# 第 3 章　多任务学习机制概述

多任务学习(multi-task learning,MTL)是机器学习的一个分支,它通过一个共享的模型同时学习多个任务。这种结构可以结合不同任务的特点,通过共享模型参数的方式提高模型的鲁棒性和学习效果[34]。MTL 具有提高数据效率、通过共享表示减少过拟合、利用辅助信息快速学习等优点。在多模态情感分析中,MTL 可以用来将 3 个模态更好地融合起来。一般情况下,多个任务之间的底层网络参数存在一定的联系,而顶层互相独立。在训练过程中,底层的全部或者部分参数受到多个任务的共同优化,以此达到多个任务联合训练的目的。根据底层参数共享方式的差异,可将其划分为硬共享(hard sharing)和软共享(soft sharing)。前者的底层参数完全共享,而后者的底层参数部分共享。

## 3.1　在计算机视觉中的多任务架构

在单任务设置中,计算机视觉架构的许多重大发展都集中在新的网络组件和连接上,以改进优化并提取更有意义的特征,如批处理归一化[35]、残差网络[36]和挤压、激励块[37]。相比之下,许多用于计算机视觉的多任务体系结构专注于将网络划分为特定于任务的共享组件,以一种允许通过共享和任务之间的信息流进行泛化的方式,同时最小化负传递。

在图 3.1 中,基本特征提取器由一系列卷积层组成,这些层在所有任务之间共享,提取的特征作为输入到特定任务的输出头。

许多工作[38-42]提出的架构是共享主干的变体。Zhang 等[39]是这些著作中最早开展相关工作的团队,其论文介绍了任务约束深度卷积网络(TCDCN),其架构如图 3.1 所示。作者提出通过共同学习头部姿态估计和面部属性推断来提高人脸地标检测任务的性能。多任务网络级联(MNC)[40]的架构与 TCDCN 类似,但有一个重要的区别:每个特定任务分支的输出被附加到下一个特定任务分支的输入,形成了层层叠叠的信息流。

在图 3.2 中,每个任务都有一个独立的网络,但十字绣单元将来自不同任务网络的平行层的信息线性组合。

在相关研究中,并非所有用于计算机视觉的 MTL 架构都包含具有特定任务输出分

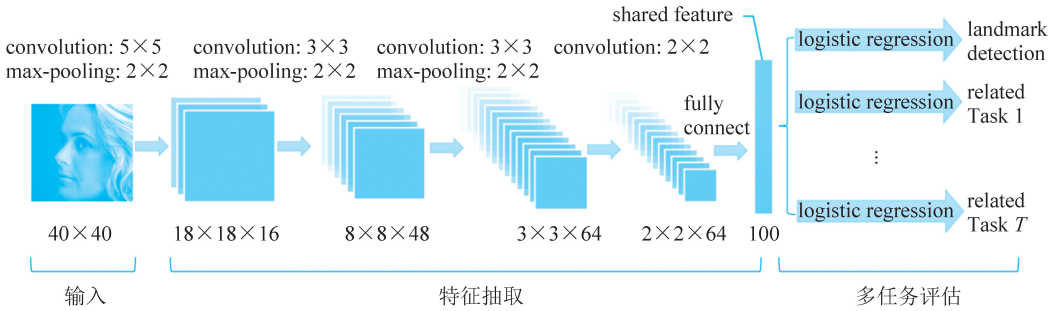


图 3.1　TCDCN 的架构[38]

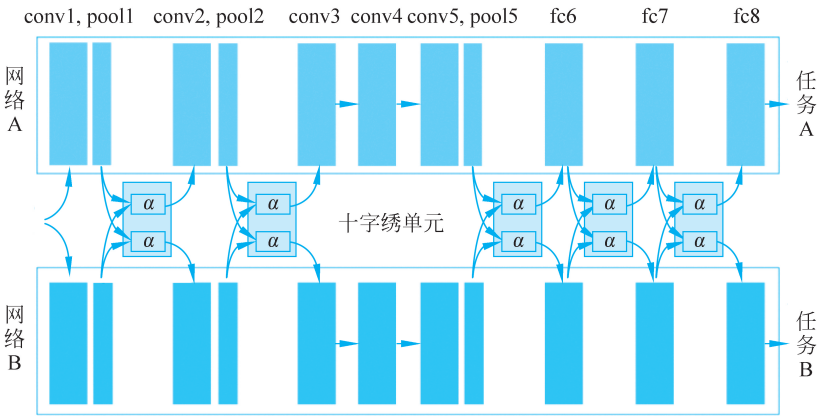


图 3.2　Cross-Stitch 网络架构[43]

支或模块的共享全局特征提取器。有些工作采取了一种单独的方法[43-45]。这些架构不是单个共享提取器，而是为每个任务提供单独的网络，在任务网络中的并行层之间具有信息流。图 3.2 的 Cross-Stitch 网络架构[43]描述了这个想法。Cross-Stitch 网络架构由每个任务的单个网络组成，但每一层的输入是每个任务网络的前一层输出的线性组合。每个线性组合的权重都是学习的，并且是针对特定任务的，这样每一层都可以选择从哪些任务中利用信息。

## 3.2　在自然语言处理中的多任务架构

自然语言处理很自然地适用于 MTL，所以人们可以就给定的一段文本提出大量相关问题，以及现代 NLP 技术中经常使用的与任务无关的表示。近年来，NLP 神经架构的发

展经历了几个阶段，传统的前馈架构演变为循环模型，循环模型被基于注意力的架构所取代。这些阶段反映在这些 NLP 架构对 MTL 的应用中。还应该指出的是，许多 NLP 技术可以被视为多任务，因为它们构建了与任务无关的一般表示（如词嵌入），并且在这种解释下，对多任务 NLP 的讨论将包括大量更广为人知的方法是通用 NLP 技术。在这里，为了实用性起见，本书将讨论限制为主要包括同时明确学习多个任务的技术，以实现同时执行这些任务的最终目标。

早期的工作都使用传统的前馈（非基于注意力的）架构来处理多任务 NLP[46-48]。许多这些架构与早期的计算机视觉共享架构具有结构相似性：一个共享的全局特征提取器，后跟特定于任务的输出分支。然而，在这种情况下，特征是单词表示。Collobert 和 Weston[48]使用共享查找表层来学习词表示，其中每个词向量的参数直接通过梯度下降学习。架构的其余部分是特定于任务的，由卷积、最大池化、连接层和 Softmax 输出组成。

现代递归神经网络的引入为 NLP 产生了一个多任务 NLP 的新模型家族，引入了新的递归架构[49-51]。序列到序列学习[20]适用于 Luong 等[50]的 MTL。在这项工作中，作者探索了用于多任务 seq2seq 模型的参数共享方案的 3 种变体，他们将其命名为一对多、多对一和多对多。在一对多模式中，编码器是所有任务共享的，解码器是特定于任务的。这对于处理需要不同格式输出的任务很有用，例如将一段文本翻译成多种目标语言。在多对一中，编码器是特定于任务的，而解码器是共享的。这与通常的参数共享方案相反，在这种方案中，较早的层是共享的，并提供给特定于任务的分支。当任务集需要以相同格式输出时，例如，图像字幕和机器翻译为相同的目标语言时，多对一变体是适用的。最后，作者探索了多对多的变体，其中有多个共享的或特定于任务的编码器和解码器。

到目前为止，上述讨论的所有 NLP 架构中，每个任务对应的子架构都是对称的。特别是，每个任务的输出分支出现在每个任务的最大网络深度，这意味着对每个任务特定特征的监督发生在相同的深度。相关研究工作[52-54]建议在早期层监督“低级”任务，以便为这些任务学习的特征可以用于更高级别的任务。通过这样做，可以形成一个明确的任务层次结构，并为来自一个任务的信息提供了一种直接的方式来帮助另一个任务的解决方案。这可以用于迭代推理和特征组合的模板称为级联信息，示例如图 3.3 所示，较低级别的任务在较早的层受到监督。

尽管 Transformers 的双向编码器表示（BERT）[31]很受欢迎，但是 MTL 文本编码方法的应用很少。Liu 等[55]通过将共享的 BERT 嵌入层添加到架构中，扩展了文献[47]的工作。整体网络架构与文献[47]非常相似，唯一的区别是在原来的架构中的输入嵌入向量之后添加了 BERT 上下文嵌入层。这种名为 MT-DNN 的新 MTL 架构实现了 SOTA 性能在 GLUE 任务[56]，相关成果发表时完成了 9 项任务中的 8 项。

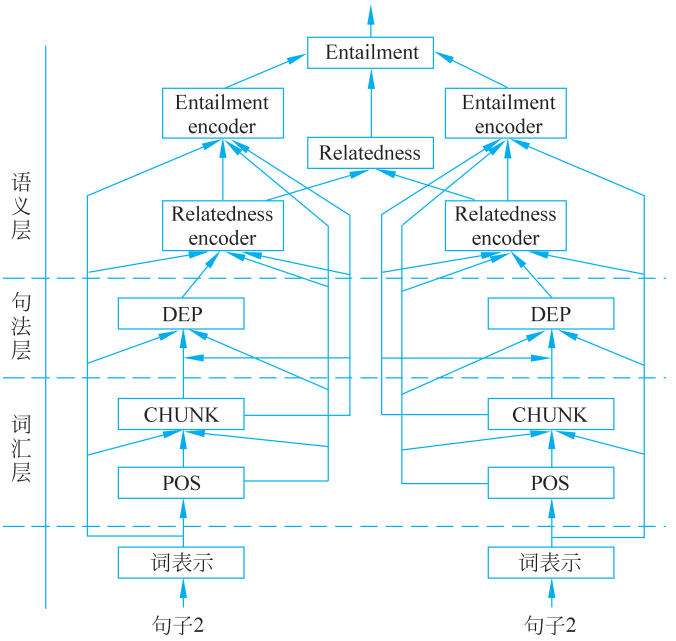


图 3.3　级联信息的各个层中的各种任务监督

## 3.3　在多模态学习中的多任务架构

多模态学习是 MTL 背后许多激励原则的有趣扩展：跨领域共享表示减少过拟合并提高数据效率。在多任务单模态情况下，表示在多个任务中共享，但在单个模态中共享。然而，在多任务多模态的情况下，表示是跨任务和跨模态共享的，这提供了另一个抽象层，通过这个抽象层，学习到的表示必须泛化。这表明，多任务多模态学习可能会增强 MTL 已经呈现出的优势。

Nguyen 和 Okatani[57] 通过使用密集的共同注意力层[58] 引入了一种共享视觉和语言任务的架构，其中任务被组织成一个层次结构，而低级任务在早期的层中受到监督。这为视觉问答开发了密集的共同注意力层，特别是用于整合视觉和语言信息。该方法对每个任务的层进行搜索，以了解任务层次结构。Akhtar 等[59] 的架构处理视觉、音频和文本输入，以对人类说话者视频中的情感进行分类，其使用双向门控循环单元（gated recurrent units，GRU）层及每对模态的成对注意力机制进行学习包含所有输入模态的共享表示。

这些工作[57,59] 专注于一组任务，所有这些任务都共享相同的固定的一组模态。相

反，另外一些工作[60-61]专注于构建一个通用的多模态多任务模型，其中一个模型可以处理不同输入域的多个任务。引入的架构[61]由一个输入编码器、一个 I/O 混频器和一个自回归解码器组成。这 3 个块中的每一个都由卷积、注意力层和稀疏的专家混合层组成。作者还证明了任务之间的大量共享可以显著提高具有有限训练数据的学习任务的性能。Pramanik 等[60]并没有使用各种深度学习模式中的聚合机制，而是引入了一种名为 OmniNet 的架构，该架构具有时空缓存机制，可以跨数据的空间维度和时间维度学习依赖关系。图 3.4 显示了一个 OmniNet 架构。每个模态都有一个单独的网络来处理输入，聚合的输出由一个称为中央神经处理器的编码器—解码器处理。然后，CNP 的输出被传递给几个特定于任务的输出头。CNP 具有空间缓存和时间缓存的编码器—解码器架构。OmniNet 在 POS 标签、图像字幕、视觉问题回答和视频活动识别方面达到了与 SOTA 方法匹敌的性能。最近，Lu 等[62]引入了一个多任务模型，可以同时处理 12 个不同的数据集，被命名为 12-in-1。他们的模型在 12 个任务中的 11 个任务上实现了优于相应的单任务模型的性能，并且使用多任务训练作为训练前步骤取得了在 7 个任务上最好的性能。该体系结构基于 ViLBERT 模型[62]，并使用动态任务调度、课程学习和超参数启发式等混合方法进行训练。

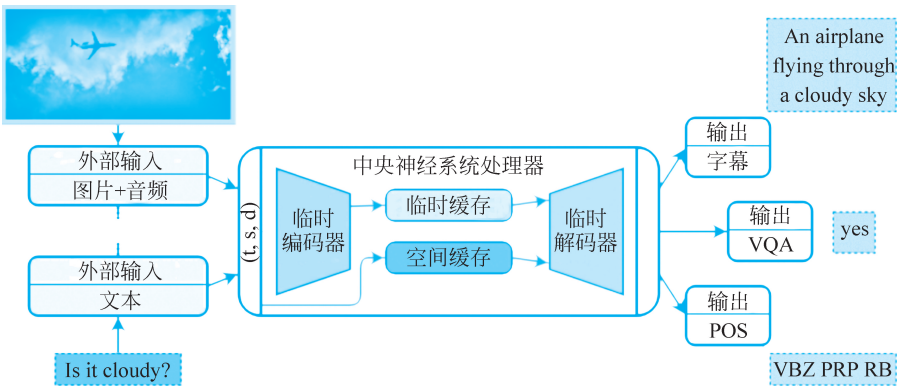


图 3.4　Pramanik 提出的 OmniNet 架构[57]

MTL 是一种近年来开始重新流行的机器学习方法，它将多个相关的单任务部分或者全部的参数共享训练，在扩大数据集规模、缓解数据稀疏问题的同时增加训练模型对于每个单任务的泛化能力。具体而言，本章中的 MTL 子任务就是针对各个单模态分支的情感分类学习。与传统的 MTL 不同，基于多任务多模态情感分类的各个子任务训练数据是完全相同的，只有单模态的标签各不相同。而正因为这些差异，使得前文提到的两个瓶颈问题得到了理想的解决：首先对于数据集模态表示不一致的问题，通过增加单独模

态的标签虽然没有解决差异性，但是从另一个角度讲，对原有单任务数据进行了二次细分，这使得细分后的同类数据之间的差异性缩小了，有利于机器进行更好的监督学习；其次，对于特征融合困难的问题，子任务分支的出现使得每一次迭代学习的过程中，单模态特征能够更好地保持自身的特点，从而扩大了模态特征之间的差异性。这样融合的特征就能够发挥其应有的作用。

## 3.4 本章小结

本章探究了机器学习中的一个重要分支——MTL。主要从计算机视觉以及自然语言处理两个领域展开综述性介绍，最后详细介绍了在多模态学习中的多任务架构，并分析了现有研究方法的优缺点。现阶段社会对于情感倾向信息的判断需求日益提升。前人提出了多种方法，但是效果都不理想，这是因为现有多模态情感分类任务存在着瓶颈问题需要解决，针对当下多模态学习的瓶颈，本书提出了将多任务机制结合到多模态情感分类的方法。

近年来，多模态情感分析任务的研究受到学术界和工业界的广泛关注。本篇主要围绕多模态表示学习和多模态表示融合两个方面对当前多模态情感分析方法进行了系统化介绍。

如何合并来自不同来源的数据、如何处理不同程度的噪声、如何处理丢失的数据及良好的表示对于机器学习模型的性能非常重要。近年来，多模态表示学习方法依据表示学习前后的对应关系可划分为两种：联合型表示和协同型表示。在多模态表示学习分类中，弱协同型表示是本书主要研究的一类表示学习方法。

在获得各单模态表示后，紧接着就是进行表示融合，即将多个单模态表示融合成一个多模态表示。研究人员就已经验证了多种模态融合后的表示可以有效地提升模型的鲁棒性。根据多模态融合在模型中所处的阶段，可以将其划分为前期融合、中期融合、后期融合和末期融合。

最后分别对多模态机器学习中的多模态表示与多模态融合的相关研究进行了综述性介绍，并详细分析了每种方法的优缺点。针对多模态表示和多模态融合方法存在的不足，本书将探究如何有效利用多任务学习机制，在多模态级别优化目标之外增加额外的单模态级别的优化目标，引导模型学到模态差异性信息。

本书内容总体结构分别从单模态信息的情感分析、跨模态信息的情感分析和多模态信息的情感分析 3 个层次上依次系统化地论述智能机器人自然交互中的情感分析问题。本书的第二篇从情感分析所使用的数据集开始讨论，分别重点介绍了常用的公开的多模态情感分析数据集，以及如何构建一个多模态多标签的中文多模态情感分析数据集，并且

深入探讨了基于主动学习的多模态情感分析数据的自动标定。在本书第三篇，分别介绍了文本、语音、图片 3 种不同模态下的情感分析方法，进一步突出每种模态所适用的方法特点。第四篇，在单模态情感分析的基础上，重点介绍了跨模态情感分析方法，并进一步优化跨模态算法的稳定性。第五篇，在跨模态情感分析的基础上，进一步探讨了多模态情感分析方法，通过先进的方法深入研究了各种多模态情感分析模型，通过多任务、自监督等方法实现多模态情感分析模型的性能鲁棒性。第六篇，在上述研究工作的基础上呈现了笔者通过开放共享的方式提供的多模态情感分析演示平台，为开展本领域研究工作的相关人员提供重要的平台支撑。本书还介绍了多模态机器学习方法在医学领域的延展性应用，为解决更多的实际问题，提供了一种思路。