西安电子科技大学
XIDIAN UNIVERSITY

计算机科学与技术学院
毕业设计（论文）中期报告

题目：基于多任务学习的文本表示方法研究

专业 ： 软件工程

班级 ： 卓越班

学号 ： 15130188018

姓名 ： 孙天祥

2019 年3 月28 日

目录

1 毕业设计的进展情况

1.1 课题工作完成情况

本课题主要研究多任务学习（Multi-Task Learning, MTL）在自然语言处理（Natural Language Processing, NLP）中的应用，旨在同时利用多个NLP任务的监督信号对模型进行训练，从而使模型超越在单一任务上的表现。

目前已经完成的工作包括：

1. **调研和学习已有的算法和框架**：通过搜集、查阅文献，了解深度学习背景下自然语言处理中的各类文本表示方法，以及多任务学习在文本表示中应用的主流模型。
2. **翻译文献**：翻译一篇超过一万字的英语文献。
3. **设计多任务学习模型**：针对已有方法的空缺或模型的缺点，设计新的多任务学习模型。
4. **搭建实验环境**：选择并学习合适的编程语言（Python）与实现框架（PyTorch），搭建实验环境。
5. **实现基线模型**：在单任务学习设定下实现基线（baseline）模型，并在常用数据集上进行实验，其结果作为新提出的多任务模型的参照。
6. **实现多任务学习模型**：编程实现自己设计的模型，通过实验对比与基线模型在各方面的性能差异。

1.2 知识学习情况

在本节中，我们会简要介绍了解本课题所需要的背景知识（包括自然语言处理中、多任务学习、自然语言处理中的多任务学习）、开发环境与工具、翻译外文资料情况。

在介绍具体的知识之前，我们首先对本课题要研究的内容进行一个粗略的定位。从题目“基于多任务学习的文本表示方法”来看，涉及到三个方面：多任务学习、表示学习、自然语言处理。其中，多任务学习与表示学习为工具，自然语言处理为应用领域。从机器学习的角度来看，深度学习本质上就是表示学习，而深度学习与多任务学

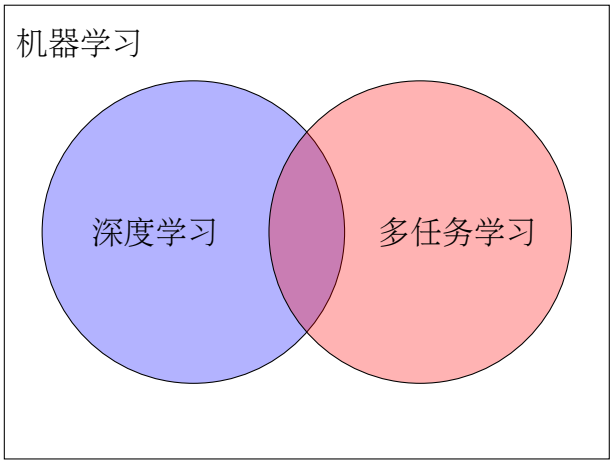


Figure 1.1: 机器学习、深度学习、多任务学习的关系

习都是机器学习的子领域，这里我们旨在使用深度学习中的多任务学习来处理自然语言文本，也即在深层神经网络中使用多任务学习来获得文本的表示。关于机器学习、深度学习和多任务学习的关系如图1.1 所示。

图中三个领域的交叉部分就是我们要使用的工具，在本课题中，我们将这一工具应用在自然语言处理中以获得更好的文本表示。下面将具体地介绍表示学习、多任务学习以及它们在自然语言处理中的应用。

1.2.1 自然语言处理中的表示学习

自然语言处理（Natural Language Processing, NLP）是一门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，随着数据量和算力的提升，深度学习在图像 [1] [2] [3]、语音 [4] [5]等多个领域得到了成功的应用，也在自然语言处理 [6] [7] [8] [9]中取得了广泛的成功。

深度学习能够成功的主要原因在于其强大的特征提取能力，因此深度学习也常常被称为表示学习（Representation Learning）。在传统的机器学习中，我们往往需要人工构造数据特征，例如N-Gram、TF-IDF等文本特征以及滤波得到的图像特征等，再将这些特征输入给学习器来进行分类或回归。而在深度学习中，神经网络自动根据任务学习合适的特征表示，并在最后一层或几层对提取出来的特征进行分类或回归。深度学习的这种学习方式被称作端到端（End-to-End）学习。

很自然地，由于深度学习强大的特征提取能力，人们将其应用在NLP领域。然而，

自然语言是离散的符号表示（这里不考虑连续的语音信号），过去的研究者处理自然语言的方式也通常是构造离散的符号处理系统。进入深度学习时代后，人们开始研究自然语言的分布式表示（Distributed Representation），从而使其便于被神经网络（在这里我们认为深度学习就是神经网络）处理。分布式表示是指将自然语言中的字、词、句子、篇章转变为向量表示。之所以被称为分布式表示，是因为语义被分散到向量的各个维度。

自2013年word2vec [10]被提出之后，自然语言的表示学习受到了研究者广泛的关注，涌现了GloVe [11]、FastText [12]等词级和字符级的向量表示。在更近期的研究中，为了解决上述词向量一词多义的缺陷，逐渐发展出更强大的表示方法，即上下文无关的表示，如ELMo [13]、CoVe [14]、GPT [15]、BERT [16]等，同时，用于学习文本表示的模型也变大越来越庞大。这些表示方法在自然语言处理中占据着非常重要的位置，在最近的五年中，除了机器翻译任务，几乎所有任务的当时最先进的（state-of-the-art）模型都使用了上述分布式表示方法。

1.2.2 多任务学习

在传统的机器学习或深度学习中，模型常常在单个任务、单个数据集上进行训练，从而学习如何执行该任务。从表示学习的角度来讲，也即是学习适用于该任务的特征表示。然而，单任务学习得到的模型常常面临着泛化能力弱、域适应（Domain Adaption）等问题。因此，有人提出了多任务学习（Multi-Task Learning, MTL）的概念，通过同时学习多个相关任务来提升在某个单一任务上的性能。

在早期的一些研究中 [17]，人们讨论了MTL的一些解释并证实了其在某些场景中的有效性。多任务学习可以被视作一种归纳偏置（Inductive Bias）迁移机制，通过利用相关任务的监督信号来提升模型的泛化性能。在机器学习中，对于一组数据，学习算法常常能找到多个模型来拟合，这些模型构成的空间被称作假设空间，由于同时使用多个任务对模型进行训练，使得算法倾向于选择一种能够同时解释多个任务的假设。从表示学习的视角来看，也即算法学习到了一种任务无关的通用表示，而这样的表示一般都具有强大的泛化能力。

在MTL的有效性上，除了表示偏置的解释之外，还有其他一些合理的解释：

- **数据增强（Statistical data amplification）** 由于多个任务的数据集通常是不

同的，使用多个任务对同一个模型进行训练相当于增大了训练数据量。

- **窃听（Eavesdropping）** 存在某些特征G 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。
- **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。

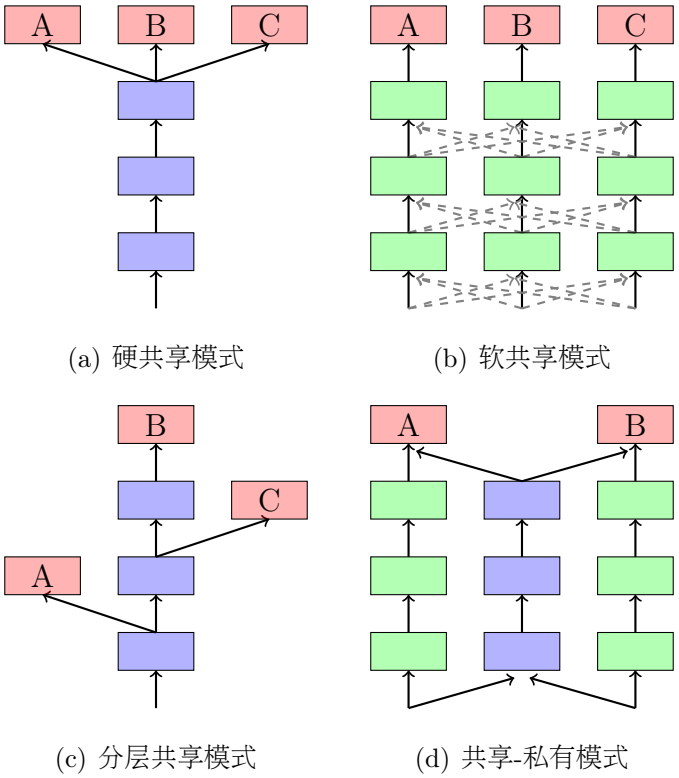
在早期的一些实验中，MTL被应用在k近邻、决策树以及简单的前馈神经网络中，在与单任务训练的对比中验证了其有效性。

随着深度学习在多个领域的成功，多任务学习也被广泛地应用在深层神经网络中，在自然语言处理、计算机视觉等多个领域上取得了较好的效果。事实上，在神经网络中引入多任务学习要比在传统机器学习模型中简单得多，图1.2 给出了目前常见的四种共享模式，图中蓝色矩形表示共享层，红色矩形表示输出层，绿色矩形表示模型私有隐层（但可能被其他任务的模型访问）。

- **硬共享模式**：让不同任务的神经网络模型共同使用一些共享模块（一般是低层）来提取一些通用特征，然后再针对每个不同的任务设置一些私有模块（一般是高层）来提取一些任务特定的特征。
- **软共享模式**：不显式地设置共享模块，但每个任务都可以从其它任务中“窃取”一些信息来提高自己的能力。窃取的方式包括直接复制使用其它任务的隐状态，或使用注意力机制来主动选取有用的信息。
- **分层共享模式**：一般神经网络中不同层抽取的特征类型不同。底层一般抽取一些低级的局部特征，高层抽取一些高级的抽象语义特征。因此如果多任务学习中不同任务也有级别高低之分，那么一个合理的共享模式是让低级任务在底层输出，高级任务在高层输出。
- **共享-私有模式**：一个更加分工明确的方式是将共享模块和任务特定（私有）模块的责任分开。共享模块捕捉一些跨任务的共享特征，而私有模块只捕捉和特定任务相关的特征。最终的表示由共享特征和私有特征共同构成。

事实上，硬共享模式和软共享模式是目前深度学习中MTL的两种最常见的模式，某

种程度上，分层共享模式可以归入硬共享模式，共享-私有模式可以归入软共享模式。



(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

Figure 1.2: 多任务学习的几种常见模式

假设有 T 个相关任务，每个任务 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 $\hat{y}_n^{(t)}$，则多任务联合学习的损失函数为

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

我们通过 λ_t 来控制任务 t 的损失函数的权重，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，也可以作为可学习的参数根据任务的不确定性来自主学习 [18]。

1.2.3 自然语言处理中的多任务学习

多任务学习在早期的基于神经网络的NLP模型中就已经得到了应用，Collobert等[19]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（Part-of-Speech Tagging）、语块标注（Chunking）、命名实体识别（Named Entity Recognition, NER）、语义角色标注（Semantic Role Labeling, SRL）、语义相

似度（semantically similar words）和语言模型（Language Modeling），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 [20]、文本分类 [21] [22]、序列标注 [23]等常见NLP任务上均取得了成功。

近期，随着迁移学习在NLP领域取得了巨大成功 [13] [15] [16]，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 [24]）以及一些多任务基准平台（DecaNLP [25]、GLUE [26]）。

2018年10月，基于迁移学习的模型（BERT [16]）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 [27] [28]。

事实上，在自然语言处理中使用的MTL框架大都也遵循上一节中介绍的几种共享模式，特别地，近年来的很多工作探索了MTL在循环神经网络上的应用 [21] [23]。

然而，目前还很少有工作在Transformer [29]上探索MTL的使用。随着Transformer在机器翻译 [29]、迁移学习 [15] [16]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。本课题探索了Transformer的几种经典的MTL架构，并设计了两种新型共享模式。

我们将在下一小节（1.2.4）中简要介绍现在流行的Transformer模型，在第1.3节中介绍新提出的基于Transformer的MTL模型。

1.2.4 基于自注意力的全连接神经网络模型

近年来，在自然语言处理与计算机视觉中，两种神经网络占据了主流地位：卷积神经网络（Convolutional Neural Network, CNN）与循环神经网络（Recurrent Neural Network, RNN）。其中，在计算机视觉中主要使用CNN，而在自然语言处理中主要使用RNN。

然而，在自然语言处理中，两种网络都存在难以解决的缺陷：1）CNN一般只能建模局部位置信息，难以捕捉长距离句子依赖；2）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。在两种网络结构都没有很好的解决语义建模问题的背景下，Transformer [29]应运而生，在计算速度、语义特征抽取和长距离依赖等方面都表现出强大的性能 [30]。Transformer完全基于自注意力（Self-Attention）机制，摒弃了之前常用的卷积结构和循环结构。下面，我们介绍Transformer的编码器部分，解码器在本课题以及一般非序列生成问题中不会被用到。

首先，Transformer编码器的目标与NLP中其他编码器没有区别，即将一组符号表示的序列 $(x_1, x_2, ..., x_n)$ 映射成一组低维稠密的向量表示 $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n)$. 给定 $\mathbf{z}$，解码器根据任务的不同生成单个标签 y，如文本分类任务，或者多个标签 $(y_1, y_2, ..., y_n)$，如序列标注任务。

Transformer模型架构

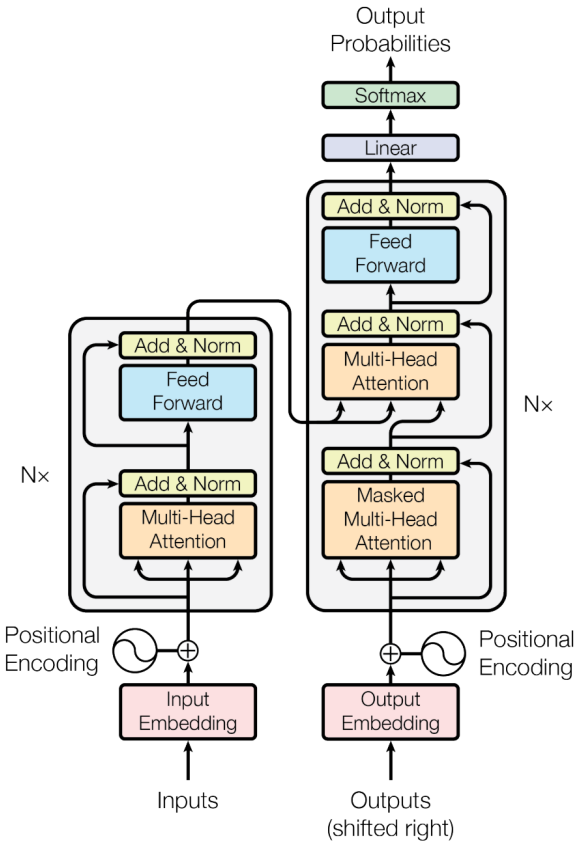


Figure 1.3: Transformer模型架构

Transformer模型架构如图1.3 所示，这里只会用到其中的编码器，即图中的左半部分。在图中，编码器由 $N = 6$ 个相同的层构成，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 [31]和层规范化 [32]。因此，每一个子层的输出为 $\text{LayerNorm}(x + \text{Sublayer}(x))$，其中 LayerNorm 为层规范化，$\text{Sublayer}(x)$ 表示对应子层实现的函数。

下面先介绍第一个子层，该子层由多头自注意力模块构成。

自注意力 注意力机制就是把一查询向量和一组键-值对映射为输出，这里的输入、查询（Query）、键（Key）、值（Value）和输出都为向量。输出向量由值向量加权得到，每个值向量的权重由查询和键向量计算得出，查询和键向量的计算方式有点积、双线性等形式。

自注意力是指查询、键、值都出自输入本身。假设我们有一个向量序列输入 $H = [\mathbf{h}_1, \ldots, \mathbf{h}_n]^\top \in \mathbb{R}^{n \times d}$，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：

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}}V)$$

其中，$Q = HW^Q, K = HW^K, V = HW^V$ 且 $W^Q, W^K, W^V \in \mathbb{R}^{d \times d_k}$.

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机制：

$$\begin{aligned}\text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_\text{h})W^O \\ \text{head}_\text{i} &= \text{Attention}(HW_i^Q, HW_i^K, HW_i^V).\end{aligned}$$

这里，每个头的维度就是 d_k.

全连接前馈网络 第二个子层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 $x \in \mathbb{R}^{n \times d}$，则第二个子层的输出为：

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

如上式所示，前馈网络的隐层使用的激活函数为ReLU函数。

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，因此需要加入额外的位置编码。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让模型自己学到，另一种是人为地设计对位置和维度敏感的编码函数，例如：

$$\mathrm{PE}_{(pos,2i)} = \sin(pos/10000^{2i/d})$$
$$\mathrm{PE}_{(pos,2i+1)} = \cos(pos/10000^{2i/d})$$

其中，pos表示记号在句子中的位置，i表示向量维度。最终，记号的表示由词向量与位置编码相加组成。

综上，Transformer实际上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。

从这种建模特点的角度，可以给出一个Transformer架构的简化图，下面我们也将基于简化图描述我们的多任务学习模型。

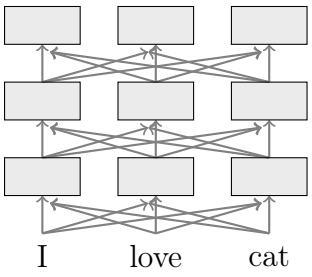


Figure 1.4: Transformer结构的一个简化版示意图

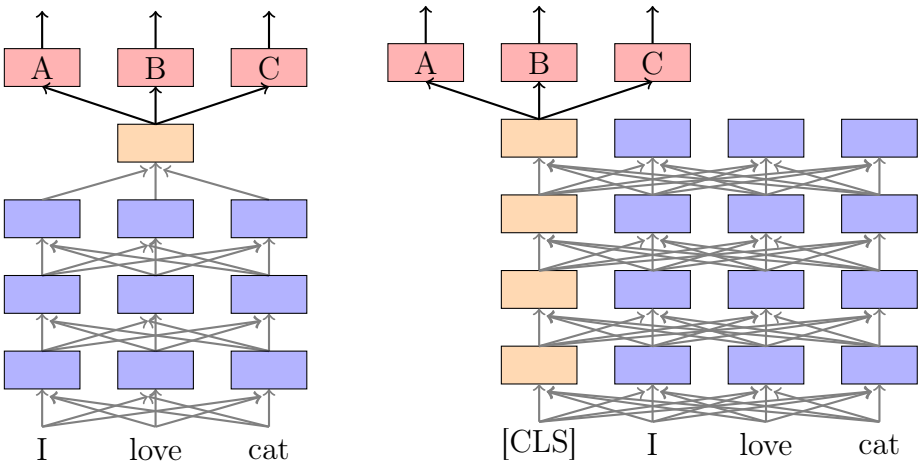
1.3 模型设计

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，属于旧方法在新模型上的应用，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提出的共享模式，属于就新模型提出的新方法，我们称之为**逐层分化**，即在每一层都形成任务特定表示。

下面我们首先介绍两种传统的硬共享模式在Transformer上的应用：第一种我

们称为**堆叠-汇聚（Stack-Pooling，下称S-P）**结构，指在Transformer的堆叠层上使用信息汇聚（也称池化）的方式来得到通用句子表示；第二种称为**堆叠-CLS（Stack-CLS，下称S-C）**结构，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 [16]中的设置一致。

S-P结构和S-C结构如图1.5所示，两种结构都在多个任务间共享相同的Transformer层，都假设存在某种句子的通用表示，不同任务的特定表示可以由该表示经过不同的任务特定层来获得。二者的不同之处在于获得最终句子表示的方式不同：S-P结构通过句子中每个记号的顶层表示的平均汇聚得到，而S-C结构通过记号[CLS]的顶层表示得到。

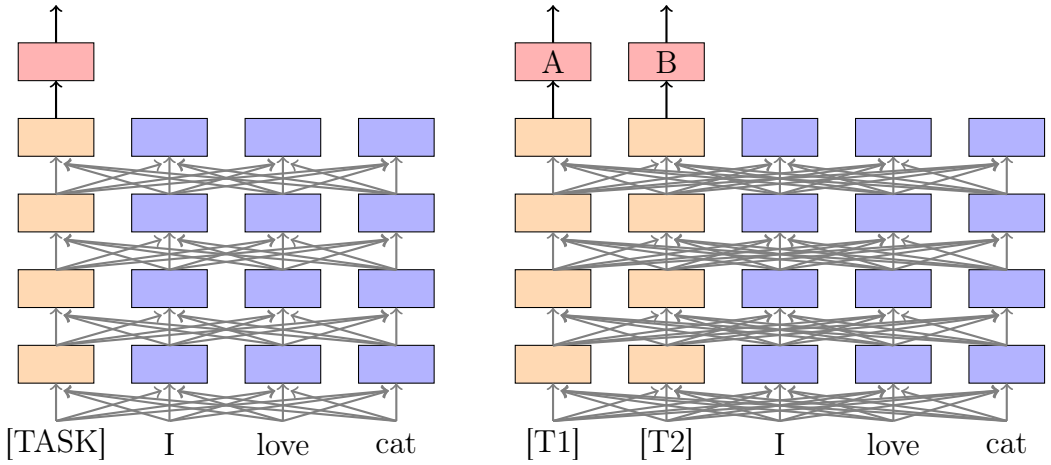


(a) Stack-Pooling结构　　(b) Stack-CLS结构

Figure 1.5: 基于Transformer的两种顶层分化共享模式

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。我们认为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，我们在每一层都形成任务特定的表示，从而提出了两种新的MTL架构：**层级-隐式（Layerwise-Implicit，下称L-I）**共享和**层级-显式（Layerwise-Explicit，下称L-E）**共享。

L-I结构与L-E结构如图1.6所示。其中，L-I结构将原来的[CLS]记号替换为[TASK]，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。同时，L-I结构不再有任务特定层，所有任务的输出层都是相同的，不同任务通过输入不同的[TASK]来控制。在L-I结构中，不同任务之间无法显式地交互，只能通过共享部分来隐式交互，但



(a) Layerwise-Implicit共享结构　　(b) Layerwise-Explicit共享结构

Figure 1.6: 基于Transformer的两种逐层分化共享模式

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。

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。如图1.6(b)所示，任务[T1]在形成自己的表示时可以访问任务[T2]的表示，因而这种架构称为**层级-显式**共享。

下面我们首先介绍开发上述四种模型以及单任务基线模型的环境与工具，接着在第1.5节展示这些模型的实验结果以及实现细节。

1.4 开发环境与工具

本课题中进行的所有实验均在以下环境中进行：

- **操作系统**　Ubuntu 16.04.2 LTS
- **CPU**　Intel(R) Xeon(R) CPU E5-2603 v4 @ 1.70GHz
- **内存**　120GB
- **显卡**　GeForce GTX 1080Ti
- **编程框架**　Python 3.6 & PyTorch 1.0 & fastNLP

1.5 实验

在本节我们首先介绍进行实验的数据集，然后介绍模型的实现细节，超参数设定以及实验结果。

1.5.1 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，每个数据集中的数据来自各自不同的领域文本，每个数据集包含两个类别。具体的数据集统计信息见表1.1。

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

Table 1.1: 数据集统计数据

1.5.2 实验结果

对比实验的对象包括五个模型：1）单任务模型；2）S-P模型；3）S-C模型；4）L-I模型；5）L-E模型。所有模型在实现过程中采取了相同超参数设定，均使用了四层Transformer，隐状态向量维度为300，使用6个注意力头，每个头的维度为50，前馈网络隐层维度为512，使用Adam优化，初始学习率为5e-4，优化批次大小为50。我们使用了预训练的300维GloVe词向量 [11]，位置编码作为模型参数自动学习。在S-P模

型中，我们使用多层感知机（Multi-Layer Perceptron, MLP）加一层Softmax作为输出层，其余模型的输出层都只使用了简单的线性层加Softmax。

实验结果如表1.2所示，可见：我们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。L-I结构和L-E结构取得的较好结果表明，在每一层都形成任务特定表示的做法相比在网络的顶层获取特征更为合理。同时，注意到S-C结构也能达到较好的准确率，这也肯定了之前被广泛使用的硬共享模式的效果，但随着任务之间差异性的增大，我们有理由认为L-I和L-E的共享架构会超越传统架构。

数据集	单任务(%)	S-P (%)	S-C (%)C	L-I (%)	L-E (%)
Books	83.5	82.5	85.25	85	87
Elec	79.5	82.5	87.5	84.75	85.75
DVD	82.75	84.5	82	85.75	84.75
Kitchen	79.5	83.5	87.25	89	86.5
Apparel	82.75	85.5	85	86	86
Camera	81.75	84.25	86.25	87	87
Health	86	85.5	87.25	88	86
Music	76.5	83	83.75	82.75	82
Toys	80	84.75	88.25	88.25	86.5
Video	84.75	81.25	85.5	86.5	84.25
Baby	81	87.75	86.5	87.25	86
Mag	89	85	89.5	89.75	90
Soft	86.5	86	88.25	86.5	86
Sports	80.25	84.25	85	86	85.75
IMDB	80.75	84.75	85.5	84.5	84.75
MR	75.25	76	75.75	78	74
AVG.	81.86	83.81	85.53	**85.94**	85.14

Table 1.2: 实验结果

1.6 翻译外文资料情况

目前已经完成了一篇综述文章《An Overview of Multi-Task Learning in Deep Neural Networks》的翻译工作，可以通过下面的地址访问译文及相关材料的在线版本：https://github.com/txsun1997/Graduation/tree/master/materials.

在该文章中，介绍了多任务学习的动机、解释、在非神经网络中的应用以及在神经网络中的应用。翻译文献的过程让我回顾了多任务学习的现有工作，并重新梳理了神经网络中的多任务学习脉络，也希望能作为毕业设计的补充材料帮助读者更好地理解这一领域。

2 存在问题与解决方案

2.1 存在的主要问题

现阶段存在的问题有：

- 仅在文本分类任务上对我们的多任务学习模型进行了实验，说服力较弱；
- 提出的L-E共享模式没有达到预期效果，平均表现甚至差于较为传统的S-C模型；
- 使用的数据集规模较小，无法验证模型的适用范围与应用前景。

2.2 解决方案与可行性研究

针对上面提到的主要问题，分别有如下解决方案：

- 在自然语言推理等其他经典NLP任务上实验，增强模型的说服力；
- 改进L-E共享模式，如为每个任务使用不同Query向量，或寻找适合L-E架构的应用场景来验证其有效性；
- 在中型、大型数据集上对模型进行测试，探索模型的适用范围。

3 下一步计划

根据上节中提到的主要问题及其重要程度，计划按如下顺序完成毕业设计及论文：

1. 开始撰写毕业论文，完成背景知识和模型的介绍；
2. 在GLUE等多任务基准数据集上使用更多样化的数据集和任务对模型进行测试；
3. 试着对模型结构进行更细粒度的改进；
4. 根据前两步的结果，完成毕业论文的实验部分及后续部分，并准备毕设答辩。

参考文献

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[2] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, 2013.

[3] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in Ghahramani *et al.* [33], pp. 1799–1807.

[4] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Cernocký, "Strategies for training large scale neural network language models," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU 2011, Waikoloa, HI, USA, December 11-15, 2011* (D. Nahamoo and M. Picheny, eds.), pp. 196–201, IEEE, 2011.

[5] X. Li, C. Hong, Y. Yang, and X. Wu, "Deep neural networks for syllable based acoustic modeling in chinese speech recognition," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2013, Kaohsiung, Taiwan, October 29 - November 1, 2013*, pp. 1–4, IEEE, 2013.

[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[7] A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," in Moschitti *et al.* [34], pp. 615–620.

[8] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On using very large target vocabulary for neural machine translation," in *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* [35], pp. 1–10.

[9] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Ghahramani *et al.* [33], pp. 3104–3112.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *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.* (C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 3111–3119, 2013.

[11] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in Moschitti *et al.* [34], pp. 1532–1543.

[12] E. Grave, T. Mikolov, A. Joulin, and P. Bojanowski, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers* (M. Lapata,

P. Blunsom, and A. Koller, eds.), pp. 427–431, Association for Computational Linguistics, 2017.

[13] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *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)* (M. A. Walker, H. Ji, and A. Stent, eds.), pp. 2227–2237, Association for Computational Linguistics, 2018.

[14] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in Guyon *et al.* [36], pp. 6297–6308.

[15] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*, 2018.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[17] R. Caruana, "Multitask Learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[18] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 7482–7491, IEEE Computer Society, 2018.

[19] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008* (W. W. Cohen, A. McCallum, and S. T. Roweis, eds.), vol. 307 of *ACM International Conference Proceeding Series*, pp. 160–167, ACM, 2008.

[20] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, "Multi-task learning for multiple language translation," in *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* [35], pp. 1723–1732.

[21] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016* (S. Kambhampati, ed.), pp. 2873–2879, IJCAI/AAAI Press, 2016.

[22] P. Liu, X. Qiu, and X. Huang, "Adversarial multi-task learning for text classification," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers* (R. Barzilay and M. Kan, eds.), pp. 1–10, Association for Computational Linguistics, 2017.

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

[24] A. Conneau and D. Kiela, "Senteval: An evaluation toolkit for universal sentence representations," in

Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018. (N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, eds.), European Language Resources Association (ELRA), 2018.

[25] B. McCann, N. S. Keskar, C. Xiong, and R. Socher, "The natural language decathlon: Multitask learning as question answering," *arXiv preprint arXiv:1806.08730*, 2018.

[26] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018* (T. Linzen, G. Chrupala, and A. Alishahi, eds.), pp. 353–355, Association for Computational Linguistics, 2018.

[27] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," *arXiv preprint arXiv:1901.11504*, 2019.

[28] Anonymous, "Bam! born-again multi-task networks for natural language understanding." anonymous preprint under review, 2018.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Guyon *et al.* [36], pp. 6000–6010.

[30] G. Tang, M. Müller, A. Rios, and R. Sennrich, "Why self-attention? A targeted evaluation of neural machine translation architectures," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018* (E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, eds.), pp. 4263–4272, Association for Computational Linguistics, 2018.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778, IEEE Computer Society, 2016.

[32] J. Lei Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[33] Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds., *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014.

[34] A. Moschitti, B. Pang, and W. Daelemans, eds., *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*, ACL, 2014.

[35] *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*, The Association for Computer Linguistics, 2015.

[36] I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, 2017.