EasyChair Preprint

№ 1092

Research on Cross-model of Transfer Learning based on Deep Learning

Bozhao Guo

June 5, 2019

基于深度学习的跨模态迁移学习研究

摘 要

迁移学习是一种新的机器学习方法，它主要利用已经有标注的数据集进行训练得到模型，从而对不同但是相关的问题进行求解。大数据时代数据规模的急速扩张导致了越来越严重的统计异构和标注缺失问题。标注数据缺失会导致传统监督学习出现严重过拟合问题。本文的研究工作如下：

1）在联合结构嵌入模型（Structured Joint Embeddings）的基础上构建了潜在嵌入模型 LatEm。尝试用分段线性的方法来代替 SJE 的线性映射。模型主要的思想是用高维语义特征代替图片的低维特征，来进行分类器的学习，从而使得训练出来的模型具有迁移性。LatEm 是一种跨模态的方法，它使用通过人工注释或从大文本语料库中以无监督的方式收集的图像和类别的信息。

2）针对 LatEm 中单个样本包含多个映射矩阵的情况。本文对每个矩阵所对应的损失函数进行排序，使用随机梯度向下和共轭梯度法来求得最优解。

3）在 MATLAB 上实现了 LatEm 模型，并在 AWA，CUB，Dogs 三个数据集上与 SJE 得到的结果进行了对比。在 CUB 和 Dogs 这两个细粒度的数据集上的无标注的分类精确度分别达到了 52.3%和 24.5%。

关键词：迁移学习，多模态，零样本学习，细粒度分类

Research on Cross-model of Transfer Learning based on Deep Learning

Abstract

Transfer learning is a new machine learning method. It mainly uses the already labeled data set to train the model to solve different but related problems. The rapid expansion of data size has led to more and more serious problems of statistical heterogeneity and labeling. The lack of annotation data can lead to serious over-fitting problems in traditional supervised learning. The research work of this paper is as follows:

1) Based on SJE, the latent embeding model LatEm is built. Try to replace the SJE linear map with a piecewise linear approach. The main idea of the model is to replace the low-dimensional features of the image with high-dimensional semantic features to learn the classifier, so that the trained model has mobility. LatEm is a cross-modal approach: it uses information from images and categories that are collected in an unsupervised manner by manual annotation or from a large text corpus.

2) For the case where a single sample in LatEm contains multiple mapping matrices. We sort the loss functions corresponding to each matrix and use the gradient down method to find the optimal solution.

3) The LatEm model was implemented on MATLAB and compared with the results obtained by SJE on the three data sets AWA, CUB and Dogs. The unlabeled classification accuracy on the two fine-grained data sets CUB and Dogs reached 52.3% and 24.5%, respectively.

Key words:transfer learning,cross-model,zero-shot learning, Fine-grained classification

目　　录

1 绪论

1.1 课题背景及目的

在人工智能领域的研究中，机器学习有着举足轻重的地位，因为这一课题是使得计算机具有智能的可能途径。它综合了概率论，统计学，逼近论等多门学科，着眼于如何用计算机来模拟人类的学习行为，使计算机能够学习新的知识和技能，并且让计算机自己能够随着知识的不断输入来重组自身的知识结构，提高自身性能。在 2010 和 2011 年连续两年的图灵奖都授予了机器学习领域的杰出学者，这也意味着在经历了三十多年的不断发展之后，机器学习已经成为计算机科学领域最重要和最活跃的研究分支之一。

所谓迁移学习，从广义上说，是一种学习对另一种学习的影响。迁移这个行为往往发生在我们将旧知识与新知识联系起来的过程中，这不过对于我们大多数人而言这是一个无意识的过程。在学习者学习新的知识时，其自身已经习得的知识、经验、态度等就会潜移默化地对这个过程产生影响。总的来说，学习和迁移这两个过程是相辅相成的，迁移是学习的继续和巩固，又是提高和深化学习的条件。具体到深度学习领域上来说，迁移学习是指利用已经训练好的模型参数（类比我们已经学习到的知识），按照一定的方式（这个过程就被称作迁移）进行整理，在另一个新的数据集上进行训练（类比学习新的知识）。

根据维基百科的定义，迁移学习是一种新的机器学习方法，它主要利用已经有标注的数据集进行训练得到模型，从而对不同但是相关的问题进行求解。传统的机器学习通常基于一个基本假设：训练集合测试集之间是满足独立同分布的关系的。并且传统的机器学习需要使用足够多的训练样本以获得好的效果。但是，在实际应用中常常会发现，想要满足这两个条件并非易事。首先，相当多数据是具有时效性的，在经过一段时间之后，原先已经被标注的样本数据不再准确，以此为基础进行训练得到的结果的准确率自然也令人怀疑。有些数据集很容易过期，换句话说不同时期的数据分布也会不同。比如对于某个用户进行室内 wifi 定位的时候，把他在一个很大的室内的数据标记好已经很难了，wifi 信号强弱还会受到时间影响，所以如果对于每个时间段都要进行一次训练那就会变得更加麻烦。除此之外，已经被标注的数据要远远少于未被标注的数据，想要获取这些数据存在不小的困难。比如在 Web 数据挖掘领域，随着新数据的不断出现，根据已有的训练样本所训练出来的分类模型就会不再可靠。但是对如此大量的数据进行标注

也是枯燥而且困难的，更糟糕的是由于数据量过大，人工标注时出错也在所难免。因此迁移学习不要求目标领域有大量已被标注的数据的特点就正好可以解决这些问题。

迁移学习的理论研究价值在于解决标注数据稀缺性和非平稳泛化误差分享这两个方面。前者来源于大数据时代数据规模的急速扩张所导致的越来越严重的统计异构和标注缺失问题。在有监督学习的情况下，如果数据集缺乏足够多正确的标签，不仅精确度令人堪忧，还会带来不可忽视的过拟合的问题。虽然在传统的机器学习领域中也有一些方法都可以有效解决标注数据缺失的问题，但是这些方法也或多或少地依赖于目标域中存在一定数量的已被标注的数据。而在这种情况下，为了额外获取人工标注的数据所要付出的代价太大，因此这个时候需要迁移学习来辅助提高目标领域的学习效果。后者则来源于经典统计学习理论给出了独立同分布条件下模型的泛化误差上界保证。所以如果不同数据域的各个样本数据之间不再服从于同一个隐含未知的分布，经典机器学习理论在这里就不适用了。当不同数据域的各个样本数据不再服从于同一个隐含的分布时。不同构的数据分析问题的研究就会出现理论上的漏洞。而使用迁移学习就可以在这种非平稳环境下规避这种风险。换句话说，在非平稳环境下，迁移学习从理论上很好的补正了传统的机器学习过程。

1.2 国内外研究现状

定义 D 为由 d 维特征空间 X 和边缘概率分布 P(x) 组成的领域，即$\mathrm{D}=\{X；P(x)\},x\in X$。给定领域 D，任务 T 定义为由类别空间 Y 和预测模型f(x)组成，即$\mathrm{T}=\{Y；f(x)\},y\in Y$，按统计观点预测模型$\mathrm{f(x)}=\mathrm{P}(y|x)$,解释为条件概率分布。

迁移学习的定义：给定一个基于数据D_t的学习任务T_t，我们可以从D_s中获取对任务T_s有用的知识。迁移学习需要找到D_s和T_s的潜在关联来提高任务T_t的预测函数的表现，其中$D_s\neq D_t$且/或$T_s\neq T_t$。 D_s的规模通常远大于D_t的规模。

1.2.1 迁移学习的分类

按特征空间、类别空间、边缘分布、条件分布等问题因素在辅助领域和目标领域间的关系，我们可以大致地将迁移学习划分为异构迁移学习和同构迁移学习两种[1,2]。

根据领域间特征空间和类别空间的是否相同，边缘概率分布和条件概率分布相同的迁移学习可以进一步被划分为异构特征空间和异构类别空间两种子类型。异构特征空间

指的是辅助领域和目标领域位于不同特征空间（即$X_{s\neq}X_t$）。典型的应用是跨语言文本分类和检索，其中训练数据和测试数据来自不同语言类型。异构类别空间指的是辅助领域和目标领域的类别空间不一致（即$Y_{s\neq}Y_t$），在文本挖掘和图像理解中受到广泛关注。

如果要在辅助领域和目标领域的类别空间不一致的条件下进行迁移学习，往往需要使用领域内的一些特定经验，例如特征空间之间的关联关系（比如建立跨语种的数据库）、数据的多个模态之间是否存在联系（如网页中的文本和图像）、或社交关联关系（如社交网络中同一个用户对同一个问题的文本和情感评价），等等。异构迁移学习在缺少领域内特定的先验知识的情况下就难以进行。为了与依赖于具有大量标注的数据样本的经典机器学习相区分，本文的主要研究内容是边缘概率分布和条件概率分布不同的迁移问题中所涉及的方法。

根据边缘概率分布和条件概率分布是否同时相同，同构迁移学习通常可以分为数据集偏移、领域适配、多任务学习三种类型。领域间的边缘概率分布和条件概率分布都不相同即 Ps(x)≠Pt(x) 且 Ps(y｜X)≠Pt(y｜X)的同构迁移学习称为数据集偏移，这个问题因为具有相当的挑战性，所以目前针对同构迁移学习的研究工作很少，常见的方法也只有实例权重法。领域间边缘概率分布不同而条件概率分布相同即 Ps(x)≠Pt(x) 的同构迁移学习称为领域适配，包括样本选择偏置和方差偏移等，是迁移学习中研究得最为充分的问题。满足领域间条件概率分布不同而边缘概率分布相同即 Ps(y｜X)≠Pt(y｜X)的同构迁移学习称为多任务学习，它通过同时学习多个任务、挖掘公共知识结构，完成知识在多个任务间的共享和迁移。

1.2.2 迁移学习的方法

本文所研究的零样本学习（zero-shot learning）的难点在于目标领域的数据样本完全没有被标记过。针对目标领域数据完全没有标签的情况，人们提出了实例权重法和特征表示法两种[3]。前者的原理在于通过调整辅助领域中已被标注的实例所占的权重来逐步提升目标领域中分布相对稠密的辅助领域所占的实例权重，通过这种方法，辅助领域和目标领域之间的数据分布就会更加接近。而后者的主要思想是通过原始数据的特征值来找到它的另外一种表示形式。这样一来我们就可以让辅助领域和目标领域之间的边缘概率分布更加接近。在一定条件下，也能够用和目标领域无关的从辅助领域提取出来的抽象特征来代替目标领域相关的具体特征。

在机器学习中，我们通常只知道辅助领域和目标领域的数据集，但是并不知道辅助领域和目标领域所分别对应的条件概率分布。除此之外，在经典的机器学习问题中想要直接估计概率密度通常是不可行的，这是因为传统机器学习中输入向量 x 的维度通常很高[4]。尽管对于实例权重法的依据的理论研究已经十分充分，实例权重法本身也更加容易推导出泛化误差上界，但是实例权重法通常只在领域间分布差异较小时有效，所以在解决自然语言处理或者图像语义分割这类问题时，我们通常采用基于特征表示的迁移学习。

1.3 论文研究内容

迁移学习的理论研究价值在于解决标注数据稀缺性和非平稳泛化误差分享这两个方面。

本文的主要研究内容如下：

1）在联合结构嵌入（Structured Joint Embeddings）的基础上构建了潜在嵌入模型（Latent Embeddings）。尝试用分段线性的方法来代替 SJE 的线性映射。模型主要的思想是用高维语义特征代替图片的低维特征，来进行分类器的学习，从而使得训练出来的模型具有迁移性。LatEm 是一种跨模态的方法：它使用通过人工注释或从大文本语料库中以无监督的方式收集的图像和类别的信息。

2）针对 LatEm 中单个样本包含多个映射矩阵的情况。本文对每个矩阵所对应的损失函数进行排序，使用随机梯度下降法和共轭梯度法来求得最优解。

3）在 MATLAB 上实现了 LatEm 模型，并在 AWA，CUB，Dogs 三个数据集上与 SJE 得到的结果进行了对比。在 CUB 和 Dogs 这两个细粒度的数据集上的无标注的分类精确度分别达到了 52.3%和 24.5%。

1.4 论文结构安排

第一章为绪论，本章首先介绍了迁移学习的相关概念，将其与传统的机器学习方法进行了比较，分析了迁移学习的理论研究价值。然后介绍了迁移学习的国内外研究现状，并从特征空间，类别空间，边缘分布和条件分布等因素上对常见的迁移学习进行了分类。最后简要概括了迁移学习的常见方法。

第二章为相关技术介绍，本章从原理以及相关应用的角度简要介绍了深度神经网络 CNN，联合结构嵌入（Structured Joint Embeddings）和常见的优化算法。为之后的两章

介绍模型的算法和实验提供了知识保障。

第三章主要介绍了基于 SJE 基础上的一种改进的模型 LatEm，这一章包括问题概述与分析，模型设计和算法流程。该章重点介绍了 LatEm 模型和理论依据，包括具体实现上的细节和调整参数的分析。

第四章为实验部分的介绍。主要介绍了模型的实验结果，并通过与其他方法的比较来证明潜在嵌入方法的优势，同时还展示了优化目标函数后得到的各种结果，表明了最后确定参数的方法。

最后为结论，本章在对前面四章进行总结的同时，对尚未实现的和想要实现的想法进行了展望，为进一步的工作明确计划。

2 相关技术

2.1 循环神经网络

2.1.1 卷积神经网络简介

使用卷积神经网络来解决图像分类问题是深度学习在计算机视觉领域的首次应用。局部连接、多层结构、池化操作和权值共享是 CNN 的四个主要特点。CNN 之所以能够代替手工设计的特征，是因为卷积神经网络具备多层非线性变换的能力，可以从数据中自动学习特征。与此同时，局部连接和多层结构的特征使它的表达能力和学习能力也非常强。卷积神经网络中结构深度的重要性已经被许多的研究实验所证明。举例来说，从发展趋势上看，从 AlexNet、VGG 到 GoogleNet、ResNet，它们在结构上的表现为深度越来越深[5,6,7,8]。为了更好的拟合目标函数来获取更好的分布特征，研究人员通过增加深度来提升网络的非线性特征。

卷积层的基本单元是神经元，每个特征面的局部区域则通过一组特征值来与神经元相连接。多个神经元组成一个特征面，而多个特征面组成卷积层。文献[10]中指出，对于卷积神经网络中每一个卷积层的神经元而言，卷积核的个数决定了下一层特征图的个数，输出特征面的大小则由卷积核大小，上一层特征图的尺寸和滑动步长共同决定[9,10]。

池化层和卷积层紧密相连，和卷积层一样，池化层也由多个特征面组成。但是池化层的每一个特征面和它上一层的一个特征面是唯一对应的，池化层的输入层就是卷积层，所以特征面的个数不会因此而改变。正因为如此，池化层的神经元也与其对应的卷积层的局部接收域相连，而不同的神经元的局部接收域之间是没有交集的。池化层之所以能够降低网络模型的计算量，是因为在池化层中进行了池化操作，从而减少了神经元的数量。

非线性激活函数是 CNN 能够学习到复杂特征的基础，它通过非线性映射来模拟人脑的非线性认知行为。

2.1.2 GoogleNet 简述

本文第四章实验部分直接使用的数据中有一部分是基于 GoogleNet 神经网络结构获得的 CNN 特征值。一般来说，为了获得高质量模型，最保险的做法就是增加模型的深度（层数）或者模型的宽度（层核或者神经元的数量），但是这种做法会导致新的问题。

一是参数过多，在这种情况下很容易出现过拟合的问题，在训练数据集有限的情况下这种问题会变得尤其突出。二是随着网络变大，计算复杂度也会变大，这会为应用带来困难。三是网络越深，那么梯度越往后传递越容易消失，这会令模型的优化变得困难。简而言之，更大的网络会更容易产生过拟合的现象，并使计算量增加。针对这些问题，GoogleNet 的解决方式是将全连接甚至一般的卷积都转化为稀疏连接。传统的神经网络使用了随机稀疏连接，而计算机在计算非均匀的稀疏数据时效率并不令人满意。GoogleNet 提出了名为 Inception 的模块化结构，其目的在于在保持神经网络结构的稀疏性的同时充分利用密集矩阵的高计算性能的优点[11]。依据是大量的文献表明可以将稀疏矩阵聚类为较为密集的子矩阵来提高计算性能。

2.2 联合结构嵌入（SJE）

联合结构嵌入（Structured Joint Embeddings）主要用于图像分类的零样本学习任务中，测试和训练图像分别属于两个不相交的集合。目标是在一个联合框架中利用输入和输出嵌入来学习嵌入中的兼容性[12]。

在输入空间 X 和结构化输出空间 Y 之间定义兼容性函数 F：X×Y→R。给定特定的输入嵌入，我们通过最大化 SJE 上的兼容性函数 F 来推导出预测，如下所示：

$$f(x;w) = \arg\max_{y \in Y} F(x,y;w)$$

参数向量 w 可以用一个 D×E 的矩阵 W 来表示，其中 D 是输入嵌入维度，E 是输出嵌入维度。显然可以看出，兼容性函数是双线性的：

$$F(x,y;w) = \theta(x)^T W \varphi(y))$$

定义这个函数的目的是对于见过或没见过的类别，衡量图像特征 θ (x)和语义表征 ϕ (y)之间的相容性。W 是所要学习的视觉-语义映射矩阵。

对于零样本学习任务，训练集和测试集是不相交的。因此，可以将 ϕ 固定到训练集的输出嵌入上来学习 W。作为预测，我们将测试图像映射到 W 上并查找对应于其中一个测试类的最接近的输出嵌入向量（使用 cosine 相似度来进行度量）。

SJE 模型的损失函数$l(x_n, y_n, y)$定义如下：

$$l(x_n, y_n, y) = \Delta(y_n, y) + \theta(x_n)^T W \varphi(y_n) - \theta(x_n)^T W \varphi(y)$$

SJE 模型使用随机梯度下降的方法来进行优化。这一部分的具体实现会在第三章第三节中进行分析。

2.3 常见的优化算法

在日常生活中，最优化问题随处可见，小到走哪条路去学校最近，大到国家对于个人所得税率的调整，这些问题其实都用到了最优化的思想。从数学的角度上看，最优化方法其实就是研究在约束确定的情况下，通过调整一些因素的值，使得另外一部分因素达到最优的方法。目前在机器学习领域，其一般方法也就是建立目标函数，添加约束，得到最优化解。比较常见的优化方法有梯度下降法，牛顿法和共轭梯度法等[13,14]。

2.3.1 梯度下降法

首先我们假设这样一个场景：一个人要从山顶走到山脚，目前不存在从山顶直接到山脚的路，所以他需要选择多条短的线路一点点往山下走。已知每次为了选择新的线路需要花费一定的时间，那么为了使从山顶到山脚所花费的时间最短，应该怎么做呢？一个很直接的想法是，从起点开始，每次出发都选择一条最陡峭的线路行动，在到达路的终点时继续选择一条新的最陡峭的路线来行动，直至到达山脚。这个行动的本身就正好模拟了梯度下降算法。

梯度下降法是一种非常常用的最优化方法，在机器学习中经常会使用它来递归地逼近最小偏差模型。梯度下降法也被称作迭代法，运算过程中针对目标函数求取最优解，当目标函数的二阶导数大于 0 时，可直接将由梯度下降法得出的解作为最优解使用。反之，当目标函数的二阶导数小于 0 时，不可直接调用梯度下降法得到的解。其主要的思想就是选择当前位置的最快下降方向作为搜索方向。梯度下降法的主要缺点在于当逼近极小值的时候收敛速度会明显减慢，原因在于每一步的步长明显变短了，此时如果利用梯度下降的方法进行求解的话就需要进行很多次的迭代迭代[15,16]。

梯度下降方法有三种不同的形式，下面我们使用线性回归算法来对这三种方法进行比较。首先定义线性回归函数的假设函数h_θ和它对应的损失函数$j(\theta)$如下：

$$h_\theta = \sum_{j=0}^{n} \theta_j x_j$$

$$j(\theta) = 1/(2m) \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

1）批量梯度下降法（BGD）：这个方法首先将损失函数$j(\theta)$对θ求偏导，得到每个θ对应的梯度。为了最小化风险函数，此时按照每个θ的梯度负方向来更新θ。这种方法能够从全局上直接得到最优解，但其复杂度较高，因为每一次的运算过程都

需要遍历数据集中的全部数据信息，所以当训练集中的数据数量大于一定值时，算法运行速度会十分缓慢[17]。

2）随机梯度下降方法（SGD）针对上一种方法做出了改进，它单方面减少了运行过程中迭代的总次数，避免了上述算法运行速度过慢的问题，虽然这样的算法在总体的训练准确度上会有所下降，但减小的量可以忽略不计，最终迭代次数增量完全在可接受范围内[18]。

3）小批量梯度下降(MBGD)是介于 BGD 和 SGD 之间的一种算法。每次迭代既不是选择全部的样本，也不是随机选择一个样本，而是选择一定规模数量的样本来更新参数[19]。

2.3.2 牛顿法

也被称为切线法，方程 f(x)=0 的根在几何意义上可表示为曲线 y=f(x)与 x 轴交点的横坐标，通过确定一个起始的 x 值作为起始切点，作曲线的切线，经过反复迭代，最终近似的得到方程 f(x)=0 的根，一般来说，这类解法在实数数域和复数数域中均有定义。因为牛顿法求解思路是通过泰勒展开求解的，故每一步求解过程都只会向目标收敛[20]。

2.3.3 共轭梯度法

虽然和梯度下降法一样，共轭梯度法也只使用了一阶导数的相关信息，但是它有效的规避了梯度下降法在逼近极小值时收敛速度明显变慢的缺点。和牛顿法相比，共轭梯度法不需要求解二阶导数,因此不需要去存储和计算 Hess 矩阵并求这个矩阵的逆矩阵。

梯度下降法会在逼近极小值时收敛速度明显变慢的原因在于，尽管梯度下降法的每一步是局部最优的，但是如果观察多次迭代中会发现在不同的迭代中会选择相似的方向，这也就说明单次的迭代没有能够一次性地将这个方向上的误差通过更新方向和步长来更新完，所以就会出现锯齿现象。因此共轭法的思路就在于，每选择一个优化方向，就选择能够一次性处理这个方向所有误差的步长，即每次都将一个方向上的误差做极小化。这样一来，对于一个 N 维的问题，我们只需将它的 N 个方向上的误差都做极小化，那么最终得到的结果一定是极小的。但是这种方法会引入新的问题，即每对一个方向上的误差做极小化时可能会影响之前优化方向上面的调整，所以每次的优化方向和之前的任何一步都应该是共轭正交的[21]。

2.4 本章小结

本章首先介绍了卷积神经网络的相关知识，简要分析了 GoogleNet 的部分结构，然后分析了 SJE 模型的相关原理，最后比较了常见的优化算法之间的优缺点，为第三章基于 SJE 模型进行改进做了理论上的准备。

3 基于深度学习的跨模态迁移学习

3.1 问题概述与分析

3.1.1 跨模态的定义

模态在信息学上指的是数据的存在形式，例如文本，图像，音频，视频等文件格式都是数据的模态。在相当一部分情况下，尽管存在形式不同，多个数据也可能是描述同一个事物的。而我们在获得数据集时往往也不是同一数据集的单一模态的数据，也可能会使用其他模态的数据来丰富我们对于同一事物或者事件的认知。这也正是跨模态迁移学习的理论研究价值所在[22]。

3.1.2 零样本学习

本文主要讨论一般的图片分类问题：

1）训练集数据X_{tr}及其标签Y_{tr}，包含了模型需要学习的类别，这里和传统的监督学习中的定义一致；

2）测试集数据X_{te}及其标签Y_{te}，包含了模型需要辨识的类别，这里和传统的监督学习的定义也是一致的；

3）训练集类别的描述A_{tr}以及测试集类别的描述A_{te}；我们将每一个类别$y_i \in \mathrm{Y}$,都表示成一个语义向量$a_i \in \mathrm{A}$的形式，而这个语义向量的每一个维度都表示一种高级的属性，当这个类别包含这种属性时，那在其维度上被设置为非零值。对于一个数据集来说，语义向量的维度是固定的，它包含了能够较充分描述数据集中类别的属性。

在零样本学习中，我们希望利用X_{tr}和Y_{tr}来训练模型，而模型能够具有识别 X_{te}的能力，因此模型需要知道所有类别的描述A_{tr}和A_{te}。

3.1.3 方法分析

零样本学习有一个十分重要的理论基础，那就是训练分类器的时候，如果用低维图片的特征被高维的语义特征所取代，那么通过这种方式得到的模型就会产生迁移性。所谓高维语义特征就是指语义向量，比如一个事物的高维语义为“两条腿，有羽毛，会咯咯叫，家禽的一种”，那么我们可以判断它是鸡。高维语义不再对事物进行细节上的描述，但是可以很好的对其进行分类。而分类正是我们所要达到的目的，因此我们就可以

舍去低维特征，转而用高维语义来代替。

如果条件允许的话，最好的描述高维特征的数据集还是通过人工得到的。这种数据集的描述准确，而且专一性强。但是如果数据集规模变得十分庞大，那么要获得一个完全由人工标注的数据集就会变得十分困难。为了解决这一问题，人们将目光投向了外部语料库，即从外部的语料库中抽取信息并进行分析，从而得到类别的高维度描述。这种方法实现了自动化提取外部语料库中的高维描述。但是由于外部语料库中数据的质量参差不齐，并且针对性不够强，从外部语料库中自动提取类别的高维描述的的结果仍然不如人工标注高维描述的效果。

3.2 方法分析

3.2.1 线性联合嵌入

在有监督的环境中，我们可以得到一个训练集 T = {(x, y)|x ∈ X ⊂ Rdx , y ∈Y ⊂ Rdy}，其中 x 是在图像特征空间 X 中定义的图像嵌入。这里的 X 可以是经过 CNN 得到的特征值。y 是在标签空间 Y 中定义的类嵌入，对类之间的概念关系进行建模。这个方法的目标是通过学习函数 f:x->y 来预测查询图像的正确类。在通常的迁移学习中，这是通过学习函数 F：X × Y → R 来完成的，对于给定的 x 和 y，预测函数会测量具有最大兼容性的函数 F(x,y)：

$$f(x) = \arg\max F(x, y)$$

通常，类嵌入使用的反映不同类的共同和区别特性的侧面信息是独立于图像提取的。使用这些嵌入，即使在训练集中没有相应标注的那些未知类也可以计算兼容性。所以零样本学习中也经常使用这个方法。在有些场合中，F（x,y）也会被简写作：

$$\mathrm{F}(x, y) = x^{\mathrm{T}}\mathrm{W}y$$

矩阵 W∈Rdx×dy 是从训练数据中学习的参数。由于 f 在 x 和 y 中的双线性，有些文献中将该模型称为双线性模型，但是也可以将其视为线性模型，因为 F 在参数 W 中是线性的。下面，这两个术语将根据上下文互换使用。

兼容性函数的线性度一直是图像分类问题中的一个难题，因为图像分类问题通常是一个复杂的非线性决策问题。但是通过使用分段线性决策函数，线性决策函数已经被非常成功地扩展到了非线性决策函数。在各种计算机视觉任务之中都广泛地应用了这种从非线性向线性决策函数扩展的方法，例如森林火灾监控和人脸识别等。相当多的建立类似模型的主要思想都是一致的：结合潜在的变量，从而使非线性的决策函数分段线性。

这样一来，这个决策函数就可以用一个线性模型的集合来代替，不同的类的特征图像，只需要从线性模型集合中选择结果最好的即可。直观上说，这种方式可以将决策函数的集合分解成为关注数据中独特“聚类”的组件。举例而言，一个组件可以关注与图像的轮廓视图而另一个组件则可以聚焦在对象的正面视图上。

3.2.2 结构性联合嵌入（SJE）

在 CVPR2016 年的会议上，Zeynep 在文献[22]提出了所谓的多线索嵌入方法（multi-cue embedding）。主要思想就是先从外部语料库中提取信息进行分析，来记录类的各种属性。然后对图片中的对象进行多个方面的描述，由于这个时候类的属性已经确定下来了，因此就可以建立起一个从样本对象到类的属性之间的映射，这个映射也就是分类器，整个过程就是一个零样本学习过程。

首先定义分类器如下。输入 x，得到分数最高的 y，即为 x 的类别。

$$f(x) = \arg\max_{y} F(x, y).$$

其中函数 F(x,y)为：

$$F(x, y) = \frac{1}{|g_x||g_y|} \sum_{i \in g_x} \sum_{i \in g_y} \max\left(0, v_i^T s_j\right)$$

g_x代表输入样本 x 在视觉上的特征的集合，g_y代表类别 y 的文本描述上的特征的集合。v_i和s_j可以是集合g_x和g_y中的任意两个元素，$|g_x|$和$|g_y|$分别表示相应集合的元素个数。

其中s_j和v_i可以进一步写作：

$$s_j = f\left(\sum_m w_m^{language} l_m + b^{language}\right).$$

$$v_i = W^{visual}\left[CNN_{\theta_c}(I_b)\right] + b^{visual}.$$

上述公式的主要思想是先将样本数据 x 和类别属性 y 分别用视觉描述向量和文本描述向量来代替，通过映射函数$w_m^{language}$，将两个向量映射到类别空间中，在这个空间中，样本 x 和其对应的类别 y 的 cosine 相似度距离越近越好。这里的距离可以利用 vi 直接乘 sj 得到。整个过程实际上就是一个经典跨模态搜索的过程。

3.2.3 潜在嵌入模型（LatEm）的构建

LatEm 是建立在结构性嵌入（SJE）的理论基础上的，本节将首先讨论 LatEm 和 SJE 的之间的差异。LatEm 通过多个 Wi 矩阵来学习分段线性函数集合，而 SJE 是线性的。线性函数集合使得在处理多个 Wi 矩阵时可以用不同的方式来处理不同类型的图像。对于一个固定的类别 y 和两个大致在视觉上不同类型的图像 x1 和 x2，举个例子，相同的两只鸟，一只在飞翔，另一只在游泳。在 SJE 中，这两个图像会通过两个单映射$w^T x_1$和$w^T x_2$来映射到类嵌入空间，单个映射矩阵 W 会将两个完全不同的图像的向量映射到同一点。而在 LatEm 中则会使用两个不同的矩阵$W_1^T x_1$和$W_2^T x_2$来映射，这样两个不同的映射是分别被分解，因此实现起来就更加容易。从预期上说，在区分视觉上具有某些相似性的两个类别时这种分解也是有利的。在 SJE 中，虽然我们可以很容易地区分红色的鸟和蓝色的鸟，但是对于不同类型的蓝色的鸟就比较难以区分了。而在 LatEm 中，Wi 中的一个可以专注于颜色，而另一个则可以专注于喙的形状（在 4.3 节中将证明这种效果是可见的）。

参照 SVM 的公式，可以构建一个非线性函数的分段线性的函数集合：

$$\mathrm{F(x,y)} = \max_{1\le i\le k} W_i^T (x \otimes y).$$

其中 i 可以取从 1 到 k 之间的任意整数，且 K 要不小于 2。W_i^T是模型中各个线性分量的参数。这里也可以写成方程 2 线性函数集合的形式：

$$\mathrm{F(x,y)} = \max_{1\le i\le k} X^T W_i y$$

对于特定的一组（X_n,Y_n）定义损失函数$L: \mathrm{X} \times \mathrm{Y} \to \mathbb{R}$如下

$$\mathrm{L}(X_n, Y_N) = \sum max\{0, \Delta(y_n, y) + F(y_n, y) - F(X_n, Y_n)\}$$

3.3 目标函数优化

本节主要分析优化目标函数的 SGD 方法。

以一定数量的时期 T 遍历所有的数据集上的样本，对于训练集中的每一个样本(xn,yn)，我们随机选择一个与 Yn 不同的 y(步骤 3)，如果随机选择的 y 超过了边界（步骤 4），那么就更新 wi 对应的矩阵。这个时候就可以找到使 Y 最大化的矩阵 wi 和使 yn 最大化的矩阵 wj。如果使 y 和 yn 最大化的矩阵相同，那么就更新矩阵 Wi。如果使 y 和 yn 最大化的矩阵不同，那么更新矩阵 wi 和 wj。

SGD 方法的 MATLAB 实现如下，相关参数会在 4.2 节中说明。

```
    for e=1:n_epoch
        perm = randperm(n_train);
        for i = 1:n_train
            ni = perm(i);
            best_j = -1;
            picked_y = labels(ni);
            while(picked_y==labels(ni))
                picked_y =    randi(n_class);
            end
            [max_score, best_j] = argmaxOverMatrices(X(ni,:), Y(:,picked_y), W);
            [best_score_yi, best_j_yi] = argmaxOverMatrices(X(ni,:), Y(:,labels(ni)), W);
            if(max_score + 1 > best_score_yi)
                if(best_j==best_j_yi)
                    W{best_j} = W{best_j} - eta * X(ni,:)' * (Y(:,picked_y) -
Y(:,labels(ni)))';
                else
                    W{best_j} = W{best_j} - eta * X(ni,:)' * Y(:,picked_y)';
                    W{best_j_yi} = W{best_j_yi} + eta * X(ni,:)' * Y(:,labels(ni))';
                end
            end
        end
    end
```

3.4 参数 K 的调整

模型中的使用的矩阵数量 K 是自由参数。我们使用两种策略来逐步微调 K 的值。第一种方法，我们首先使用标准的交叉验证策略，即将数据集随机分成不相交的部分（在零样本设置中）并选择具有最佳交叉验证性能的 K。尽管这是一个完善的策略，我们发现它在实验中运作良好，但本文也使用了一种基于剪枝的方法，这个方法在特定情况下能够更快地进行训练。作为第二种方法，我们从大量矩阵开始，并按如下方式修剪它们。随着训练的进行，每个采样的训练样本选择一个矩阵进行评分。将这些信息记录下来，

我们记录这些信息就可以统计出每个训练样例选择各个矩阵的次数。这个步骤是通过在 SGD 的步骤中将 Wj *的计数器增加 1 来完成的。利用这些信息，在经过五次训练数据之后，如果这个选择这个矩阵的样本的数量占总样本数量的比小于 5%，那么就删去这个矩阵。到目前为止，这是基于一个猜想：如果只有极少数的训练样本选择了这个矩阵，则这个矩阵本身可能对性能并不产生太大的影响，故将其舍去。通过这种方法，我们只需要训练一个适应自身的模型，而不是训练多个模型来交叉验证 K，然后用选择的 K 训练最终模型。

3.5 本章小结

本章对论文中所涉及到的问题进行了分析。3.1 节主要介绍了关于跨模态和零样本学习的相关概念。3.2 节主要介绍了 BJE，SJE 和 LatEm 三种模型，分析了它们的工作原理。3.3 节主要介绍了目标函数优化所采用的随机梯度下降方法（Stochastic Gradient Descent），分了了算法的具体实现步骤。3.4 节主要介绍了如何选择 LatEm 模型中矩阵的数量 K

4 系统与实验

4.1 数据集与评估指标

4.1.1 数据集的简单介绍

本文使用了 AWA，CUB 和 Dogs 三个数据集试验和验证 LatEm 模型。试验直接使用的是用 GoogleNet 从这些数据集上提取的 CNN 特征值。

1）Animal with Attributes（AwA）：这个数据集全部由动物的图片组成，总共有 50 个类，其中 40 个类被划作训练集，10 个作为测试集。AWA 数据集的类别的语义有 85 维，共有 30475 张图片

2）加利福尼亚理工学院鸟类数据库（以下简称为 CUB）：这个数据集全部由鸟类的图片组成，总共有 200 类，其中 150 类被划归训练集，剩下的 50 类则被划归测试集，这是一个细粒度的数据集。CUB 数据集的类别的语义有 312 维，共有 11788 张图片。

3）Stanford Dogs Dataset（Dogs）：这个数据集全部由狗的图片组成，总共有 120 个类，其中 100 个类被划作训练集，20 个作为测试集。Dogs 的类别的语义有 113 维，共有 20580 张图片

4.1.2 评估指标

训练得到的结果会与正确的结果相比对，结果分为四类，即“判定为正确且判断正确”、“判定为正确且判断错误”、“判定为错误且判断正确”和“判定为错误且判断错误”。这四种结果的样本数目分别记作 TP，FP，TN，FN。通常来说，衡量模型好坏有精确率，召回率和准确率三个指标。在本实验中采用准确率来衡量模型的分类结果。

4.2 系统的体系结构

4.2.1 各部分说明

1）main 函数:分为设置参数，导入数据、训练模型并验证三个步骤

首先设置参数 param.eta、param.nepoch、param.K、param.cls_emb 四个参数。param.eta 是学习率，初始设置为 1e-1；param.nepoch 是 epoch 的值；param.K 潜在嵌入的数量也就是 3.2 节中所提到的 K 值。param.cls_emb 是要评估的类入的名称，有“word2vec”、‘golve’、‘wordnet’和‘cont’四种。

然后是导入数据，此处导入的是 CUB 数据集的 CNN 特征值。

最后是训练模型，将结果与正确值相比对，输出准确率来评估训练效果。

2）argmaxOverMatrices 函数：[best_score,best_idx] = argmaxOverMatrices(x, y, W)

输入分析：输入三个参数，x,y,w。其中 x 是一个图像嵌入实例，y 是一个类嵌入，w 就是从 x 到 y 的映射矩阵。这是训练过程中的主要函数，选择了 K 个矩阵来进行比较，计算损失函数，得到分数最高的那个矩阵。

输出分析：输出两个参数，best_score 和 best_idx，best_score 是所有输入中的最高的双线性得分，best_idx 则是具有最高分数的嵌入的索引。

3）latEm_test 函数：[mean_class_accuracy] = latEm_test(W, X, Y, labels)

输入分析：输入四个参数，W，X，Y，labels。其中 W 是潜在的嵌入，X 是图像嵌入矩阵，每一行都是一个图像实例，Y 是类嵌入矩阵，每列用于一个类。label 是所有图像的地面实况标签。

输出分析：输出一个参数，mean_class_accuracy。其中 mean_class_accuracy 是所有类别的平均分类精度

4）latEm_train 函数：W = latEm_train(X, labels, Y, eta, n_epoch, K)

输入分析：输入六个参数：X，labels，Y，eta，n_epoch，K。其中 X 是图像嵌入矩阵，每行是一个图像实例，Y 是类嵌入矩阵，每列用于一个类。labels 是所有图像实例的地面实况标签。eta 是随机梯度下降法（SGD）的学习率。n_epoch 是要学习的嵌入数量。

输出分析：输出一个参数 W，W 是具有 K 个嵌入的矩阵。

4.2.2 基于 MATLAB 上的一些简单优化

MATLAB 的帮助文档中提及了一些改善代码性能的一些手段。比较通用的包括向量的预先分配内存，这一点在编辑器里也会提示。有时候预先分配内存与否和性能关系很大，譬如以下两段代码：

```
命令行窗口
>> tic
x = 0;
for i = 2:1000000
x(i) = x(i-1)+5;
end
toc
时间已过 0.168480 秒。
fx >>
```

```
命令行窗口
toc
时间已过 0.168480 秒。
>> tic
x = zeros(1,1000000);
for i = 2:1000000
x(i) = x(i-1)+5;
end
toc
时间已过 0.013095 秒。
fx >>
```

运行结果显示为"时间已过 0.168480 秒"和"时间已过 0.013095 秒"。另外在声明变量的时候不使用原有变量，而创建新变量也可以减少运行时间。

MATLAB 还提供了一些改善性能的手段，包括将长脚本拆开成小段，调用执行；将大的代码块分开为独立的函数；将过分复杂的函数或是表达式采用简单的来代替；采用函数，而不是脚本；向量化代码，采用 MATLAB 自带的函数；采用矩阵的稀疏结构；运行 MATLAB 的时候不要在后台运行其他大的程序；不要重载任何 MATLAB 的内建函数或数据类型。这些技巧中有很多并不那么实用，其中向量化是最有效的一种方法之一。向量化代码中包含有很多常用的函数，比如 all，any，cumsum，diff，find 等。

类型转换：MATLAB 中的运算符支持多种类型，譬如矩阵乘法中多用 double 型变量，但如果一个矩阵是逻辑输入也没有关系。但运算速度差异较大，譬如

```
>> Gc_logic = Gc>0;
>> a=randi([0 1],1,16384);
>> tic;b = a*Gc;toc
时间已过 0.107618 秒。
>> tic;b = a*Gc_logic;toc
时间已过 0.503132 秒。
```

观测结果类型为 double，我们可以大胆推测实际上逻辑型变量在运算过程中先转化为了 double 型。另一个实验结果是：

```
>> tic;Gt=double(Gc_logic);b = a*Gt;toc
```

时间已过 0.373506 秒。

通过创建新变量，可以使运行速度有些许提高

4.3 实验结果与分析

在我们的潜在嵌入（LatEm）模型中，图像嵌入（图像特征）和类嵌入（边信息）是两个基本组件。简而言之，作为图像嵌入，使用从整个图像中提取的预训练 GoogleNet 的顶层合并单元的 1,024 维输出。我们不对图像执行任何特定于任务的预处理。

作为类嵌入，我们评估了四种不同的替代方案，即 attributes（att），word2vec（w2v），golve（glo）和 hierarchies（hie）。attributes 是通过人工注释获得的对象的区别属性。

对于细粒度数据集，如 CUB 和 Dogs，由于对象在视觉上彼此非常相似，因此需要大量的属性来加以区别。在使用的三个数据集中，CUB 包含 312 个属性，AWA 包含 85 个属性而 Dogs 不包含属性注释。我们的属性类嵌入是每个类的向量，根据人类判断来测量每个属性的强度。

除了人类注释之外，可以从大的未标记文本语料库或通过类之间的层次关系自动构造类嵌入。这种方式优点在于不需要任何昂贵的人工注释，而缺点在于这类标注的质量要低于有监督的属性。论文的研究背景之一是从大型文本语料库中提取类嵌入和包含类之间的潜在关系，我们想自动学习这些。因此，我们评估了构建无监督文本嵌入的三种常用方法。 Word2Vec 是一个双层神经网络，它预测单词的方法是用一个滑动的窗口来找寻它的上下文。它为已经学习过的词汇表中的每个单词构建一个向量。 Glove 是另一种分布式文本表示方法，它使用文档中单词的同现统计。文中所使用的为类别构建向量结构的另一种方法是使用诸如 WordNet 之类的层次结构。这里所说的层次结构向量是基于 WordNet 中的子节点和父节点之间的层次距离，对应于我们的类名。为了直接比较，我们再次使用[2]提供的层次向量。在尺寸方面，w2v 和 glo 是 400 维，而 hie 大约是 200 维。

图像特征会令函数归一化，使得每个维度具有零均值和单位方差。所有类嵌入都是归一化的。在开始时将矩阵 Wi 随机初始化为均值为 0 的一组矩阵。epoch 的数量固定为 150。CUB，AWA 和 Dog 数据集的学习率分别选择为 t = 0.1,0.001,0.01，并且在迭代时保持恒定。对于每个数据集，这些参数在默认数据集拆分的验证集上进行调整，并对所有其他数据集折叠和所有类嵌入保持不变。如 3.4 节所述，我们使用交叉验证和剪枝

两种策略来选择潜在矩阵的数量 K。当使用交叉验证时，K 在{2,4,6,8,10}中变化，并且基于验证集上的准确度选择最佳的 K。当使用剪枝时，K 最初设置为 16，然后在训练期间的每五个时期，如果选择这个矩阵的样本数量占样本总数的比例不超过 5%，那么就将这个矩阵去掉。

现在对 LatEm 和前文中提到的 SJE 方法之间进行比较。 SJE（3.2.2 节）通过学习双线性函数来最大化图像和类嵌入之间的兼容性。与之不同的是，LatEm 通过在图像和类嵌入之间定义的多个兼容性函数来学习非线性函数（即分段线性函数）。结果如表 4.1 所示。使用通过人工注释获得的文本嵌入，即 attributes（att），在 AWA 数据集上 LatEm 相较于 SJE 有了显著改善（从 67.8%到 72.3%）。然而，由于我们的目标是减少有监督和无监督类嵌入之间的准确性差距，因此这里关注无监督嵌入，即 w2v，glo 和 hie。在所有数据集中，使用 w2v，glo 和 hie 时 LatEm 对比 SJE 有了明显提高。使用 w2v，LatEm 在 CUB 上的准确率达到了 32.1%（相对于 27.6%），在 AWA 上达到了 62.4%的准确率（相对于 52.4%），Dogs 上的准确率为 23.1%（相对于 18.9%）。同样，使用 glo，LatEm 在 CUB 上的准确率达到 32.6%（相对于 23.7%），在 AWA 上达到 63.1%的准确率（相对于 58.7%），在 Dogs 上达到 21.3%的准确率（相对于 18.2%）。虽然使用 hie 的时候在 Dogs 的结果从 24.3%提高到 25.2%，但在 CUB（从 20.6%增加 24.2%）和 AWA（51.5%增加 57.5%）的改善更为显着。这个结果定量地表明，学习分段线性潜在嵌入确实捕获了类嵌入空间的潜在语义。

表 4.1 使用不同类嵌入在三个数据集上所得到的精确度

	CUB		AWA		Dogs	
	SJE	LatEm	SJE	LatEm	SJE	LatEm
attribute	49.8	46.2	67.8	72.3	——	——
word2vec	27.6	32.1	52.4	62.4	18.9	23.1
glove	23.7	32.6	58.7	63.1	18.2	21.3
hierarchies	21.4	24.3	51.4	57.6	24.2	25.3

4.3.1 稳定性评估

由于缺乏标注的训练数据，零样本学习是一个具有挑战性的问题。换句话说，在训练期间，即使没有测试类的图像也没有类关系。因此，零样本学习在参数设置上比较困难，因为训练和测试类属于不相交的集合。为了获得对预测的稳定估计，除了标准分析

之外，我们还对其他（在我们的例子中为四个）独立和随机选择的数据分割进行了实验。对 LatEm 和 SJE 都重复了五次实验。

结果显示在表 4.2 中。对于所有数据集，SJE 和 LatEm 之间的所有结果的比较都保持不变，因此结论是相同的。尽管 SJE 在 CUB 上的监督属性优于 LatEm，但 LatEm 在 AWA 上的监督属性优于 SJE，而在使用无监督类嵌入时 LatEm 所有结果始终优于 SJE。结果详情如下。在 AWA 上使用有监督的类嵌入（即 attributes），LatEm 的准确率为 72.5％（相对于 70.5％）。在使用无监督嵌入时，使用 w2v 观察到最高准确度，52.3％（对比 49.3％）。在 CUB 上，使用 w2v 的 LatEm 在无监督类嵌入中获得最高准确度，其中 33.1％（相对于 27.7％）在 Dogs 上，LatEm 在所有类嵌入中获得最高准确度，即 25.6％（对比 24.6％）。这些结果确保我们在表 4.2 中提到的准确度提升不是由于数据集偏差。

表 4.2 使用不同类嵌入在三个数据集上所得到的精确度

	CUB		AWA		Dogs	
	SJE	LatEm	SJE	LatEm	SJE	LatEm
attribute	49.5	45.6	70.7	72.5	——	——
word2vec	27.7	33.1	49.3	52.3	23.0	24.5
glove	24.8	30.7	50.1	50.7	14.8	20.2
hierarchies	21.4	23.7	43.4	46.2	24.6	25.6

为了实验的完整性，在本节中，我们记录了 LatEm 使用所有类嵌入（包括监督属性）与 SJE 的完整比较。但是，使用 attributes 有两个缺点。首先，由于细粒度对象类共享许多共同属性，因此我们需要大量的属性，这些属性的获取成本很高。其次，属性注释需要在数据集的基础上完成，即为鸟类收集的属性不适用于狗。因此，基于属性的方法不能跨数据集推广。对无监督的文本嵌入设置更加重要，即 w2v，glo，hie。此外，在使用无监督的词向量时，LatEm 在所有数据集中的 9 个案例中全部优于 SJE。 对于以下部分，我们将仅使用 w2v，glo 和 hie 的结果。

4.3.2 潜在嵌入的可解释性

在之前的分析中，我们已经证明 LatEm 相较于 SJE 在鸟类和狗的两个细粒度数据集（即 CUB 和 Dogs）和动物的一个数据集（即 AWA）上有所提高。 在本节中，我们将针对 CUB 数据集，研究单个 Wi 是否学习图像和类之间的视觉一致性和是否存在可

解释的潜在关系。使用 word2vec 和 glove 作为文本嵌入。 图 4.1 和图 4.2 给出了对于两个嵌入，即 w2v 和 glo，由三个不同的矩阵 Wi 所检索的最高得分图像。

图 4.1 在 CUB 上使用 word2dev 根据矩阵所得到的得分最高的图像

图 4.2 在 CUB 上使用 glove 根据矩阵所得到的得分最高的图像

对于 w2v，我们观察到由相同 Wi（在同一行中）得分高的图像在视觉方面具有一些相似之处。这里要注意的是，尽管他们属于不同的类，但是有长而尖的喙是这些类鸟类的共同方面之一。类似地，对于第二行，检索到的图像是具有棕色头部和浅色胸脯的小鸟，最后一行包含身上的羽毛全部都是黑色的大型鸟类。这些结果很有意思，因为 w2v

是以无监督的方式在维基百科上训练得到的，因此词向量没有属性概念，但是 LatEm 模型不仅能够推断出类的隐藏公共属性，而且用视觉证据支持它们。

4.3.3 剪枝和模型选择的交叉验证

在本节中，我们评估模型中矩阵数量 K 所获得的性能是通过剪枝与交叉验证来确定的。

表 4.3 和表 4.4 显示了通过两种方法选择的矩阵的数量以及它们在三个数据集上的性能。在性能方面，两种方法是相差不大的。剪枝在五个案例中优于交叉验证，在其余六个案例中交叉验证的方法要优于剪枝。性能差距通常在 1-2%之内，除了 AWA 数据集 att 和 w2v 分别为 72.5%和 70.7%以及 52.3%和 49.3%，其中前面的精确度是通过剪枝的方法得到，后面的精确度是通过交叉验证的方法得到的。 因此，这两种方法在性能方面都没有明显的优势，但是交叉验证稍微好一些。

表 4.3 选择的矩阵的数量

	CUB		AWA		Dogs	
	剪枝	交叉验证	剪枝	交叉验证	剪枝	交叉验证
att	3	4	7	2	——	——
w2v	8	10	8	4	6	8
glo	6	10	7	6	9	4
hie	8	2	7	2	11	10

表 4.4 使用表 4.3 中矩阵数量时得到的精确度

	CUB		AWA		Dogs	
	剪枝	交叉验证	剪枝	交叉验证	剪枝	交叉验证
att	43.8	45.6	63.0	72.5	——	——
w2v	33.9	33.1	48.9	52.3	25.0	24.5
glo	31.5	30.7	51.6	50.7	18.8	20.2
hie	23.8	23.7	45.5	46.2	25.2	25.6

就模型尺寸而言，交叉验证似乎略有优势。它选择一个较小的模型，因此在空间和时间效率上都更好。这种趋势对于 AwA 数据集的所有类嵌入都是一致的，但对于 CUB 和 Dogs 来说是混合的。剪枝相对于交叉验证的优势在于训练要快得多，因为交叉验证需要使用多个模型进行训练和测试（每次可能选择 K 一次），修剪只需要训练一次。 然

而，在修剪中存在另一个自由参数，即选择支持矩阵的训练数据量以使其在修剪中存在。可以说，它比直接设置要使用的矩阵数而不是交叉验证更直观。

4.3.4 评估潜在嵌入的数量

在之前的叙述中，当我们使用多个数据划分时，虽然现有技术与我们的方法之间的相对性能差异没有改变，但在某些情况下我们观察到精度的某种增加或减少。在本节中，我们研究了在 CUB 数据集上进行五次重复的实验，并对不同数量的 K 进行了进一步分析。为了完整性分析，我们还评估了单矩阵情况，即 K∈{1,24,6,8,10}。实验使用的是无监督嵌入，即 w2v，glo，hie。

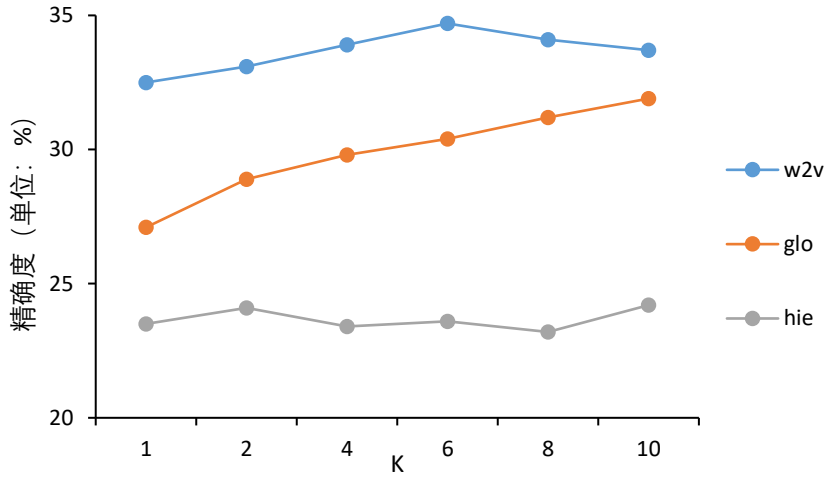

图 4.2 当 K 取不同的值时分别使用 w2v,glo,hie 所得到的精确度

图 4.3 显示了具有不同数量矩阵的模型的性能。这里可以观察到最初性能通常随着 K 的增加而增加，但是之后的趋势则随着使用不同的嵌入而不同。在使用 w2v 时，性能不断增加，直到 K＝6，然后开始下降，可能是由于模型过度拟合所导致的。在使用 glo 时，精确度随着 K 的增加而增加，在 K＝10 时，最终精度比 K＝1 时提高约 5%。而在使用 hie 时，在任何情况下标准误差都不会显著增加，精确度对 K 取上文中提到的所有值都是相似的，精确度随 K 变化的趋势并不明显。总之，当 K 发生变化时所表现出来的结果似乎取决于所使用的嵌入，但是，在零样本设置中，根据数据分布，结果可能变化高达 5%。

4.3.5 针对优化的改进

针对目标函数的优化，表 4.5 中记录了使用随机梯度向下法（SGD）和共轭梯度法时与表 4.2 中数据的对比。在使用共轭梯度法时，除了在 CUB 数据集的 attribute 上准确

率略低于 SGD，其他所有情况下均有所提高，但是都不大，平均在 1%~2%之间。

表 4.5 使用随机梯度下降法和共轭梯度法所得到的精确度

	CUB		AWA		Dogs	
	LatEm	LatEm_imp	LatEm	LatEm_imp	LatEm	LatEm_imp
attribute	45.6	44.8	72.5	73.8	——	——
word2vec	33.1	34.2	52.3	53.4	24.5	25.6
glove	30.7	31.4	50.7	51.6	20.2	22.4
hierarchies	23.7	24.5	46.2	47.1	25.6	27.1

除了使用随机梯度下降法和共轭梯度法外，本文还尝试了小批量下降法和拉格朗日法进行处理。前者由于和随机梯度下降法本出一源，因此准确率并未出现较大变化，通常都在 1%以内。而使用拉格朗日法时准确率出现了下降，因此这两种方法的数据并未在文中记录。

4.4 本章小结

本章首先简要介绍了实验使用的数据集的相关信息，分析了整个模型及其测试程序的输入和输出参数，记录了 LatEm 和 SJE 在 AWW，CUB，Dogs 三个数据集上的精确度差异。简要分析了潜在嵌入的可解释性以及参数 K 的选择过程，并对两种目标函数的优化方法进行了比较。

本章也为本文的核心章节之一，通过对实验结果进行分析，为下一章的总结与展望提供思路。

结论

本文对机器学习中经常出现的标注数据稀缺的问题进行了研究，期望能够通过使用外部文本数据来完成图片分类工作，实现跨模态的迁移学习。

本文的主要工作总结如下：

1）在 SJE 的基础上构建了潜在嵌入模型 LatEm。尝试用分段线性的方法来代替 SJE 的线性映射。模型主要的思想是用高维语义特征代替图片的低维特征，来进行分类器的学习，从而使得训练出来的模型具有迁移性。LatEm 是一种跨模态的方法：它使用通过人工注释或从大文本语料库中以无监督的方式收集的图像和类别的信息。

2）针对 LatEm 中单个样本包含多个映射矩阵的情况。本文对每个矩阵所对应的损失函数进行排序，使用随机梯度下降法和共轭梯度法来求得最优解。

3）在 MATLAB 上实现了 LatEm 模型，并在 AWA，CUB，Dogs 三个数据集上与 SJE 得到的结果进行了对比。在 CUB 和 Dogs 这两个细粒度的数据集上的分类精确度全面优于 SJE。

未来的工作展望主要集中在优化方法的改进上面。本实验中只采用了随机梯度下降法和共轭梯度法进行优化。但是在使用拉格朗日法时出现了负优化的情况，与之前的预期不符，因此有必要分析到底是由于算法本身不适合这个问题还是人为的算法实现理解错误所导致的。除此之外，在查阅文献时，还看到了使用共轭梯度法和最速下降法的混合方法，这种方法不仅提高了共轭梯度法的收敛速度，而且解决了梯度下降法难以求解“性态不好”的目标函数的缺陷。这种方法有尝试的价值。

致谢

转眼间，大学四年也走到了尽头。于我而言，这四年在知识水平的提升之外，我还学到了很多做人的道理。毕业设计便是这四年时光的最后一个句点，尽管这个句点划的并不完美，即便有再多的不甘，也只得就此停笔。

首先，我要由衷地感谢我的导师王德庆老师在我毕业设计完成过程中的督促和批评，感谢他在我整个毕业设计期间提出宝贵的意见，经常关心我的工作进度，在我拖拉懒散时督促我抓紧时间，可以说没有他的督促和指导，就没有今天的这篇论文。

其次，我要感谢我的同学们。在开始着手毕设之前我对机器学习的了解也仅仅停留在看了西瓜书的前几章，读过四五篇相关的文献的程度。因此在开题阶段我遇到了比较大的困难，几乎完全不知道如何下手。在这个期间，我的同学们耐心解答了我一些很基础的问题，告诉我应该从哪些重要的文献开始看起。不管是学习还是生活，他们都对我关怀备至，正因为有了他们，我的大学生活才丰富多彩，摇曳生姿。

我还要感谢我的家人。正因为有了他们在背后的默默支持，我才能从一次次的挫折中挣扎着爬起来。其实我是个很容易气馁的人，即便在撰写论文时我都一度想过放弃，正是你们耐心的开导使得我重新坚定信念，奋力向前。

最后，感谢各位评审老师能够抽出时间来审阅这篇论文，祝你们一切顺利！

参考文献

[1] Xian Y, Akata Z, Sharma G, et al. Latent Embeddings for Zero-Shot Classification[J]. 2016:69-77.
[2] 龙明盛. 迁移学习问题与方法研究[D]. 北京：清华大学, 2014.
[3] Zhang Z, Saligrama V. Zero-Shot Learning via Joint Latent Similarity Embedding[C]// Computer Vision and Pattern Recognition. IEEE, 2016:6034-6042.
[4] Wang Q, Chen K. Multi-Label Zero-Shot Human Action Recognition via Joint Latent Embedding[J]. 2017.
[5] Ziming Zhang, Venkatesh Saligrama. Zero-Shot Learning via Semantic Similarity Embedding[J]. 2015.
[6] Zhang L, Xiang T, Gong S. Learning a Deep Embedding Model for Zero-Shot Learning[J]. 2016:3010-3019.
[7] Bucher M, Herbin S, Jurie F. Hard Negative Mining for Metric Learning Based Zero-Shot Classification[C]// European Conference on Computer Vision. Springer International Publishing, 2016:524-531.
[8] Weston J, Bengio S, Hamel P. Large-Scale Music Annotation and Retrieval: Learning to Rank in Joint Semantic Spaces[J]. Computer Science, 2011.
[9] Qin J, Wang Y, Liu L, et al. Beyond Semantic Attributes: Discrete Latent Attributes Learning for Zero-Shot Recognition[J]. IEEE Signal Processing Letters, 2016, 23(11):1667-1671.
[10] Yanwei Fu, Timothy M. Hospedales, Tao Xiang, et al. Learning Multimodal Latent Attributes[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(2):303-316.
[11] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Labelembedding for image classification. IEEE TPAMI, 2015.
[12] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of Output Embeddings for Fine-Grained Image Classification. In CVPR, 2015.
[13] H. Chen, A. Gallagher, and B. Girod. What's in a name? firstnames as facial attributes. In CVPR, 2013.
[14] K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. ML, 2002.
[15] J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In CVPR, 2013
[16] Chen L , Zheng J , Gao M , et al. TLRec:Transfer Learning for Cross-Domain Recommendation[C]// 2017 IEEE International Conference on Big Knowledge (ICBK). IEEE, 2017.
[17] S. Huang, M. Elhoseiny, A. M. Elgammal, and D. Yang. Learning hypergraph-regularized attribute predictors. In CVPR, 2015.
[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In CVPR, 2015.
[19] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In NIPS, 2013.
[20] Z. Zhang and V. Saligrama. Zero-shot learning via joint latent similarity embedding. In

CVPR, 2016.
[21] 张腾腾. 基于 Fisher 矢量编码与稀疏约束的数据分类[D].西安：西安电子科技大学，2017
[22] Akata Z , Malinowski M , Fritz M , et al. [IEEE 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Las Vegas, NV, USA (2016.6.27-2016.6.30)] 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Multi-cue Zero-Shot Learning with Strong Supervision[J]. 2016:59-68.