班　级　1513018
学　号　15130188018

西安电子科技大学

本科毕业设计论文

题　　目　基于多任务学习的文本表示方法研究

学　　院　计算机科学与技术学院

专　　业　软件工程

学生姓名　孙天祥

导师姓名　刘西洋 教授

摘要

文本表示是自然语言处理的必要任务，表示方法的好坏对于模型性能起着至关重要的作用。为得到泛化能力强的文本表示，近年来多任务学习被广泛应用在各大自然语言处理任务中，通过联合学习多个相关任务来共享任务间的知识，从而提升模型在各个任务上的表现。

近年来，一种新型的神经网络模型 Transformer 因其在机器翻译和迁移学习等方面取得的巨大成功开始在自然语言处理领域流行。然而，之前的工作大都在卷积网络和循环网络上研究多任务学习的共享结构，目前还很少有工作探究 Transformer 上的多任务共享架构。本文在 Transformer 上探索了句子级的多任务文本表示方法：首先设计了两种传统的硬共享结构，接着提出了两种逐层共享结构，能够在每一层根据输入动态地抽取其他任务的特征来形成任务特定表示。我们在 16 个情感分析任务上进行了实验，对比单任务模型，四种多任务模型的准确率均取得了较大提升，其中，本文提出的两种逐层共享结构均超越了传统共享结构。实例可视化分析的结果解释了模型的有效性，并展示了任务之间的相关性。

最后，总结了本文工作与已有研究的异同，概括了本文的创新性与局限性，并展望了未来多任务学习在自然语言处理领域的发展前景。

关键词： **自然语言处理** **多任务学习** Transformer

ABSTRACT

Language representation is an essential task for natural language processing(NLP). Representation method plays a crucial role in the performance of the model. To obtain a general representation, multi-task learning(MTL) methods, which allow models to share knowledge between related tasks, are widely applied to many NLP tasks.

Recently, Transformer, a novel neural network based on self-attention mechanism, has become popular in the field of NLP due to its great success in machine translation and transfer learning. However, most of the previous work has focused on designing multi-task sharing scheme for convolutional networks and recurrent networks. There is little exploration on multi-task learning in Transformer. In this paper, we propose 4 sentence-level multi-task transformer architectures: two traditional hard sharing architectures and two layerwise sharing architectures, which can dynamically extract features of related tasks and form task-specific representation based on the input at each layer. We conduct experiments on 16 sentiment analysis tasks. Our 4 multi-task transformers consistently outperform transformers trained with single task. Besides, the proposed layerwise sharing architectures achieve better accuracy than traditional architectures. Case study and visualization explain the validity of the proposed models and demonstrate the relationship between tasks.

Finally, we summarize the similarities and differences between our proposed models and existing work, point out our contributions and defects.

Key words: **Natural Language Processing** **Multi-Task Learning** **Transformer**

目录

第一章 绪论

本章首先介绍基于深度学习的自然语言处理的研究背景，阐述在该背景下多任务学习的研究价值及意义。接着简要介绍自然语言处理和多任务学习的研究进展，并指出目前存在的问题。然后介绍本文的研究内容及目标，并概括了本文工作的创新之处。本章的最后给出了论文的主要内容和章节安排。

1.1 研究背景及意义

1950 年，阿兰·图灵（Alan Turing）提出了著名的图灵测试[1]，直接推动了人工智能从哲学探讨的层面上升到科学研究。随后不久，在 1956 年举办的达特茅斯会议上，人工智能的概念被正式提出，John McCarthy 将这一新兴领域的研究目标定义为：“让机器的行为看起来像人类所表现出来的智能行为一样”。自 1956 年至今的六十余年中，研究人员尝试了多种方法来实现这一愿景，人工智能领域也随着这些方法的成功与失败经历了数次热潮与低谷。近年来，随着数据量的增加和算力的增强，以神经网络为代表的深度学习异军突起，在语音识别[1][2]、计算机视觉[3][4][5] 等众多应用场景中取得了巨大突破，也为自然语言处理领域带来了深刻的变革。

自然语言，即文明发展过程中自然形成的语言，是最能体现人类智慧和文明的产物。自然语言处理（Natural Language Processing, NLP）被很多人认为是“人工智能皇冠上的明珠”，致力于使用计算机技术处理、理解和生成人类语言。随着深度学习技术的发展，越来越多的研究者开始使用深度学习的技术解决自然语言处理中的难题，在很多任务上远远超越了之前的传统方法[6][7][8][9]。

然而，对 NLP 问题的研究常常被划分为多个任务，如命名实体识别、阅读理解、机器翻译等。目前一般的做法是为当前关注的某一个任务设计特定的神经网

[1]图灵测试是指，一个人在不接触对方的情况下，通过某种方式和对方进行一系列的问答。若在相当长时间内，他无法根据问答的情况判断对方是人还是机器，那么可以认为该机器具备智能。

络模型，在此单一任务及数据集上进行训练。遗憾的是，这样设计出来的模型常常具有较大的局限性，在某个数据集上表现优秀的模型可能在另一个数据集上就会表现很差，即使这两个数据集来自同一任务。并且，由于 NLP 任务及数据集众多，一时之间各种神经网络结构层出不穷。深度学习刚刚帮助人们从“特征工程”中脱离出来，很快又陷入了所谓的网络“结构工程”。同时，这也将模型限制在了特定领域，难以发展出更为通用的智能系统。

为解决这一问题，很多研究人员转而寻求泛化能力更加强大、能够提取数据更一般性的特征表示的模型，而不是为每一个新的任务甚至数据集设计新的模型。很快，人们发现通过迁移学习和多任务学习能够得到这样的模型。事实上，迁移学习和多任务学习能够有效的原因都在于模型参数共享，或者说知识共享。其中，迁移学习的主要形式就是预训练一个较为通用的模型，再在目标任务和数据集上进行微调。在计算机视觉领域，在 ImageNet[10] 这样的大规模图像分类数据集上预训练的模型能够在很多图像分类任务上表现良好；在 NLP 领域，通过在大规模无标注文本上预训练一个语言模型也通常能够给各种下游任务带来很大收益[11][12]。这些发现表明，共享模型在其他任务上学习到的知识能够显著提升模型的性能。

在这一背景下，人们开始思考：是否可以训练单个模型来处理几乎所有的任务？要想得到这样的单一模型，几乎无可避免地需要借助多任务学习的方法。近年来，多任务学习被广泛地应用在自然语言处理领域中，在序列标注、文本分类、机器翻译等多个经典 NLP 任务上都取得了令人鼓舞的效果。随着多任务学习的引入，人们发现很多 NLP 任务可以归纳为统一的模型范式，如问答范式[13]、分类范式[12][14]。同时，越来越多的研究者开始关注模型在多任务上的表现，出现了 decaNLP[13]、GLUE[15] 等大规模多任务评测数据集。随着算法和评测基准的逐渐成熟，多任务学习模型的开发受到越来越多的研究者关注。同时也需要注意到，多任务学习的研究历史不过二三十年，对基于深层神经网络的多任务学习的研究甚至更短。因此，如何为各神经网络模型及任务设计合适的共享结构，仍然是亟待探索的问题。

1.2 相关研究进展

本节将简要介绍深度学习背景下的自然语言处理、多任务学习、深度学习以及它们相结合的研究进展及现状，并阐述了它们之间的联系以及目前存在的不足。

自然语言处理（NLP）是一门旨在使得计算机具备处理、理解和生成自然语言（人类语言）能力的学科。近年来，以神经网络为代表的深度学习在自然语言处理[6][7][8][9]中取得了广泛的成功。然而，不同于语音、图像等连续实值信号，自然语言是由离散的符号构成，这使其难以直接作为神经网络的输入。为解决这一问题，人们使用低维稠密向量来表征文本的语义信息[16][17]，由于语义信息被分布到向量的各个维度，因此这种方法被称为分布式表示。随着分布式表示的引入，深度学习在自然语言处理领域得到了广泛的应用，卷积神经网络（CNN）、循环神经网络（RNN）相继被用于处理文本数据，近年来又提出了完全基于自注意力机制的全连接网络 Transformer。这些神经网络的应用使得过去很多难以解决的 NLP 问题上取得了巨大进展。

事实上，文本的分布式表示的好坏对于模型的性能起着至关重要的作用。以分类任务为例，给定数据的一个好的表示，即使简单的线性分类器也能取得非常高的分类准确率[18][19]。进入深度学习时代以来，自然语言处理领域中取得的许多突破都来自于对文本的通用表示方法的研究，如 word2vec[16]，ELMo[11]，BERT[14]等。然而，相较于语音和图像数据，由于文本数据本身的离散性和歧义性，以及标注成本高、难度大等问题，如何得到一个好的文本表示仍然是自然语言处理领域的重大难题。

从机器学习的角度来看，一个好的表示方法除了能够在对应任务上表现良好，还应当具备良好的可迁移性与泛化能力，即能够在相似任务和新数据上获得较好的效果。在自然语言处理领域，常常使用*多任务学习*（Multi-Task Learning, MTL）和*迁移学习*（Transfer Learning）的方法来得到这样的文本表示[14][20]。

对多任务学习较系统的研究可以追溯到 1993 年[21]，它是指同时使用多个任务对模型进行训练，使其学习到数据的某种泛化表示，该表示能够同时解释这多

个任务。多任务学习作为一种模型无关的技术，在很多传统的机器学习模型以及神经网络上都可以应用。特别地，由于神经网络易于扩展的特性，多任务学习在神经网络上的应用更为方便和灵活。

在过去的几年里，很多研究人员探索了多任务学习在 CNN 和 RNN 上的应用模式，验证了基于神经网络的多任务学习在文本表示上的有效性。Collobert 等人[20] 使用一个简单的卷积网络来同时学习词性标注、语块标注、命名实体识别、语义角色标注、语义相似度、语言模型等多个任务，超越了使用单任务训练的效果。随着循环神经网络在 NLP 上的广泛应用，研究者开始基于循环网络构造多任务学习框架，在机器翻译[22]、文本分类[23][24]、序列标注[25] 等常见 NLP 任务上均取得了成功。

多任务学习的一个关键问题在于如何设计一个高效的共享模式来允许模型共享多个任务的知识。上述提到的工作也大都致力于为所要解决的问题以及采用的网络结构来设计合适的共享模式，如硬共享模式、软共享模式、分层共享模式、共享-私有模式等。

同时，也有大量工作致力于使用迁移学习的范式来获取文本的通用表示，一般做法是利用语言模型[11][12]、机器翻译[26] 或其他无监督任务[14] 来预训练一个可迁移的模型。并且，迁移学习与多任务学习本身并不互斥，因此可以同时利用二者的方法，使用多任务预训练迁移模型[27]，也可以在预训练得到的模型的基础上再使用多任务来微调[28][29]。

事实上，多任务学习和迁移学习本质上都是通过共享参数来迁移模型在不同任务中学习到的知识，并以此来提升泛化能力。因此，通常在迁移学习中效果很好的模型也可以应用在多任务学习中。近期，以 Transformer 为预训练网络结构得到的迁移模型 GPT[12] 和 BERT[14] 在诸多自然语言处理任务上取得了极大的提升，这证明了 Transformer 强大的文本表示能力。然而，不同于 CNN 和 RNN，目前还很少有工作研究多任务学习在 Transformer 上的应用模式，已有的少量工作也只是将最传统的多任务共享模式简单地应用在 Transformer 中[28]。

1.3 本文研究内容

本文试图在一定程度上填补目前多任务学习在 Transformer 结构下的研究空缺，探索基于 Transformer 的多任务共享模式，并通过实验比较几种多任务 Transformer 结构的效果。首先，本文考察了传统的硬共享模式在 Transformer 上的应用效果，然后，根据 Transformer 自身的结构特点设计了新的多任务架构。

在 CNN 及 RNN 结构中应用多任务学习通常是“纵向”的，如硬共享模式和分层共享模式，即在网络结构的某一层上堆叠任务特定层[25][30]。这种架构蕴含着一个假设：存在某种通用的文本表示，特定的任务表示可以由该通用表示通过简单的变换得到。然而，这样的假设限制了能够同时学习的任务的多样性，难以处理弱相关任务及不同难度的任务。此外，任务特定层的位置通常需要根据任务特点来人为设定，这增大了多任务学习的应用难度。也有一些工作采用了“横向”共享的架构，如软共享模式和共享-私有模式，即允许模型使用其他任务的模型在同一层的隐状态[23][31][32]，然而这些架构也存在两个问题：（1）通常需要为每个任务训练一个模型，参数量大且难以扩展；（2）各任务模型之间的信息交互难以控制。

而在 Transformer 中，可以很容易地在“横向”以记号的形式进行扩展，并且扩展的记号可以像普通单词一样与句子中的每个单词进行交互。同时，相比软共享、共享-私有模式等“横向”共享方法，这种共享结构只需增加少量参数。基于这一特性，本文给出了两种新型多任务共享模式：层级-隐式共享模式（L-I 结构）和层级-显式共享模式（L-E 结构）。在十六个文本分类数据集上进行的实验表明，本文提出的层级-隐式共享结构只需要增加 0.5% 的参数量就可以提升约 5% 的平均分类准确率。

此外，建模多任务之间的联系一直是多任务学习领域的研究重点。层级-显式共享结构可以直接利用注意力机制对任意任务之间的关系进行建模。并且，由于注意力机制的引入，本文的多任务 Transformer 结构具备良好的可解释性。对注意力得分矩阵的可视化为模型的预测结果提供了证据，展示了任务之间的交互关系。

1.4 论文结构

本文主要内容包括介绍已有的基于多任务学习的文本表示方法，几种新型的多任务文本表示模型及其实验结果，工作总结及对未来的展望。全文分为五个章节进行介绍，具体结构安排如下：

第 1 章，介绍研究背景及意义，概括本文的研究内容。

第 2 章，介绍相关的理论基础及前沿进展。在 2.1 节介绍深度学习的相关概念；在 2.2 节介绍基于神经网络的多任务学习；在 2.3 节介绍深度学习背景下自然语言处理的发展现状；在 2.4 节介绍多任务学习在自然语言处理中的应用。

第 3 章，详细介绍本文研究的模型结构及实现细节。在 3.1 节介绍 Transformer 的模型结构；在 3.2 节介绍几种多任务 Transformer 架构；在 3.3 节给出本文所使用的超参数设置以及实现细节。

第 4 章，介绍实验相关的信息。在 4.1 节描述模型应用的具体任务；在 4.2 节给出本文使用的各个数据集的有关信息；在 4.3 节介绍模型在各数据集上的实验结果；最后在 4.4 节对模型进行了实例可视化分析。

第 5 章，对本文的贡献和不足进行总结，回顾相关领域面临的机遇与挑战，并给出了未来的研究方向。

第二章 相关工作

本文研究的内容包含了深度学习、多任务学习与自然语言处理三个主题，这里首先阐述这些主题之间的联系，接着分别介绍它们各自的相关概念与研究进展，最后介绍了三者交叉的一些重要研究工作。

我们利用深度学习与多任务学习来解决自然语言处理中的问题，具体的，即在深层神经网络中使用多任务学习来获得文本的泛化表示。从这一角度来看，深度学习与多任务学习是我们的工具，而自然语言处理为我们的应用场景。同时，深度学习与多任务学习都可以归为机器学习问题，而他们二者又有交叉，如图 2.1 所示。

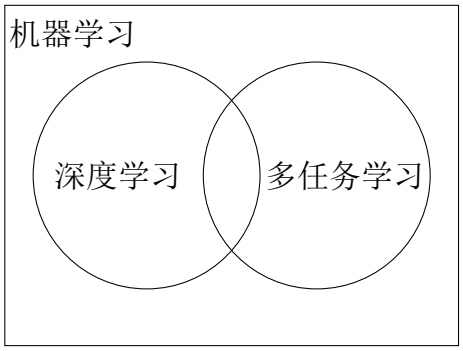


图 2.1　机器学习、深度学习、多任务学习的关系

2.1　深度学习与神经网络

近年来，深度学习（Deep Learning）发展十分迅速，在人工智能的很多子领域上取得了巨大成功，如语音识别[1][2]、计算机视觉[3][4][5]、自然语言处理[6][7][8][9]等。

从要研究的问题来看，深度学习属于机器学习的一个分支，都是研究如何从有限个样例中通过某种算法总结出一般性规律或模式，并将其应用到新的未知数据上去。例如，通过机器学习或深度学习算法，可以根据历史病例总结出症状与疾病之间的规律，当遇到新的病人时可以利用算法总结出来的规律来帮助判断这

个病人得了什么疾病[1]。

同时，深度学习又与机器学习有很大的不同。从模型的构成来说，深度学习模型一般更加复杂，参数量更多。由于数据从输入到输出需要经过多个线性或非线性组件，信息传递路径较长，因此被称作深度学习。深度学习模型的一个最典型代表就是人工神经网络（Artificial Neural Network，ANN），下面简称神经网络。神经网络是一种受人脑神经系统启发的复杂数学模型，由神经元以及它们之间的连接组成。通过这些神经元的加工，原始数据从底层特征开始经过一道道加工逐渐转换为抽象的高层语义特征。

总的来说，神经网络可以被看做一个包含大量参数的复合函数，该函数描述了输入和输出之间的复杂关系，因此可以被用于处理很多模式识别任务，如语音识别、图像识别等。形式化地，神经网络可以这样描述：

$$y = \mathcal{F}(\mathbf{x} \mid \theta). \qquad \text{式 (2-1)}$$

其中，函数 $\mathcal{F}$ 受参数 θ 控制。

给定包含大量样例的集合，参数 θ 可以通过随机梯度下降等优化算法得到，求解参数的优化过程就被称为学习过程，通过学习得到的参数被称为可学习参数，常简称为参数。还有一部分不可以被学习的参数，例如神经网络的层数，被称为超参数，意为控制参数的参数，超参数通常根据经验指定或根据实验效果来确定。

前面提到，参数学习需要一个包含大量样例的集合，该集合就是数据集。在有监督学习中，每个样例一般包含输入 $\mathbf{x}$ 和输出 y，输出也常被称为标签。其中，输入一般为一个向量，向量的每一个维度表示了样本的一个属性；输出一般为一个标量。包含 n 个样例的集合 $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ 就是数据集。在实际使用时，数据集通常被划分为训练集、验证集（也叫开发集）和测试集。其中，用训练集来让算法学习参数，用验证集来调整算法的超参数，用测试集来衡量模型的优劣。

给定数据集 $\mathcal{D}$ 和模型结构，即 $\mathcal{F}$ 的形式，优化算法可以得到参数 θ 进而确

[1]例子来源：《神经网络与深度学习》，邱锡鹏，https://nndl.github.io

定模型 $\mathcal{F}(\theta)$. 下面介绍一种简单的前馈神经网络结构：*多层感知机*（Multi-Layer Perceptron, MLP）。一个单隐层的 MLP 可以记作 $\mathcal{F}(\mathbf{x};\theta) = \mathbf{w}_2(f(\mathbf{w}_1\mathbf{x} + b_1)) + b_2$，其中参数 $\theta = \{\mathbf{w}_1, b_1, \mathbf{w}_2, b_2\}$，函数 $f(\cdot)$ 为非线性激活函数，如 ReLU. 多层感知机的结构如图 2.2 所示，为简单起见，图中省略了偏置项 b_1, b_2.

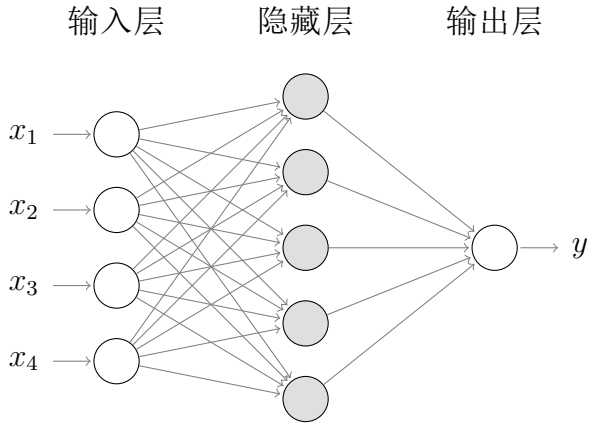


图 2.2　多层感知机

随着深度学习的发展，神经网络的结构也变得日益多样和复杂。上面的多层感知机属于全连接前馈网络，而前馈网络的另一典型代表就是在计算机视觉中被广泛使用的*卷积神经网络*（Convolutional Neural Network, CNN），它与全连接网络的区别在于权重共享和局部连接。除前馈网络外，还有一类网络称为反馈网络，因为反馈网络的神经元可以接收来自本身的反馈信号，从而具备一定的记忆能力，因此也被称为记忆网络。反馈网络的一种典型代表就是*循环神经网络*（Recurrent Neural Network, RNN），它在自然语言处理中得到了广泛应用。另外，近年来*图神经网络*（Graph Neural Network, GNN）因其在处理图结构数据上的优势也得到了迅速发展，近期也被用于处理文本和图像数据。

深度学习能够成功的重要原因在于其强大的特征表示能力。在传统的机器学习中，通常需要人为地构造数据特征，再训练一个学习算法来总结这些构造出来的特征输入与输出之间的关系。例如，在文本分类任务中，传统机器学习的做法一般是利用词袋模型或 TF-IDF 来构造文本特征，再将其作为支持向量机等分类器的输入来进行分类。而在深层神经网络中，算法自动学习原始数据的特征表示，

数据从输入到输出的过程 $\mathbf{x} \rightarrow \mathcal{F} \rightarrow y$ 可以人为地划分为两个阶段：*表示*和*预测*[2]。所谓表示，就是神经网络中间隐层的状态，上文中 MLP 的中间隐层就可以当作某种浅层表示；而预测，比如分类或回归，一般是在上一层得到的特征表示的基础上进行简单变换得到易于预测的任务特定表示。因此，深度学习又可以看作是一种表示学习，而深度学习的成功也证明了学到好的特征表示的重要意义。

2.2 多任务学习

在机器学习中，我们通常关心模型在某个任务上的某个指标，并通过在某个数据集上训练单个模型或一组集成模型来达成此目标。事实上，有时我们可以同时利用其他数据集甚至其他任务来联合训练模型，从而进一步提升性能，这种方法就是*多任务学习*（Multi-Task Learning, MTL）。

从定义上来说，多任务学习就是一种归纳转移（Inductive Transfer）方法，通过利用包含在相关任务训练信号中的领域特定信息来提升模型的泛化能力[33]。在机器学习中，给定训练集，存在多个假设能够解释训练数据，这些假设构成的空间被称为假设空间，假设空间中的一个假设就对应了一个模型。不同的机器学习算法倾向于选择不同的假设，这种选择偏好被称为*归纳偏置*（Inductive Bias）。直观上，多任务学习使得算法倾向于选择一种能够同时解释多个任务的假设，也即引入了多个任务的归纳偏置。从表示学习的角度看，这样能够同时适用于多个任务的表示就是泛化的表示。图 2.3 较为直观地给出了多任务学习的一种解释。

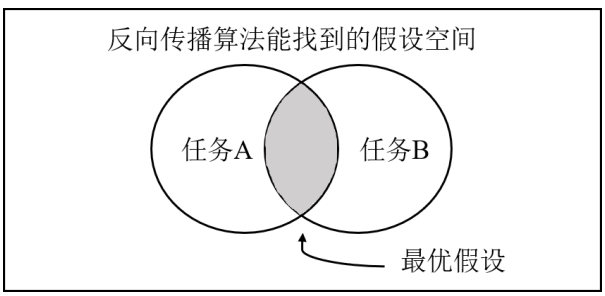


图 2.3 多任务学习的一种直观解释

关于多任务学习为何有效，除了上面的解释（归纳偏置）之外，Caruana 还给

[2]有时也被称为编码和解码

出了几种合理的解释[33]：

- **数据增强（Statistical data amplification）** 由于多个任务的数据集通常是不同的，使用多个相关任务对同一个模型进行训练相当于增大了训练数据量。
- **特征选择（Attribute selection）** 如果任务噪声较大，或者数据高维而数据量有限，模型很难分辨相关特征和无关特征，而多任务学习可以帮助模型选出相关特征，因为这些特征通常在多个任务中是共用的。也就是说，其他任务为模型选择特征提供了额外的证据。
- **窃听（Eavesdropping）** 存在某些特征对于任务 A 易于学习，而对于任务 B 则难以学习。通过多任务学习，模型可以在执行任务 B 时使用任务 A 学到的特征。

在深度学习之前，多任务学习已经被应用在线性模型、核方法、决策树、多层感知机等传统机器学习算法上，有大量的研究集中在稀疏正则化[34][35] 以及学习任务之间的关系[36][37] 上。

随着深度学习的发展，多任务学习开始应用在深层神经网络中，并在自然语言处理[20]、计算机视觉[31]、语音识别[38]、药物设计[39] 等众多应用场景中取得了成功。同时，多任务学习作为一种模型无关的方法，在卷积神经网络[20][31]、循环神经网络[23]、图网络[40] 上都可以应用。

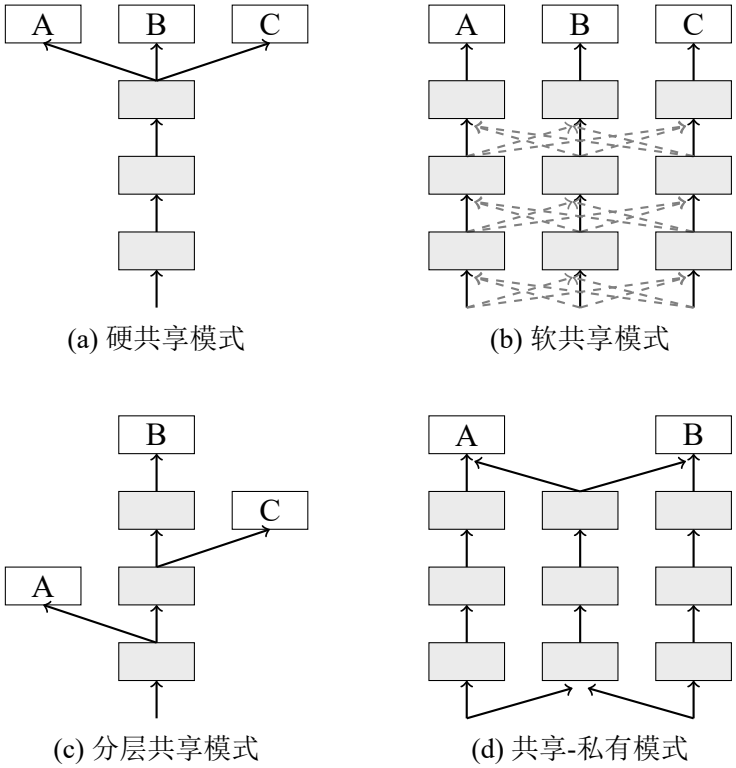
然而，无论是在传统机器学习算法上还是在深层神经网络上，多任务学习的核心观点都在于知识共享。如何为特定任务和学习算法设计合适的共享模式一直是多任务学习的重要问题。从多任务学习的应用方式来看，可以大概分为下面四种共享模式：

- **硬共享模式**：为多个任务联合训练单个模型，让不同任务共享相同的神经网络层来提取任务无关的通用特征表示，每个任务通过自己的任务特定层在通用表示基础上进行加工得到任务特定表示。
- **软共享模式**：为每个任务训练自己的模型，但每个任务都可以窃听其他任务的模型学习到的特征表示。窃听的方式可以是注意力机制、门控机制，也可

以是简单的线性加权。

- **分层共享模式**：为多个任务联合训练单个模型，但简单任务在神经网络低层施加监督信号，困难任务在神经网络的高层施加监督信号。
- **共享-私有模式**：为每个任务训练单个模型，同时使用多个任务训练一个共享模型。每个任务的模型在提取自己任务特定表示的时候也可以从共享模型中提取任务无关的通用特征。

四种共享模式如图 2.4 所示，图中灰色模块为共享层。关于基于神经网络的多任务学习共享模式的类型，有的文献[41][42] 有不同的分类方式，但硬共享和软共享是两种公认的模式。



(a) 硬共享模式　(b) 软共享模式

(c) 分层共享模式　(d) 共享-私有模式

图 2.4　多任务学习的几种常见共享模式

具体的，我们以最常见的硬共享模式为例，给出多任务学习的形式化描述。假设有 T 个相关任务，任务 t 的数据集为 $\mathcal{D}_t = \{\mathbf{x}_n^{(t)}, y_n^{(t)}\}_{n=1}^{N_t}$，包含 N_t 个样本。假设模型在任务 t 的第 n 个样本上的输出为

$$\hat{y}_n^{(t)} = \mathcal{G}^{(t)}(\mathcal{F}(\mathbf{x}_n^{(t)}; \theta_{\mathcal{F}}); \theta_{\mathcal{G}^{(t)}}), \qquad \text{式 (2-2)}$$

其中 $\mathcal{F}$ 为神经网络共享层，$\mathcal{G}^{(t)}$ 为任务 t 的特定层。模型总参数包含共享层的参数以及任务特定层的参数，即 $\theta = [\theta_{\mathcal{F}}, \{\theta_{\mathcal{G}^{(t)}}\}_{t=1}^{T}]$. 多任务联合学习的损失函数为

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \lambda_t \sum_{n=1}^{N_t} \mathcal{L}_t(\hat{y}_n^{(t)}, y_n^{(t)}), \qquad \text{式 (2-3)}$$

其中，$\mathcal{L}_t(\cdot)$ 为第 t 个任务的损失函数，λ_t 为第 t 个任务损失函数的权重。λ_t 通常被看作是超参数，根据任务 t 的重要程度或困难程度来确定，也可以作为可学习参数根据任务的不确定性来自主学习[43]。

最后，通过优化如下目标来得到模型参数

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\theta). \qquad \text{式 (2-4)}$$

多任务学习使用的优化算法与常见单任务学习的优化算法没有什么不同，可以采用随机梯度下降等方法。

2.3　自然语言处理

自然语言处理（Natural Language Processing, NLP）是一门涵盖计算机科学、语言学、数学等多个领域的交叉学科，也是人工智能的核心领域之一，旨在使用计算机技术来处理、理解和生成自然语言文本。同时，自然语言处理也是一个包括词性标注、句法分析、语义角色分析、机器翻译、阅读理解、问答系统、对话系统等在内的庞大的研究领域。从定义上，任何与处理文本数据相关的问题都可以归为自然语言处理的问题。

事实上，人们对 NLP 的研究早在人工智能被提出之前就开始了。上世纪上半叶，Claude Shannon 使用概率模型来描述人类语言，提出用熵来表示语言的不确定性。1956 年，Noam Chomsky 提出生成语法理论，给出了语言句法结构的符号规则，此后，基于规则的符号系统开始在 NLP 领域流行。研究人员根据研究工具的不同分为统计和规则两个流派。到上世纪末，随着隐马尔科夫模型、核方法等

方法的出现，统计自然语言处理开始因其灵活性和通用性渐渐成为主流。随着深度学习的兴起，这一趋势被再次加强，基于神经网络的方法开始在各大 NLP 任务上取得 state-of-the-art 结果。

在深度学习背景下，一个使得 NLP 取得重大进展的关键概念是*分布式表示*（Distributed Representation）[44]。在传统机器学习方法中，普遍采用 one-hot 方法表示文本特征，即在一向量中用 1 来表示出现的单词，用 0 来表示未出现的单词。然而这种表示方法下向量维度与词表大小一致，不便于扩展，且任意两单词之间正交，无法计算语义相似度。而分布式表示则将文本语义分布到向量的各个维度，或者说分布到多个神经元上进行处理。在分布式表示下，文本用低维稠密向量表示，使得语义组合和语义相似度的计算都非常灵活。分布式表示的概念是整个深度学习技术的核心，如何学到一个好的分布式表示是决定各个深度学习算法性能的关键。

在 NLP 中，文本的分布式表示方法一直是研究重点。分布式表示的应用引出了*词向量*（也被称作*词嵌入*）的概念，在词嵌入矩阵中，每个单词用一个低维稠密词向量来表示。最初，Bengio 等人[45] 将分布式表示引入语言建模，让神经网络在反向传播时同时更新网络参数和词向量。这里的词向量作为模型的参数，训练前采用随机初始化。2013 年，Mikolov 等人[16] 提出了 word2vec 方法，可以在大规模无标注文本上预训练一组词向量，使用预训练的词向量初始化神经网络的词嵌入层可以显著提升模型效果。2014 年，Pennington 等人[17] 使用类似的思路发明了 GloVe，这种词向量具备更好的线性性质。从此，在绝大多数 NLP 任务中，人们通常使用预训练好的词向量来作为模型输入，而不再是随机初始化。然而，这些文本表示方法将单词映射为固定的词向量，难以处理一词多义的问题，因此是*未语境化*（non-contextualized）的。在现实场景中，一个词的意思常常要通过其所在的语境来确定，例如“苹果”在某些语境中表示一种水果，而在某些语境中可能表示一家公司。为处理这一问题，人们提出了*语境化*（contextual）的文本表示方法，如 ELMo[11]，GPT[12]，BERT[14] 等，这些上下文敏感的文本表示方法使得

神经网络模型在各大 NLP 任务上取得了巨大的提升。

文本的分布式表示一般被认为是神经网络的输入，即第一层（词嵌入层）。为执行特定的任务，通常需要在文本表示的基础上再进行加工得到易于预测的任务特定表示。这种加工可以采用不同的神经网络来实现。在 NLP 中，目前普遍使用的神经网络为卷积神经网络（CNN）[20][46][47] 和循环神经网络（RNN）[48][49][50]。然而，这二者还都存在难以解决的缺陷：CNN 一般只能建模局部位置信息，难以捕捉长距离句子依赖；而 RNN 每一时间步的计算都依赖上一时间步的状态，导致其运算速度缓慢，难以并行，无法在大规模工业场景中使用。需要注意的是，本文提到的 RNN 包含原始版本及其变种，如 LSTM[51] 和 GRU[52].

另外，人们在对自然语言处理、计算机视觉等问题的研究过程中也提出了一些模型无关的机制，其中的一个典型代表就是注意力机制。在 NLP 中，注意力机制最初出现在机器翻译中，显著提升了翻译质量[53]。随后，注意力机制开始在 NLP 的各个子问题中被广泛使用，基于双向 LSTM 和注意力机制的神经网络模型一度成为处理包括阅读理解、自然语言推理等在内的各个 NLP 任务的标配。甚至在 2017 年，Google 提出了完全基于自注意力的神经网络：Transformer[54]，在机器翻译任务上取得了新的 state-of-the-art 结果。

Transformer 在语义信息提取和长距离依赖任务上都取得了富有竞争力的结果[55]，并且因其可以并行计算的优点在 NLP 领域迅速流行。近期，随着基于 Transformer 的预训练模型 BERT[14] 取得的巨大成功，Transformer 开始受到越来越多的研究者关注，出现了 Universal Transformer[56]、Gaussian Transformer[57]、Star-Transformer[58] 等变种。

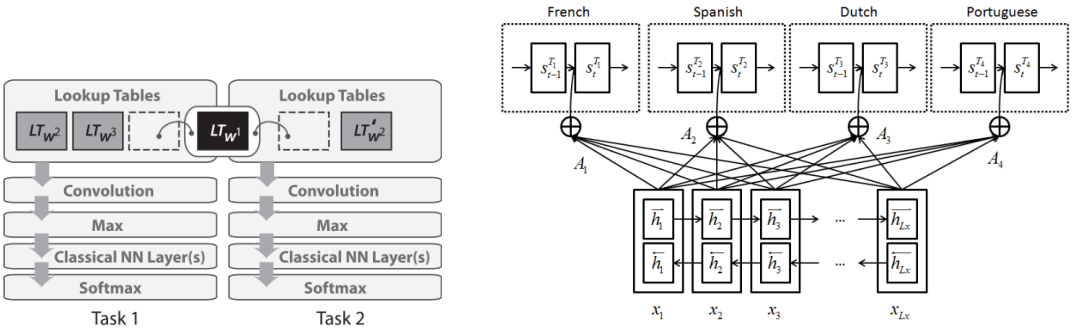
2.4　神经多任务自然语言处理

在前文中，介绍了深度学习的基本概念，以及多任务学习和自然语言处理在深度学习背景下的研究现状。在这一基础上，本节将介绍使用基于神经网络的多任务学习在自然语言处理中的主要进展。

事实上，早在十年前就已经有人在神经网络模型中应用多任务学习来解决 NLP 的问题：Collobert 等人[20] 使用一个简单的卷积神经网络（Convolutional Neural Network, CNN）来同时学习词性标注、语块标注、命名实体识别、语义角色标注、语义相似度和语言模型，超越了 CNN 使用单任务训练时的表现。

随着循环神经网络（Recurrent Neural Network, RNN）在 NLP 上的广泛应用，研究者开始基于 RNN 构造多任务学习框架，在机器翻译[22]、文本分类[23][24]、序列标注[25] 等常见 NLP 任务上均取得了成功。

图 2.5 分别给出了多任务学习在 CNN 和 RNN 上的两种模型结构：(a) 使用多任务 CNN 处理词性标注、命名实体识别等多个序列标注任务和语义相似度任务；(b) 使用多任务 RNN 处理多语言机器翻译问题。两模型均属于多任务学习中的硬共享结构，且 (a) 模型仅共享词表。



(a) 一种多任务 CNN 模型[20]　　(b) 一种多任务 RNN 模型[22]

图 2.5　多任务学习在 CNN 和 RNN 上的应用示例

从共享架构上看，基于神经网络的多任务学习架构涵盖了前文提到的各种共享模式。首先，最简单的硬共享模式率先得到了应用[20]，并且直到今天还在被广泛的使用[28]；随后，软共享模式因其灵活性也被广泛地应用在 NLP 任务中，并取得了出色的表现[32]；由于 NLP 任务对文本表示的要求不同（例如比较简单的 NLP 任务仅需要简单的语法知识而难度较大的任务需要复杂的语义信息），分层共享在 RNN 中得到了应用[25]，取得了优于硬共享架构的效果；最后，Liu 等人[23] 在 RNN 中使用共享-私有模式，在文本分类任务上取得了较大提升。

近期，随着迁移学习在 NLP 领域取得了巨大成功[11][12][14]，人们发现学习一

个通用的任务无关的表示能够给特定任务带来的收益远远大于根据任务特点对模型结构的改进。为了评测模型的通用表示能力，研究者们开始使用多个任务的性能来测试单一模型，开发了通用表示能力评测工具（SentEval[59]）以及一些多任务基准平台（DecaNLP[13]、GLUE[15]）。

2018 年 10 月，BERT[14]，一个预训练的 Transformer 模型，在 GLUE 的多个任务上的表现都大幅度超越了之前的模型。最近，人们发现在 BERT 的基础上使用多任务学习对模型进行微调能够取得进一步提升[28][29]。

然而，目前还很少有工作在 Transformer 上探索多任务学习的使用。随着 Transformer 在机器翻译[54]、迁移学习[12][14] 中取得了巨大成功，如何使其通过多任务学习取得额外的收益，以及如何对其设计新的共享模式都是值得探索的问题。

第三章 模型

本章将详细介绍本文所使用的模型，首先在 3.1 节介绍 Transformer 模型，它将作为本文实验中的单任务学习基线模型。接着在 3.2 节介绍多任务 Transformer 的几种架构，最后在 3.3 节描述了一些具体的实现细节。

在描述模型结构之前，首先阐述即将用到的几个概念：

记号 模型在处理输入的句子时，能够区分的最小单位即是记号。在自然语言处理中，一个记号或位置常常是一个单词，但也可以是字符，还可以是词片。将一个输入句子切分为多个记号的过程称为*记号化*，不同粒度模型的记号化过程对比可见表 3.1，其中中文单词级的记号化需要借助分词技术。本文中使用的是单词级模型，因此下文中提到的记号和单词所指的概念是一致的。

词表 用于训练模型的数据集中出现的记号集合被称为词表，该集合中元素的个数就是词表大小。词表大小是衡量数据集规模和模型参数量的一个重要指标。词表大小与模型有关，通常来说，字符级的模型词表较小，单词级的模型词表较大。

任务 本文中的任务是对于模型而言的，并不一定是通常意义上的任务。例如，模型需要优化多个目标函数，每个目标函数就可以看作一个任务。因此，联合训练词性标注、命名实体识别、语言模型等多个 NLP 任务可以被视作多任务学习；联合训练多个数据集，即使这些数据集被用于处理同一个 NLP 任务（如中文分词），也可以被视作多任务学习。

隐编码 神经网络的中间层被称为*隐藏层*，其前为输入层，其后为输出层。在很多用于处理 NLP 的神经网络（如循环网络和 Transformer）中，输入句子的每一个记号在隐藏层都有对应的向量表示，我们将记号在隐藏层的向量表示称为隐编码。

接着，我们从数据的角度介绍神经网络模型处理 NLP 问题时的一般过程。给定一输入句子，首先按照预先定义的规则对其进行记号化，得到记号序列后通

表 3.1 不同粒度的记号化方式

语言	粒度	输入	输出
英文	字符级	she looks lovely.	s h e l o o k s l o v e l y .
	单词级	she looks lovely.	she looks lovely .
	词片级	she looks lovely.	she looks love ly .
中文	字符级	她看起来很可爱。	她 看 起 来 很 可 爱 。
	单词级	她看起来很可爱。	她 看起来 很 可爱 。

过查词表得到各记号对应的索引，该索引序列 $x = (x_1, x_2, ..., x_n)$ 即模型的输入。模型的输入层一般为词嵌入（也称词向量）矩阵，假设词表大小为 $|V|$，词嵌入维度为 d_w，则词嵌入矩阵 $\mathbf{W}^e \in \mathbb{R}^{|V| \times d_w}$，$x_i$ 的取值范围为 $0 \leq x_i < |V|$. 模型根据 x 查询词嵌入矩阵的对应行，再经过线性投影矩阵 $\mathbf{W}^p \in \mathbb{R}^{d_w \times d}$ 得到隐藏层的输入 $\mathbf{z}^{(0)} \in \mathbb{R}^{n \times d}$，其中 d 为隐层维度。在很多情形下，若 $d_w = d$，也可不经过线性投影。神经网络的隐藏层将 $\mathbf{z}^{(0)}$ 逐层加工，形成隐编码 $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, ..., \mathbf{z}^{(N)}$，其中 N 为隐层个数。最后，模型的输出层根据 $\mathbf{z}^{(N)}$ 得到预测结果。以文本分类任务为例，模型在输出时首先由 $\mathbf{z}^{(N)} \in \mathbb{R}^{n \times d}$ 通过某种信息汇聚手段（如最大池化、平均池化等）得到文本编码 $\mathbf{z}_s \in \mathbb{R}^{d_o}$，再将其通过分类矩阵映射为各类别对应的分数，假设类别数为 c，则分类矩阵为 $\mathbf{W}^o \in \mathbb{R}^{d_o \times c}$. 最后，使用 Softmax 函数对各类别分数进行规范化，从而得到样本被模型判定为各个类别的概率。图 3.1 以文本二分类任务为例描述了这一过程。

形式化地，给定文本编码 $\mathbf{z}_s$，模型将文本判断为类别 j $(0 \leq j < c)$ 的概率为

$$\begin{aligned} P(y = j|\mathbf{z}_s) &= \text{Softmax}_j(\mathbf{z}_s \mathbf{W}^o) \\ &= \frac{e^{\mathbf{z}_s \mathbf{W}^o_j}}{\sum_{k=1}^{c} e^{\mathbf{z}_s \mathbf{W}^o_k}}. \end{aligned} \qquad \text{式 (3-1)}$$

其中，$\mathbf{W}^o_k$ 表示矩阵 $\mathbf{W}^o$ 的第 k 列。

而对于序列标注任务，即为句子中的每个词进行分类标注，通常不需要得到句子级的文本编码 $\mathbf{z}_s$，而只需要对 $\mathbf{z}^{(N)}$ 执行式 (3-1) 进行分类即可。其中 $\mathbf{z}^{(N)}$ 可

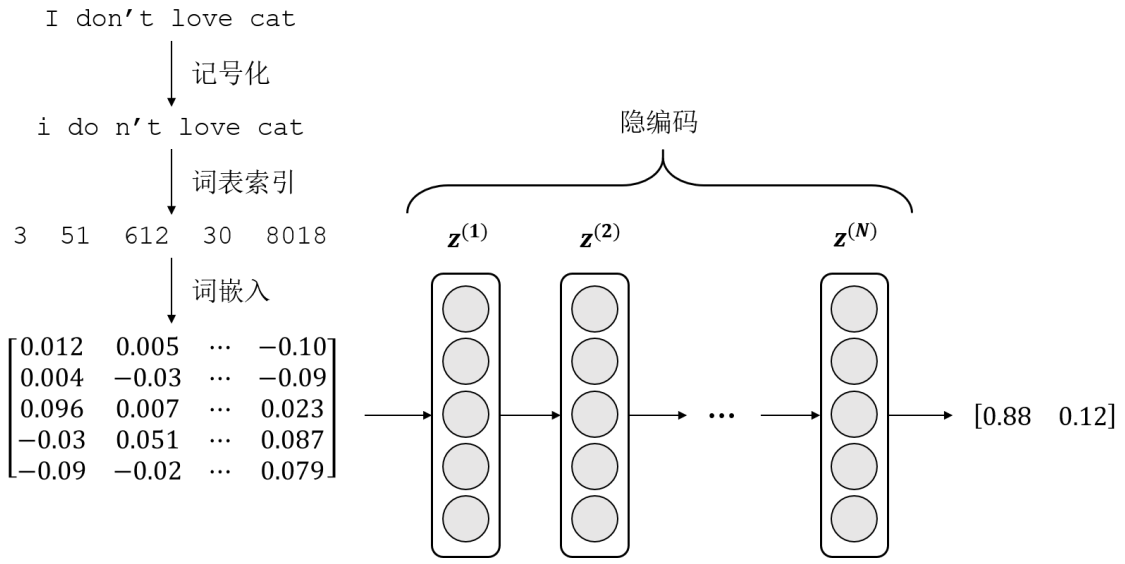


图 3.1　神经网络处理 NLP 问题的一般过程

以展开为 $\mathbf{z}_1^{(N)}, \mathbf{z}_2^{(N)}, ..., \mathbf{z}_n^{(N)}$，分别表示每个记号的第 N 层隐编码。

以上即是使用神经网络处理 NLP 问题的一般输入输出过程，该过程通常在各神经网络模型中是通用的，不同的神经网络模型的区别在于如何由 $\mathbf{z}^{(0)}$ 进行逐步抽象加工得到 $\mathbf{z}^{(N)}$ 或 $\mathbf{z}_s$.

3.1　Transformer

Transformer 是 Google 于 2017 年提出的一种完全基于*自注意力*（self-attention）的全连接神经网络[54]，摒弃了之前常用的循环计算和卷积结构，又同时具备了卷积网络的并行计算特性以及循环网络的处理长距离依赖的能力。Transformer 最初被用在机器翻译中，因此包含一个*编码器*和一个*解码器*，这里仅介绍本文用到的编码器部分[1]。

作为编码器，一个 N 层的 Transformer 可以将输入的句子序列 $(x_1, x_2, ..., x_n)$ 编码得到 N 组低维稠密的向量表示 $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, ..., \mathbf{z}^{(N)}$，其中每一层 $\mathbf{z}^{(i)}$ 都包含句子中对应记号的隐编码 $(\mathbf{z}_1^{(i)}, \mathbf{z}_2^{(i)}, ..., \mathbf{z}_n^{(i)})$. Transformer 模型的输入输出形式与前文描述的一般过程完全一致，本节主要描述 Transformer 处理隐编码的过程。

Transformer 编码器的结构如图 3.2 所示，图中阴影部分表示 Transformer 的

[1]解码器的构成与编码器十分类似，只是增加了与输入端的注意力交互。

一层，左侧的 N 表示层数。每一层包含两个子层：第一个子层是一个多头自注意力模块，第二个子层是一个简单的全连接前馈网络。在每个子层都有残差连接[60] 和层归一化[61] 来优化训练过程。假设输入为 x，每一个子层的输出为 LayerNorm$(x + \text{Sublayer}(x))$，其中 LayerNorm$(\cdot)$ 为层归一化，Sublayer$(\cdot)$ 表示对应子层实现的函数，即多头自注意力和前馈网络。

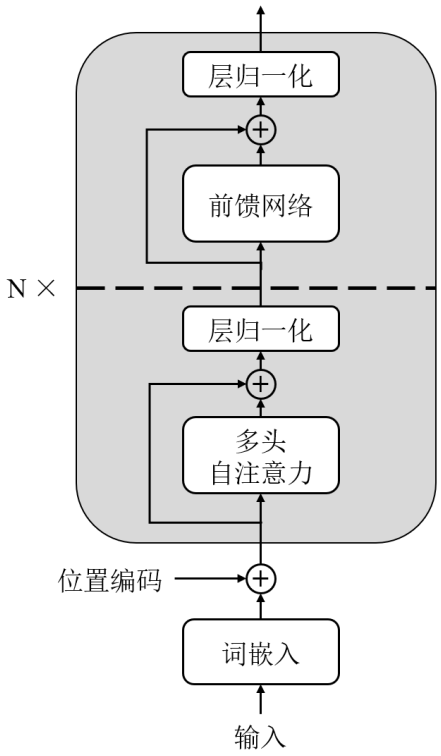


图 3.2 Transformer 编码器架构

下面先后介绍两个子层：第一个子层主要由多头自注意力模块构成，首先介绍自注意力机制的概念，接着描述 Transformer 中使用的多头自注意力计算方式；然后介绍第二个子层的主要构成部分——逐点前馈网络。最后介绍位置编码。

3.1.1 自注意力

注意力机制（attention mechanism）是一种强大的信息抽取方法，能够帮助模型动态地根据输入注意到重要的信息。在自然语言处理领域，注意力机制最初被用于机器翻译中[53]，使用注意力机制的编码器-解码器模型能够更好地进行单词对齐，从而提高翻译质量。

注意力机制首先根据查询（Query）向量和键（Key）向量计算得到一组注意

力分数，再利用注意力分数对值（Value）向量进行加权得到输出。其中，查询向量和键向量的计算方式有点积、双线性等形式，Transformer 使用的是一种缩放的点积形式。

自注意力是指查询、键、值都出自输入本身。假设有一个向量序列输入 $H = [h_1, h_2, \ldots, h_n]^\top \in \mathbb{R}^{n\times d}$，其中 n 为句子长度，d 为输入维度，则自注意力的计算方式为[2]

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^\top}{\sqrt{d_k}})V \qquad \text{式 (3-2)}$$

其中，$Q = HW^Q, K = HW^K, V = HW^V$ 且 $W^Q, W^K, W^V \in \mathbb{R}^{d\times d_k}$. Softmax 函数的形式参见式 (3-1)，这里未指定下标，因此函数的输出不是标量，而是样本被判定为各个类别的概率构成的向量或矩阵。特别地，在上式上，Softmax 函数的输出为一个 $n \times n$ 的矩阵，该矩阵被称为**注意力矩阵**。

3.1.2 多头自注意力

为了捕获更丰富的语义模式，提取句子元素之间更多的交互信息，Transformer 使用了**多头自注意力**（multi-head self-attention）机制：

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \qquad \text{式 (3-3)}$$

其中每个注意力头的结果使用式 (3-2) 计算得到：

$$\text{head}_i = \text{Attention}(HW_i^Q, HW_i^K, HW_i^V). \qquad \text{式 (3-4)}$$

这里，Concat 表示拼接，每个头的维度就是 d_k，输出矩阵 $W^O \in \mathbb{R}^{d_k h\times d}$. 在提出 Transformer 的论文中，作者设置 $d_k = d/h$，即令隐层维度被头的个数整除。

因此，第一个子层即是通过多头自注意力对整个输入句子进行建模，通过输出矩阵 W^O，该子层的输入和输出的维度保持一致。同时，这种建模方式是全局的，即句子中的每个记号都可以看到其他所有记号，因此在时间步上是全连接的。

[2]为简便起见，下面的公式中不再对向量加粗表示。

3.1.3 逐点前馈网络

第二个子层主要由一单隐层全连接前馈网络构成，其输入为第一个子层的输出。假设第一个子层的输出为 $x \in \mathbb{R}^{n \times d}$，则前馈网络的输出为：

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2. \quad \text{式 (3-5)}$$

其中，$W_1 \in \mathbb{R}^{d \times d_{ff}}, W_2 \in \mathbb{R}^{d_{ff} \times d}$，$d_{ff}$ 为前馈网络的隐层神经元个数。前馈网络的隐层使用的激活函数为 ReLU 函数，即 $\text{ReLU}(\cdot) = \max(0, \cdot)$.

需要注意的是，该前馈网络在输入句子的每个位置是共享的，即对每个位置（或者说每个点）使用相同的前馈网络，因此被称为逐点前馈网络（Point-wise Feedforward Network），在计算机视觉中，这种计算方式也被称为 1×1 卷积。

容易发现，第二个子层的输入和输出也保持了相同的维度。因此，Transformer 的每一层处理过程 $z^{(i)} \to z^{(i+1)}$ 并不改变数据维度。

3.1.4 位置编码

由于 Transformer 完全使用自注意力机制来对输入句子进行建模，无法将时序关系考虑进去，例如，对于不使用位置编码的 Transformer 来说，“猫坐在椅子上”和“椅子坐在猫上”两句话的表示并没有什么不同。因此需要在输入时加入额外的*位置编码*（position encoding）。通常来说，加入位置编码有两种方式：一种是作为可学习的参数使用梯度下降算法训练得到，另一种是人为地设计对位置和维度都敏感的编码函数，例如：

$$\text{PE}_{(pos,2i)} = \sin(pos/10000^{2i/d}) \quad \text{式 (3-6)}$$

$$\text{PE}_{(pos,2i+1)} = \cos(pos/10000^{2i/d}) \quad \text{式 (3-7)}$$

其中，pos 表示记号在句子中的位置，i 表示向量维度。

两种加入位置编码的方式互有优劣：将位置编码作为参数在数据量大时非常有效，但增加了需要学习的参数量；人为设计一种比较好的位置编码方式避免了

参数量的增长，并且在很多情形中也能取得很好的效果，但这种方式常常难以与预训练好的词向量兼容。

最终，句中每个单词的表示由词向量与位置编码相加组成，词向量可以随机初始化，也可以使用预训练好的词向量，如 word2vec，GloVe 等。

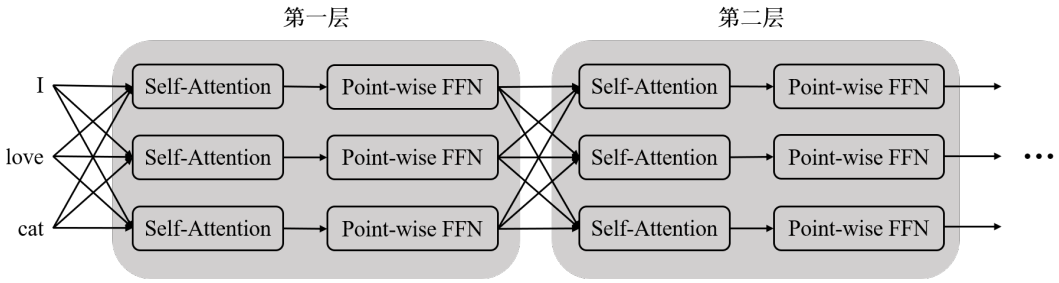


图 3.3　Transformer 编码过程

事实上，Transformer 就是利用注意力机制来建模句子中任意单词与其他单词之间的关系，是一种在时间步上全连接的全局建模方法。不同于传统的全连接网络，Transformer 可以处理变长的句子，因为连接权重是根据输入单词来动态生成的。图 3.3 给出了 Transformer 的编码过程，可见其第一个子层在输入句子的各个位置是全连接的，第二个子层则是逐点计算的。

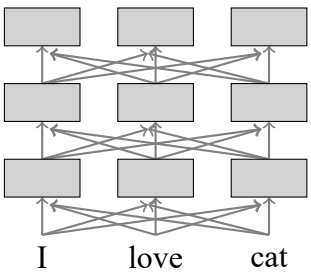


图 3.4　Transformer 结构的一个简化版示意图

本文中我们主要关注句子中各个位置之间的交互，由于对每一位置使用了相同的前馈网络，因而可以进一步忽略掉前馈网络部分，从而可以给出一个 Transformer 架构的简化图 3.4，下一节中也将基于类似的简化图来描述我们的多任务 Transformer 结构。

3.2 多任务 Transformer

在前文基础上，本节将展示四种基于 Transformer 的共享架构，其中两种为传统的硬共享模式，由于这种架构的任务特定层堆叠在共享层上面，在神经网络的顶层形成不同任务的表示，因此我们将其归纳为*顶层分化*；另外两种为针对 Transformer 的结构特点提出的共享模式，我们称之为*逐层分化*，也叫*逐层共享*，即在每一层都形成任务特定表示。

对于顶层分化模式，我们给出了两种具体的实现结构：S-P 结构和 S-C 结构。对于逐层分化模式，我们也给出了两种架构方式：L-I 结构和 L-E 结构。

3.2.1 S-P 结构

S-P 意为 Stack-Pooling，即*堆叠-池化结构*。S-P 结构是指在 Transformer 的共享层上使用*池化*（pooling）的方式来进行信息汇聚，从而得到通用句子表示。常用的池化方式一般有*平均池化*（mean pooling）和*最大池化*（max pooling）。以平均池化为例，假设包含 n 个单词的句子输入为 x，经过 Transformer 的 N 层共享层之后输出的隐编码为 $z^{(N)} = \mathcal{F}(x) \in \mathbb{R}^{n \times d}$，那么对于分类任务，采用平均池化的 S-P 结构的预测为

$$\hat{y} = \text{Softmax}(\frac{1}{n}\sum_{i=1}^{n} z_i^{(N)} \cdot W^t + b). \qquad \text{式 (3-8)}$$

其中任务 t 的分类矩阵 $W^t \in \mathbb{R}^{d \times c}$，$c$ 为分类个数。在实际实现时，平均池化后先接一多层感知机（MLP）再输入 Softmax 层常常能取得更好的效果。若 MLP 中隐层激活函数采用 ReLU，则式 (3-8) 可写为

$$\hat{y} = \text{Softmax}(\text{MLP}(\frac{1}{n}\sum_{i=1}^{n} z_i^{(N)})), \qquad \text{式 (3-9)}$$

$$\text{MLP}(x) = \max(0, xW_1^t + b_1^t) \cdot W_2^t + b_2^t. \qquad \text{式 (3-10)}$$

图 3.5 给出了 S-P 结构的示意，浅灰色模块表示共享部分，深灰色模块代表整个句子的表示，A, B, C 代表三个不同的任务。与 2.2 节中的图示不同的是，这里

的一个矩形代表一个单词的特征表示，而不是神经网络的一层。

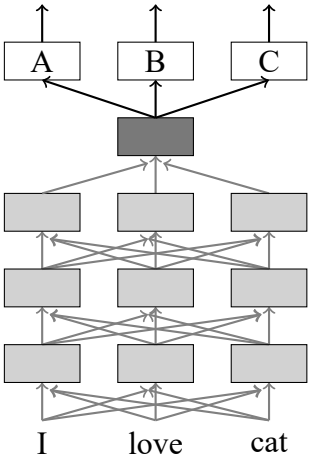


图 3.5　Stack-Pooling 结构

3.2.2 S-C 结构

S-C 意为 Stack-CLS，即堆叠-CLS 结构。S-C 结构是指在 Transformer 的输入时添加一个 CLS 记号用来捕捉句子在每一层的表示，最后使用 CLS 的顶层表示来作为句子的通用表示。CLS 为 Classification 的简写，该方法与 BERT[14] 中的设置一致。

假设记号 CLS 总是放在输入的最左端，那么最顶层的句子隐编码为 $z_{CLS}^{(N)} = z_0^{(N)} \in \mathbb{R}^d$，则 S-C 结构的输出为

$$\hat{y} = \text{Softmax}(z_0^{(N)} \cdot W^t + b). \qquad \text{式 (3-11)}$$

其中任务 t 的分类矩阵 $W^t \in \mathbb{R}^{d \times c}$，$c$ 为分类个数。图 3.6 给出了 S-C 结构的示意图，浅灰色为共享模块，深灰色为共享句子表示。

然而，在神经网络顶层形成的句子表示上进行分化得到任务特定表示的方法限制了任务特定表示对底层语义信息的使用。不同任务关注的句子信息可能在语法和语义层面都非常不同，为鼓励不同任务表示的差异化，可以在每一层都形成任务特定的表示，这就是逐层分化模式。下面介绍该模式的两种实现结构：L-I 结构和 L-E 结构。

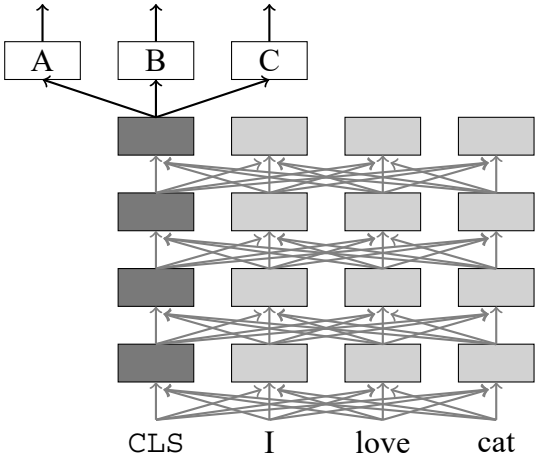


图 3.6 Stack-CLS 结构

3.2.3 L-I 结构

L-I 意为 Layerwise-Implicit，即*层级-隐式共享结构*。首先，L-I 结构将原来的 CLS 记号替换为 TASK_ID，该记号表示任务编号，每个任务编号对应一个不同的任务向量，执行不同任务通过输入不同的任务编号 TASK_ID 来控制。

在 L-I 结构中，不同任务之间无法显式地交互，只能通过共享模块来隐式地交互，但不同任务在每一层都有自己的表示，因而称为层级隐式共享。L-I 结构如图 3.7 所示。

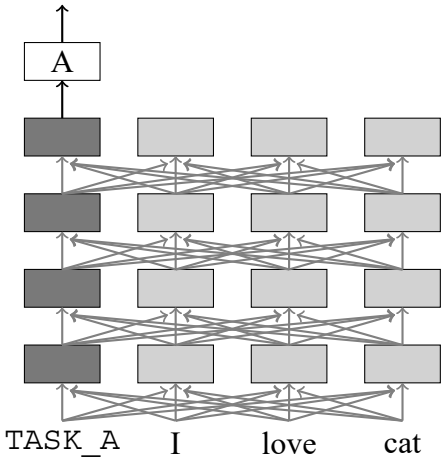


图 3.7 L-I 结构

L-I 结构的输入为 $x = (task\_id, x_1, x_2, ..., x_n)$，其中 $0 \leq task\_id < T$ 且 $0 \leq x_i < |V|$，其中 T 为联合学习的任务个数，词表大小为 $|V|$. 模型的输入层除词嵌入矩阵 $W^{word} \in \mathbb{R}^{|V| \times d_w}$ 外还需设置一任务嵌入矩阵 $W^{task} \in \mathbb{R}^{T \times d_t}$，这里 d_t 为任

务向量的维度，可与词向量维度 d_w 保持一致。假设词嵌入和任务嵌入都被线性投影到隐层维度 d，那么原始输入经过 L-I 结构的嵌入层后得到隐层的输入

$$z^{(0)} = W^{task}_{task\_id} \oplus W^{word}_{x_1} \oplus W^{word}_{x_2} \oplus \cdots W^{word}_{x_n}. \qquad \text{式 (3-12)}$$

其中 $\oplus$ 为拼接操作，拼接得到的 $z^{(0)} \in \mathbb{R}^{(n+1)\times d}$.

在输入到 Transformer 后，记号 `TASK_ID` 的计算过程与其他位置单词相同，模型应当学会根据输入的 `TASK_ID` 的不同来在每一层使用注意力机制提取相应任务关注的单词信息。假设 `TASK_ID` 总是作为第一个记号放置在输入单词之前，则模型预测输出的过程与式 (3-11) 相同。

3.2.4 L-E 结构

在 L-I 结构基础上，我们进一步提出了 L-E 结构，允许任务在每一层形成自己的特定表示时能够访问其他任务的表示。L-E 意为 Layerwise-Explicit，即*层级-显式共享结构*。L-E 结构如图 3.8 所示，任务 A 在形成自己的表示时可以访问任务 B 的表示，因而这种架构是显式共享的。各任务的特定句子编码由相应的 `CLS_ID` 形成。

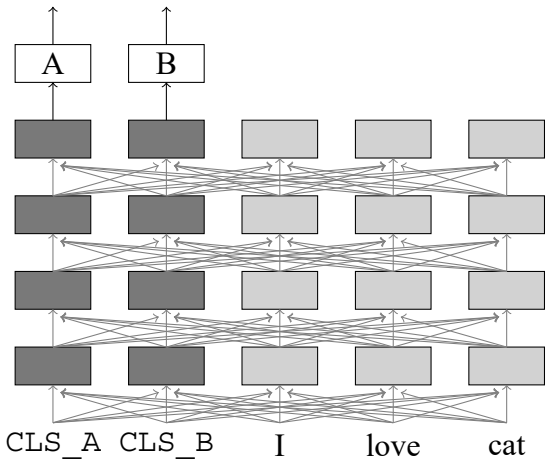


图 3.8　L-E 结构

需要注意的是，这里的 `CLS_ID` 与 L-I 结构中的 `TASK_ID` 不同，`CLS_ID` 是一个可学习的向量参数，无需在嵌入矩阵中查找。在 L-E 结构输入时，为每一个任务并列地设置一个不同的 `CLS_ID` 记号，因此可以看作是 S-C 结构的横向扩

展，但 L-E 结构的任务特定模块被堆叠在该任务的 CLS_ID 相对应的那一列。虽然每一个任务的 CLS_ID 都被同时输入给模型，但在训练和推断过程中都只使用目前关注的其中的一个任务记号。L-E 结构允许任务在形成自己每一层的句子表示时窃听其他任务是如何抽象句子特征的。

假设联合学习的任务数为 T，输入句子中记号个数为 n，隐层维度为 d，那么原始输入经过 L-E 结构的嵌入层后得到隐层输入及其后各隐层的隐编码为 $z^{(0)}, z^{(1)}, ..., z^{(N)} \in \mathbb{R}^{(T+n)\times d}$. L-E 结构根据当前执行的任务输出对应的预测结果：

$$\hat{y} = \text{Softmax}(z^{(N)}_{task\_id} \cdot W^t + b). \qquad \text{式 (3-13)}$$

其中 $0 \leq task\_id < T$，任务 t 特定的分类矩阵为 W^t.

3.2.5 模型对比

最后，在本小节简要地概括上述四种多任务 Transformer 结构的特点以及它们与已有共享模式的异同，

从共享模式上看，S-P 结构和 S-C 结构都属于硬共享模式，二者结构的低层都为任务共享层，任务特定的句子表示只能在网络顶层的任务特定模块形成。二者的区别在于 S-P 结构使用池化的方式根据各个记号的顶层隐编码得到句子表示，而 S-C 结构利用 CLS 的顶层隐编码作为其句子表示。

而 L-I 结构和 L-E 结构是逐层共享的，允许模型在每一层都形成自己任务特定的句子表示，增大了共享的灵活性。但 L-I 结构在抽取自己的句子特征时无法访问其他任务抽取的特征，因此是隐式共享的；而 L-E 结构允许同时形成多个任务的句子表示，从而使得每个任务在抽取自己需要的特征时可以窃听其他任务的特征，因此是显式共享的。

在形式上，L-I 结构和 L-E 结构与软共享模式类似，都允许某个任务在形成每一层隐编码时窃听其他任务的特征。然而，L-I 结构和 L-E 结构与之前的很多软共享方法[31][32] 的不同之处在于，L-I 结构和 L-E 结构可以根据输入样本的不同动态地决定与哪些任务共享以及共享多少特征。

3.3　实现细节

3.3.1 训练过程

为了避免可能出现的个别任务长时间不被选中的情况，本文采取的训练方式与一般的做法[3]稍有不同。下面的算法 1 给出了具体的训练算法。

算法 1 多任务联合训练过程

输入： M 个任务的数据集 $\mathcal{D}_m, 1 \leq m \leq M$；每个任务的批量大小 $K_m, 1 \leq m \leq M$；最大迭代次数 T；学习率 α.

输出： 模型参数 θ.

function TrainModel($\mathcal{D}_m$, K_m, T, α)
    初始化模型参数 θ_0
    初始化任务列表 L
    for $t = 1 \cdots T$ **do**
        for $m = 1 \cdots M$ **do**
            将 $\mathcal{D}_m$ 划分为 $c = N_m/K_m$ 个小批量集合：$\mathcal{B}_m = \{\mathcal{I}_{m,1}, \cdots, \mathcal{I}_{m,c}\}$
        end for
        $i = 1$
        while $|L| > 0$ **do**
            打乱任务列表 L 顺序
            for each $m \in L$ **do**
                if $\mathcal{I}_{m,i}$ 存在 **then**
                    计算小批量样本 $\mathcal{I}_{m,i}$ 上的损失 $\mathcal{L}$
                    更新参数：$\theta_t = \theta_{t-1} - \alpha \cdot \nabla_\theta \mathcal{L}(\theta)$
                else
                    将 m 从任务列表 L 中删除
                end if
            end for
            $i = i + 1$
        end while
    end for
    return θ_T
end function

在多任务联合训练时，也存在一部分较为复杂的训练策略。例如，除了通过均匀采样的方式挑选任务外，还可以按数据集比例为不同任务分配被选中的概

[3]更一般的训练方法可见文献[33]。

率[62]；也可以预先定义任务采样策略[63]；还可以采用不确定性来为不同任务的损失函数分配权重[43]。

3.3.2 超参数设定

实验中使用的均为 4 层 Transformer，模型维度为 300，包含 6 个注意力头，每个头为 50 维，前馈网络隐层维度为 512 维。Transformer 中的位置编码作为模型参数自动学习。使用 Adam 算法[64] 进行参数学习，初始学习率为 5e-4，最多训练 30 个轮次。训练使用的小批次（mini-batch）大小为 50.

实验中采用的词向量为在 840B 词汇量的 Common Crawl 预料集上预训练的 300 维 GloVe[17]。为避免学习算法对 GloVe 进行过多的修改，我们为词嵌入层设置了更小的学习率：5e-5.

对于 S-P 结构，实现时在输出预测前叠加了一层多层感知机（MLP）以增强其效果，其余结构无此设置。前文提到的神经网络均基于 fastNLP[4]和 PyTorch 实现，所有实验可在一张 NVIDIA TITAN Xp 上进行。

[4]https://github.com/fastnlp/fastNLP

第四章 实验

本章介绍验证模型结构所进行的实验。首先，使用 Transformer 作为基线模型实验了单任务学习下的模型性能，接着使用相同的超参设定实验了 3.2 节介绍的四种多任务 Transformer 的性能。具体地，在第 4.1 节介绍实验任务，在第 4.2 节介绍数据集相关信息，在第 4.3 节介绍实验结果。最后， 4.4 节给出了实验分析。

4.1　任务描述

我们在文本分类（Text Classification）这一经典 NLP 任务上进行实验。事实上，很多 NLP 问题都可以归为文本分类的范畴，例如情感分析（Sentiment Analysis, SA）、自然语言推理（Natural Language Inference, NLI）等。在目前被广泛使用的多任务基准数据集 GLUE[15] 中，所有任务都可以被归为文本分类任务。

文本分类即将一段文本归到某个特定的预先定义的类别。待分类的文本通常是一个句子（如情感分析）或一个句子对（如自然语言推理）。需要注意的是，这里的一个句子并不一定是常规意义上以一个句号为结束标识符的一句话，也有可能包含多句话。文本分类任务在现实生活中有着广泛的应用，如自动分析产品评价、微博情感分析、文档归类等等。文本分类任务需要模型能够抽取出易于分类的句子表示，即需要句子级的文本表示模型。

具体地，本文在情感分析任务上进行实验：对于一段用户生成的文本，模型需要判断其情感极性为正向还是负向。然而对于不同领域的文本通常需要关注不同的特征，例如在食物类的评论文本中模型应当更加关注“好吃”、“美味”等词，而在电影评论文本中应当更加关注“好看”、“烂片”等词语。同时，不同领域的文本分类任务常常也需要某些通用的特征，如“很棒”、“失望”等词在所有与情感倾向有关的分类任务中都是应当被关注的。而事实上，在某个特定的领域内，文本数据量常常是有限的，这种情况下单任务学习通常难以取得很好的效果，可以通过多任务学习来利用其他领域的相关知识帮助分类。

4.2 数据集

我们的基线单任务模型以及多任务模型在 16 个文本分类数据集上进行了对比实验，其中的前 14 个数据集来自亚马逊的产品评论[1]，但是来自各自不同的领域，如图书、电子、光盘等。该部分数据由 Blitzer 等人[65] 收集而成，其余两个数据集 IMDB[66] 和 MR[67] 则来自电影评论。每个数据集包含约 2000 个样本，其中 70% 划分为训练集，10% 划分为验证集，20% 划分为测试集。数据集的具体统计信息见表 4.1。

表 4.1 数据集统计数据

数据集	训练集大小	验证集大小	测试集大小	类别数	平均长度	词表大小
Books	1400	200	400	2	159	19K
Elec	1398	200	400	2	101	11K
DVD	1400	200	400	2	173	20K
Kitchen	1400	200	400	2	89	9K
Apparel	1400	200	400	2	57	7K
Camera	1397	200	400	2	130	9K
Health	1400	200	400	2	81	9K
Music	1400	200	400	2	136	17K
Toys	1400	200	400	2	90	10K
Video	1400	200	400	2	156	17K
Baby	1300	200	400	2	104	8K
Mag	1370	200	400	2	117	11K
Soft	1315	200	400	2	129	11K
Sports	1400	200	400	2	94	10K
IMDB	1400	200	400	2	269	25K
MR	1400	200	400	2	21	7K

这里所有数据集中的样本都被标注为两个类别，分别表示情感极性的正向和负向。如表 4.1 所示，虽然各数据集均为情感分析任务，且数据规模相当，但其样本来自不同的产品领域，平均句子长度和词汇量大小也各不相同，因此各个任务的难度也有所差别。

表 4.2 给出了其中几个数据集中的部分样例。

[1] https://www.cs.jhu.edu/~mdredze/datasets/sentiment/

表 4.2　数据集中的部分样本

领域	样例	类别
Books	It is a very dry book and hard to stay interested in. I am barely able to stay awake while reading it. It does have some interesting things.	negative
Mag	The magazine was shipped in a timely manner, i would use this vendor again.	positive
Elec	Very pleased with the high capacity cartridge with my epson stylus cx6600.	positive
MR	Just a big mess of a movie, full of images and events, but no tension or surprise.	negative
Health	Its quality is cheap and poorly made. I bought two thinking it was a good deal but I threw them out after 2 days because the on/off switch didn't work.	negative

4.3　实验结果

使用训练集对单任务基线模型以及四种多任务 Transformer 模型进行训练，选择在验证集上表现最好的模型进行测试。测试集上各模型的分类准确率如表 4.3 所示。可见，本文的四个多任务模型在十六个文本分类任务上均超过了单任务训练的表现，验证了多任务学习在 Transformer 模型上的有效性。其中，L-I 结构在四种多任务架构中表现最好，S-P 结构表现最差。注意到两种逐层共享的结构：L-I 结构和 L-E 结构，取得了相对传统硬共享结构（S-P 结构和 S-C 结构）更高的分类准确率，这说明在每一层都形成任务特定表示的模式相比在网络的顶层获取特征更为合理。随着任务之间差异性的增大，我们有理由认为 L-I 结构和 L-E 结构的共享模式将表现出更大的优越性。

值得注意的是，本文的四种多任务 Transformer 结构相比单任务学习下的设定并不会增加过多的参数，这一特点与硬共享模式类似，相对的，传统的软共享模式常常需要很多额外的参数。表 4.4 给出了单任务 Transformer 以及四种多任务 Transformer 结构的参数量，各模型的网络层数均为四层，参数量的统计忽略了词嵌入、位置编码等模型无关的参数。

表 4.3 模型在测试集上的分类准确率

数据集	单任务	多任务			
		S-P 结构	S-C 结构	L-I 结构	L-E 结构
Books	83.50	82.50	84.00	85.00	84.50
Elec	79.50	82.50	83.50	84.75	85.75
DVD	82.75	84.50	85.50	85.75	85.75
Kitchen	79.50	83.50	85.00	89.00	87.75
Apparel	82.75	85.50	86.75	86.00	85.75
Camera	81.75	84.25	85.00	87.00	89.00
Health	86.00	85.50	87.50	88.00	86.75
Music	76.50	83.00	83.00	82.75	81.50
Toys	80.00	84.75	86.25	88.25	86.50
Video	84.75	81.25	85.50	86.50	84.25
Baby	81.00	87.75	85.50	87.25	87.50
Mag	89.00	85.00	91.00	89.75	89.25
Soft	86.50	86.00	88.75	86.50	87.75
Sports	80.25	84.25	83.75	86.00	85.50
IMDB	80.75	84.75	85.00	84.50	84.50
MR	75.25	76.00	75.75	78.00	76.50
AVG.	81.86	83.81	85.11	**85.94**	85.53

可见，本文提出的几种多任务 Transformer 结构相比单任务模型并不会增加过多的参数，即使在十六个任务的情形下，最复杂的 L-I 结构和 L-E 结构也仅增加了 0.5% 的参数量。

另外，在实践中，神经网络的层数通常是一个比较重要的超参数，模型性能有时会由于层数的不同而产生较大差异。之前的实验中使用的模型均为 4 层 Transformer，为探究多任务 Transformer 结构对网络层数的敏感性，我们实验了四

表 4.4 各模型参数量对比

类型	模型	参数量	相对增量
单任务	Transformer	2,773,150	-
多任务	S-P 结构	2,782,180	+0.32%
	S-C 结构	2,782,480	+0.34%
	L-I 结构	2,786,980	+0.50%
	L-E 结构	2,786,980	+0.50%

种结构在不同层数下的平均分类准确率，结果如图 4.1 所示。

在各个层数设定中，逐层共享结构（L-I 结构和 L-E 结构）均优于传统的硬共享结构（S-P 结构和 S-C 结构），且 L-I 结构在三种设定中都取得了最优准确率，这一实验结果证明了模型的鲁棒性，也表明了表 4.3 数据的可信性。另外，在二、四、六层三种设定中，模型被设定为四层在除 S-P 结构中都在测试集上表现出了最高分类准确率，而 S-P 结构则随着层数增长而准确率下降，没有获得模型复杂度增长带来的收益。

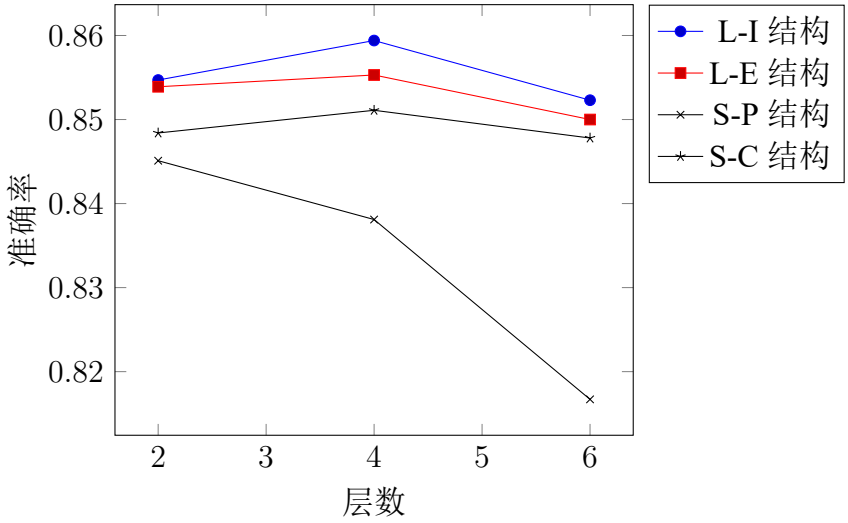
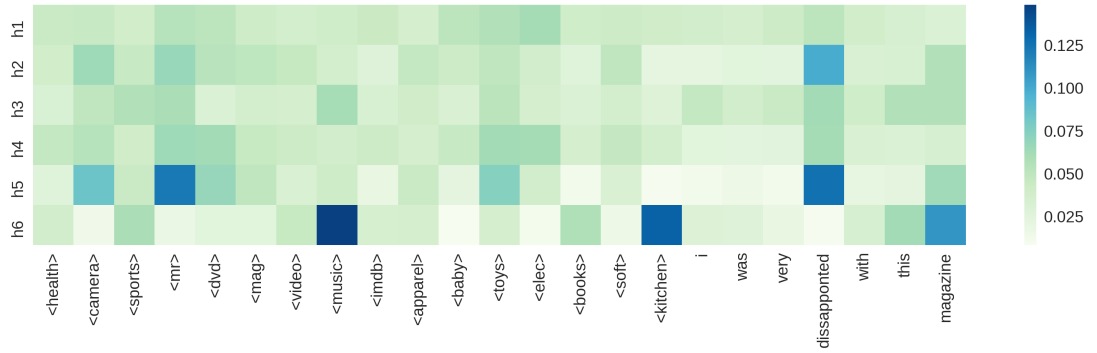


图 4.1　网络层数对测试集准确率的影响

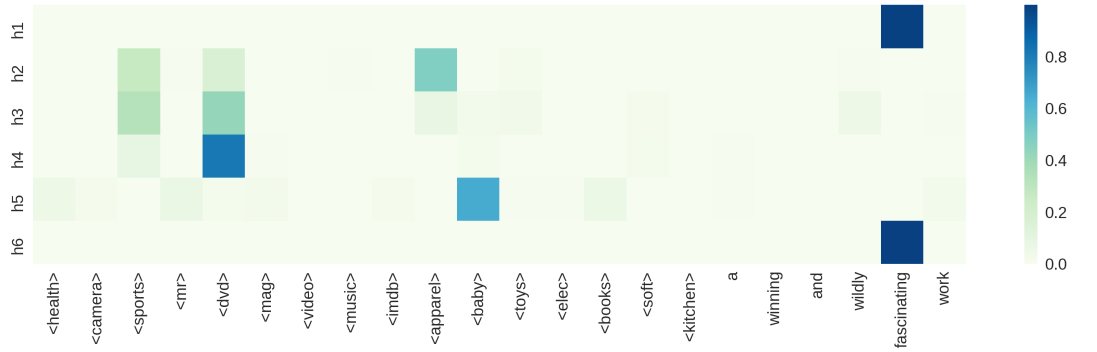
4.4　实验分析

Transformer 模型由于基于注意力机制，可以将注意力分数矩阵进行可视化来揭示模型的机理，因而相比其他深度模型具备较好的可解释性。另外，在多任务学习中，相关任务之间的交互关系一直是很多研究人员关注的重点问题。

考虑到以上两点，本节根据两个来自不同领域的示例输入对 L-E 结构中最后一层的注意力进行可视化分析，以探究模型有效的原因以及任务之间的关系。注意力矩阵的可视化结果如图 4.2 所示，每一行代表一个注意力头，每一列代表输入中的一个位置，其中前 16 个位置为任务标识符，后续位置为句子中的各个单词，两段输入文本分别来自杂志（Mag）数据集和电影评论（MR）数据集。



(a) 杂志（Mag）



(b) 电影评论（MR）

图 4.2　L-E 结构注意力可视化

可见，杂志领域情感分析任务与音乐和厨具领域的情感分析任务有很强的相关性，而电影评论情感分析任务与影碟产品领域的任务最为相似。同时还可以注意到，L-E 模型还注意到了输入句子中具有较强情感指示性的单词，如图 4.2 (a) 中的 disappointed（失望）以及图 4.2 (b) 中的 fascinating（惊艳的），这在一定程度上解释了模型有效的原因。

第五章 总结与展望

本章首先对本文的研究内容及其与已有工作的异同进行了总结，概括了本文工作的贡献和不足。随后，对多任务自然语言处理的研究前景进行了展望。

5.1　工作总结

本文研究了四种多任务 Transformer 架构，其中包含两种传统的硬共享结构，也包含两种新颖的逐层共享结构。据我们所知，本文是首次较为系统地在 Transformer 模型上研究多任务共享架构。近期，Liu 等人[28] 在 BERT[14] 的基础上提出了一种多任务 Transformer 模型 MT-DNN，在多个 NLP 任务上相较原始的单任务学习模型提升了性能。然而，MT-DNN 的模型架构仍是简单的硬共享模式，并没有为 Transformer 设计新的共享结构。本文的工作除了验证硬共享模式在 Transformer 上的有效性之外，还针对 Transformer 的结构特点设计了两种简单高效的逐层共享模式：L-I 结构和 L-E 结构。

逐层共享结构使得模型能够在每一层都形成自己的任务特定表示，而不局限于神经网络的顶层。事实上，多任务学习中另一种被广泛使用的共享模式——软共享——也可以在每一层形成任务特定表示。本文的 L-I 结构和 L-E 结构与传统的软共享模式的区别在于，L-I 结构和 L-E 结构基于注意力机制，能够动态地根据输入来决定与哪个任务共享，以及共享特征的程度。简单地说，L-I 结构和 L-E 结构更加灵活，能够自行决定何时共享、与谁共享、共享多少。另外，随着共享任务数量的增多，L-I 结构和 L-E 结构只需要增加少量的参数，而软共享往往需要增加大量参数，难以扩展。

本文在 16 个文本分类数据集上进行了实验，结果表明多任务 Transformer 模型一致性地超越了单任务学习模型的性能，并且 L-I 结构和 L-E 结构相对传统的硬共享结构取得了更好的效果。最后，讨论了四种结构对于网络层数的敏感性，并通过可视化实例分析解释了模型的工作机理，揭示了任务之间的相关性。

然而，本文提出的多任务学习模型架构也具有一定的局限性，只能适用于句子级别的任务，如文本分类、自然语言推理，难以应用于序列标注任务中。如何为序列标注任务设计有效的多任务 Transformer 结构仍是需要解决的问题。

5.2 未来展望

考察进入深度学习时代以来自然语言处理领域的研究进展，容易发现很多模型性能的重大突破都来自于预训练方法的改进，如 word2vec[16]、ELMo[11] 和 BERT[14]，这些预训练模型往往是通用的、可迁移的，因而在很多任务中都能带来明显的性能提升。随着这些通用模型的发展，越来越多的人开始思考一个令人兴奋的命题：是否能够训练单一模型来处理几乎所有的 NLP 任务？

然而，目前最强大的 BERT 也还无法成为这样的“单一模型”。近期的相关实验表明，BERT 在语义任务以及需要任务特定语法知识的任务上表现欠佳，对实体和指代现象的处理还不够好[18][19]，这说明，BERT 在某些情况下也并非如此“通用”。如何赋予预训练模型更多知识呢？

近期，百度对 BERT 的训练任务进行了改进，提出了基于中文的预训练模型 ERNIE，它具备了更多的实体知识。但其并未在根本上解决 BERT 面临的局限性。

造成这一局限性的一个关键原因在于，用来训练 BERT 的任务——完形填空和预测下一句话——并不能对所有 NLP 任务都带来很大提升。事实上，可能不存在某个单一预训练任务在所有目标任务下都非常有效。在这一背景下，一个很自然的想法便是利用多任务学习方法来训练。

本质上，迁移学习和多任务学习都是通过参数共享的方式来起作用。BERT 采用的模型结构便是 Transformer，因而 BERT 的成功证明了 Transformer 强大的知识存储能力和可迁移性，这为多任务 Transformer 的有效性提供了保障。我相信，随着对多任务 Transformer 结构的探索，越来越多的任务可以被同时训练，也会有越来越多的任务知识可以共存于单一模型中，对基于多任务学习的文本表示方法的研究将会带来更加强大的通用模型。

致谢

四载匆匆，倏忽而至，终于要毕业了。

依稀记得那段穿着军装、踢着正步、烈日永远不会缺席的日子，记得敲下的第一行代码，也还记得和队友们一起通宵做竞赛的疲惫，更记得无数个平凡的日子里坐在教室里一起听课的同学们和站在讲台上的教授们……

回首这段岁月，我只觉得幸运。何其幸运，能遇到这么多学识渊博、兢兢业业的老师；何其幸运，能遇到这么多优秀又谦虚的同学们。借此机会，我想向你们致以真挚的感谢。

首先，我想感谢即将成为我博士阶段导师的邱锡鹏老师和黄萱菁老师，他们提供的悉心指导和实验环境为本文的顺利完成提供了保障。在几个月的相处中，邱老师的学术品味和研究热情都让我受益匪浅，我相信我也将度过同样充实愉快的博士生涯。感谢刘西洋教授，为本文的完成提供了宝贵的修改意见。

然后，我要感谢本科阶段给予我巨大帮助的老师们和同学们。感谢陈慧婵老师和郭艳艳老师，他们为我打下了扎实的数理基础；感谢张淑平老师、张立勇老师和顾新老师，让我掌握了必要的专业知识；感谢周水生老师，在我的数模和科研上提供了大量指导。感谢一路同行的同学们，王磊、余天焕、徐之浩、班浩等等，你们一直是我成长路上的榜样；还要感谢陪伴我四年的室友，王敏锐、王许丞和冯尧，感谢你们一直以来的包容与帮助。

最后，我要感谢我的家人。感谢我的父母，在我的求学路上面临的每一次选择，你们都给予了最大的鼓励与信任，没有你们一如既往的支持，也不会有现在的我；还要感谢我的女朋友曲雪纯，遇到你是我本科阶段最大的收获之一。你们一直，也将永远是我最强大的后盾！

孙天祥

二零一九年五月于西电

参考文献

[1] Mikolov T, Deoras A, Povey D, et al. Strategies for training large scale neural network language models[C] // 2011 IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU 2011, Waikoloa, HI, USA, December 11-15, 2011. 2011 : 196 – 201.

[2] Li X, Hong C, Yang Y, et al. Deep neural networks for syllable based acoustic modeling in Chinese speech recognition[C] // Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2013, Kaohsiung, Taiwan, October 29 - November 1, 2013. 2013 : 1 – 4.

[3] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Commun. ACM, 2017, 60(6) : 84 – 90.

[4] Farabet C, Couprie C, Najman L, et al. Learning Hierarchical Features for Scene Labeling[J]. IEEE Trans. Pattern Anal. Mach. Intell., 2013, 35(8) : 1915 – 1929.

[5] Tompson J J, Jain A, LeCun Y, et al. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation[C] // Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. 2014 : 1799 – 1807.

[6] Collobert R, Weston J, Bottou L, et al. Natural Language Processing (Almost) from Scratch[J]. Journal of Machine Learning Research, 2011, 12 : 2493 – 2537.

[7] Bordes A, Chopra S, Weston J. Question Answering with Subgraph Embeddings[C] // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. 2014 : 615 – 620.

[8] Jean S, Cho K, Memisevic R, et al. On Using Very Large Target Vocabulary for Neural Machine Translation[C] // Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers. 2015 : 1 – 10.

[9] Sutskever I, Vinyals O, Le Q V. Sequence to Sequence Learning with Neural Networks[C] // Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. 2014 : 3104 – 3112.

[10] Deng J, Dong W, Socher R, et al. ImageNet: A large-scale hierarchical image database[C] // 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA. 2009 : 248 – 255.

[11] Peters M E, Neumann M, Iyyer M, et al. Deep Contextualized Word Representations[C] // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers). 2018 : 2227 – 2237.

[12] Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training[J]. URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf, 2018.

[13] McCann B, Keskar N S, Xiong C, et al. The natural language decathlon: Multitask learning as question answering[J]. arXiv preprint arXiv:1806.08730, 2018.

[14] Devlin J, Chang M-W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[15] Wang A, Singh A, Michael J, et al. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding[C] // Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018. 2018 : 353 – 355.

[16] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality[C] // Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.. 2013 : 3111 – 3119.

[17] Pennington J, Socher R, Manning C D. Glove: Global Vectors for Word Representation[C] // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. 2014 : 1532 – 1543.

[18] Tenney I, Xia P, Chen B, et al. What do you learn from context? Probing for sentence structure in contextualized word representations[J], 2018.

[19] Liu N F, Gardner M, Belinkov Y, et al. Linguistic Knowledge and Transferability of Contextual Representations[J]. arXiv preprint arXiv:1903.08855, 2019.

[20] Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning[C] // Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008. 2008 : 160 – 167.

[21] Caruana R. Multitask Learning: A Knowledge-Based Source of Inductive Bias[C] // Machine Learning, Proceedings of the Tenth International Conference, University of Massachusetts, Amherst, MA, USA, June 27-29, 1993. 1993 : 41 – 48.

[22] Dong D, Wu H, He W, et al. Multi-Task Learning for Multiple Language Translation[C] // Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers. 2015 : 1723 – 1732.

[23] Liu P, Qiu X, Huang X. Recurrent Neural Network for Text Classification with Multi-Task Learning[C] // Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016. 2016 : 2873 – 2879.

[24] Liu P, Qiu X, Huang X. Adversarial Multi-task Learning for Text Classification[C] // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers. 2017 : 1 – 10.

[25] Søgaard A, Goldberg Y. Deep multi-task learning with low level tasks supervised at lower layers[C] // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers. 2016.

[26] McCann B, Bradbury J, Xiong C, et al. Learned in Translation: Contextualized Word Vectors[C] // Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA. 2017 : 6297 – 6308.

[27] Subramanian S, Trischler A, Bengio Y, et al. Learning General Purpose Distributed Sentence Representations via Large Scale Multi-task Learning[C] // 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. 2018.

[28] Liu X, He P, Chen W, et al. Multi-Task Deep Neural Networks for Natural Language Understanding[J]. arXiv preprint arXiv:1901.11504, 2019.

[29] Anonymous. BAM! Born-Again Multi-Task Networks for Natural Language Understanding[H]. 2018.

[30] Zheng R, Chen J, Qiu X. Same Representation, Different Attentions: Shareable Sentence Representation Learning from Multiple Tasks[C] // Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.. 2018 : 4616 – 4622.

[31] Misra I, Shrivastava A, Gupta A, et al. Cross-Stitch Networks for Multi-task Learning[C] // 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. 2016 : 3994 – 4003.

[32] Ruder S, Bingel J, Augenstein I, et al. Latent Multi-task Architecture Learning[J], 2017.

[33] Caruana R. Multitask Learning[J]. Machine Learning, 1997, 28(1) : 41 – 75.

[34] Argyriou A, Evgeniou T, Pontil M. Multi-Task Feature Learning[C] // Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006. 2006 : 41 – 48.

[35] Lounici K, Pontil M, Tsybakov A B, et al. Taking Advantage of Sparsity in Multi-Task Learning[C] // COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009. 2009.

[36] Evgeniou T, Micchelli C A, Pontil M. Learning Multiple Tasks with Kernel Methods[J]. Journal of Machine Learning Research, 2005, 6 : 615 – 637.

[37] Jacob L, Bach F R, Vert J. Clustered Multi-Task Learning: A Convex Formulation[C] // Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008. 2008 : 745 – 752.

[38] Deng L, Hinton G E, Kingsbury B. New types of deep neural network learning for speech recognition and related applications: an overview[C] // IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013. 2013 : 8599 – 8603.

[39] Ramsundar B, Kearnes S M, Riley P, et al. Massively Multitask Networks for Drug Discovery[J]. CoRR, 2015, abs/1502.02072.

[40] Liu P, Fu J, Dong Y, et al. Multi-task Learning over Graph Structures[J]. arXiv preprint arXiv:1811.10211, 2018.

[41] Ruder S. An overview of multi-task learning in deep neural networks[J]. arXiv preprint arXiv:1706.05098, 2017.

[42] Meyerson E, Miikkulainen R. Beyond Shared Hierarchies: Deep Multitask Learning through Soft Layer Ordering[C] // 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. 2018.

[43] Kendall A, Gal Y, Cipolla R. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics[C] // 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. 2018 : 7482 – 7491.

[44] Hinton G E, McClelland J L, Rumelhart D E. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1[G] // . Cambridge, MA, USA : MIT Press, 1986 : 77 – 109.

[45] Bengio Y, Ducharme R, Vincent P. A Neural Probabilistic Language Model[C] // Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA. 2000 : 932 – 938.

[46] Kim Y. Convolutional Neural Networks for Sentence Classification[C] // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. 2014 : 1746 – 1751.

[47] Gehring J, Auli M, Grangier D, et al. Convolutional Sequence to Sequence Learning[C] // Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. 2017 : 1243 – 1252.

[48] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model[C] // INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010. 2010 : 1045 – 1048.

[49] Wen T, Gasic M, Mrksic N, et al. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems[C] // Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015. 2015 : 1711 – 1721.

[50] Ma X, Hovy E H. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF[C] // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers. 2016.

[51] Gers F A, Schmidhuber J, Cummins F A. Learning to Forget: Continual Prediction with LSTM[J]. Neural Computation, 2000, 12(10) : 2451 – 2471.

[52] Chung J, Gülçehre Ç, Cho K, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. CoRR, 2014, abs/1412.3555.

[53] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[C] // 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015.

[54] Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need[C] // Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA. 2017 : 6000 – 6010.

[55] Tang G, Müller M, Rios A, et al. Why Self-Attention? A Targeted Evaluation of Neural Machine Translation Architectures[C] // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018. 2018 : 4263 – 4272.

[56] Dehghani M, Gouws S, Vinyals O, et al. Universal transformers[J]. arXiv preprint arXiv:1807.03819, 2018.

[57] Guo M, Zhang Y, Liu T. Gaussian Transformer: a Lightweight Approach for Natural Language Inference[J], 2019.

[58] Guo Q, Qiu X, Liu P, et al. Star-Transformer[J]. arXiv preprint arXiv:1902.09113, 2019.

[59] Conneau A, Kiela D. SentEval: An Evaluation Toolkit for Universal Sentence Representations[C] // Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.. 2018.

[60] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C] // 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. 2016 : 770 – 778.

[61] Lei Ba J, Kiros J R, Hinton G E. Layer normalization[J]. arXiv preprint arXiv:1607.06450, 2016.

[62] Sanh V, Wolf T, Ruder S. A hierarchical multi-task approach for learning embeddings from semantic tasks[J]. arXiv preprint arXiv:1811.06031, 2018.

[63] Kiperwasser E, Ballesteros M. Scheduled Multi-Task Learning: From Syntax to Translation[J]. TACL, 2018, 6 : 225 – 240.

[64] Kingma D, Ba J. Adam: a method for stochastic optimization (2014)[J]. arXiv preprint arXiv:1412.6980, 2015, 15.

[65] Blitzer J, Dredze M, Pereira F. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification[C] // ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic. 2007.

[66] Maas A L, Daly R E, Pham P T, et al. Learning Word Vectors for Sentiment Analysis[C] // The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA. 2011 : 142 – 150.

[67] Pang B, Lee L. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales[C] // ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA. 2005 : 115 – 124.