量，接着求出它在子空间上的投影，然后用T算出分类器的参数，即可得到w_{trans}。而w_{model}得到的方法与原论文一致。

3.4.3.2. 改进2.

论文中的方法有两个明显的问题：

- 在得到每一类别的feature时，采取了平均所有feature的办法。但这一做法有个前提条件，需要保证alexnet的fc7输出的feature是线性空间上比较“聚合”的点，

  也就是要保证同一类的点距离较近、不同类的点距离较远。

  更进一步，本质上讲，这里求平均feature的操作用的是线性空间R^{4096}上的加法，但是fc7输出的feature可能并不是线性空间，因此这种求平均的方法可能会丧失同一类feature的特性。
- 在求出feature到embedding空间的像时，loss函数保证这个映射具有“保留内积”（余弦函数可以看成单位向量的内积）的特性。但这个内积依然是线性空间的内积，并不一定适用feature所在的某个特殊的空间。

正因为我们不知道fc7层的分布函数，所以我们人为定义的度量（论文中是余弦相似度）都是不准确的。因此，我们可以有两种可能的方向：

1. 学习一种距离度量，可以准确刻画fc7 feature中分布的距离。
2. 对fc7的分布进行限制，从而可以用已知的度量方式进行度量。

4. 模型概述

基于上面对已有主流思想的调研与分析，我们提出了以下的创新模型：

4.1. 基于度量学习的PrototypicalNetwork

4.1.1. 模型思想

由于PrototypicalNetwork中使用的Bregman divergence函数是欧氏距离函数，而实际上，对于两种向量，我们可以通过深度学习的方法学习一种度量，使得这种度量满足Bregman divergence的条件，并且符合fc7的feature域的指数分布族。

由于原论文是在假定了度量函数后去限制指数分布族，这样可能对参数空间的限制不准确。但我们不需要限制指数分布族的参数，而是让度量去尽可能适应已有的一种指数分布族，从而理论上可以得到更精确的结果。

4.1.2. 实现结果

由于我们得到的fc7的特征是向量，无法通过卷积层的计算，因此我们只是简单地使用MLP进行计算。

对于loss的限制，一方面是需要使用cross entropy保证分类结果的正确，另一方面需要**限制度量函数是Bregman divergence**，即：

对于任意随机变量X，度量函数$d(x,y)$满足：

$$\min_y E(d(X,y)) = E(X)$$

对于样本的学习，我们使用样本均值代替期望，用样本中所有数据与均值点的距离的平均值来代替$E(d(X,y))$，从而只需要使均值点在$E(d(X,y))$的计算中最小即可。

然而，由于网络结构的限制，使得度量函数刻画地十分不精确，最终这种方法并不能很好地收敛。如果我们可以利用fc7之前的特征，或许这种方法是一种十分有效果的方法。

4.2. 基于线性空间正交投影方法的Network Embedded Regression

4.2.1. 基本思路

由于论文中通过机器学习的方法学了一种投影V，而在线性空间中的正交投影能满足论文中对V的所有限制，因此我们可以直接用数值计算的方法，构造一个线性空间的正交投影V。对于T，只需要用最小二乘法求得方程VT=W的解即可。

具体地，我们定义一个正交投影V，使得feature空间投影到base class的feature张成的子空间中。从数学角度而言，本质上要保证：

1. base class中的feature之间的内积在这个投影下恒定不变（论文已保证，即$||A - VV^T||$要尽可能小）
2. 新class的feature向量中，与base class正交的部分，在这个投影下投影成了0.（论文未保证）

接着，再利用最小二乘法定义一个映射T，保证在T的作用下，这个子空间内的点可以被投影到分类器参数空间内对应的点。即最小化$||VT - W||$。

其实，本质上而言，如果想定义一个这样的正交投影映射，可以这样定义：将原feature空间的所有列向量进行施密特正交化，得到一个标准正交向量组，再找它们的正交补空间，从而直接可以构造对应的映射向量，并且满足上面两条性质。

至此，我们得到了正交投影V，以及最小二乘法得到的T。

而对于一张新的图片而言，我们可以首先得到它的feature向量，接着求出它在子空间上的投影，然后用T算出分类器的参数，即可得到w_{trans}。而w_{model}得到的方法与原论文一致。

4.2.2. 算法实现

首先，我们采用朴素的 softmax-regression和500张新类，对fc7层输出feature向量和最终label训练出一个基础的50类分类器，其参数称

$$W_{model}(shape = 4096 * 50)$$

同样，我们采用朴素的 softmax-regression和6w张旧类，对fc7层输出feature向量和最终label训练出一个对旧类分类器，其参数称

$$W_{base}(shape = 4096 * 1000)$$

对于每一个新类，求出其在fc7层中平均特征向量，并归一化

$$feature_{novel-average}(shape = 50 * 4096)$$

对于每一个旧类，求出其在fc7层中平均特征向量，并归一化

$$feature_{base-average}(shape = 1000 * 4096)$$

我们发现，在一定精度范围中，$feature_{base-average}$中向量线性不相关，所以对于每一个新类，其对应的$feature_{novel-average}[i]$, 可以求其在$feature_{base-average}$中各向量组成的线性空间中的投影，并将投影用1000维坐标表示为旧类feature的线性组合，$V[i](shape = 1000)$使得$feature_{novel-average}[i] = V[i] * feature_{base-average}$

组成矩阵$V(shape = 50 * 1000)$。对于每一个新类，我们可以由旧类的分类器参数，求出一对应的分类器参数，具体方法为：将旧类的分类器$W_{base}(shape = 4096 * 1000)$依$V[i]$求线性组合，得到

$$W_{novel}[i] = V[i] * W_{base}^T$$

组成矩阵

$$W_{novel} = V * W_{base}^T$$

最终，得到两模型（新经验，旧经验）组合成的分类器参数

$$W = W_{model} * \alpha + W_{base} * \beta$$

4.3. 基于限制特征分布的VANER

4.3.1. 基本思路

由于VANER对输入特征进行了一些假定（近似满足线性空间），但fc7的输出并不一定符合这个假设，因此可能会有问题。基于这个问题，我们对模型进行修改，结合PrototypicalNetwork的思想，首先对feature进行限制：

- 将fc7的输出接一个MLP，其中loss与PrototypicalNetwork的loss完全一致，通过预先训练好这个网络，我们可以得到一种新的feature，这种feature在欧式距离的度量下，每一类与均值点的距离最近，而与其他类别的均值点较远。因此，这一步相当于对fc7的feature进行变换，得到符合欧式空间度量的feature。
- 接着，将这种feature作为VANER的输入，从而VANER可以使用余弦距离度量来刻画feature之间的相似性，从而可以更高地提高VANER的准确度。

4.3.2. 二分类器

我们二分类器选用了1000个二分类器，每个二分类器c_i是一个全连接层，输入是特征向量，输出是一个2维向量，分别表示这个特征向量属于/不属于第i类的权重。

4.3.4. VANER模型

在 Vaner 模型中，我们考虑把 W 替换成 V，T 矩阵的乘积，其中 $V \in \mathbb{R}^{n\times q}$ 矩阵是把基类进行降维的矩阵，$T \in \mathbb{R}^{q\times p}$ 是对降维后的向量进行的线性变化。具体地，$n = 1000$ 表示基类的个数；q 是降维后的向量维数，实践中在 $q = 600$ 时效果最好；$p = 4096$ 表示网络 fc7 层中提取出的向量的个数。

为了训练 V, T 两个矩阵，我们还需要定义一个合理的损失函数。令 A 表示基类向量间的**余弦距离**(cosine distance)邻接矩阵，即

$$A_{i,j} = \frac{\overline{x_i^B} \cdot \overline{x_j^B}}{||\overline{x_i^B}||_2 \cdot ||\overline{x_j^B}||_2}$$

其中，$\overline{x_i^B}$ 代表基类 i 的特征向量的平均值（这实际上并不合理），对于 V, T 矩阵的优劣，我们认为 V, T 矩阵一方面要保证 V, T 乘积和原映射 W 的相似性，同时也要保证降维过程保留原有特征，所以我们有以下损失函数：

$$\mathscr{L}(V, T) = ||VT - W||_F^2 + \lambda||A - VV^T||_F^2$$

其中，λ 是参数，在实践过程中，由于损失函数 $\mathscr{L}(V, T)$ 可以降到非常小的值，所以 λ 的取值并不关键。得到合理的 V, T 以后，我们每加入一个新类相当于将损失函数拓展为

$$\mathscr{L}(v_{new}) = ||\begin{bmatrix} A & a_{new}^T \\ a_{new} & 1 \end{bmatrix} - \begin{bmatrix} V \\ v_{new} \end{bmatrix}\begin{bmatrix} V^T & v_{new}^T \end{bmatrix}||_F^2$$

可以证明，当 $\mathscr{L}$ 取最小值时，

$$v_{new} = a_{new}(V^T)^+,$$
$$M^+ = (M^T M)^{-1} M^T$$

其中，a_{new} 表示新类别的特征向量，M^+ 是矩阵的伪逆，定义如上，最终的映射矩阵 w_{new}^N 为

$$w_{new}^N = v_{new} T$$

结合实际情况，我们在产生结果时利用了集成学习的思路，将原矩阵 W 和新矩阵 w_{new}^N 得到的结果进行了voting，提升了结果准确率。

4.3.3. 实现结果

由于PrototypicalNetwork的准确率只有63%左右，由它提出的feature并不能很好地对novel class符合"均值点最近"的特性，从而最终结果与标准VANER的结果相差不大。

5. 实际运行情况

在运行我们的模型之前，我们先对目前已有的主要的算法进行了测试，以确定baseline。然而，最终我们发现，通过baseline算法得到的效果却更加出色。为了准确测试我们的算法，我们将Caltech256的原数据集进行了下载，并通过上面的标注进行了测试（没有使用其中的任何数据用来训练）。下面是不同baseline算法所能获得的最佳效果：

Baseline算法	准确率
DecisionTree	0.282
LinearRegression	0.286
Prototypical Network	0.674
KNN	0.681
Bayes	0.688
Alexnet fineTune	0.710
SVM	0.728
SoftmaxRegression	0.738

另外一方面，我们使用了数据增强的手段进一步提高准确率。对于这次实验而言，最大的瓶颈实际上在于输入数据的不足。为了弥补这一缺失，我们不得已用变换的方式来补充数据。我们采用的是公开的库imgaug，它提供了一系列不同种类的图像变换方法，而在经过多次尝试后，我们最后选择了其中的旋转、放缩、切变、翻转对比度改变以及亮度改变这六种。对于每一张输入图片，我们都相应生成了三张变换后的图片，将每类的10张图片变为了40张，对于效果提升有着很大的帮助（经测试SoftmaxRegression的准确率提升了0.01左右）。

相比于Baseline的算法，我们事先设计的模型因为过于复杂还没能完全完成，目前所能取得的最好成绩为0.72左右。正因如此，我们最终采用的是将Baseline的算法和我们的算法进行合并Voting的方式获得最终成绩，并最终将准确率提升到了0.74。