**分类号**  **学号**

**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）**

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

**Candidate : GAO YE**

**Major : Computer Software and Theory**

**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

近年来，深度学习模型在诸多领域都取得了巨大的成功。然而，深度学习模型规模庞大，具有数百万（甚至数十亿）个参数，需要大量计算能力并且难以部署在边缘设备上。为了获得更快的训练速度并解决由于缺乏标记数据的问题，人们已经提出知识蒸馏（KD）将从一个模型学到的信息转移到另一个模型。知识蒸馏模型的结构可以表示为学生-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。由于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，为了解决这个问题，人们提出了自知识蒸馏技术。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。

过去的自知识蒸馏模型（比如Be your own teacher: Improve the performance of convolutional neural networks via self distillation，BYOT）将作为学生模型的各个block层一视同仁，忽略了各个层提取的特征对结果的不同作用。为了解决这个问题，在BYOT模型的基础上，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（Self-Knowledge Distillation with Self-Attention, SKDSA）由此被提出。SKDSA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。SKDSA还可以从理论上证明了等价于集成学习。

SKDSA在多个数据集上的图像分类任务上进行了实验并和当前主流的自蒸馏效果进行了对比。SKDSA不仅比其他自知识蒸馏效果要好，而且训练时间也更短。消融实验进一步证明了SKDSA的各个部件都有必要。

**关键词：**注意力机制；自注意力机制；知识蒸馏; 自知识蒸馏

# Abstract

The recent two decades have witnessed a dramatic advent of deep learning, which has achieved excellent performance in various applications in the fields of computer vision, natural language processing, speech recognition, etc.. However, these models are huge in size with millions (and even billions) of parameters, demanding heavy computation power and failing to be deployed on edge devices. To achieve faster speeds and to handle the problems caused by the lack of labeled data, knowledge distillation (KD) has been proposed to transfer information learned from one model to another. Knowledge distillation model can be described as “student-teacher” learning framework, in which teacher networks provide knowledge to student networks. Since training the teacher network may consume enormous computational resources, self-distillation technique is proposed to address this problem. Without training an extra teacher model, the self-knowledge distillation technique utilizes the student network's block layers to distill itself and emerges as a new and powerful technique for the knowledge distillation.

Self-knowledge distillation techniques proposed in the past (i.e. Be your own teacher: Improve the performance of convolutional neural networks via self distillation, namly BYOT) treat all block layers (act as student) equally, neglecting their different impact to the final layer(act as teacher). In this work, we modify BYOT and 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. Moreover, SKDSA takes less training time than other self-distillation methods.

**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](#_Toc98946887)

[A Dissertation Submitted in Partial Fulfillment of the Requirements II](#_Toc98946888)

[for the Master Degree in Engineering/Science （工学/理学硕士） II](#_Toc98946889)

[摘 要 I](#_Toc98946890)

[Abstract II](#_Toc98946891)

[目 录 III](#_Toc98946892)

[1 绪论 1](#_Toc98946893)

[1.1 研究背景与意义 1](#_Toc98946894)

[1.2 国内外研究现状 2](#_Toc98946895)

[1.3 存在的问题 5](#_Toc98946896)

[1.4 论文主要内容 6](#_Toc98946897)

[2 相关理论 9](#_Toc98946898)

[2.1 知识蒸馏 9](#_Toc98946899)

[2.2 作为对比的自知识蒸馏的方法 10](#_Toc98946900)

[2.3 注意力机制 18](#_Toc98946901)

[2.4 数据增强 27](#_Toc98946902)

[2.5 Bagging 29](#_Toc98946903)

[2.6 本章小结 30](#_Toc98946904)

[3 基于自注意力机制的自知识蒸馏 31](#_Toc98946905)

[3.1 基于自注意力机制的自知识蒸馏 31](#_Toc98946906)

[3.2 SKDSA和bagging的等价性证明 33](#_Toc98946907)

[3.3 本章小结 35](#_Toc98946908)

[4 实验过程 36](#_Toc98946909)

[4.1 实验环境 36](#_Toc98946910)

[4.2 实验数据集 36](#_Toc98946911)

[4.3 实验对比算法基准 38](#_Toc98946912)

[4.4 训练算法和实现细节 39](#_Toc98946913)

[4.5 对比结果 39](#_Toc98946914)

[4.6 消融实验及其分析 42](#_Toc98946915)

[4.7 本章小结 44](#_Toc98946916)

[5 总结与展望 46](#_Toc98946917)

[5.1 主要工作总结 46](#_Toc98946918)

[5.2 主要创新点 46](#_Toc98946919)

[5.3 未来工作展望 47](#_Toc98946920)

[参考文献 49](#_Toc98946921)

[附录1 攻读硕士学位期间取得的研究成果 56](#_Toc98946922)

[附录2 攻读学位期间参加的科研项目 57](#_Toc98946923)

[附录3 中英文缩写对照表 58](#_Toc98946924)

[附录4 其它附录 59](#_Toc98946925)

# 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，一种新的基于自注意力机制的自知识蒸馏框架（Self-Knowledge Distillation with Self-Attention, SKDSA）由此被提出。这个框架可以计算出各个学生层的重要性差异，并根据重要性的差异赋予它们不同的注意力分数。为了测试SKDSA的泛化能力，在第四章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SKDSA比当前最好的基准还要好。

## 国内外研究现状

### 知识蒸馏

传统知识蒸馏的主要思想是让学生模型直接模仿教师模型的最终预测，其示意图如图1.1所示。由于传统知识蒸馏对于模型压缩简单而有效，目前已经广泛用于不同的任务和应用。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。Ba等人[10]发现浅层前馈网络可以学习以前由深度网络学习的复杂函数，并实现以前只有深度模型才能实现的精度。在某些情况下，浅层神经网络可以使用与原始深度模型相似的参数总数来学习这些深度函数。Kim等人[11]提出类距离损失，帮助教师网络形成密集聚类的向量空间，使学生网络更容易从中学习。



图1.1 传统的知识蒸馏

### 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，高容量的教师模型的训练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。

一种解决方案是做基于特征的自知识蒸馏。由于模型神经网络的深层比浅层包含更高阶、更抽象的信息，深层部分可以用来向浅层部分蒸馏知识。换言之，基于特征的自知识蒸馏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，其思路如图1.2所示。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。而Hou等人[12]的工作中，相比于直接蒸馏特征，他们使用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。



图1.2 深层教导浅层的自知识蒸馏

基于特征的自知识蒸馏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。因此，另外一些人想到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]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。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。



图1.3 基于数据增强的自知识蒸馏

### 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，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格外关注输入的重点部分，从而使模型更加有效。

在自然语言处理领域，注意力机制在很多任务中都扮演着至关重要的角色，比如机器翻译[60]、语言建模[\*165]、自然语言推断[\*193]、智能问答[\*198]、情感分析[\*220]、语义分析[\*235]、摘要归纳[\*258]等。Vaswani[60]等人提出Transformer架构具有划时代的意义：它彻底摒弃了递归和卷积，完全基于注意力机制进行建模。Dehghani等人[\*165]提出了Universal Transformer (UT)作为Transformer 模型的推广，它将前馈序列模型（如 Transformer）的并行性和全局视野与RNN模型结合起来。Shen等人[\*193]提出了“定向自注意力网络” （Directional selfattention network, DiSAN）来学习句子嵌入，它仅基于所提出的注意力而没有任何 RNN/CNN 结构。DiSAN由具有时间顺序编码的定向自注意力组成，然后将序列压缩为向量表示的多维注意力。Xing等人[\*198] 提出了“分层循环注意网络”（Hierarchical recurrent attention network, HRAN）来在一个统一的框架中对这上下文和关联回答进行建模。HRAN分别使用单词级别的注意力和话语级别的注意力来提取话语内部和话语之间的重点。

注意力机制同样可以应用在计算机视觉任务中，比如动作识别、图像分类[\*142]、图像生成[\*119]、目标检测[\*375]、人物识别[\*380]、分割[\*393]、显着性检测[\*411]等等。Fu等人[\*142] 提出了循环注意卷积神经网络（Recurrent attention convolutional neural network, RA-CNN），它以强化学习的方法在多个尺度上递归地学习的判别区域注意力和基于区域的特征表示。Zhang等人[\*119] 提出了自注意力生成对抗网络（Self-Attention Generative Adversarial Network, SAGAN），它能够完成基于注意力的、远距离依赖的图像生成任务。Kong等人[\*375] 提出特征重配架构，以高度非线性但有效的方式将低级表示与高级语义特征相结合。他们的架构由全局注意力和局部重构组成，能够在全局和局部的不同空间位置和尺度上收集面向任务的特征。Li等人[\*380]提出Harmonious Attention CNN (HA-CNN)模型，通过联合学习软像素注意力和硬区域注意力以及同时优化特征表示，以便优化不受控制图像中人员的识别。

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，SKDSA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。

## 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，SKDSA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。

## 论文主要内容

论文共分为5章内容，章节内容之间的关系图如图1.4所示。

图1.4 组织结构图

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SKDSA算法。

第二章 需要用到的工具和技术：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：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SKDSA。再证明SKDSA和bagging的等价性。

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

第五章 总结与展望：SKDSA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。随后证明了SKDSA和bagging的等价性。随后进行了消融研究。

# 基础理论与相关知识

这一章节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，基于自注意机制的自知识蒸馏(Self-Knowledge Distillation with Self-Attention，简称SKDSA)方法由此提出，并随后证明SKDSA可以被视为集成学习中的bagging。

下文中的记号。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

SKDSA方法在大类上属于知识蒸馏，所以先介绍知识蒸馏的概念。



图2.1 知识蒸馏示意图

首先介绍传统知识蒸馏的概念。如图2.1所示，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，一张图片的属于第个类别概率为：

|  |  |  |
| --- | --- | --- |
|  |  | （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，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。

## 作为对比的自知识蒸馏的方法

### BYOT

SKDSA方法是对BYOT方法的修改，所以在这里介绍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，SKDSA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，SKDSA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”。BYOT模型将“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。BYOT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） |

BYOT使用两个超参数和来平衡式（2-3）（2-4）（2-5）。



图2.2 BYOT的框架图[6]

### DDGSD

SKDSA对比试验的基准之一是Xu等人[32]提出的DDGSD。

Xu等人[32]提出了一种自蒸馏机制，它无需其他模型的帮助，即可根据数据表示的一致分布有效地优化单个网络。通常，常见的数据失真技术（例如，图片的随机镜像/裁剪、旋转、多尺度）仅从现有训练样本的邻域扩大额外的虚拟样本，这使得网络可以看到更多失真的训练样本以提高模型泛化能力。在另一个观点中，良好的泛化要求模型能够在相同训练数据的不同扭曲版本中保持相似或不变的类概率预测（后验分布）和全局特征表示（特征分布）。因此，为了进一步挖掘单个网络的潜在容量，他们设计了一种新的数据失真引导自提炼机制，以最小化数据到数据知识转移的后验概率和特征分布的差异。具体说来，他们利用Kullback Leibler (KL)散度约束来衡量不同扭曲版本的后验概率分布之间的匹配，其方式类似于不同模型之间的知识蒸馏[4]，并采用经验最大平均差异(MMD)作为衡量这些失真版本之间特征分布一致性的非参数指标。在训练基线分类任务时，他们仍然使用硬标签保留传统的交叉熵损失项（即 softmax 损失）。上述三个损失项可以集成在一起作为优化目标，以端到端的方式从头开始实现单网络的自蒸馏。这种新机制可以驱动单个网络自动学习更多固有表示进行泛化，有效抑制学习到的特征朝向单方向依赖。

### FRSKD

SKDSA对比试验的基准之一是Ji等人[8]提出的FRSKD。

过去的知识蒸馏方法依赖于辅助网络，其复杂度与分类器网络相同或更低；因此，无论是通过作为卷积层输出的特征还是分类器网络的软标签，都很难生成精炼的知识。此外，基于数据增强的方法容易丢失样本之间的局部信息，例如不同扭曲的样本或旋转的样本。因此，利用特征蒸馏面临一定的困难。

为了解决现有自知识蒸馏的局限性，Ji等人[8]提出了一种新颖的自知识蒸馏方法，即通过自知识蒸馏（FRSKD）进行特征细化，它引入了一个辅助的自学网络，可以将精炼的知识转移到分类器网络。图2.3显示了 FRSKD 与现有知识蒸馏方法之间的差异。Ji等人[8]提出的方法 FRSKD 可以利用软标签和特征图蒸馏来进行自知识蒸馏。



图2.3 FRSKD和其他知识蒸馏方法的区别[8]

### SLA-SD

SKDSA对比试验的基准之一是Lee等人[9]提出的SLA-SD。

Lee等人[9]发现，多任务学习方法强制原始任务的主要分类器在自监督任务的转换方面保持不变。例如，当使用旋转作为自监督时（将每个图像旋转 0、90、180、270 度同时保留其原始标签），可以强制初级分类器学习对图像旋转不变的表示方法。强制这种不变性可能会导致任务的复杂性增加，因为转换可能会在很大程度上改变样本的特征和/或识别对象的有意义信息。因此，这可能会损害整体表示学习，并降低主要全监督模型的分类准确性（参见第 3.2 节中的表 1）。

为了解决这一问题，Lee等人[9]提出了一个简单而有效的想法，即学习一个关于原始标签和自监督标签的联合分布的统一任务，而不是像先前文献一样将其视为的两个独立任务。例如，当在 CIFAR-10（10 个标签）上使用旋转自监督（4 个标签）进行训练时，作者学习了所有可能组合（即 40 个标签）的联合概率分布。

作者将这种标签增强方法称为自监督标签增强 (SLA)，在不假设原始标签和自监督标签之间的关系的情况下，不会具有任何转换不变性。此外，由于作者为每个转换分配不同的自监督标签，因此可以在测试时通过聚合所有转换来进行预测，如图2.4所示。这可以使用单个模型提供（隐式）集成效果。



图2.4 聚合和自蒸馏[9]

### CS-KD

SKDSA对比试验的基准之一是Yun等人[7]提出的FRSKD。

正则化深度神经网络的预测分布可能是有效的，因为它包含模型最简洁的知识。Yun等人[7]利用暗知识的概念（即DNN 做出的错误预测的知识）为深度模型开发一种新的输出正则化器。它的重要性通过所谓的知识蒸馏 (KD) [4]得到了证明。

虽然大多数工作使用知识蒸馏将教师网络学到的暗知识转移到学生网络，作者在训练单个网络时对暗知识本身进行正则化，即自知识蒸馏。具体来说，Yun等人[7]提出了一种新的正则化技术，即按类别自知识蒸馏 (CS-KD)，它匹配或提取相同标签的不同样本之间的深度神经网络的预测分布，如图2.5所示。可以预期，如果样本属于同一类，作者所提出的正则化方法会迫使深度神经网络产生类似的错误预测，而传统的交叉熵损失没有考虑预测分布的这种一致性。此外，它可以同时实现两个理想的目标：防止过度自信的预测和减少类内变化。



图2.5 CS-KD示意图

### DIANet

SKDSA对比试验的基准之一是Huang等人[67]提出的DIANet。

注意力是一种选择性关注一小部分信息而忽略其他可感知信息的认知过程，目前已被用于有效地缓解神经网络从句子中学习大量信息上下文的问题、图像和视频。特别是在计算机视觉中，深度神经网络 (DNN) 结合了模仿注意力机制的特殊算子，可以有效地处理图像中的信息区域。这些算子被模块化并作为注意力模块插入网络。

过去的注意力方法是设计一个新的注意力模块并将它们单独插入网络。一般来说，注意力模块可以分为三个部分：提取、处理和重新校准。首先，插件模块提取网络的内部特征，可以压缩通道信息或空间信息。接下来，该模块处理提取并生成掩码，以通过全连接层、卷积层来衡量特征的重要性。最后，将掩码重新校准特征。以前的注意力模型将重点放在设计有效的方法来处理提取的特征，他们的共同特点是将注意力模块被单独插入整个深度神经网络的每一层。

Huang等人[67]提出了一个新颖而简单的框架，该框架在不同的网络层中共享一个注意力模块，以鼓励分层信息的集成，这个参数共享模块被称为密集和隐式注意 (DIA) 单元。DIA 单元的结构和计算流程如图2.6所示。在 DIA 单元中也有三个部分：(1)提取、(2)处理、(3)重新校准。其中第二部分是 DIA 单元中对网络注意力进行建模的主要模块，是DIANet方法的关键创新，其中注意力模块的参数是共享的。

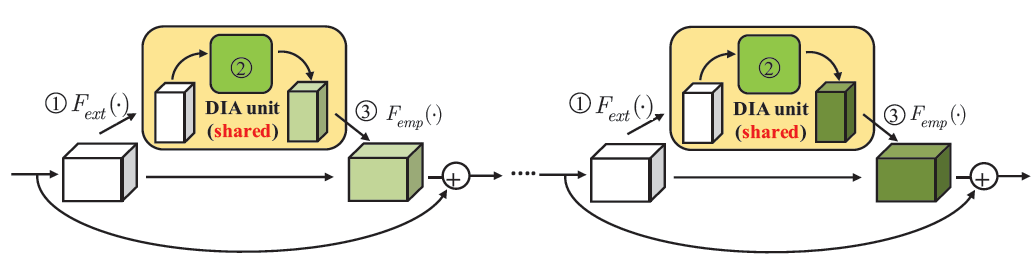


图2.6 残差网络中的DIA单元

此方法的优点如下：(1) 如图2.6所示，DIA 单元与网络主干平行放置，并与同一阶段的所有层共享（具有相同空间大小的连续层的集合)以改善不同深度层的交互。 (2) 由于 DIA 单元是共享的，随着网络深度的增加，来自 DIA 单元的参数增量的数量大致保持不变。

### SAN

SKDSA对比试验的基准之一是Zhao等人[68]提出的SAN。

在用于图像识别的卷积网络中，网络层执行两个功能。第一个是特征聚合，其中卷积操作通过组合来自内核挖掘的所有位置的特征来执行。第二个功能是特征变换，它由连续的线性映射和非线性标量函数执行：这些连续的映射和非线性操作会破坏特征空间并产生复杂的分段映射。

SAN的基础之一是特征聚合和特征转换这两个功能可以解耦。如果有一种执行特征聚合的机制，那么特征变换可以由分别处理每个特征向量（每个像素）的感知器层来执行。感知器层由线性映射和非线性标量函数组成。因此，SAN侧重于特征聚合。

卷积算子通过一个固定内核执行特征聚合，该内核应用预训练的权重来线性组合来自一组附近位置的特征值。权重是固定的，不适应特征的内容。而且由于每个位置都必须使用专用的权重向量进行处理，因此参数的数量与聚合特征的数量呈线性关系。Zhao等人[68]提出了一些替代的聚合方案，并构建了高性能的图像识别架构，将特征聚合（通过自注意）和特征转换（通过元素感知器）交错。Zhao等人[68]提出了许多替代的聚合方案，并构建了高性能的图像识别架构，将特征聚合（通过自注意力）和特征转换（通过元素感知器）结合在一起。

## 注意力机制

SKDSA是添加了自注意机制的BYOT方法，所以这里介绍一下注意力机制。

### 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，假设一个人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，他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，他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。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。 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。

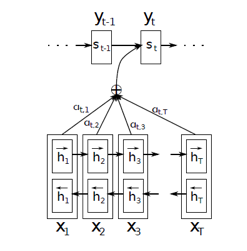


图2.7 基于注意力机制的翻译模型

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，下面将列出它们的数学方程。

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。这是因为模型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，由此引出局部注意力概念。

直觉上，当一个人试图从任何给定的信息中推断出某些东西时，他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：

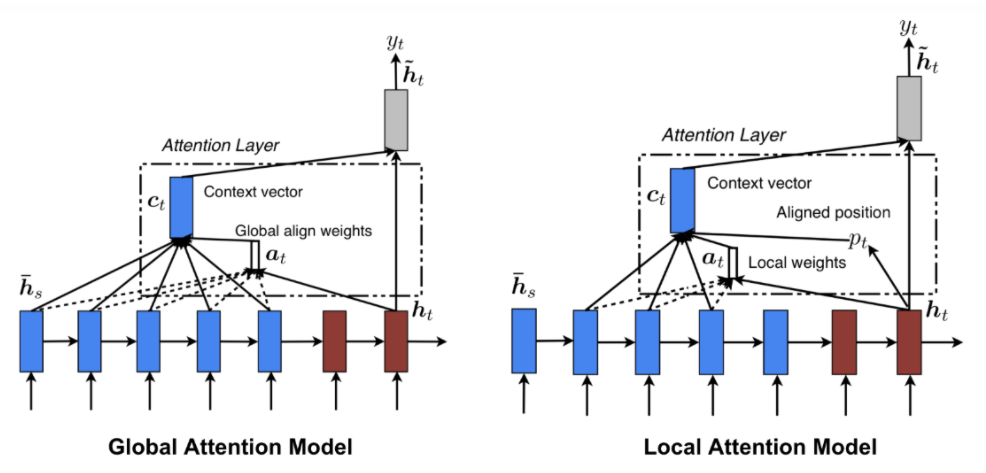


图2.8 全局注意力模型和局部注意力模型

### 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，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。

当一个人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。

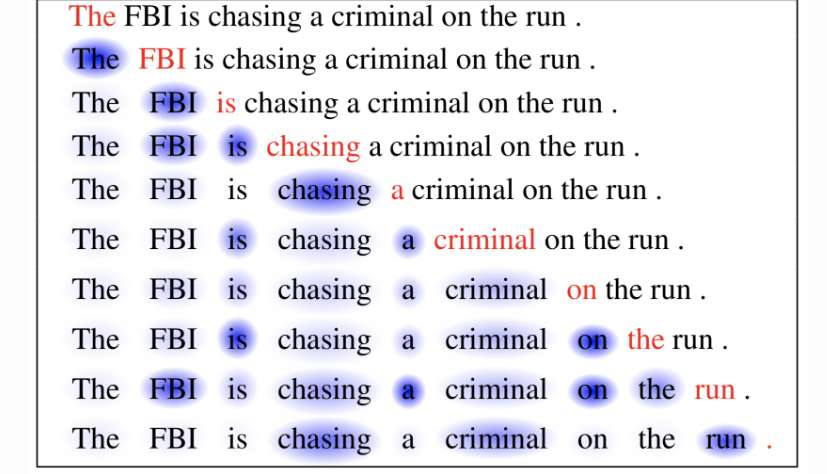


图2.9 自注意力机制

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。可以使用矩阵乘法计算出这些向量。

每当需要计算目标词相对于输入嵌入的注意力时，首先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。

如果嵌入的维度是而想要维度的键向量，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，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。最终可以得到不同的矩阵，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。

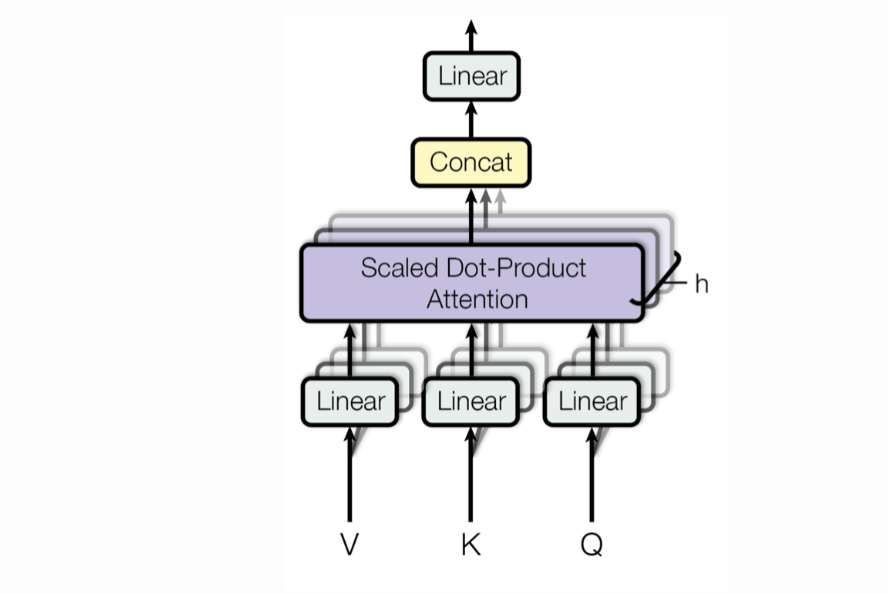


图2.10 多头注意力机制包含几个并行的注意力层

图2.6展示了transformer的结构。从中可以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。

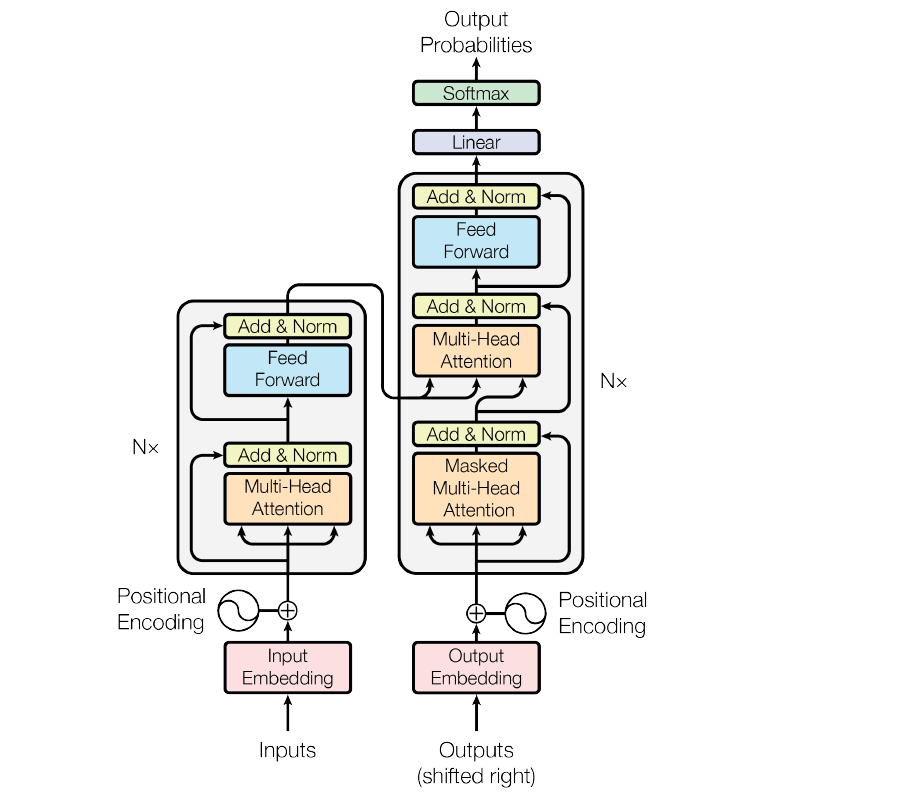


图2.11 Transformer模型的架构

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，可以得到解码器的输出概率。

## 数据增强

SKDSA使用了图像裁剪的数据增强方法，所以介绍相关的数据增强方法。

有许多研究希望通过将深度卷积网络应用于计算机视觉任务来改进当前的基准。提高这些模型的泛化能力是最困难的挑战之一。泛化性是指模型在以前见过的数据（训练数据）与以前从未见过的数据（测试数据）上评估时的性能差异。泛化性差的模型过度拟合了训练数据。发现过拟合的一种方法是在训练期间绘制每个时期的训练和验证准确度。

为了构建有用的深度学习模型，验证误差必须随着训练误差而不断减小。数据增强是实现这一目标的一种非常强大的方法。增强的数据将代表一组更全面的可能数据点，从而最小化训练集和验证集之间的距离，以及任何未来的测试集。

数据增强是减少过度拟合的技术之一，它的特点是从问题的根源，即训练数据集开始处理过度拟合。这是基于可以通过增强从原始数据集中提取更多信息的假设。这种数据增强增强通过数据扭曲或过采样人为地增加了训练数据集的大小，方式包几何变换、颜色转换、随机擦除、对抗训练和神经风格迁移等等。过采样增强创建合成实例并将它们添加到训练集中。例如，GAN 样本可以与随机裁剪堆叠在一起，以进一步扩大数据集。图像数据增强的大致分类如下图所示。



图2.12 图像数据增强分类

在讨论图像增强技术之前，首先需要考虑为什么图像识别任务如此困难。在经典的图像分类任务比如猫狗分类中，图像识别软件必须克服视点、照明、遮挡、背景、比例等问题。数据增强的任务是将这些平移不变性增加到数据集中，这样即使存在这些挑战，生成的模型也能表现良好。

一个普遍接受的概念是，更大的数据集会产生更好的深度学习模型。然而，由于收集和标记数据的手动工作，组装庞大的数据集可能是一项非常艰巨的任务。有限的数据集是医学图像分析中一个特别普遍的挑战。鉴于大数据，深度卷积网络已被证明对于医学图像分析任务非常强大。这激发了 CNN 在医学图像分析任务中的使用，例如肝脏病变分类、脑部扫描分析、皮肤病变分类的持续研究等等。医学图像处理所研究的许多图像都来自计算机断层扫描 (CT) 和磁共振成像 (MRI) 扫描，这两种方法的收集都非常昂贵且劳动强度大。由于疾病的罕见性、患者隐私、医学专家对标记的要求以及进行医学成像过程所需的费用和人工工作，构建大型医学图像数据集尤其困难。这些困难促使人们从医学图像分类的应用角度对图像数据增强，特别是基于 GAN 的过采样进行了许多研究。

许多关于数据增强有效性的研究利用流行的学术图像数据集来对结果进行基准测试。这些数据集包括 MNIST 手写数字识别、CIFAR-10/100、ImageNet、tiny-imagenet-200、SVHN（街景门牌号码）、Caltech-101/256、MIT places、MIT-Adobe 5K 数据集、Pascal VOC 和 斯坦福汽车。开源数据集的扩展为研究人员提供了各种各样的案例来比较数据增强技术的性能结果。大多数这些数据集（例如 ImageNet）将被归类为大数据。许多实验将自己限制在数据集的一个子集上，以模拟有限的数据问题。

除了关注有限的数据集之外，人们还考虑类不平衡问题和如何用数据增强解决过采样问题。类不平衡描述了多数样本与少数样本的比例偏斜的数据集。接下来将说明如何使用数据增强来完成图像数据中的类平衡过采样。

深度学习和神经网络模型的许多方面都与人类智能进行了比较。比如说，如果两个人正在尝试学习如何弹奏吉他，而一个人已经知道如何弹奏钢琴，那么弹奏钢琴的人似乎会更快地学会弹奏吉他。类似于学习音乐，可以对 ImageNet 图像进行分类的模型在 CIFAR-10 图像上的表现可能比具有随机权重的模型更好。

数据增强类似于想象或梦境。人类根据过往的经验想象出不同的场景。想象帮助人们更好地了解这个的世界。诸如 GAN 和风格转换之类的数据增强方法可以“想象”对图像的更改，以便更好地理解它们。

## Bagging

由于需要证明SKDSA方法和bagging的等价性，所以需要先介绍bagging的概念。

在训练模型时，无论处理的是分类问题还是回归问题，都可以获得一个函数，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，可以获得不同的模型。

Bagging的思路很简单：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，不可能拟合完全独立的模型，因为它需要太多的数据。因此，可以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。

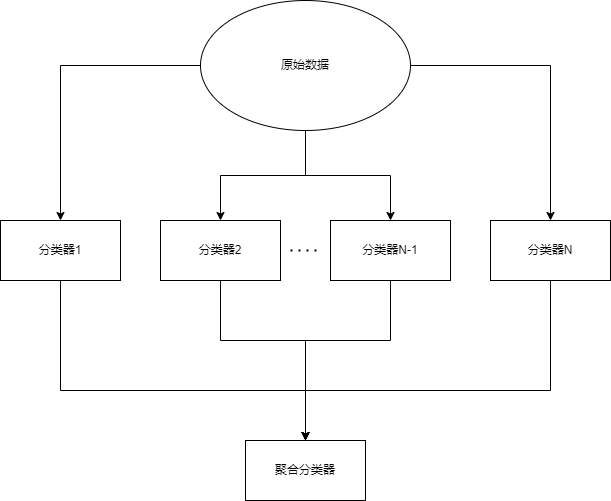


图2.13 bagging示意图

Bagging的过程如图2.8所示。首先，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，可以为每个样本拟合一个弱学习器，最后聚合它们，这样就可以“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，它们可以拟合 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，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。

最后，bagging 的一大优势是它可以并行化。由于不同的模型彼此独立，因此如果需要，可以使用密集的并行化技术。

## 其他术语

### 交叉熵

交叉熵是一种常用于机器学习中的损失函数。它是对给定随机变量或事件集的两个概率分布之间差异的度量。

在信息论中，低概率事件有更多的信息量，高概率事件有更少的信息量。假设事件的概率为，则事件包含的信息量如下所示：

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

熵是表示某个随机事件所需的比特数。不均匀分布具有较低的熵，而均匀的概率分布具有较高的熵。

具有个离散状态中的一组的随机变量和它们的概率的熵值为：

|  |  |  |
| --- | --- | --- |
|  |  | （2-14） |

交叉熵建立在信息论中的熵的概念之上。它计算一个概率分布与另一个概率分布相比，表示平均事件所需的比特数。

两个概率分布和之间的交叉熵，可以记为，其中是交叉熵函数，代表是目标分布，代表目标分布的估计。由下式计算：

|  |  |  |
| --- | --- | --- |
|  |  | （2-15） |

此计算适用于离散概率分布。交叉熵值将是一个以比特为单位的正数，如果两个概率分布和相同，则结果将等于分布的熵。

交叉熵被广泛用作分类模型的损失函数。分类问题是涉及一个或多个输入变量和类别标签预测的问题。分类模型训练集中的每个样本都有一个已知的类标签，概率为1，所有其他的类标签的概率为0。模型可以估计样本属于每个类别标签的概率。 然后可以使用交叉熵来计算两个概率分布之间的差异。

### KL散度

在很多情况下，我们需要比较两个概率分布。具体来说，我们可能有一个随机变量和该变量的两个不同概率分布，例如真实分布和对真实分布的估计。在这种情况下，量化分布之间的差异可能很有必要——这通常被称为计算两个概率分布之间的距离的问题。

人们想到的一种方法是计算两个概率分布之间的散度。散度是不对称的，计算分布和的散度将给出与和散度不同的值。

散度是信息论中许多不同计算的重要基础，普遍用于机器学习中的计算。 例如，互信息（信息增益）和交叉熵。散度也直接用作复杂建模问题的工具，例如在优化生成对抗网络 (GAN) 模型时逼近目标的概率分布。

Kullback-Leibler散度就是信息论中常用的一种散度，它以 Solomon Kullback 和 Richard Leibler方法的两位作者命名，有时也被称为“相对熵”。它量化了一个概率分布与另一个概率分布的差异程度。两个分布和之间的KL散度通常表示为。它的计算方式为：

|  |  |  |
| --- | --- | --- |
|  |  | （2-16） |

KL散度值的直观理解是，当中某个事件的概率很大，而中相同事件的概率很小时，则存在很大的散度；当来自的概率很小而来自的概率很大时，也存在很大的散度，但没有第一种情况那么大。它可用于测量离散和连续概率分布之间的差异，在后一种情况下，计算事件的积分而不是离散事件的概率之和。

## 本章小结

本章介绍了SKDSA方法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，一种基于自注意力机制的自知识蒸馏方法来考虑各个block层的不同重要性由此被提出。



图3.1 SKDSA的框架图

SKDSA的框架图如图3.1所示，(1) 神经网络根据深度被分成几个block。(2) 在每一个block层下方，添加上额外的bottleneck层和全连接层，再导向多个分类器。(3) 然后，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。(4) 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。

SKDSA模型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。SKDSA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。

这里给出计算出自注意力机制权重的具体步骤。第一步，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） |

SKDSA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3.2所示，SKDSA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；然后通过查询和键计算出注意力分数。下式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，SKDSA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，通过式（2-1）引入温度缩放机制。加下来通过式（3-1）得到查询和键的相似性，以此来计算注意力权重。SKDSA的总损失函数可以写成：

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

其中代表式（2-2）中提到的知识蒸馏损失系数(kd loss)，代表基于自注意力机制的自知识蒸馏(SKDSA)损失函数。此外，通过修正式（2-6）中的特征损失函数，使损失函数正则化。SKDSA的损失函数可以写成：

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

在推断阶段，SKDSA把神经网络最深层的输出作为模型预测的结果。SKDSA的算法步骤如表3.1所示。



图3.2 对SKDSA中自注意力模块的说明

表3.1 SKDSA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 |

## SKDSA和bagging的等价性证明

这一部分将证明SKDSA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，假设有个神经网络记为，它们的投票权重依次记为，最终预测结果可以表示成下式：

|  |  |  |
| --- | --- | --- |
|  |  | （3-5） |

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

|  |  |  |
| --- | --- | --- |
|  |  | （3-6） |

其中满足。令式(3-5)中的(因为神经网络有层)，，，再令，

|  |  |  |
| --- | --- | --- |
|  |  | （3-7） |

把式（3-6）和（3-7）代入(3-5)，可以得到：

|  |  |  |
| --- | --- | --- |
|  |  | （3-8） |

由此证明了SKDSA模型的输出可以被视为不同模型的集成，这提升了SKDSA模型的泛化能力。而SKDSA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

这一章从理论上介绍了知识蒸馏、自知识蒸馏以及由此导出SKDSA方法。接下来还从理论上证明了SKDSA方法和bagging的等价性。

# 实验过程

实验使用一个较小的、自知识蒸馏的神经网络在几个图像分类任务上测试SKDSA模型。接下来为了更进一步测试SKDSA还进行了消融研究。

## 实验环境

SKDSA算法使用python中的pytorch工具编写实现。实验在一块Nvidia 2080ti显卡上进行训练和测试。

## 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)。

在实验中，CIFAR-10和CIFAR-100被分为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100000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。

由于实验室的计算资源有限，SKDSA难以在数据集ImageNet上进行测试；只好在Tiny ImageNet数据集上进行了测试，它是ImageNet的一个很好的替代品。

### 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，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

### 实验对比算法基准

将SKDSA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)[32]，通过教育自己来提纯自己(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]。实验同样把SKDSA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**[32]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。

实验中的FRSKD[8]和SLA-SD[9]使用了其官方代码，根据相应的文章实现了其对应的模型。这里的实验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

第一名命中率被用来作为衡量的指标。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。实验使用第一名命中率来衡量模型的泛化能力，实验的优化的目标是尽可能提高验证集中的第一名命中率。

## 训练算法和实现细节

### 神经网络架构

SKDSA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。优化方法选择随机梯度下降方法（Stochastic gradient descent, SGD）。

### 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（weight decay）设为，动量（momentum）设为0.9。初始学习率(learning rate)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250。设一批训练样本的数量（batch size）为128。实验只使用了一种常规的数据增强方法，即随机剪裁（cutout）。蒸馏温度从{1，2，3，4}中选取，超参数从{0.5，1，1.5，2，3}中选取，超参数从{50，100，200，500}中选取。

## 对比结果

### 在数据集CIFAR-100和Tiny-ImageNet上的实验结果

为了进行结果对比，每个实验都重复了3次。实验结果记录了最后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4.1 在数据集CIFAR-100和Tiny-ImageNet上的实验结果对比

| 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4.1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，SKDSA将准确率提升了3.38%到5.71%; 在数据集TinyImageNet上，SKDSA将准确率提升了1.94%到5.42%。如果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，SKDSA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SKDSA方法和数据增强兼容。

需要说明的是，尽管 SLA 引入了很多数据增强策略，但SKDSA侧重于自注意力机制而不是数据增强，所以表4.1中没有FRSKD+SLA这项对比。由于FRSKD+Aug的训练时间比SKDSA的训练时间要长得多，所以比较不公平。因此，表4.1中没有加入FRSKD+Aug这项对比。

### 在细粒度数据集上的实验结果

表4.2 在细粒度数据集上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** |

实验还在细粒度数据集上测试了SKDSA和挖剪图像的增强方法。类似于表4.1中的结果，从表4.2中可以看到，在细粒度数据集上SKDSA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，SKDSA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，SKDSA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SKDSA方法比其他自知识蒸馏方法表现要好。

除了预测的准确率高，SKDSA模型的训练速度和推理速度也比前三个基准线方法（DDSGD, BYOT, CS-KD）略快，比后两个基准线方法快很多。尤其需要说明的是，SKDSA方法的训练时间仅仅是FRSKD训练速度的四分之一。

### 不同自注意力机制的实验结果

表4.3 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 |

接下来把SKDSA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4.3所示，SKDSA方法显著好于其他两种自注意力方法，这彰显了SKDSA方法的优越性。

## 消融实验及其分析

在消融研究中，每个实验都被重复了三次，实验结果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4.4 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，实验使用一个去掉了自注意力模块的模型作为对照。表4.4第一行的w/o SA是去掉自注意力机制的SKDSA（也就是BYOT），第三行的MixUp代表另一种数据增强方法。可以看出，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SKDSA的每一部分都很有必要。

### 自注意力方法与数据增强方法的相容性

正如表4.4所示，在给MixUp方法（一种数据增强方法）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%。可以发现，通过给SKDSA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，SKDSA方法同样和数据增强方法兼容。



图4.1 不同蒸馏温度下的结果以及是否使用温度缩放的结果

### 蒸馏温度分析

为了研究蒸馏温度对实验结果的影响，实验测试了SKDSA方法在不同蒸馏温度{1，2，3，4}下的表现，实验同样测试了给自注意力模块使用温度缩放策略的有效性。如图4.1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SKDSA模型可以从知识蒸馏中获益。图4.1中绿柱代表的是自注意力模块没有使用温度缩放的结果，黄柱代表的是自注意力模块使用温度4缩放的结果，由此可以看出使用温度缩放策略来做自注意力机制使结果更好。

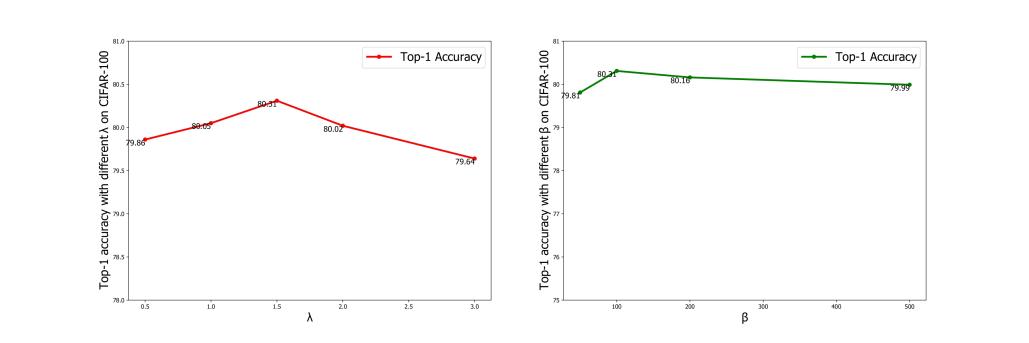


图4.2 不同超参数和下的第一名命中率

### 敏感性分析

这部分实验测试了两个超参数和在不同取值下对实验结果的影响。测试的取值范围是{0.5，1，1.5，2，3}，取值范围是{50，100，200，500}。实验以ResNet作为骨干在数据集CIFAR-100和TinyImageNet上做了实验。图4.2说明了不同超参数下SKDSA方法的分类准确性。除了超参数以外，所有的设置都和文章4.3部分中的设置相同。可以发现SKDSA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

这一章讨论了SKDSA算法的效果。使用SKDSA在多个学术界和工业界的数据集(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，并和当前几种主流的自知识蒸馏算法（包括DDGSD，BYOT，CS-KD，SLA+SD，FRSKD等）作为对比，可以发现SKDSA方法的性能比其他已有的自知识蒸馏方法结果都要好。

另外，这一章还对框架中的自注意力模块、自注意力机制与数据增强机制的相容性、蒸馏温度等分别进行了消融实验分析。消融分析证明了自注意机制对提升自知识蒸馏性能很重要，SKDSA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

## 主要工作总结

为了解决深度学习规模庞大难以训练的问题，在迁移学习的启发下，人们已经提出知识蒸馏将从一个较大模型（教师模型）学到的信息迁移到另一个较小模型（学生模型）。为了教师网络规模庞大难以训练的问题，人们又提出了自知识蒸馏技术。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。

过去的自知识蒸馏模型（比如BYOT）将作为学生模型的各个block层一视同仁，忽略了作为教师模型的网络最深层对它们的不同影响。为了解决这个问题，在BYOT模型的基础上，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（SKDSA）由此被提出。在SKDSA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。SKDSA还可以从理论上被证明等价于集成学习。

SKDSA在多个数据集上的图像分类任务上进行了实验并和当前主流的自蒸馏效果进行了对比。SKDSA不仅比其他自知识蒸馏效果要好，而且训练时间也更短。消融实验进一步证明了SKDSA的各个部件都有必要。

## 主要创新点

本论文的主要创新点可以归纳如下：

由于过去的基于特征的自知识蒸馏方法忽视了不同层特征信息之间重要性的差别，这可能会导致大部分暗知识的损失。SKDSA模型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。多项实验结果表明SKDSA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。

SKDSA方法看似很简单，但有充分的根据。尽管过去人们对自知识蒸馏和自注意力机制都有了充分的研究，但SKDSA是第一个将它们结合起来的。SKDSA方法使用注意力机制作为一种手段来考虑每个中间层对最终特征表示的不同的重要性，从而挖掘端到端深度学习中的暗知识。

SKDSA进一步被证明了可以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。虽然这个证明看起来显而易见，但是这个证明是必要的，因为它展示SKDSA方法与 bagging 的相似之处，其中弱学习器和投票过程可以提供更稳定和可靠的模型。

## 未来工作展望

SKDSA算法虽然在实验上取得了不错的效果，但仍然还有很多地方值得改进和扩展。未来可能的改进大致有以下方案。

1. 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（2-6）所示。实验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，SKDSA模型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。

第二，尝试把学生层模型映射到较小的空间。

在传统的知识蒸馏模型中，教师模型比学生模型大很多；但是在自知识蒸馏中，作为学生模型的block层和作为教师模型的最深层常常大小相近。之后或许可以考虑把学生模型映射到新的较小的空间中，或许可以取得更好的效果。

第三，尝试更多的数据增强方式。

因为时间所限，实验只采用了图像裁剪这一种数据增强的方式。未来可能尝试采用更多的数据增强（比如几何变换、颜色转换、随机擦除、对抗训练和风格迁移等）策略，进一步提高实验结果。

第四，尝试其他深度网络架构。

因为时间所限，SKDSA的深度网络架构只尝试了残差网络和宽残差网络两种架构。未来可以尝试更多的残差网络变种，比如ResNeXt[74]、DensetNet[75]等等。

第五，在更多数据集上进行实验。

因为实验室计算资源和资金有限，SKDSA只在几个较小的数据集上进行了实验。未来可以在更大的数据集上进行实验，比如ImageNet。另外，FRSKD和SLA方法的结合在一些数据集上已经取得了更好的效果。未来可以考虑将BYOT和SLA方法结合在一起。

第六，尝试把SKDSA应用到半监督任务和弱监督任务中。

SKDSA方法可能适用于半监督学习和弱监督学习任务。未来有机会的话可以测试这一点。

第七，在语言模型上尝试SKDSA方法。

虽然SKDSA在图像分类任务上取得了成功，但是还没有在语言模型上测试它的可行性。未来可以思考将SKDSA应用到语言模型上的方法策略。

# 参考文献

[1] 卢宏涛, 张秦川. 深度卷积神经网络在计算机视觉中的应用研究综述. 数据采集与处理. 2016, 31(1): 1-7

[2] 奚雪峰, 周国栋. 面向自然语言处理的深度学习研究. 自动化学报. 2016, 42(10): 1445-65

[3] 侯一民, 周慧琼, 王政一. 深度学习在语音识别中的研究进展综述. 计算机应用研究. 2017, 34(8): 2241-6

[4] G. Hinton, O. Vinyals, J. Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015

[5] S. J. Pan, Q. Yang. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 2009, 22(10):1345–1359

[6] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, K. Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In International Conference on Computer Vision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 3713-3722

[7] S. Yun, J. Park, K. Lee, J. Shin. Regularizing class-wise predictions via self-knowledge distillation. In: Conference on Computer Vision and Pattern Recognition, 2020: 13876-13885

[8] M. Ji, S. Shin, S. Hwang, G. Park, I. Moon. Refine myself by teaching myself: Feature refinement via self-knowledge distillation. In: Conference on Computer Vision and Pattern Recognition, 2021: 10664-10673

[9] H. Lee, S. J. Hwang, J. Shin. Self-supervised label augmentation via input transformations. In: International Conference on Machine Learning, 2020: 5714-5724

[10] Ba J, Caruana R. Do deep nets really need to be deep? In: Conference on Neural Information Processing Systems, 2014: 27

[11] Kim, S. W., Kim, H. E.. Transferring knowledge to smaller network with class-distance loss. In: International Conference on Learning Representations workshop, 2017

[12] S. Hou, X. Pan, C. C. Loy, Z. Wang, D. Lin. Learning a unified classifier incrementally via rebalancing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019: 831–839

[13] S. Hegde, R. Prasad, R. Hebbalaguppe, and V. Kumar, “Variational student: Learning compact and sparser networks in knowledge distillation framework,” International Conference on Acoustics, Speech, and Signal Processing, 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 Institute of Electrical and Electronics Engineers 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 Institute of Electrical and Electronics Engineers 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,” Conference on Computer Vision and Pattern Recognition, 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 Institute of Electrical and Electronics Engineers 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,” International Conference on Machine Learning, 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,” Association for Computational Linguistics, 2019

[30] Y. Tian, D. Krishnan, and P. Isola, “Contrastive representation distillation,” in International Conference on Learning Representations, 2019

[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 Association for the Advancement of Artificial Intelligence, 2019

[33] H. Lee, S. J. Hwang, and J. Shin, “Rethinking data augmentation: Self-supervision and self-distillation,” International Conference on Machine Learning, 2020

[34] ?????

[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. Institute of Electrical and Electronics Engineers 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. Institute of Electrical and Electronics Engineers 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 Association for the Advancement of Artificial Intelligence, 2021

[49] Dasarathy B V, Sheela B V. A composite classifier system design: concepts and methodology. Proceedings of the Institute of Electrical and Electronics Engineers, 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. Proceedings of Machine Learning Research, 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 Conference on Computer Vision and Pattern Recognition, 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. Institute of Electrical and Electronics Engineers, 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, Institute of Electrical and Electronics Engineers 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 Association for the Advancement of Artificial Intelligence, 2020

[68] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In CVPR, 2020

[69] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017

[70] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems,pages1097–1105,2012 [\*1]

[71] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556,2014 [\*3]

[72] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proceedings of the Institute of Electrical and Electronics Engineers Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015 [\*4]

[73] R. Srivastava, K. Greff and J. Schmidhuber. Training Very Deep Networks. arXiv preprint arXiv:1507.06228v2,2015 [\*5]

[74] S. Xie, R. Girshick, P. Dollar, Z. Tu and K. He. Aggregated Residual Transformations for Deep Neural Networks. arXiv preprint arXiv:1611.05431v1,2016 [\*8]

[75] G. Huang, Z. Liu, K. Q. Weinberger and L. Maaten. Densely Connected Convolutional Networks. arXiv:1608.06993v3,2016 [\*9]

[76] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. arXiv preprint arXiv:1406.6247, 2014. [\*46]

[77] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4476–4484, Honolulu, HI, July 2017. IEEE. [\*142]

[78] Dongbin Zhao, Yaran Chen, and Le Lv. Deep reinforcement learning with visual attention for vehicle classification. IEEE Transactions on Cognitive and Developmental Systems, 9(4):356–367, 2016. [\*129]

[79] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. arXiv:1807.03819 [cs, stat], July 2018. arXiv: 1807.03819. [\*165]

[80] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. Disan: Directional selfattention network for rnn/cnn-free language understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, 2018. [\*193]

[81] Chen Xing, YuWu,WeiWu, Yalou Huang, and Ming Zhou. Hierarchical recurrent attention network for response generation. In Proceedings of the AAAI Conference on Artificial Intelligence, 2018. [\*198]

[82] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In International conference on machine learning, pages 7354–7363. PMLR, 2019. [\*119]

[83] Tao Kong, Fuchun Sun, Chuanqi Tan, Huaping Liu, and Wenbing Huang. Deep feature pyramid reconfiguration for object detection. In Proceedings of the European conference on computer vision (ECCV), pages 169–185, 2018. [\*375]

[84] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018. [\*380]

[85] K. Mangalam and M. Salzamann, On compressing u-net using knowledge distillation. arXiv preprint arXiv:1812.00249, 2018

[86] Q. Ding, S. Wu, H. Sun, J. Guo, and S.-T. Xia, “Adaptive regularization of labels,” arXiv preprint arXiv:1908.05474, 2019

# 附录1 攻读硕士学位期间取得的研究成果

**投稿论文**

[1] \*\*\*, **\*\*\***, and \*\*\*. In International Joint Conference on Artificial Intelligence, 2022, under review. (CCF A类会议).

# 附录2 攻读学位期间参加的科研项目

**1. 中央高校基本科研业务费学科交叉专项**

项目名称: 大数据智能与先进计算基础研究平台建设

项目编号: No. 2019kfyXKJC021

起止时间: 2019年8月至2021年8月

担任角色：研究成员

# 附录3 中英文缩写对照表

BYOT Be your own teacher

CS-KD Class-wise self-knowledge distillation

DDGSD Data-Distortion Guided Self-Distillation for Deep Neural Networks

DIANet Dense-and-Implicit Attention Network

DNN Deep neural network

FRSKD Feature Refinement via Self-Knowledge Distillation

GAN Generative adversarial networks

LSTM Long term short memory

MMD Maximum Mean Discrepancy

RNN Recurrent neural network

SAN Self-attention network

SGD Stochastic gradient descent

SKDSA Self-knowledge distillation based on self-attention

SLA-SD Self-supervised label augmentation and self-distillation

# 附录4 其它附录

可包括详细的公式推导、实验数据、计算程序、援引他人的原始资料、数据及其设备条件等。

如果有，更改附录标题为合适的题目。