**分类号**  **学号 M201973167**

**学校代码10487 密级**

****

**硕士学位论文**

**（学术型☑ 专业型□）**

**基于自注意力机制的自知识蒸馏**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **学位申请人** | **：** |  | **高也** |  |  |
| **学科专业** | **：** |  | **计算机软件与理论** |  |  |
| **指导教师** | **：** |  | **何琨 教授** |  |  |
| **答辩日期** | **：** |  | **2022年XX月XX日** |  |  |

**A Dissertation Submitted in Partial Fulfillment of the Requirements**

**for the Master Degree in Engineering/Science （工学/理学硕士）/the Professional Master Degree (专业学位)**

**Self-Knowledge Distillation with Self-Attention**

**Candidate : GAO YE**

**Major : Computer Science**

**Supervisor : Prof. He Kun**

**Huazhong University of Science and Technology**

**Wuhan 430074, P. R. China**

**May, 2022**

**独创性声明**

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

**学位论文版权使用授权书**

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保 密□，在 年解密后适用本授权书。

本论文属于

不保密☑。

（请在以上方框内打“√”）

学位论文作者签名： 指导教师签名：

日期： 年 月 日 日期： 年 月 日

# 摘 要

[分三段，

研究背景和意义，不足或局限

本文所做工作，提出的方法，创新性

实验结果及分析]要占4/5页

知识蒸馏是一种将知识从教师神经网络迁移到学生神经网络的技术。我们之所以需要将知识从教师神经网络迁移到学生神经网络，是因为教师神经网络常常非常复杂巨大，在训练时需要消耗大量的运算资源。和传统的知识蒸馏技术相比，自知识蒸馏技术不需要外部的教师模型，而是把神经网络最深层的信息视为教师模型，把神经网络的一些浅层(也就是block层)的信息视为学生模型来达到知识蒸馏的效果。在本篇论文中，我们提出一种新的自知识蒸馏模型——基于自注意力机制的自知识蒸馏模型。本模型利用自注意力机制，为作为学生模型的不同中间层计算出不同的权重。我们还从理论上证明了我们的模型等价于集成学习。我们的模型在多个图像识别任务中都取得了迄今为止最好的实验结果。

**关键词：** 注意力机制；自注意力机制；知识蒸馏; 自知识蒸馏

# Abstract

Knowledge distillation is a widely used technique that could effectively transfer knowledge from a teacher network to a student network. However, training the teacher network may consume enormous computational resources. Without training an extra teacher model, the self-distillation technique utilizes the student network's block layers to distill itself and emerges as a new and powerful technique for the knowledge distillation. In this work, we propose a novel Self-Knowledge Distillation with Self-Attention (SKDSA) framework, which utilizes the self-attention mechanism to adaptively obtain the coefficients of distillation for each block layer of the network. We also provide theoretical proof that the proposed approach is essentially an ensemble modeling strategy. Extensive experiments show that our method achieves compelling results on various visual tasks and outperforms the state-of-the-art baselines.

**Key words:** attention, self-attention, knowledge distillation, self-knowledge distillation

# 目 录

[基于自注意力机制的自知识蒸馏 I](#_Toc97905685)

[A Dissertation Submitted in Partial Fulfillment of the Requirements II](#_Toc97905686)

[for the Master Degree in Engineering/Science （工学/理学硕士）/the Professional Master Degree (专业学位) II](#_Toc97905687)

[摘 要 1](#_Toc97905688)

[Abstract 2](#_Toc97905689)

[目 录 3](#_Toc97905690)

[1 绪论 5](#_Toc97905691)

[1.1 研究背景与意义 5](#_Toc97905692)

[1.2 国内外研究现状 6](#_Toc97905693)

[1.3 存在的问题 11](#_Toc97905694)

[1.4 论文组织结构 12](#_Toc97905695)

[2 需要用到的技术工具 13](#_Toc97905696)

[2.1 知识蒸馏 13](#_Toc97905697)

[2.2 自知识蒸馏 14](#_Toc97905698)

[2.3 注意力机制 16](#_Toc97905699)

[2.4 集成学习中的bagging 25](#_Toc97905700)

[2.5 网络模型 27](#_Toc97905701)

[2.6 本章小结 28](#_Toc97905702)

[3 基于自注意力机制的自知识蒸馏 29](#_Toc97905703)

[3.1 基于自注意力机制的自知识蒸馏 29](#_Toc97905704)

[3.2 ??和集成学习的关系 31](#_Toc97905705)

[3.3 本章小结 32](#_Toc97905706)

[4 实验过程 33](#_Toc97905707)

[4.1 实验数据集 33](#_Toc97905708)

[4.2 实验对比算法基准 35](#_Toc97905709)

[4.3 训练算法和实现细节 36](#_Toc97905710)

[4.4 对比结果 36](#_Toc97905711)

[4.5 消融实验及其分析 38](#_Toc97905712)

[4.6 本章小结 40](#_Toc97905713)

[5 总结与展望 41](#_Toc97905714)

[5.1 主要工作总结 41](#_Toc97905715)

[5.2 主要创新点 41](#_Toc97905716)

[5.3 未来工作展望 41](#_Toc97905717)

[致 谢 42](#_Toc97905718)

[参考文献 43](#_Toc97905719)

[附录1 攻读硕士学位期间取得的研究成果 49](#_Toc97905720)

[附录2 攻读硕士学位期间参与的科研项目 50](#_Toc97905721)

# 绪论

## 研究背景与意义

近年来深度学习技术获得了迅猛的发展和广阔的应用，在计算机视觉[1]、自然语言处理[2]、语音识别[3]等领域都取得了非凡的成果。虽然在诸多任务上都有着亮眼的表现，受训练的神经网络常常既深又宽并且包含海量的参数，这使得他们难以应用在计算资源紧缺的场合(比如移动设备或嵌入式设备)。为了解决这个问题，Hinton等人[4]提出知识蒸馏技术，这种技术可以把知识从一个复杂的、预训练的教师模型迁移到一个轻量级的学生模型上。

受迁移学习[5]的启发，知识蒸馏技术首先训练一个更加复杂和庞大的教师模型，然后把其中的知识压缩到一个更小的学生模型上。然而，传统的知识蒸馏技术存在一系列问题。首先，预训练大型的教师模型需要更多的训练时间。其二，大的教师模型和小的学生模型存在容量上的差异，而且学生模型高度依赖于教师模型。因此，人们提出一系列自知识蒸馏技术，即不利用外部的教师模型，利用学生模型自身的知识来做知识蒸馏。

自知识蒸馏技术[6][7][8][9]使用神经网络中间层的信息来做知识蒸馏，以便免去使用额外教师模型的开销。然而，当前已有的自知识蒸馏模型也存在一些问题。其中，Zhang等人[6]把神经网络的深层视为教师模型，把神经网络的浅层视为学生模型，用神经网络深层的信息来训练神经网络浅层。然而，这种方法将各个作为学生的神经网络浅层的信息一视同仁，可能会造成一些信息的丢失。

为了解决这个问题，我们提出一种新的基于自注意力机制的自知识蒸馏框架。通过这个框架，我们可以计算出各个学生层的重要性差异，根据重要性的差异赋予它们不同的注意力分数。为了测试我们模型的泛化能力，我们使用当前流行的两种卷积神经网络在一些视觉数据集上做了实验。实验结果显示我们的方法比当前最好的基准还要好。

## 国内外研究现状

### 知识蒸馏

在深度学习中，知识蒸馏是一种被广泛使用的有效的技术，用来将知识从一个神经网络迁移到另一个神经网络。知识蒸馏主要有两个用途： 模型压缩和知识迁移。对于模型压缩任务，我们训练一个小的模型来模仿一个预训练的大的模型。虽然根据不同的知识形式人们设计出不同的模型，知识蒸馏模型的共同特点可以归纳为学生-教师结构，其中提供知识的结构被称为教师模型，学习知识的结构被称为学生模型。Hinton等人[4]提出通过学习软化标签的类分布，将知识从教师模型迁移到学生模型。软化的标签通过引入温度缩放来提升低概率类别的概率。这个方法在图片分类任务上取得了令人惊喜的效果。Mangalam等人[10]提出一种特别的方法，通过重新计算类权重将神经网络压缩到一个更小的版本中；再计算类的权重，通过阻碍固有类的不平衡来软化标签的分布。Ding等人[11]提出利用残差标签和残差损失使学生模型使用训练过程中产生的错误经验，这可以减小过拟合并提高性能。类似的，Tian等人[12]将教师模型的知识视为结构性的知识，在训练学生的过程中通过对比学习来获取更多的交互性信息。Hegde等人[13]提出通过增加基于变分推断的稀疏正则化器来训练一个变化的学生模型。学生训练的稀疏性降低了过拟合并提升了分类准确率。Wen等人[14]注意到来自教师的信息是重要的，但是不确定的监督同样影响结果。因此，他们提出通过平滑正则来修正错误的预测，通过动态温度来避免不确定的监督。

也有一些工作通过尝试不同的正则化方法来避免过拟合和欠拟合。Cho等人[15]发现当教师模型的尺寸远远大于学生模型时，早早停止的教师模型可以训练出更好的学生模型。早早停止训练教师类似于给教师正则化，在收敛点停止知识蒸馏使学生更好地契合训练。Liu等人[16]提出将参数的分布建模为先验知识，即聚合来自教师神经网络的分布空间；然后用一个稀疏记录惩罚先验知识来限制学生，避免过正则化。Mishra等人[17]通过知识蒸馏将神经网络的量化和模型的压缩结合起来，他们发现低精度神经网络的性能可以通过蒸馏教师的logits得到显著提升。

也有一些方法从噪声标签中学习。在Li等人[18]的文章中，蒸馏的目的是使用大的有噪声的数据集来增强小的干净的数据集，以便学到更好的视觉表示，获得更好的分类效果。由于原始训练集往往太大很难整体上传到云端，Xu等人[19]提出一个正定的无标签的分类器来解决这个问题。Xie等人[20]通过三步训练一个噪声学生模型： 第一步，用有标签的数据训练一个教师模型；第二步，利用教师模型在无标签图像上生成伪标签；第三步，利用有标签图像和伪标签图像来训练学生模型，同时给学生模型注入噪声以获得更好的泛化能力和稳定性。通过这种方法，学生模型可以比教师模型获得更好的泛化性能。类似的，Sarfraz等人[21]把对抗性扰动视为提升学生模型泛化性和稳定性的重要因素。基于人类学习的方法，人们提出两种教师-学生模型： 多变教师模型和软正则化。多变教师模型将教师模型的不确定性通过教师神经网络的Dropout[22]机制传递给学生。软正则化方法则是在知识蒸馏中添加高斯噪声来提升模型的对抗稳定性。

给蒸馏添加限制也是一种可行的思路。Yang等人[23]的方法给教师模型增加限制条件。Yang等人发现，除了学习正确的标签类别，学习高置信度的类别可以降低学生模型的过拟合风险。他们据此提出根据代际优化神经网络的方法，过程是由原始数据集监督训练出模型，再以模型作为教师监督训练出模型，以此类推。Yu等人[24]将Hinton等人[4]的工作进一步扩展到度量学习中，通过嵌入神经网络将从图片中学到的信息投射到嵌入空间。嵌入坐标一般用来计算教师-学生数据对之间的距离。从这个观点来看，基于嵌入神经网络计算的知识就是真实的知识——因为它代表着数据分布。他们设计了两种不同的教师： 绝对的教师和相对的教师。绝对教师的目的是尽可能缩短教师嵌入坐标和学生嵌入坐标的距离，而相对教师的目的是迫使学生模型去学习数据点间类似距离的嵌入坐标。Arora等人[25]提出一个嵌入模型来为问答获取查询和文本信息之间的互动，其中输入表示的嵌入包含一个查询编码器、一个历史探测编码器、一个历史回复编码器和一个文本编码器。Zagoruyko等人[26]通过诱导学生模型模仿教师模型的注意力图提升了蒸馏的效果。

### 自知识蒸馏

虽然传统的知识蒸馏方法取得了很好的效果，但是它们还是有许多缺陷亟待解决。首先，传统的知识蒸馏方法效率低下，因为学生模型很少需要用到教师模型的全部知识。第二，设计和训练高容量的教师模型依然面临许多困难。第三，两阶段的蒸馏过程需要大量的计算和存储资源。为了解决这些问题，人们提出一些自知识蒸馏的方法。自知识蒸馏的目标是让学生模型蒸馏自己内部的知识，而不需要借助外部的教师模型。

Furlanello等人[27]第一次提出自知识蒸馏的概念。他们首先通过顺序教学，让学生神经网络持续性更新，在这一过程的最后部分，通过聚合多个学生神经网络的结果可以令最终结果获得更好的性能。Hahn等人[28]将重生蒸馏[27]应用到自然语言处理中。Yang等人[23]发现目前学生-教师模型优化的机制尚不清楚，因此他们选择给教师模型增加一些限制条件(比如说给交叉熵损失函数加上一个额外项)，这样学生模型可以更好地学习类之间的相似性，避免过拟合。Clark等人[29]将Furlanello方法[27]的单任务学习扩展到多任务学习，利用模型中的单任务模型依次训练去教导一个多任务模型。由于重生蒸馏的方法基于多步训练，它们比“深层教导”模型更加低效和耗时。

由于模型神经网络的深层比浅层包含更多有用的信息，深层部分可以用来向浅层部分蒸馏知识。在Hou等人[30]的工作中，相比于直接蒸馏特征，他们使用Zagoruyko等人[26]的基于注意力的方法来迫使神经网络浅层去模仿神经网络深层的注意力图。Luan等人[31]将每个层分支打造成一个分类器，他们用最深层的分类器的信息(特征图和logits)蒸馏到浅层分类器。

大多数方法把关注点放在如何更好地在自知识蒸馏过程中“教导监督”学生模型，用于训练学生的数据却没有被充分挖掘和利用。为此，Xu等人[32]和Lee等人[33]通过训练样本的数据增强来做知识蒸馏。这样的方法带来如下好处：首先，优化一个没有其他模型或分支的单个学生神经网络更加迅速和有效；第二，通过数据到数据的自提炼，学生模型可以学到更多固有的泛化表示；第三，在计算和存储成本相对较低的前提下，学生模型的性能能够显著提高。Xu等人[32]对训练数据中的图像批量使用随机镜像和裁剪。Lee等人[33]利用了两种类型的数据增强方式(图片旋转和颜色变化)，又用集成学习把学生模型的logits聚合到一起，再由学生模型进行自知识蒸馏。

还一种类型的自知识蒸馏是基于数据增强来实现。Xu等人[34]设计了一种通过数据失真实现自知识蒸馏的方法，即使用一种一致规则化损失函数来迫使输入和它的变种能够产生相似的预测结果。另有一种类型的自知识蒸馏是基于特征的自知识蒸馏，即通过在不同神经网络层之间添加额外的连接来实现利用中间层信息的目的。BYOT[6]就是一种这样的方法。它通过加入一系列辅助的神经网络层来进行自知识蒸馏，以便把信息从神经网络深层传递到神经网络浅层。

### 自注意力机制

注意力机制最早由Bahdanau等人[35]提出，现已变成神经网络中的一种流行的方法。注意力在人工智能的诸多方向已经有了广泛的应用，比如自然语言处理[36]，语音识别[37]和计算机视觉[38]。对注意力机制的直观理解可以用人类的生物学机制来解释。比如说，我们的人体视觉系统倾向于选择性地关注图片的某些部分，同时忽视其他不相关的信息[39]。类似的，在语言、语音或图像等任务中，输入的某些部分比其他部分更重要。比如说，在机器翻译或摘要提取的任务中，只有输入里面的某些特定单词对预测下一个单词有帮助。类似地，在图像处理任务中，输入图像只有的某些部分和图像标签有关联(比如一张大熊猫的图片里，大熊猫只占图片的一部分)。注意力机制能够让模型重点关注输入的重点部分，从而使模型更加有效。

Yang等人[40]提出层次注意力网络(HAN)来获取文档的结构。文档有一个层次结构： 单词组成句子，句子组成文档。文档中不同的单词和句子包含不同程度的信息。另外，某个单词或句子的重要性和上下文语境密切相关，当上下文不同时，相同的单词和句子在不同的上下文语境中重要性可能截然不同。为了解决这个问题，HAN使用了包含单词和句子两个层面的注意力机制，即让模型在构建文档的表示时赋予单词和句子不同的注意力权重。Xiong等人[41]创建了一个可以获取问题和文档互动方式的编码器。Ptr-Net[42]修改了基于RNN搜索的注意力机制来表示变长的字典，它把注意力机制作为指针使用。See等人[43]使用了一个经典序列到序列注意力模型和Ptr-Net[42]模型的混合模型来抽取文本摘要。混合指针产生器[43]利用指针将单词从文本中复制出来，这既可以保留通过生成器产生新单词的能力，又可以辅助信息复制的准确性。它追踪哪些信息已经被抽取，从而避免了重复抽取的问题。FusionNet[44]利用一个新的概念“单词历史”来特化注意力信息，包括从最底层的单词嵌入到最高层的语义表示。这个概念考虑到输入数据是逐步转化为更抽象的表示，在人的记忆流中逐步形成每个单词的历史。FusionNet使用了全感知多层次注意力机制和一个使用单词历史的注意力分数机制。Rocktäschel等人[45]使用了两点注意力来识别文本蕴含，这个机制使模型能够参加过去的输出向量。包含注意力的LSTM并不需要获取前提的全部语义，而是由注意力产生输出向量。Luong等人[46]提出基于全局和局部注意力的机器翻译。全局注意力机制类似于软注意力机制，而局部注意力则是对前者的一种提升使之可微分。

自注意力机制是注意力机制的一种特例。Parikh等人[47]利用自注意力机制将大问题分解成能够分而治之的几个子问题，从而增加处理问题的并行性。在表示学习领域，自注意力机制同样被用于发现不同神经网络层之间的隐含关系。Ji等人[48]首先将自注意力机制应用于知识蒸馏以便更加准确地抽取知识特征。然而，之前提到的方法都需要一个预先训练好的教师模型。由于教师模型大而复杂，需要大量的计算资源，所以很难被广泛地应用。有鉴于此，我们采用了一种修正后的自注意力机制来提高抽取知识的效率，避免使用外部的教师模型。

### 集成学习

集成学习利用多个机器学习算法分别产生较弱的预测结果，再通过投票机制产生相对单个算法较好的性能。多项实验证明，随着模型复杂度的增加，模型的整体错误率先是逐步降低，到达最低点后再逐步升高。集成学习最早的研究可以追溯到上个世纪。Dasarathy等人[49]利用不同种类的分类器组成一个复合分类器，由此可以提升整体的分类准确率。Kearns等人[50]研究了弱学习算法和强学习算法在PCA模型中的关联性。Schapire等人[51]研究了将多个弱学习模型并入一个高精度模型的可行性。具体说来，集成学习致力于将多个不同的机器学习算法无缝集成到一个统一的框架中。因此，每个算法的互补信息可以被有效利用来获得整体模型更好的表现。从这个角度看，集成学习具有非常好的扩展性，因为它可以合并各种类型的机器学习算法(比如分类任务、聚类任务等)。总的说来，现有的集成学习方法可以被分为四类： 监督集成分类，半监督集成分类，集成聚类，半监督集成聚类。

一个典型的集成分类模型包含两步： (1) 使用多个弱的分类器生成分类结果，(2) 集成多个结果到一个投票模块得到最终结果。广泛使用的集成分类方法包括 bagging[52]，AdaBoost[53]，随机森林[54]，随机子空间[55]，梯度提升[56]。Bagging通过从训练集中随机采样产生样本子集，然后利用这些样本子集来训练集成模型。Bagging模型中的训练是通过并行方式进行的。AdaBoost通过迭代调整样本的权重来处理错误分类的样本，由此提升模型的分类准确率。AdaBoost模型的训练是串行的而不是并行的。随机森林从两个角度训练多个决策树模型： 采样角度和特征角度。因此，它解决了投票集成多个决策树时易于出现的过拟合问题。随机森林中单个模型的训练是并行的，这一点和bagging颇为类似。随机子空间通过随机采样特征构造了一系列特征子空间，然后训练这些子空间中的基础分类器来产生多个结果，再集成到最终结果。随机子空间的基本模型是并行训练的。梯度上升算法通过随机采样得到样本子集，再构建和训练每一个学习者以降低前一个学习者产生的残差。因此，梯度上升算法可以使集成模型的最终残差之和足够小，从而迫使预测结果接近实际值。类似于AdaBoost方法，梯度提升中的基本模型以串联方式进行训练。

我们的方法可以等价为bagging[52]的一种，即通过实现一系列同质的机器学习算法来降低泛化的差距。我们会在下面的章节证明SASD和bagging的等价性。

## 存在的问题

虽然BYOT提供了基于特征的自知识蒸馏方法，但是它把不同的学生神经网络层一视同仁，忽略了他们的不同贡献。为了提高自知识蒸馏的有效性，我们修正了BYOT模型，即通过加入一个自注意力机制模块来计算不同中间层(学生神经网络层)和最深层(教师神经网络层)特征的相关性。

## 论文组织结构

全文共分为4章。主要内容如下：

第一章 绪论：本论文研究背景是深度学习在诸多领域的广泛应用。接下来我们介绍了知识蒸馏、自知识蒸馏、自注意力机制、集成学习的国内外研究现状以及存在的问题，由此引出本文的主要研究内容。

第二章 需要用到的工具和技术：我们介绍本篇论文中需要用到的工具和技术，包括自知识蒸馏、注意力机制、集成学习中的bagging和残差网络。

第三章 理论和算法：我们在理论层面介绍了知识蒸馏和自知识蒸馏，把自知识蒸馏和自注意力机制结合起来，由此导出我们的算法。再证明我们方法和bagging的等价性。

第四章 实验过程：我们在CIFAR-100[57]等数据集上进行图像分类实验，发现我们方法的性能比其他已有的自知识蒸馏方法结果都要好。我们之后还做了消融研究。

第五章 总结与展望：我们的方法将自注意力机制和自知识蒸馏结合起来，获得了比其他已有的自知识蒸馏方法结果都要好的图像分类结果。我们证明了我们的方法和bagging的等价性。我们还进行了消融研究。

# 需要用到的技术工具

在这一章节，我们首先介绍知识蒸馏和自知识蒸馏的基本概念，介绍利用中间层知识的重要性然后利用中间层预测结果的集成学习来提高模型的有效性和可靠性。受到自注意力机制的启发我们提出基于自注意机制的自知识蒸馏(Self-Knowledge Distillation with Self-Attention，简称SKDSA)方法，并证明我们的方法可以被视为集成学习中的bagging。

本文中的记号。我们记为一系列标签实例，其中N是标签的数量; 记为最深层的特征，记为最深层的输出。假设该模型有个Block层，记为第个block层的特征，记为第个block层的输出(的范围是从1到)。

## 知识蒸馏

我们首先介绍传统知识蒸馏的概念。知识蒸馏是一项在大的教师模型的帮助下训练小的学生模型的技术。和其他神经网络压缩方式不同的是，知识蒸馏可以无视教师模型和学生模型结构上的不同。在Hinton等人[4]提出的模型中，知识由教师模型迁移到学生模型，通过最小化教师模型产生的logits和学生模型产生的logits的差异来实现。

然而在许多情况下，教师模型logits的softmax的输出在正确类型上概率很高，在非正确类型上概率趋近于零。在这种情况下，除了数据集中已有的正确标签外，模型无法提供更多的信息。为了解决这个问题，Hinton[4]提出温度归一化的概念，即令标签软化。假定代表由模型产生的logit向量，一张图片的某个类别概率为：

|  |  |  |
| --- | --- | --- |
|  |  | (2-1) |

其中是蒸馏温度参数，代表教师模型中标签软化的程度。当时，我们可以得到正规的softmax公式(也就是没有蒸馏)。随着增加，softmax函数产生的概率分布越来越软，提供越来越多的关于教师认为哪个类别和预测的类别更为相似的信息。教师模型提供的信息被称为暗信息。是暗信息在影响整体蒸馏的信息流。当计算蒸馏损失函数的时候，在教师模型中使用的相同的温度也用来计算学生模型中的logits。对于有正确标签的图像，Hinton[4]指出训练学生模型时把正确标签和教师的软化标签放在一起训练更好。因此，学生模型的损失函数可以定义为学生模型预测出的标签和真实标签的差距。总损失函数由学生损失函数和蒸馏损失函数相加而成：

|  |  |  |
| --- | --- | --- |
|  |  | (2-2) |

其中函数代表的是由软标签和计算出的交叉熵损失函数，代表知识蒸馏的系数。

## 自知识蒸馏的BYOT方法

传统知识蒸馏方法非常直接，但是教师模型非常庞大复杂，训练起来很花时间。此外，由于教师模型和学生模型的容量差距巨大，所以学生模型重度依赖教师模型。 为了解决这个问题，人们提出许多自知识蒸馏(即利用模型自身的知识信息)的方法。特别地，Zhang等人[6]提出自知识蒸馏机制，方法是把模型的神经网络深层的信息蒸馏到模型的神经网络浅层。他们根据以下思维方式构建自知识蒸馏框架： 首先，目标卷积神经网络根据其深度分为几个浅层。例如，将ResNet50 根据 ResBlocks 分为 4 个部分。第二，在每个浅层之后设置一个分类器，结合一个瓶颈层和一个仅用于训练并且可以在推理中移除的全连接层。添加瓶颈层的主要考虑是减轻每个浅分类器之间的影响，并添加来自Block层信息的引导。在训练阶段，我们将所有具有相应分类器的block层视为学生模型，而将最深层视为教师模型。为了提高学生模型的性能，在训练过程中引入了三种损失函数：

* 损失函数1： 真实标签与最深层分类器的交叉熵损失。它使用来自训练数据集的真实标签和最深层分类器的输出来计算。通过这种方法，我们把隐含在数据集中的知识从真实标签引入到最深层分类器。
* 损失函数2： 老师指导下的KL（Kullback-Leibler）散度损失函数。KL散度使用学生和教师之间的softmax输出计算，并引入每个block层分类器的softmax层。通过引入 KL 散度，自知识蒸馏框架将教师神经网络(最深的神经网络)的影响传递到每个block层分类器。
* 损失函数3： 特征图之间的L2损失函数。它可以通过计算最深层分类器和每个block层分类器的特征图之间的L2损失来获得。通过 L2 损失函数，模型将特征

图中的隐含知识引入每个block层分类器的瓶颈层，从而引导所有的分类器在其瓶颈层中的特征图拟合最深层分类器的特征图。

需要说明的是，所有新添加的层(图2-1中ResBlock下方的部分)仅在训练期间应用。它们不对推理过程产生影响。

损失函数1由式(2-3)表示。它由最深层的分类器输出和真实标签计算得出：

|  |  |  |
| --- | --- | --- |
|  |  | (2-3) |

损失函数2由式(2-4)表示。它是和的KL散度，目的是让添加的block层分类器能够模拟最深层的预测结果，其中代表第个分类器经过softmax层的输出，代表最深层分类器经过softmax层的输出，代表蒸馏温度。

|  |  |  |
| --- | --- | --- |
|  |  | (2-4) |

损失函数3由式(2-5)表示。它代表最深层分类器对浅层分类器的“提示”。我们将“提示”定义为教师模型隐藏层的输出，目的是指导学生模型的学习，如Romero等人[58]的工作一样。它的工作原理是缩短block层分类器的特征图和最深层分类器的特征图之间的距离。 然而，由于不同深度的分类器的特征图具有不同的大小，因此需要添加额外的神经网络层来对齐它们。我们没有像Romero等人[58]一样使用卷积层，而是使用了瓶颈架构，该架构能够更好地提升模型性能。式(2-5)中表示第个block层分类器的特征，表示最深层分类器的特征。

|  |  |  |
| --- | --- | --- |
|  |  | (2-5) |

综合上述三种损失函数，也就是式(2-3)(2-4)(2-5)，神经网络整体的损失函数可以写成：

|  |  |  |
| --- | --- | --- |
|  |  | (2-6) |

我们使用两个超参数和来平衡式(2-3)(2-4)(2-5)。



图2-1 BYOT的框架图。

## 注意力机制

### 注意力机制的提出

在心理学中，注意力是选择性地专注于一件或几件事情而忽略其他事情的认知过程。比如，假设我们正在查看一张小学生的合影照。通常情况下，会有一群孩子坐在几排之间，而老师会坐在中间的某个地方。如果想要知道照片中有几个人，我们仅仅需要关注照片中有几个脑袋就可以了，而不需要在意照片中的其他事物。如果想要知道照片中的老师，我们就会去寻找照片中符合成年人特征的信息，而忽略了其他信息。这就是我们的大脑的“注意力”机制。

深度学习中最早使用的注意力机制是对自然语言处理 (NLP) 中基于编码器-解码器的神经机器翻译系统的改进。后来，注意力机制及其变种被用于深度学习的其他领域，包括计算机视觉、语音处理等。

在注意力机制被引入自然语言处理(NLP)之前，机器翻译主要使用基于RNN/LSTM的编码器-解码器机制。也就是说，编码器和解码器都是由一个个RNN/LSTM单元组成的。它的实现主要通过以下两步：

* 编码器 LSTM 用于处理整个输入句子并将其编码为上下文向量，这是 LSTM/RNN 的最后一个隐藏状态。我们可以将此视为对输入句子的一个很好的表征。编码器的所有中间状态都被忽略，最终状态就是解码器的初始隐藏状态。
* 解码器的LSTM或RNN 单元一个接一个地生成(翻译出)句子中的单词。

简而言之，有两套RNN/LSTM。 我们将第一套称之为编码器——它会读取输入句子并尝试理解它，然后再对其进行“总结”。它将摘要（上下文向量）传递给解码器(也就是第二套RNN/LSTM)，解码器通过查看输入句子来翻译它。

这种方法的主要缺点是显而易见的。如果编码器对输入句子“总结”不好，翻译结果就会很糟糕。实验证实，编码器在尝试理解较长的句子时会创建一个糟糕的“总结”。这就是RNN机制面临的长距离依赖难题。RNN面临长距离依赖难题的原因是梯度消失和爆炸问题。随着输入句子长度的增加，基于RNN的编码器-解码器网络的性能会迅速下降。尽管 LSTM 比 RNN 更好地处理远程依赖问题，但在特定情况下它往往会变得健忘。另一个问题是，在翻译句子时，没有办法凸显不同单词重要性的差别。

假设我们要做一个预测下文的任务。“尽管他在波兰出生，因为他是在日本长大的，所以他日语说得很好”。在这组句子中，如果我们要预测单词“日语”，单词“长大”和“日本”在预测时应该给予这两个更大的权重。虽然“波兰”也是一个地名，但是在预测时应该被忽略掉。

那么有什么方法可以在创建上下文向量的同时保持输入句子中的所有相关信息完整无缺？Bahdanau 等人[35]提出了一个简单优雅的模型，此模型不仅可以在上下文向量中考虑所有输入词，还应该对每个输入词给予相对重要性。每当这个模型生成一个句子时，它都会在编码器隐藏状态中搜索最相关信息所在的位置。这个模型就是注意力机制。

图2-2是 Bahdanau 等人[35]展示的注意力模型的示意图。图中使用的双向 LSTM 为每个输入句子生成一系列标注()。其中向量 等是编码器中前向和后向隐藏状态的串联。

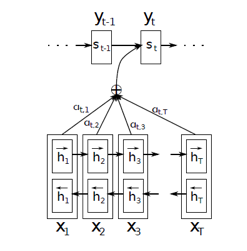


图2-2 基于注意力机制的翻译模型

简单来说，所有的向量 都是输入句子中个单词的表示。在简单的编码器和解码器模型中，仅使用编码器 LSTM 的最后一个状态（在本例中为）作为上下文向量。但是 Bahdanau 等人[35]在创建上下文向量时将重点放在输入的所有单词的嵌入（由隐藏状态表示）。他们通过对隐藏状态进行加权求和来做到这一点。

现在的问题是如何计算加权求和所需要的权重？权重也是通过前馈神经网络学习的，我们在下面提到了它们的数学方程。

输出词的上下文向量是使用标注的加权和生成的，如式2-7所示：

|  |  |  |
| --- | --- | --- |
|  |  | (2-7) |

权重由式2-8给出的 softmax 函数计算：

|  |  |  |
| --- | --- | --- |
|  |  | (2-8) |

是由函数 a 描述的前馈神经网络的输出分数，该函数试图捕获处的输入和处的输出之间的对齐。

|  |  |  |
| --- | --- | --- |
|  |  | (2-9) |

一般而言，如果编码器产生个“标注”（隐藏状态向量），每个具有维度，则前馈网络的输入维度为（假设解码器的先前状态也具有维度且和这两个向量是连接的）。该输入与维的矩阵相乘（当然后面加上偏置项）以获得分数(维数为 (Tx, 1))。

在这些分数的顶部，使用正切双曲函数，然后使用softmax函数以获得输出的归一化对齐分数：

|  |  |  |
| --- | --- | --- |
|  |  | (2-10) |

因此，是一个维向量，它的元素依次对应于输入句子中每个单词的权重。

### 全局注意力机制和局部注意力机制

术语“全局”注意力是指所有输入都被赋予了重要性。 最初，全局注意力与我们之前讨论的注意力概念有一些细微的差别。差别就在于它考虑了编码器 LSTM 和解码器 LSTM 的所有隐藏状态来计算“可变长度上下文向量”，而 Bahdanau等人[35]提出的注意力机制则是使用编码器 LSTM 的所有隐藏状态和单向解码器 LSTM 的先前隐藏状态来计算上下文向量。在编码器-解码器架构中，分数通常是编码器和解码器隐藏状态的函数。 任何函数都是有效的，只要它捕捉到输入词相对于输出词的相对重要性。

使用“全局”注意力层时需要大量计算。这是因为我们必须考虑所有隐藏状态，将其串联成一个矩阵，并乘以正确维度的权重矩阵，以获得前馈连接的最后一层。因此，随着输入大小的增加，矩阵大小也会增加。 简单来说，前馈连接中的节点数量增加了，因此增加了计算量。为了减小计算量，我们使用了局部注意力概念。

直觉上，当我们试图从任何给定的信息中推断出某些东西时，我们的大脑往往会通过只接受最相关的输入来缩小搜索空间。全局和局部注意力的想法受到主要用于计算机视觉任务的软注意力和硬注意力概念的启发, 软注意力和硬注意力由Xu等人[59]在2015年提出。软注意力是全局注意力，其中所有图像块都被赋予了一定的权重；但在硬注意力中，一次只考虑一个图像块。

但是局部注意力与图像捕获任务中使用的硬注意力不同。相反，它是这两个概念的混合，不用考虑所有编码的输入，而是只考虑一部分用于上下文向量的生成。这不仅避免了在软注意力中产生的昂贵计算，而且比硬注意力更容易训练。

那么如何实现这一目标？在这里，模型试图预测输入词嵌入序列中的位置。在位置周围，它考虑一个大小为的窗口。因此，上下文向量生成为位置中输入的加权平均值，其中是根据经验选择的。

此外，可以有两种类型的“对应”方式：

1. 单调对应，其中设置为，假设在时间，只有附近的信息很重要。
2. 预测对齐，其中模型本身预测对齐位置式2-7所示：

|  |  |  |
| --- | --- | --- |
|  |  | (2-11) |

其中和是训练期间学习的模型参数，是输入句子长度。显然，范围在。

图2-3展示了全局注意力机制和局部注意力机制之间的区别。全局注意考虑所有隐藏状态（蓝色），而局部注意只考虑一个子集：

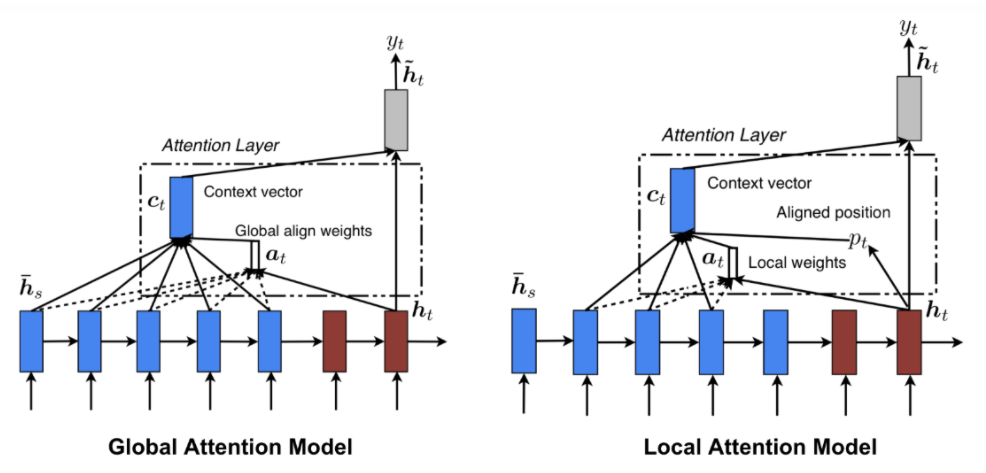


图2-3 全局注意力模型和局部注意力模型

### Transformer

Vaswani 等人[60]的名为“Attention is All You Need”的论文是迄今为止对 Attention 最重要的贡献之一。他们重新定义了注意力机制，即基于键、查询和值提供了一个非常通用和广泛的注意力定义。他们引用了另一个称为多头注意力的概念。

首先，让我们定义什么是“自注意力”。Cheng 等人[61]在他们名为“Long Short-Term Memory-Networks for Machine Reading”的论文中，将 self-Attention 定义为关联单个句子的不同位置以获得更好的表示的机制。

机器阅读器是一种算法，可以自动理解提供给它的文本。如图2-4所示，红色字是当前瞬间读取或处理的，蓝色字是“记忆”。不同的阴影代表“记忆”激活的程度。

当我们逐字阅读或处理句子时，先前看到的单词也会被注意到，并从阴影中推断出来，这正是机器阅读器中的自注意力所做的。

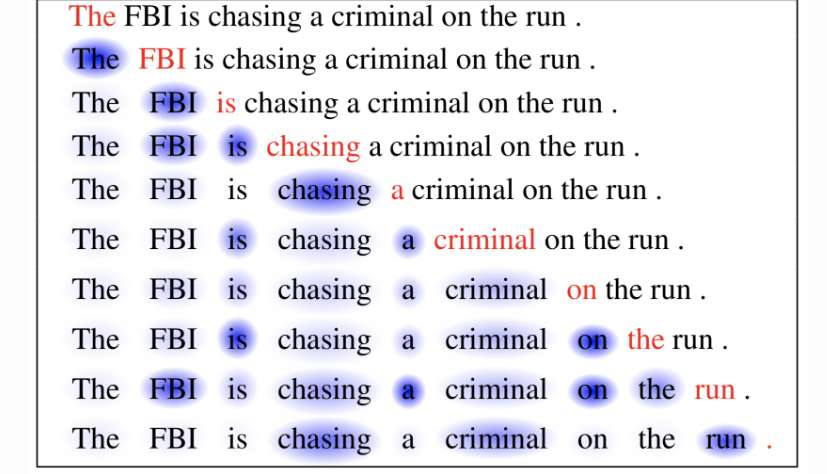


图2-4 自注意力机制

以前，要计算句子中单词的注意力分数，计算的机制是使用点积或单词的其他函数与先前看到单词的隐藏状态表示。Cheng等人[61]提出了一个基本相同但更通用的概念。假设我们要计算“chasing”这个词的注意力。该机制将计算嵌入“chasing”与嵌入每个先前看到的单词（如“The”、“FBI”和“is”）的点积。现在，根据广义注意力的定义，每个词的embedding应该有对应的三个不同的向量，分别是键、查询和值。我们可以使用矩阵乘法计算出这些向量。

每当我们需要计算目标词相对于输入嵌入的注意力时，我们应该使用目标的查询和输入的键来计算匹配分数，然后这些匹配分数作为权重求和过程中的值向量。其中键、查询和值向量是不同子空间中嵌入向量的抽象。可以这样理解：你提出一个问题；查询命中输入向量的键。键可以与读取的记忆位置进行比较，值就是要从记忆位置读取的值。

如果嵌入的维度是并且我们想要维度的键向量，我们必须将嵌入乘以维度的矩阵。所以，键向量。 类似地，对于查询和值向量，等式将是，（E 是任意单词的嵌入向量）。

现在，为了计算单词“chasing”的注意力，我们需要将“chasing”嵌入的查询向量与之前每个单词的键向量进行点积，对应的键向量单词依次为“The”、“FBI”和“is”。然后再将这些值除以 （嵌入的维度），加之以softmax函数。所以，过程如下所示：

|  |  |  |
| --- | --- | --- |
|  |  | (2-12) |

基本上，这是目标词的查询向量和输入嵌入的关键向量的函数 。它不一定是和的点积。任何人都可以根据需要选择函数。

实际上，将所有嵌入的输入向量组合在一个矩阵中，然后将其与常见的权重矩阵、、相乘，分别得到、和矩阵。所以结合后的方程变为：

|  |  |  |
| --- | --- | --- |
|  |  | (2-13) |

因此，上下文向量是键、查询和值的函数。Bahdanau等人[35]的注意力是这里描述的注意力机制的特例，其中突出的特点是单个嵌入向量同时用作键、查询和值向量。

在多头注意力机制中，矩阵与不同的、和矩阵相乘，分别得到不同的、Q和矩阵。我们最终得到不同的矩阵，即每个输入词的嵌入被投影到不同的“表示子空间”中。

例如，在三头自注意力中，对应于“chasing”这个词，将有 3 个不同的矩阵，也称为“注意头”。这些注意力头被连接起来并与单个权重矩阵相乘，得到一个注意力头，该注意力头头将捕获来自所有注意力头的信息。图2-5展示了多头注意力机制。可以看到，不同的、、向量产生了多个注意力头，它们是串联的。

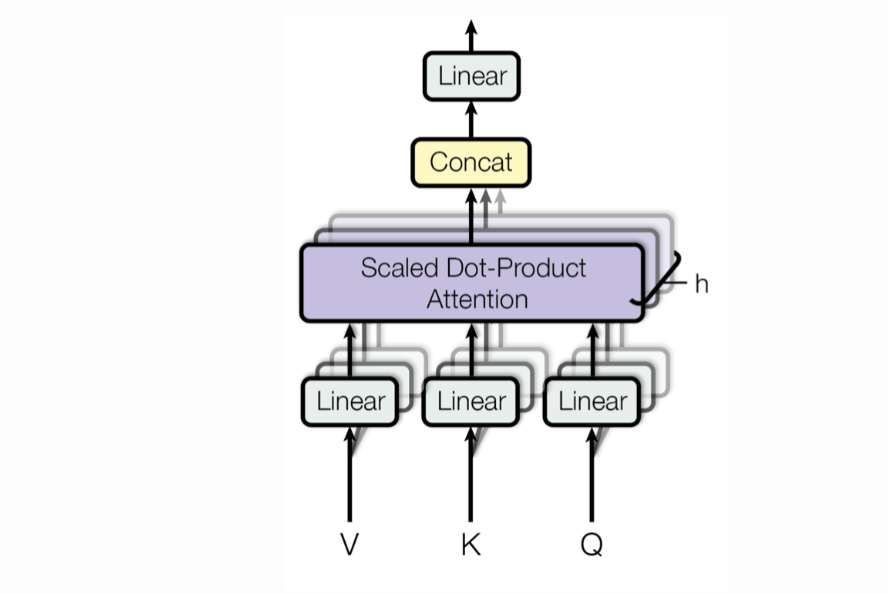


图2-5 多头注意力机制包含几个并行的注意力层

图2-6展示了transformer的结构。我们看到已经使用了一种称为“位置编码”的东西，并添加了在编码器和解码器中的输入嵌入。这是因为之前描述的模型无法解释输入单词的顺序。Transformer试图通过位置编码来捕捉这一点。Transformer为每个输入嵌入添加了一个向量，这些向量是为了表征每个单词的位置或输入中不同单词之间的距离。

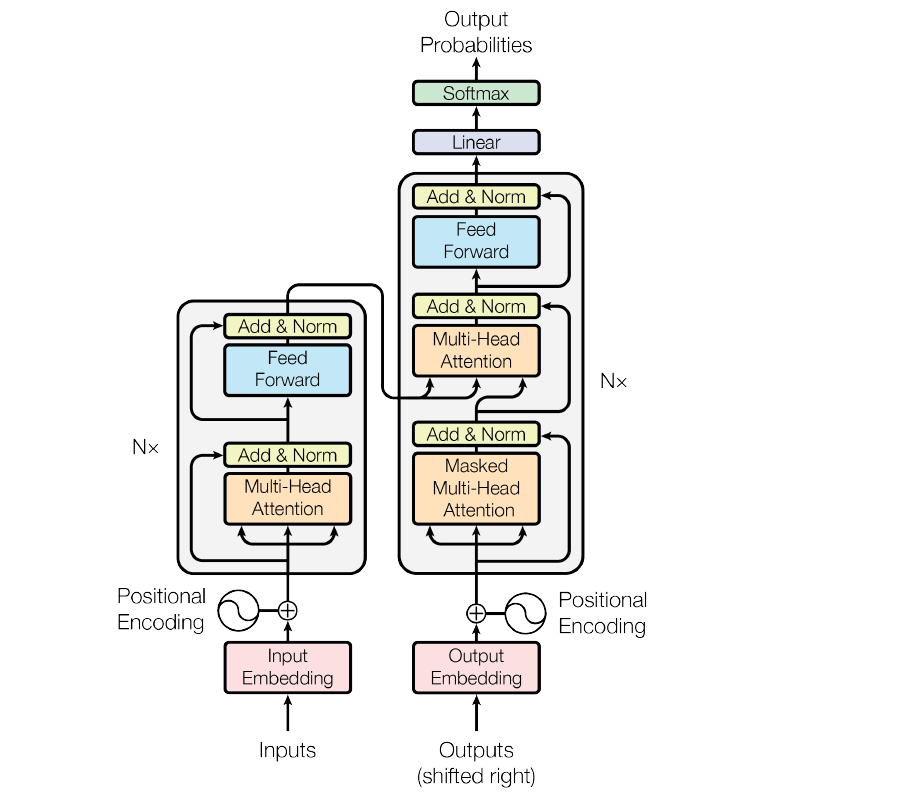


图2-6 Transformer模型的架构

如图2-6所示，在这个位置编码+输入嵌入层之上，还有两个子层：在第一个子层中，有一个多头自注意力层。从位置编码的输出到多头自注意力的输出之间存在一个附加的残差连接，在此之上他们使用了一个归一化层。层归一化是一种类似于批量归一化的技术，在计算归一化统计时不考虑整个小批量数据，而是在计算中考虑了网络同一层中的所有隐藏单元。在第二个子层中，代替多头自注意力，有一个前馈层（如图），其他所有连接都一样。

在解码器方面，除了上述两层之外，还有另一层在编码器结构的顶部使用多头注意力。然后，在一个子层之后是一个线性层和一个softmax层，我们得到了解码器的输出概率。

## 集成学习中的bagging

在训练模型时，无论我们处理的是分类问题还是回归问题，我们都会获得一个函数，该函数接受输入，返回输出，并根据训练数据集进行定义。由于训练数据集的理论差异（因为数据集是来自真实未知潜在分布的观察样本），拟合模型也会有所不同：如果观察到另一个数据集，我们将获得不同的模型。

Bagging的思路很简单：我们想要拟合几个独立的模型并“平均”它们的预测，以获得具有较低方差的模型。然而，在实践中，我们不能拟合完全独立的模型，因为它需要太多的数据。 因此，我们依靠引导样本的良好“近似属性”（代表性和独立性）来拟合几乎独立的模型。

首先，我们创建多个 bootstrap 样本，以便每个新的 bootstrap 样本将充当另一个（几乎）独立的数据集，该数据集来自真实分布。然后，我们可以为每个样本拟合一个弱学习器，最后聚合它们，这样我们就可以“平均”它们的输出，从而获得一个方差小于单个弱学习器的集成模型。粗略地说，由于 bootstrap 样本是近似独立且同分布（i.i.d.）的，因此学习的基本模型也是如此。然后，“平均”弱学习器输出不会改变预期答案，但会减少其方差（就像平均 i.i.d. 随机变量保留预期值但减少方差一样）。

因此，假设我们有 L 个大小为 B 的引导样本（L 个独立数据集的近似值），表示为，我们可以拟合 L 个几乎独立的弱学习器。然后将它们聚合到某种平均过程中，以获得具有较低方差的集成模型。

有几种可能的方法可以聚合并行拟合的多个模型。对于回归问题，可以对所有单个模型的输出取平均以获得集成模型的输出。对于分类问题，每个模型输出的类可以看作是一个投票，获得多数票的类由集成模型返回（这称为硬投票）。仍然对于分类问题，我们还可以考虑所有模型返回的每个类的概率，对这些概率进行平均，并保持具有最高平均概率的类（这称为软投票）。如果可以使用任何相关权重，则平均值或投票可以是简单的或加权的。投票结果可以通过直接相加或者加权得出。

## 网络模型

为了搭建我们的神经网络，我们使用一种名为残差网络的模型。

### 残差神经网络(ResNet)

残差神经网络（ResNet）[62]是一种人工神经网络。残差神经网络利用跳过连接或“抄近路”来跳过神经网络中的某些层。典型的 ResNet 模型是通过包含非线性 (ReLU) 和批归一化(BN)的双层或三层跳跃实现的。在残差神经网络的语境中，非残差神经网络可以描述为普通神经网络。

添加跳跃连接有两个主要原因：(1) 避免梯度消失的问题，或缓解 Degradation（精度饱和）问题；(2) 在适当深度的模型中添加更多层会导致更高的训练错误。在最简单的情况下，仅调整相邻层连接的权重，而上游层没有明确的权重。当跨过单个非线性层或中间层都是线性时，这种方法效果最好。如果不是，那么应该为跳过的连接学习一个显式的权重矩阵。

跳过有效地简化了神经网络，因为在初始训练阶段使用更少的层。同时由于减少了梯度消失的影响和传播的层数更少，神经网络学习速度更快。然后，神经网络在学习特征空间时逐步恢复之前跳过的层。在训练结束时，所有层都扩展，模型会更接近流形，因此可以学习得更快。如果神经网络没有残差部分，那么它就需要探索更多的特征空间，这使得它更容易受到导致其离开流形的扰动的影响，并且需要额外的训练数据来恢复。

### 宽残差神经网络(Wide ResNet)

深度残差神经网络被证明能够扩展到数千层并且仍然具有改进性能的空间。然而，每提高百分之一的准确率都会增加几乎一倍的层数，因此训练非常深的残差神经网络存在减少特征重用的问题，这使得这些神经网络的训练速度非常慢。为了解决这些问题，宽残差神经网络(Wide ResNet)[63]在残差神经网络块的架构的基础上，使用了一种减少深度和增加残差神经网络宽度的新架构。实验证实即使是一个简单的 16 层的宽残差神经网络在准确性和效率上也优于所有以前的残差神经网络在多个数据集上的结果。

## 本章小结

本章我们介绍了本文需要用到的技术工具，分别是知识蒸馏，自知识蒸馏，注意力机制，自注意机制，集成学习中的bagging，以及我们实验中需要用到的网络模型(主要是残差神经网络和宽残差神经网络)。

# 基于自注意力机制的自知识蒸馏

## 基于自注意力机制的自知识蒸馏

正如前文所提到的，一个自知识蒸馏模型的block层的特征图对于提升模型的稳定性非常重要。然而，式子(2-3)中的函数只是把由block层和最深层计算出的损失函数直接相加，而没有考虑各个block层蒸馏信息的不同重要性，这可能会导致一定的信息损失。为了解决这个问题，我提出一种基于自注意力机制的自知识蒸馏方法来考虑各个block层的不同重要性。



图3-1 我们方法的框架图。(i) 神经网络根据深度被分成几个block。(ii) 在每一个block层下方，我们添加上额外的bottleneck层和全连接层，再导向多个分类器。(iii) 然后，我们加上注意力连接，用来衡量每个block层和最深层特征的相关性，注意力模块的作用是计算block层和最深层的相似度。(iv) 其中每个分类器都是单独训练，训练采用三种类型的监督方法。预测出的分数是作为衡量每个block层对总损失函数贡献度的系数。

如图3-1所示，在我们的框架里，我们通过自注意力机制把模型的表示和block层的语义信息联系起来。我们通过利用由自注意力机制得出的系数(或者说权重)来更有效地从模型block层获取信息，以便提升模型的稳定性和泛化能力。

这里我们给出计算出自注意力机制权重的具体步骤。在第一步，我们基于伪教师查询和block层键来计算出权重系数，然后把它们加起来。记相应的权重为，则它们满足。每一个实例都为第个block层保存关联系数。注意力权重由下式计算：

|  |  |  |
| --- | --- | --- |
|  |  | (3-1) |

我们使用softmax函数来衡量每一个block层对于最深层的蒸馏重要性，其中代表最后一层表示所产生的查询，代表第个block层产生的键。正如图2-2所示，我们通过添加投影层计算出和：

|  |  |  |
| --- | --- | --- |
|  |  | (3-2) |

投影层的目的是降维，把教师层的维数降到。同样的道理，为了计算注意力权重，我们使用第个投影层来处理第个block层的键特征，使每一个查询的键和查询的维数相符。此外，在计算注意力权重前，我们通过等式(2-1)引入温度缩放机制。加下来我们通过等式(2-4)得到查询和键的相似性，以此来计算注意力权重。我们的总损失函数可以写成：

|  |  |  |
| --- | --- | --- |
|  |  | (3-3) |

其中代表等式(2-2)中提到的知识蒸馏损失系数(kd loss)，代表我们提出的基于自注意力机制的自知识蒸馏(SKDSA)损失函数。此外，我们修正式(2-3)中的特征损失函数，使损失函数正则化。我们方法(SKDSA)的损失函数可以写成：

|  |  |  |
| --- | --- | --- |
|  |  | (3-4) |

在推断阶段，我们把神经网络最深层的输出作为我们模型预测的结果。本文的主要算法如表3-1所示。



图3-2 自注意力机制的说明。我们在block层(学生层)和最深层(教师层)上加一个适配层，用来提取查询和键。然后我们利用查询和键计算出注意力分数。

表格3-1 算法过程

|  |
| --- |
| 基于自注意力机制的自知识蒸馏。 |
| **输入**： 训练集; 模型参数随即初始化。  **输出**： 模型参数经过充分的训练和更新。  1： while 模型参数还没有收敛：  2： 从训练集中挑出一批数据记为。  3： 将放入特征译码器进行前向传播，获得block层的表示和最深层表示。  4： 根据式(2-5)计算和。  5： 计算的注意力分数和学生键。  6： 利用式(2-7)中损失函数的梯度反向传播来更新参数，其中注意力分数作为权重。  7： end while |

## 和集成学习的关系

在这一部分，我们证明我们的方法和集成学习中bagging[52]的关系。

Bagging是一种是机器学习中的一种集成算法，旨在提高用于统计分类和回归的机器学习算法的稳定性和准确性。它还可以减少方差并有助于避免过度拟合。虽然它通常应用于决策树方法，但它可以与任何类型的方法一起使用。Bagging 是模型平均方法的一个特例。

Bagging的过程是这样的。 (1) 假设训练数据集中有 N 个观测值和 M 个特征。 从训练数据集中随机抽取一个样本进行替换。(2) 随机选择 M 个特征的子集，并使用给出最佳分割的任何特征迭代地分割节点。(3) 让这棵树长到最大。(4) 上述步骤重复 n 次，并根据 n 棵树的预测聚合给出预测。

Bagging的核心思想是通过分开训练多个弱学习者，再通过一个权重投票机制得到一个更加稳定和强大的模型。对于有投票机制的bagging，假设我们产生个神经网络记为，它们的投票权重依次记为，最终预测结果可以表示成下式：

|  |  |  |
| --- | --- | --- |
|  |  | (2-8) |

正如章节2-2所述，因为我们使用注意力机制去将各个block层的信息到最深层，我们模型的最终预测表示可以写成：

|  |  |  |
| --- | --- | --- |
|  |  | (2-9) |

其中满足。令式(2-8)中的(因为我们的神经网络有层)，，，再令，

|  |  |  |
| --- | --- | --- |
|  |  | (2-10) |

把式(2-9)和(2-10)代入(2-8)，我们容易得到：

|  |  |  |
| --- | --- | --- |
|  |  | (2-11) |

由此我们证明了我们模型的输出可以被视为不同模型的集成，这提升了我们模型的泛化能力。而我们模型中的注意力系数又恰好对应于集成学习中的投票权重。

## 本章小结

在这一章中，我们从理论上介绍了知识蒸馏、自知识蒸馏以及由此导出我们方法的框架。我们还从理论上证明了我们方法和bagging的等价性。

# 实验过程

我们使用一个较小的、自知识蒸馏的神经网络在几个图像分类任务上测试我们的模型。我们定量分析了我们的方法。最后，为了更进一步我们的方法我们还进行了消融研究。

## 实验数据集

### CIFAR-10和CIFAR-100数据集

CIFAR-10[57]和CIFAR-100[57]是由8000 万个微小图像数据集的标记子集。 它们由 Alex Krizhevsky、Vinod Nair 和 Geoffrey Hinton 收集。

CIFAR-10 数据集由 10 个类别的 60000 个 32x32 彩色图像组成，每个类别包含 6000 个图像。 有 50000 个训练图像和 10000 个测试图像。

数据集分为五个训练批次和一个测试批次，每个批次有 10000 张图像。测试批次包含来自每个类别的 1000 个随机选择的图像。训练批次包含随机顺序的剩余图像，但一些训练批次可能包含来自一个类的图像多于另一个。在它们之间，训练批次恰好包含来自每个类别的 5000 张图像。

这些类是完全互斥的。比如，汽车类和卡车类之间没有重叠。汽车类包括轿车、SUV 之类的东西，卡车类只包括大卡车。两者都不包括皮卡车。

CIFAR-100类似于CIFAR-10，不同之处是它有 100 个类，每个类包含 600 张图像。每个类有 500 个训练图像和 100 个测试图像。CIFAR-100 中的 100 个类分为 20 个超类。每个图像都带有一个“精细”标签(它所属的类)和一个“粗略”标签(它所属的超类)。

在我们的实验中，我们将CIFAR-10和CIFAR-100分为50000个数据的训练集和10000个数据的验证集。

### Tiny ImageNet数据集

ImageNet 项目是一个大型视觉数据库，用于视觉对象识别软件研究。该项目已经对超过 1400 万张图像进行了手工标注。ImageNet 包含 20，000 多个类别，每个典型类别由数百张图像组成。第三方图像 URL 的注释数据库可直接从 ImageNet 免费获得，但实际图像不属于 ImageNet。自 2010 年以来，ImageNet 项目每年举办一次软件竞赛，即 ImageNet 大规模视觉识别挑战赛 (ILSVRC)，软件程序将竞争正确分类和检测对象和场景。挑战赛使用一千个非重叠类的“修剪”列表。

ImageNet 大规模视觉识别挑战赛(ILSVRC)始于 2010 年，现已成为图像识别的标准基准。Tiny ImageNet Challenge 是一个类似的挑战，数据集较小而且图像类别较少。它包含 200 个图像类，训练集包含100，000个图像，验证集包含10，000个图像，测试集包含10，000个图像。所有图像的大小都是64×64。

### Caltech-UCSD Birds 200数据集

Caltech-UCSD Birds 200 (CUB-200)[64]是一个图像数据集，包含 200 种鸟类（主要是北美）的6033张照片。

### Stanford 40 Actions数据集

Stanford 40 Actions[65]数据集包含人类执行 40 项动作的图像。数据集总共有 9532 张图像，每个动作类有 180-300 张图像。

### Stanford Dogs数据集

Stanford Dogs[66]数据集包含来自世界各地的 120 种狗的图像。该数据集是使用 ImageNet 中的图像和标注构建的，用于细粒度图像分类任务。数据集中包括120种类别的20580张图像。

### MIT Indoor Scene Recognition数据集

室内场景识别是高层次视觉中一个具有挑战性的开放性问题。大多数适用于室外场景的场景识别模型在室内领域表现不佳。主要困难在于，虽然一些室内场景（例如走廊）可以通过全局空间属性很好地表征，但其他的（例如书店）可以通过它们包含的对象更好地表征。更一般地说，为了解决室内场景识别问题，我们需要一个可以利用局部和全局判别信息的模型。

该数据库包含 67 个室内类别，共 15620 张图像。图片的数量因类别而异，但每个类别至少有 100 张图片。

## 实验对比算法基准

### 基准线

我们将我们方法的结果和五种已有的知识蒸馏方法相比较： 做你自己的老师(BYOT)[6]，数据失真引导的自知识蒸馏(DDGSD)[34]，通过教育我自己来提纯我自己(FRSKD)[8]，通过输入转换来做自监督标签增强(SLA-SD)[9]，通过自知识蒸馏预测类别(CS-KD)[7]。我们同样把我们的方法和两种自注意力机制方法相比较：密集而隐含的注意力神经网络(DIANet)[67]，自注意力神经网络(SAN)[68]。

* **BYOT** [6]通过辅助的分类器来利用block层的输出，再通过真实标签和神经网络自身的信号(比如预测的logit和特征图)来训练这些辅助的分类器。
* **DDGSD**[34]对于同一个样本产生不同的变形，在训练神经网络让同一个样本的不同变形产生一致性的预测。
* **FRSKD**[8]利用一个辅助的自教育神经网络为分类神经网络传递精炼后的知识。为了进行自知识蒸馏它同时使用了软标签和特征图蒸馏。
* **SLA-SD**[9]通过输入转换和把输出聚合成教师标签实现自监督标签增强。
* **CS-KD**[7]使用隶属于同一类别的其他样本的logits来做正则化，训练目标是让同一类别的样本产生相似的预测结果。
* **DIANet**[67]给不同的神经网络层添加一个注意力组件，目的是更有效地利用神经网络层间的信息。
* **SAN**[68]利用成对的注意力组件去抽取更有用的信息来引导模型训练。

为了实现FRSKD[8]和SLA-SD[9]我们使用了其官方代码。我们根据相应的文章实现了其对应的模型。我们使用了相同的训练集并且微调了基线模型的超参数。

### 衡量标准

我们使用第一名命中率作为衡量的指标。第一名命中率的意思是测试集中预测概率最大的标签是正确标签的概率。我们使用第一名命中率来衡量模型的泛化能力。

## 训练算法和实现细节

### 神经网络架构

我们使用了当前流行的卷积神经网络架构ResNet[62]和WideResNet[63]，其中包含64个滤波器，第一个卷积层的卷积核大小为，步长为1，填充为1。

### 超参数

所有的神经网络都是从零开始训练，优化方法中随机梯度下降的动量值设为0.9，权重衰减设为，学习率设为0.1。到第100个epoch时，学习率降为最初的1/10; 到第150个epoch时，学习率降为最初的1/100; epoch总数为100。我们设一批数据的数量为128。我们使用常规的数据增强方法，比如图片的反转和随机剪裁。蒸馏温度从{1，2，3，4}中选取。优化的目标是尽可能提高验证集中的第一名命中率。

## 对比结果

### 和其他自知识蒸馏方法的对比

为了进行结果对比，每个实验都重复了3次。我们记录了最后epoch的第一名命中率的平均数和标准差。最好的结果用粗体标注。

表格3-1 在数据集CIFAR-100和Tiny-ImageNet上自知识蒸馏方法的对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法 | CIFAR-100 | | TinyImageNet | |
| WRN-16-2 | ResNet18 | WRN-16-2 | ResNet18 |
| 基准线 | 70.42±0.08 | 74.80±0.52 | 51.25±0.20 | 54.60±0.41 |
| DDGSD | 71.98±0.06 | 76.68±0.56 | 52.30±0.27 | 57.76±0.22 |
| BYOT | 70.28±0.23 | 76.87±0.13 | 51.43±0.08 | 56.76±0.41 |
| CS-KD | 72.64±0.62 | 78.01±0.23 | 52.23±0.16 | 57.72±0.10 |
| SLA+SD | 73.00±0.45 | 77.88±0.30 | 51.64±0.35 | 58.67±0.42 |
| FRSKD | 73.27±0.45 | 78.61±0.30 | 53.08±0.33 | 59.61±0.31 |
| SKDSA | 74.80±0.26 | 80.51±0.12 | 53.21±0.35 | 60.42±0.24 |
| SKDSA +Aug | **76.03±0.21** | **81.77±0.20** | **53.42±0.28** | **62.61±0.38** |

表格3-1显示了两种神经网络架构下不同方法的第一名命中率。在数据集CIFAR-100上，和其他神经网络结构相比，我们的方法将准确率提升了3.38%到5.71%; 在数据集TinyImageNet上，我们的方法将准确率提升了1.94%到5.42%。如果我们仅仅看自知识蒸馏方法，几乎所有的自知识蒸馏模型都可以提升分类准确率。和所有的基准线方法相比，我们的方法比其他自知识蒸馏方法都要好，在图片分类任务中做到了当前最好。这个结果说明使用自注意力机制的自知识蒸馏机制，模型能够更有效地利用block层的知识。此外，增加数据增强模块(比如我们在实验中使用的由DeVries等人[69]提出的挖剪图像方法)，可以在数据集CIFAR-100上提升 1.96%到2.23%，在数据集TinyImageNet上提升0.21%到2.59%，这说明我们的方法和数据增强兼容。因此，我们在细粒度数据集上测试了我们的方法和挖剪图像的增强方法。

表3-2 在细粒度数据集上使用ResNet18架构的各种自知识蒸馏方法效果对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法 | CUB200 | MIT67 | Dogs | Stanford40 |
| 基准线 | 52.72±1.34 | 56.07±1.02 | 64.75±0.20 | 44.30±0.21 |
| DDGSD | 58.79±0.78 | 59.68±0.86 | 69.20±0.27 | 45.84±1.72 |
| BYOT | 58.86±0.51 | 58.41±0.71 | 68.92±0.18 | 48.51±1.02 |
| CS-KD | 64.86±0.1 | 57.42±0.37 | 69.02±0.40 | 47.34±0.31 |
| SLA+SD | 56.47±0.81 | 61.59±1.30 | 67.64±0.34 | 54.67±0.38 |
| FRSKD | 65.39±0.45 | 61.61±0.55 | 70.77±0.33 | 56.00±0.21 |
| SKDSA | 65.20±0.56 | 61.90±0.42 | 71.21±0.35 | 58.02±0.24 |
| SKDSA +Aug | **67.03±0.31** | **65.77±0.45** | **72.84±0.37** | **62.61±0.61** |

类似于表3-1中的结果，从表3-2中可以看到，在细粒度数据集上我们的方法比其他自知识蒸馏方法表现要好。特别要说明的是，在CUB-200-2011数据集中和基准线相比，我们方法将第一名命中率从52.72%提升到67.03%; 在Stanford Dogs数据集中和基准线相比，我们方法将第一名命中率从64.75%提升到72.84%。结果说明在细粒度数据集上我们的方法比其他自知识蒸馏方法表现要好。

表3-3 基于自注意力机制的视觉模型在数据集CIFAR-100上的对比

|  |  |  |  |
| --- | --- | --- | --- |
| 结构 | SKDSA | SAN | DIANet |
| ResNet18 | **80.51±0.23** | 75.90±0.41 | 76.62±0.21 |
| ResNet34 | **80.81±0.34** | 76.40 ±0.41 | 77.1 ±0.26 |
| ResNet50 | **81.13±0.31** | 77.2±0.4 | 78.6±0.32 |

接下来，我们把我们的方法和两种自注意力模型在数据集CIFAR-100上的表现相比。正如表3-3所示，我们的方法显著好于其他两种自注意力方法，这彰显了我们方法的优越心。

## 消融实验及其分析

在消融研究中，每个实验都重复了三次，我们记录第一名命中率在最后一个epoch的期望和标准差。最好的实验结果用粗体字标注。

表3-4 不同方法下，添加和不添加自注意力模块的实验结果对比。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法 | CIFAR-100 | | TinyImageNet | |
| WRN-16-2 | ResNet18 | WRN-16-2 | ResNet18 |
| w/o SA | 71.71±0.15 | 77.80±0.32 | 51.92±0.34 | 57.20±0.47 |
| SKDSA | **74.80±0.21** | **80.51±0.20** | **53.21±0.28** | **60.02±0.38** |
| MixUp | 72.21±0.23 | 78.33±0.38 | 52.82±0.34 | 58.43±0.42 |
| MixUp+SA | **76.73±0.24** | **81.41±0.35** | **53.68±0.22** | **61.44±0.32** |
| SLA | 73.00±0.45 | 77.52±0.30 | 50.77±0.34 | 58.48±0.47 |
| SLA+SA | **76.83±0.52** | **82.81±0.44** | **53.89±0.22** | **63.02±0.31** |

### 自注意力机制的重要性

为了证明自注意力模块在利用block层关系信息的重要性，我们使用一个去掉了自注意力模块的模型作为对照。正如表3-4所示，在去掉了自注意力模块之后，每个实验的分类准确性都大幅下降，这证实了我们的猜测。

### 自注意力方法与数据增强方法的相容性

正如表3-4所示，在给MixUp方法增加自注意力模块后，在架构依次为WRN-16-2，ResNet18，WRN-16-2，ResNet18的模型中，第一名命中率依次提高4.52%，3.08%，0.86%，3.01%。对于SLA方法，结果依次提高3.83%，5.29%，3.12%，4.54%。我们还发现，通过给我们的方法加上增强策略，实验表现同样可以大幅提高。 以上这些结果说明，我们的方法同样和数据增强方法兼容。



图3-1 不同蒸馏温度下数据集CIFAR-100上的的第一名命中率。我们使用ResNet18作为我们的主干。绿柱代表的是自注意力模块没有使用软化标签的结果，黄柱代表的是自注意力模块使用温度4软化标签的结果。

### 蒸馏温度分析

为了研究蒸馏温度对实验结果的影响，我们测试了我们的方法在不同蒸馏温度{1，2，3，4}下的表现。我们同样测试了给自注意力模块使用软化策略的有效性。正如图3-1所示，当蒸馏温度由1提升到2时，预测准确率有所提升，说明我们的模型从知识蒸馏中获益。我们还可以从图3-1中看到使用软化策略来做自注意力机制使结果更好，说明我们的方法可以具备更强的泛化能力，说明温度缩放。

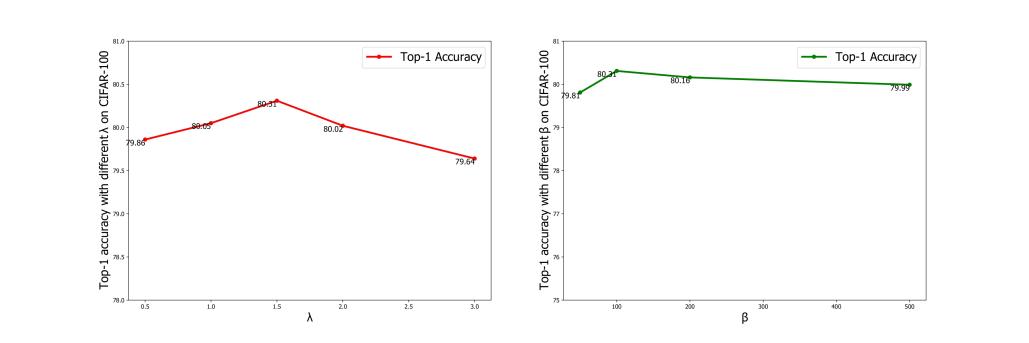


图3-2 在数据库CIFAR-100中，以ResNet18为主干，不同超参数和下的第一名命中率。

### 敏感性分析

我们测试了两个超参数和在不同取值下对我们实验结果的影响。我们测试了取值{0.5，1，1.5，2，3}，取值{50，100，200，500}。我们用ResNet作为骨干在数据集CIFAR-100和TinyImageNet上做了实验。图3-2说明了不同超参数下我们方法的准确性。除了超参数以外，所有的设置都和3.1部分中的设置相同。我们发现我们的方法在超参数的变化下表现稳定，表现最好的超参数是，。

## 本章小结

在这一章中，我们在多个数据集(包括CIFAR-10，CIFAR-100，Caltech-UCSD Bird(CUB200)[64]，Stanford 40[65]，Stanford Dogs (Dogs)[66]，and MIT Indoor Scene Recognition (MIT67))上进行图像分类实验，发现我们方法的性能比其他已有的自知识蒸馏方法结果都要好。我们在消融研究部分说明了自注意机制对提升自知识蒸馏性能很重要，我们的方法和数据增强策略相容，温度缩放可以增强模型的泛化能力，以及找到了相对较好的超参数。

# 总结与展望

## 本文主要内容及结论

在本篇文章中，我们提出一种新的基于自注意力机制的自知识蒸馏方法(SKDSA)并通过和其他现有的自知识蒸馏方法比较证明了它的优越性。在我们的方法中，神经网络最深层扮演教师模型和神经网络block层扮演的学生模型通过自注意力机制联系起来，通过这种方式把神经网络深层的知识迁移到浅层。我们还证明了我们的方法和集成学习的等价性。我们的方法在图片分类任务上做到了当前最优，多项实验证明了我们方法的有效性。

## 本文的主要创新点

本论文的主要创新点可以归纳如下：

我们提出一种新的基于自注意力机制的自知识蒸馏(SKDSA)策略，给位于神经网络深层的教师和神经网络浅层的学生添加自注意力连接，以便区分不同位置的block层对最深层特征表示的不同影响。

我们进一步证明了我们的方法可以被视为一种集成学习机制并为此提供了理论证明，这意味着具有更好的泛化性能。

多项实验结果表明我们方法的图片分类准确性比现有的其他自知识蒸馏方法更好。

## 展望

我们考虑了未来可能的改进，大致有以下方案。

自动调整新引入的超参数。为了平衡交叉熵损失、KL 散度和提示损失，引入了两个超参数和，如式2-3所示。通过实验，我们发现这两个超参数对模型性能有影响。由于计算资源有限而且论文截止时间临近，我们还没有在更广大的范围搜索合适的超参数。在不久的将来，可以使用类似学习率衰减或受动量启发的算法来探索两个超参数的自动调整。

# 致 谢

感谢何琨老师的耐心指导。我作为一名跨专业考来计算机的学生，本身计算机基础较为薄弱，对于自身研究方向的选择也一度困惑。何老师带我尝试了很多方向的小组，包括自然语言处理、知识图谱、社交网络，最终来到了知识蒸馏的小组，并且遇到了意气相投的学长一起做课题。在论文初稿完成后，何老师多次给出中肯的修改建议，甚至熬夜熬到零点半左右。

感谢何老师的博士生张硕玺学长。张硕玺学长数学系出身，理论功底深厚实战能力强。本篇文章的方向性建议是张学长提出来的，在编程和调试的过程中也提出了很多有价值的建议。兴趣爱好上我和张师兄也有很多共同点，包括英超、欧冠、世界杯、篮球、古代史、游戏、金庸小说、娱乐圈八卦等。这些共同的兴趣爱好在科研之余给了我们很多乐趣，深化了我们的友谊，这反过来也让我们科研的合作更加顺畅。

感谢父母对我一直以来物质和精神方面的支持。像我这么大年纪的人，一般的家长早就逼着赶紧结婚生孩子了。但是家父家母非常开明，支持我勇敢追逐梦想，做自己想做的事。我也由衷地希望他们身体健康，幸福吉祥。

三年的研究生生涯紧张而充实，回首往事感慨良多。这三年的学习生涯令我从一个人工智能的门外汉到对这个学科有了略微的了解，当然我对人工智能的热爱是一直不变的。我打算继续在这个方向上攻读博士学位，希望能够从对这个学科有些许的了解，进化成能够对这个学科做出一定的贡献。

# 参考文献

1. Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopadpadakis. Deep learning for computer vision: A brief review. Computational intelligence and neuroscience, 2018.
2. Li Deng and Yang Liu. Deep learning in natural language processing. Springer, 2018.
3. Li Deng and John Platt. Ensemble deep learning for speech recognition. In Proc. Interspeech, 2014.
4. Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.
5. Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10):1345–1359, 2009.
6. Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In ICCV, 2019.
7. Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In CVPR, 2020.
8. Mingi Ji, Seungjae Shin, Seunghyun Hwang, Gibeom Park, and Il-Chul Moon. Refine myself by teaching myself: Feature refinement via self-knowledge distillation. In CVPR, 2021.
9. Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. Self-supervised label augmentation via input transformations. In ICML, 2020.
10. K. Mangalam and M. Salzamann, “On compressing u-net using knowledge distillation,” arXiv preprint arXiv:1812.00249, 2018.
11. Q. Ding, S. Wu, H. Sun, J. Guo, and S.-T. Xia, “Adaptive regularization of labels,” arXiv preprint arXiv:1908.05474, 2019.
12. Y. Tian, D. Krishnan, and P. Isola, “Contrastive representation distillation,” in International Conference on Learning Representations, 2019.
13. S. Hegde, R. Prasad, R. Hebbalaguppe, and V. Kumar, “Variational student: Learning compact and sparser networks in knowledge distillation framework,” ICASSP, 2020.
14. T. Wen, S. Lai, and X. Qian, “Preparing lessons: Improve knowledge distillation with better supervision,” arXiv preprint arXiv:1911.07471, 2019.
15. J. H. Cho and B. Hariharan, “On the efficacy of knowledge distillation,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 4794–4802.
16. I.-J. Liu, J. Peng, and A. G. Schwing, “Knowledge flow: Improve upon your teachers,” Seventh International Conference on Learning Representations, 2019.
17. A. Mishra and D. Marr, “Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy,” arXiv preprint arXiv:1711.05852, 2017.
18. Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, “Learning from noisy labels with distillation,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1910–1918.
19. Y. Xu, Y.Wang, H. Chen, K. Han, X. Chunjing, D. Tao, and C. Xu, “Positive-unlabeled compression on the cloud,” in Advances in Neural Information Processing Systems, 2019, pp. 2561–2570.
20. Q. Xie, E. Hovy, M.-T. Luong, and Q. V. Le, “Self-training with noisy student improves imagenet classification,” CVPR, 2020.
21. F. Sarfraz, E. Arani, and B. Zonooz, “Noisy collaboration in knowledge distillation,” openreview.net, 2019.
22. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” The journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.
23. C. Yang, L. Xie, S. Qiao, and A. L. Yuille, “Training deep neural networks in generations: A more tolerant teacher educates better students,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 5628–5635.
24. L. Yu, V. O. Yazici, X. Liu, J. v. d. Weijer, Y. Cheng, and A. Ramisa, “Learning metrics from teachers: Compact networks for image embedding,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2907–2916.
25. S. Arora, M. M. Khapra, and H. G. Ramaswamy, “On knowledge distillation from complex networks for response prediction,” in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 3813–3822.
26. S. Zagoruyko and N. Komodakis, “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer,” International Conference on Learning Representations, 2016.
27. T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, “Born again neural networks,” ICML, 2018.
28. S. Hahn and H. Choi, “Self-knowledge distillation in natural language processing,” arXiv preprint arXiv:1908.01851, 2019.
29. K. Clark, M.-T. Luong, U. Khandelwal, C. D. Manning, and Q. V. Le, “Bam! born-again multi-task networks for natural language understanding,” ACL, 2019.
30. S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, “Learning a unified classifier incrementally via rebalancing,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 831–839.
31. Luan, H. Zhao, Z. Yang, and Y. Dai, “Msd: Multiself-distillation learning via multi-classifiers within deep neural networks,” arXiv preprint arXiv:1911.09418, 2019.
32. Ting-Bing Xu and Cheng-Lin Liu. Data-distortion guided self-distillation for deep neural networks. In AAAI, 2019.
33. H. Lee, S. J. Hwang, and J. Shin, “Rethinking data augmentation: Self-supervision and self-distillation,” ICML, 2020.
34. Ting-Bing Xu and Cheng-Lin Liu. Data-distortion guided self-distillation for deep neural networks. In AAAI, 2019.
35. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
36. Andrea Galassi, Marco Lippi, and Paolo Torroni. 2020. Attention in Natural Language Processing. IEEE Transactions on Neural Networks and Learning Systems (2020), 1–18.
37. Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. 2015. Describing multimedia content using attention-based encoder-decoder networks. IEEE Transactions on Multimedia 17, 11 (2015), 1875–1886.
38. Feng Wang and David MJ Tax. 2016. Survey on the attention based RNN model and its applications in computer vision. arXiv preprint arXiv:1601.06823 (2016).
39. Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. web (Proceedings of Machine Learning Research, Vol. 37). 2048–2057.
40. Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pages 1480–1489, 2016.
41. Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. arXiv preprint arXiv:1611.01604, 2016.
42. Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In Advances in neural information processing systems, pages 2692–2700, 2015.
43. Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368, 2017.
44. Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, and Weizhu Chen. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. arXiv preprint arXiv:1711.07341, 2017.
45. Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Koˇcisk`y, and Phil Blunsom. Reasoning about entailment with neural attention. arXiv preprint arXiv:1509.06664, 2015.
46. Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1412–1421, Lisbon, Portugal, 2015. Association for Computational Linguistics.
47. Ankur P Parikh, Oscar T ̈ackstr ̈om, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. arXiv preprint arXiv:1606.01933, 2016.
48. Mingi Ji, Byeongho Heo, and Sungrae Park. Show, attend and distill: Knowledge distillation via attention-based feature matching. In AAAI, 2021.
49. Dasarathy B V, Sheela B V. A composite classifier system design: concepts and methodology. Proceedings of the IEEE, 1979, 67(5): 708–713.
50. Kearns M. Learning boolean formulae or finite automata is as hard as factoring. Technical Report TR-14-88 Harvard University Aikem Computation Laboratory, 1988.
51. Schapire, Robert E. The strength of weak learnability. Machine Learning, 1990, 5(2): 197–227.
52. Breiman L. Bagging predictors. Machine Learning, 1996, 24(2): 123–140.
53. Hastie T, Rosset S, Zhu J, Zou H. Multi-class adaboost. Statistics and its Interface, 2009, 2(3): 349–360.
54. Breiman L. Random forests. Machine Learning, 2001, 45(1): 5–3.
55. Ho T K. Random decision forests. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition. 1995, 278–282.
56. Friedman J H. Stochastic gradient boosting. Computational Statistics and Data Analysis, 2002, 38(4): 367–378.
57. Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
58. A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In International Conference on Learning Representations, 2015.
59. Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." International conference on machine learning. PMLR, 2015.
60. Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
61. Cheng, Jianpeng, Li Dong, and Mirella Lapata. "Long short-term memory-networks for machine reading." arXiv preprint arXiv:1601.06733 (2016).
62. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.
63. Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. arXiv preprint arXiv:1605.07146, 2016.
64. C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.
65. Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. Human action recognition by learning bases of action attributes and parts. In 2011 International conference on computer vision, pages 1331–1338. IEEE, 2011.
66. Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, June 2011.
67. Zhongzhan Huang, Senwei Liang, Mingfu Liang, and Haizhao Yang. Dianet: Dense-and-implicit attention network. In AAAI, 2020.
68. Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In 写会议全称CVPR, 2020.
69. Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017.

# 附录1 攻读硕士学位期间取得的研究成果

**投稿论文**

1. Shuoxi Zhang#, **Ye Gao**#**,** Kun He\*.Self-Knowledge Distillation with Self-Attention. 写一下会议全称，IJCAI , 2022, under review. (CCF A类会议)

# 附录2 攻读硕士学位期间参与的科研项目

[1] 中央高校基本科研业务费学科交叉专项：大数据智能与先进计算基础研究平台建设(2019kfyXKJC021), 2019-2021.