西安电子科技大学
XIDIAN UNIVERSITY

基于多任务学习的文本表示方法研究

Language Representation with Multi-Task Learning

孙天祥
txsun@stu.xidian.edu.cn

2019年5月

几个问题…

- 题目　基于多任务学习的文本表示方法研究

Q1	为什么要做文本表示?
Q2	为什么要用多任务学习做文本表示?
Q3	如何用多任务学习做文本表示?

目录 – 研究意义

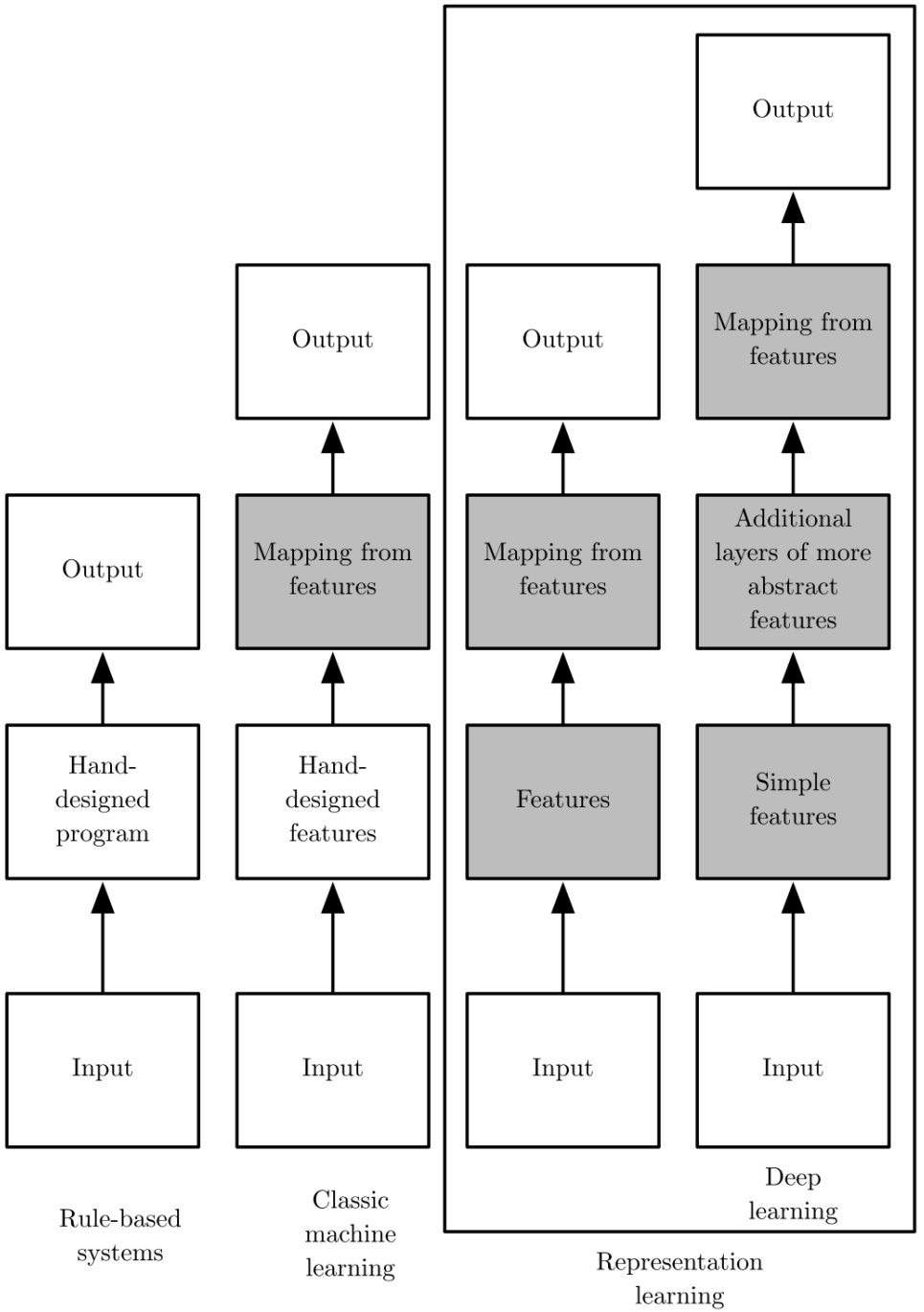
- 研究意义
  - 深度学习
  - 多任务学习
  - 自然语言处理
- 相关研究进展
- 模型与实验
- 总结

深度学习（deep learning）

- 深度学习是实现人工智能（AI）的一个可能手段
- 深度学习≈神经网络
  - 前馈网络（MLP、CNN）
  - 反馈网络（RNN、LSTM、GRU）
  - 图网络（GNN）
- AI的很多子问题的最优解决方案

深度学习（deep learning）

- 深度学习是一种表示学习！
- 信息表示的重要性
  - 210 ÷ 6 = ?
  - CCX ÷ VI = ?
- 分布式表示
  - RGB

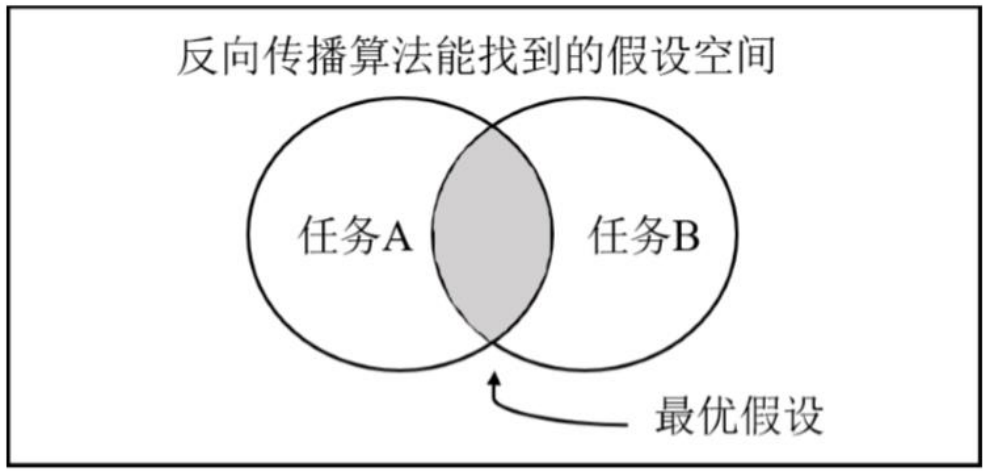


深度学习（deep learning）

- **什么是好的表示?**
  - 使后续任务简单
  - 任务无关的一般表示
- **深度学习能否得到好的表示?** 不一定!
- **如何得到好的表示?**
  - 多任务学习

多任务学习（multi-task learning, MTL）

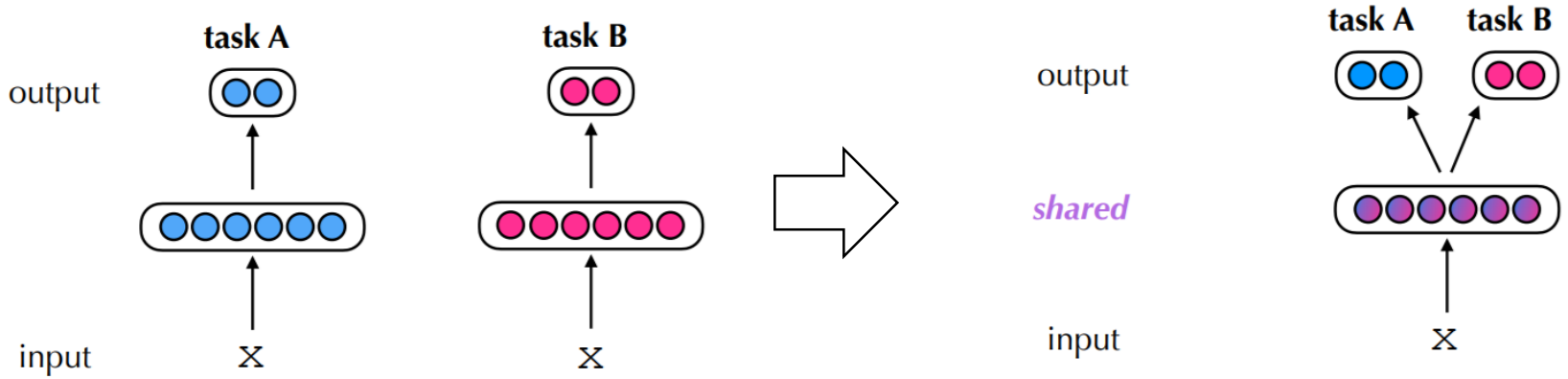
- 多任务学习是一种归纳转移（inductive transfer）方法，通过利用包含在相关任务训练信号中的领域特定信息来提升泛化能力[1]
- 使用多个任务对模型进行训练，使该模型学习到多个任务的**共享表示**



[1] R. Caruana, “Multitask Learning,” Machine Learning, vol. 28, no. 1, pp. 41–75, 1997.

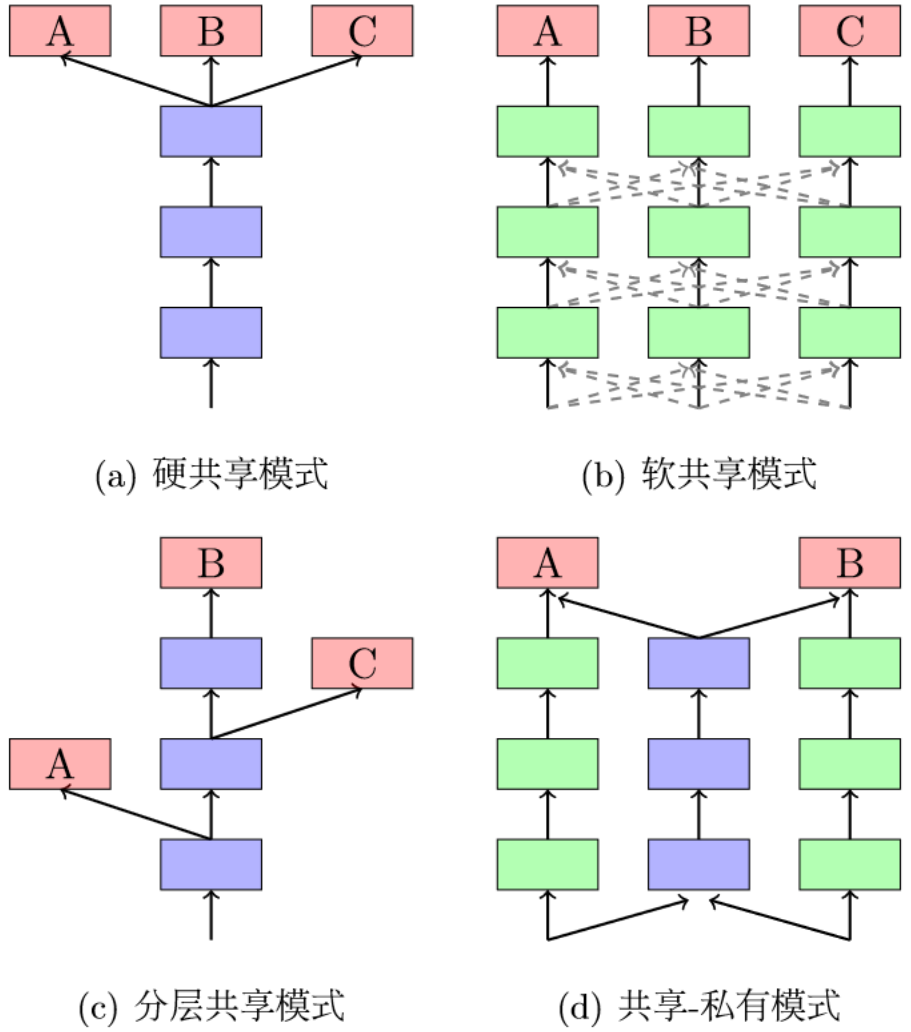
多任务学习（multi-task learning, MTL）

- MTL是模型无关的，可以用在各种机器学习模型中
  - 在神经网络中使用MTL更加简单



多任务学习（multi-task learning, MTL）

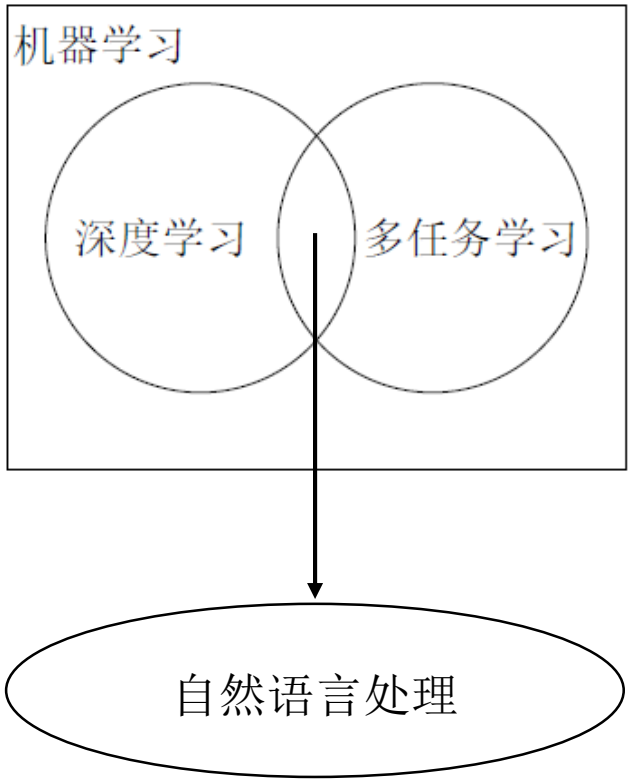
- 难点：设计合适（?）的**共享模式**
- 目前已有的共享模式：
  - 硬共享
  - 软共享
  - 分层共享
  - 共享-私有
  - 函数共享
  - 主辅共享
  - …



(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

自然语言处理（natural language processing, NLP）



自然语言处理（natural language processing, NLP）

- **自然语言**
  - ≈人类语言
  - ≠程序语言
- **使用计算机技术处理、理解和生成自然语言**
- **相似概念**
  - 计算语言学（computational linguistics, CL）
  - 自然语言理解（natural language understanding, NLU）
- **一切与处理文本语言相关的问题，都可以归为NLP的问题**

自然语言处理（natural language processing, NLP）

- 自然语言处理中的神经网络
  - 卷积网络（CNN）
  - 循环网络（RNN）
  - Transformer[2]
- 表示对于深度学习非常重要，文本表示对于NLP模型更为重要!

[2] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*. 2017.

自然语言处理（natural language processing, NLP）

- 文本是离散的符号，表示难度更大

(a) 图像数据

(b) 语音数据

我喜欢猫

I love cat

Amo el gato

أنا أحب القطط

(c) 文本数据

- 文本标注成本高，特定领域文本数据量有限，难以学习到好的表示

自然语言处理（natural language processing, NLP）

- 预训练文本的分布式表示
  - **Non-textualized** representation
    - word-level: word2vec, GloVe, etc.
    - char-level: fastText, etc.
  - **Contextual** representation
    - RNN: ELMo, CoVe, etc.
    - Transformer: GPT, BERT, etc.
- 性能提升显著！

Q1 为什么要做文本表示?

自然语言处理（natural language processing, NLP）

- 然而，文本的表示问题仍未被完全解决…
  - 通用表示的不足
    - 实体指代、复杂语法…
    - 需要任务/领域为什么要用
  - 通用模型的需求
    - 特征工程 → 结构工程
    - decaNLP、GLUE…
- **文本表示需要多任务学习的方法**

Q2 为什么要用多任务学习做文本表示?

目录 – 相关研究进展

- 研究意义
- 相关研究进展
  - 基于深度学习的自然语言处理
  - 自然语言处理中的多任务学习
- 模型与实验
- 总结

基于深度学习的自然语言处理（DL based NLP）

- 文本的分布式表示
- 神经网络模型
  - 卷积神经网络（convolutional neural network, CNN）
  - 循环神经网络（recurrent neural network, RNN）
  - Transformer
- 数据量有限时，常常难以学到好的表示

自然语言处理中的多任务学习（MTL for NLP）

- MTL with CNNs
  - Collobert et al. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning
  - Misra et al. Cross-stitch Networks for Multi-task Learning
- MTL with RNNs
  - Dong et al. Multi-Task Learning for Multiple Language Translation
  - Liu et al. Recurrent Neural Network for Text Classification with Multi-Task Learning
  - Hashimoto et al. A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks
- MTL with Transformer
  - ?

Q3 如何用多任务学习做文本表示?

目录 – 模型与实验

- 研究意义
- 相关研究进展
- 模型与实验
  - Transformer
  - 多任务Transformer
  - 实验任务
  - 实验结果
- 总结

Transformer

- 自注意力

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

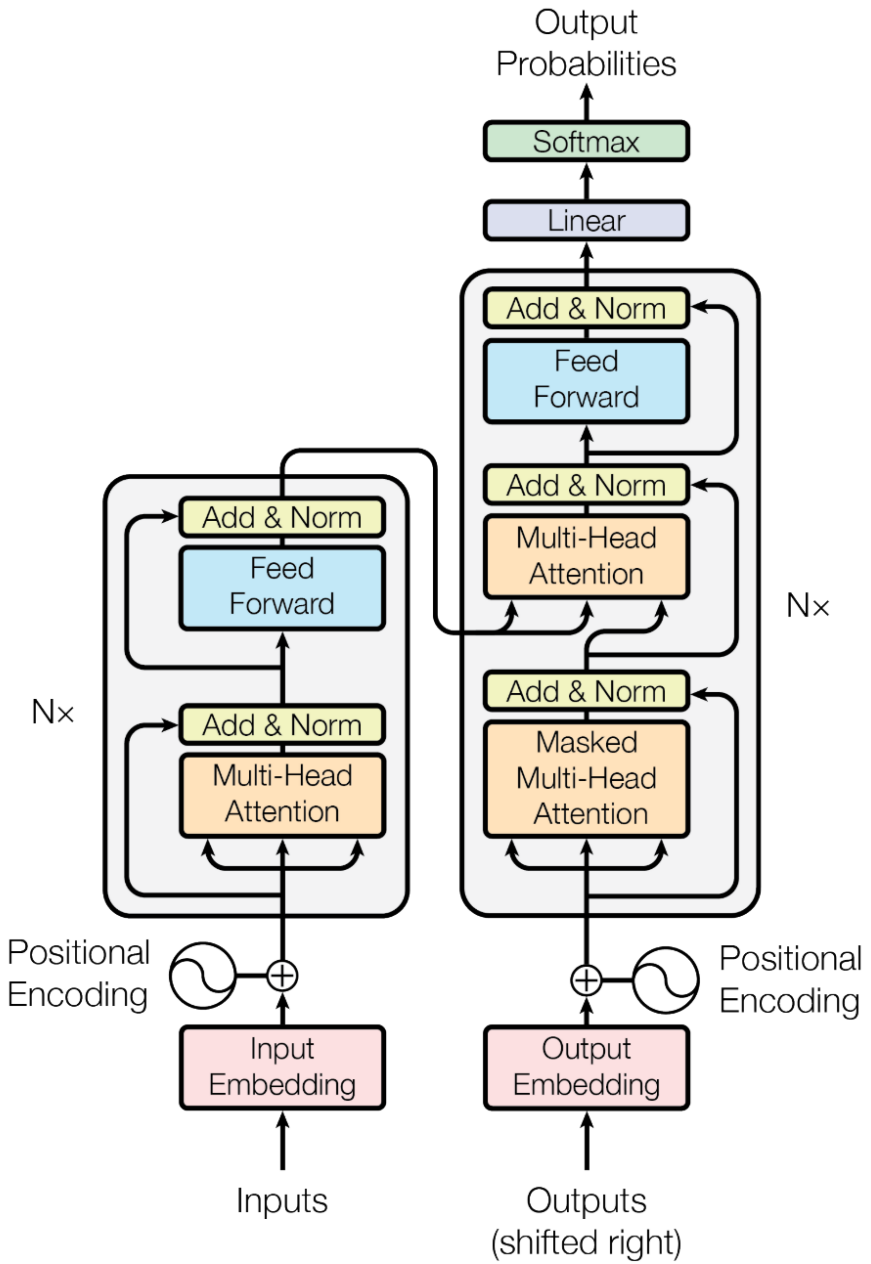
- 多头自注意力

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

$$\text{head}_i = \text{Attention}(HW_i^Q, HW_i^K, HW_i^V).$$

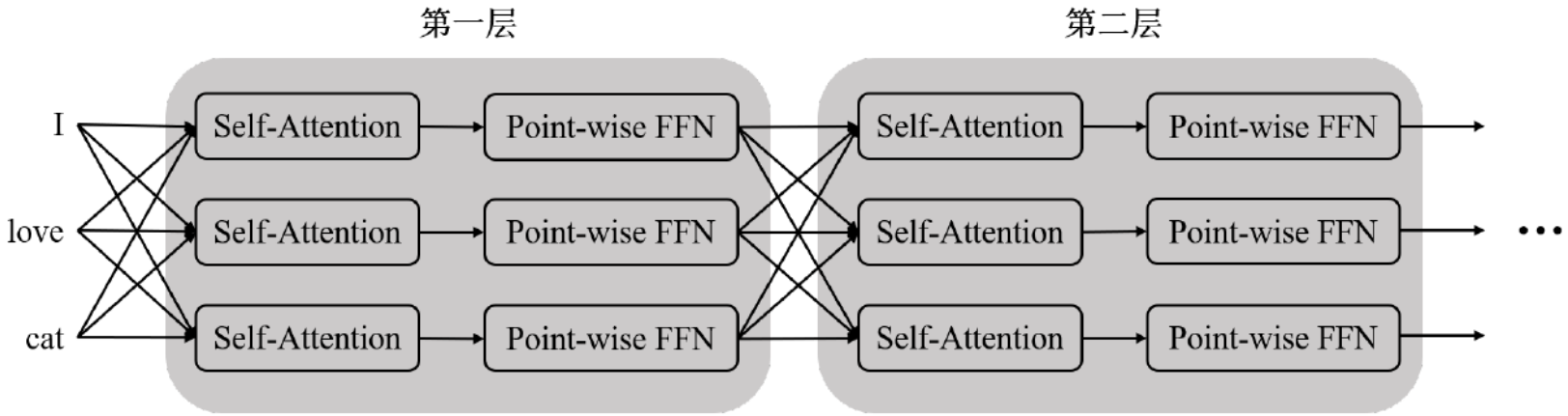
- 逐点前馈网络

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

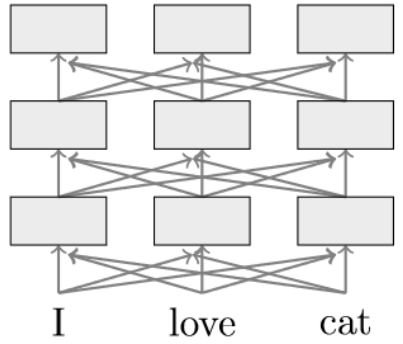


[2] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*. 2017.

Transformer



Transformer编码过程



一个简化版本

多任务Transformer

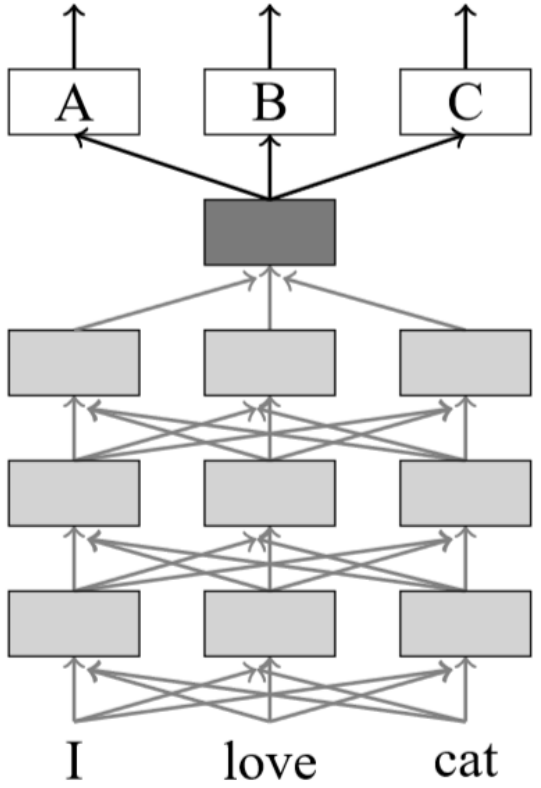
- **顶层分化**
  - S-P结构（Stack-Pooling，堆叠-池化）
  - S-C结构（Stack-CLS，堆叠-CLS）
- **逐层分化**
  - L-I结构（Layerwise-Implicit Sharing，逐层-隐式共享）
  - L-E结构（Layerwise-Explicit Sharing，逐层-显式共享）

S-P结构

- 句子表示通过池化（pooling）得到
- 共享层形成通用表示，顶层堆叠任务特定层形成任务特定表示
- 预测方式：

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

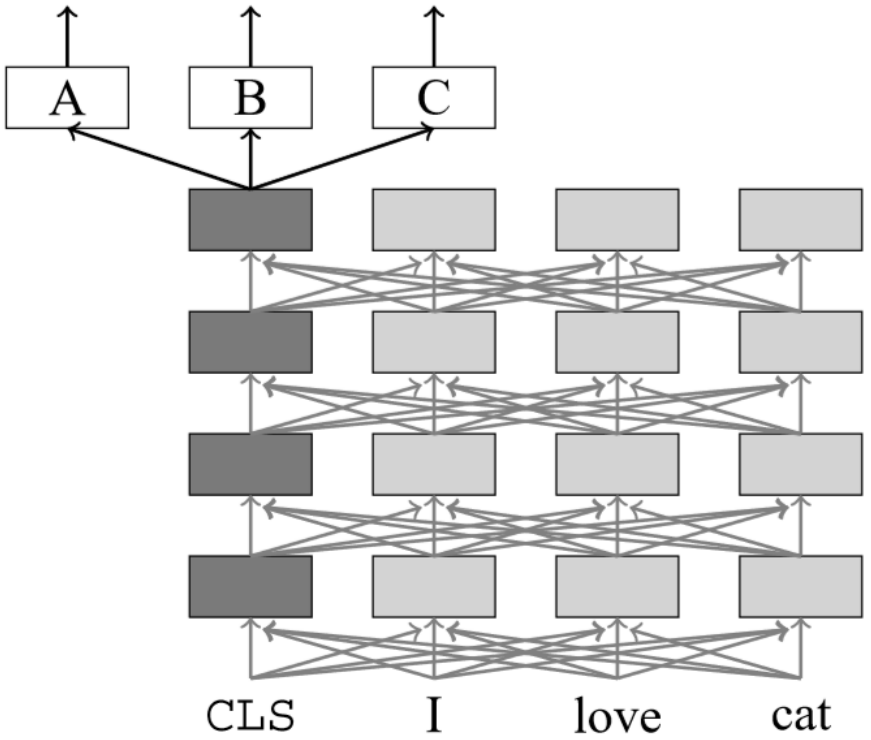
$$\text{MLP}(x) = \max(0, xW_1^t + b_1^t) \cdot W_2^t + b_2^t.$$



S-C结构

- 句子表示通过 [CLS] 记号得到
- 共享层形成通用表示，顶层堆叠任务特定层形成任务特定表示
- 预测方式：

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



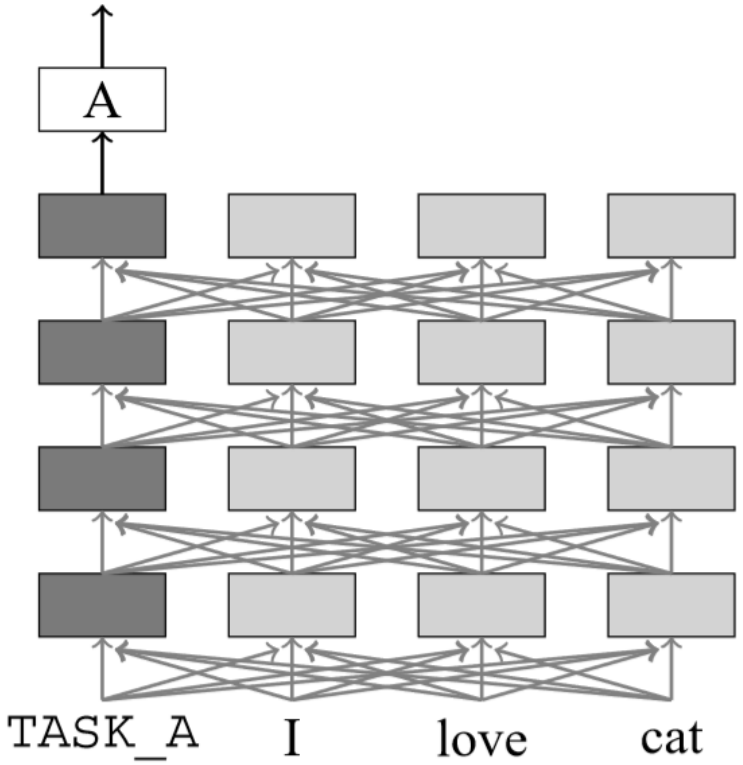
L-I结构

- 句子表示通过 [TASK_ID] 得到
- 在每一层形成任务特定表示
- 隐层的输入为：

$$z^{(0)} = W^{task}_{task\_id} \oplus W^{word}_{x_1} \oplus W^{word}_{x_2} \oplus \cdots W^{word}_{x_n}.$$

- 预测方式：

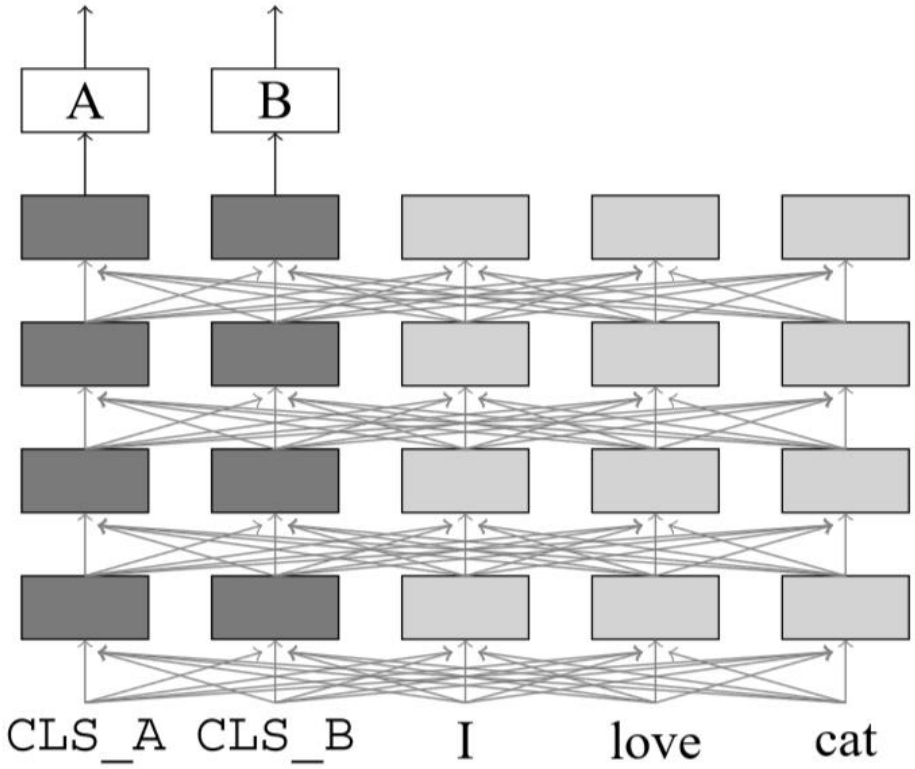
$$\hat{y} = \text{Softmax}(z_0^{(N)} \cdot W^t + b).$$



L-E结构

- 句子表示通过 [CLS_ID] 得到
- 在每一层形成任务特定表示
- 预测方式：

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



模型对比

模型	形成句子表示方式	共享模式	共享方式
S-P结构	池化	硬共享	隐式共享
S-C结构	特定记号	硬共享	隐式共享
L-I结构	特定记号	逐层共享	隐式共享
L-E结构	特定记号	逐层共享	显式共享

- 软共享也可以在每一层形成任务特定表示，因此逐层共享可以归为软共享的范畴，但与传统的软共享模式由有一些不同
  - 参数量大大减少
  - 根据输入动态共享（与谁共享、共享多少）

模型实现

开发环境

软件环境	操作系统	Ubuntu 16.04.2 LTS
	编程框架	PyTorch 1.0 & fastNLP 0.3
	编程语言	Python 3.6
硬件环境	CPU	Intel(R) Xeon(R) E5-2603 @ 1.70GHz
	内存	120GB
	显卡	NVIDIA TITAN Xp

训练过程

算法 1 多任务联合训练过程

输入: M 个任务的数据集 $\mathcal{D}_m, 1 \le m \le M$; 每个任务的批量大小 $K_m, 1 \le m \le M$; 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

实验任务

- 文本分类（情感分析）
- 16个数据集，样本分别来自不同领域
- 每个数据集约2k样本，按7-1-2划分为训练集、验证集、测试集

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

实验任务

- 部分样本

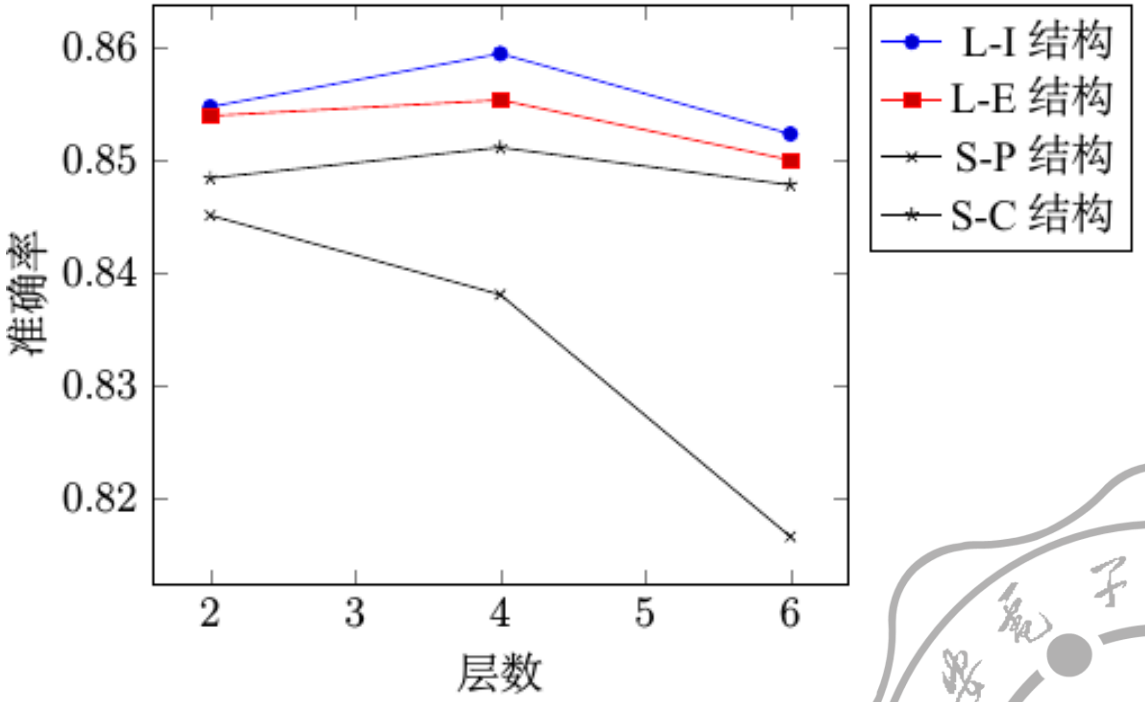
领域	样例	类别
Books	It is a very dry book and hard to stay interested in. I am barely able to stay awake while reading it. It does have some interesting things.	negative
Mag	The magazine was shipped in a timely manner, i would use this vendor again.	positive
Elec	Very pleased with the high capacity cartridge 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

实验结果

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

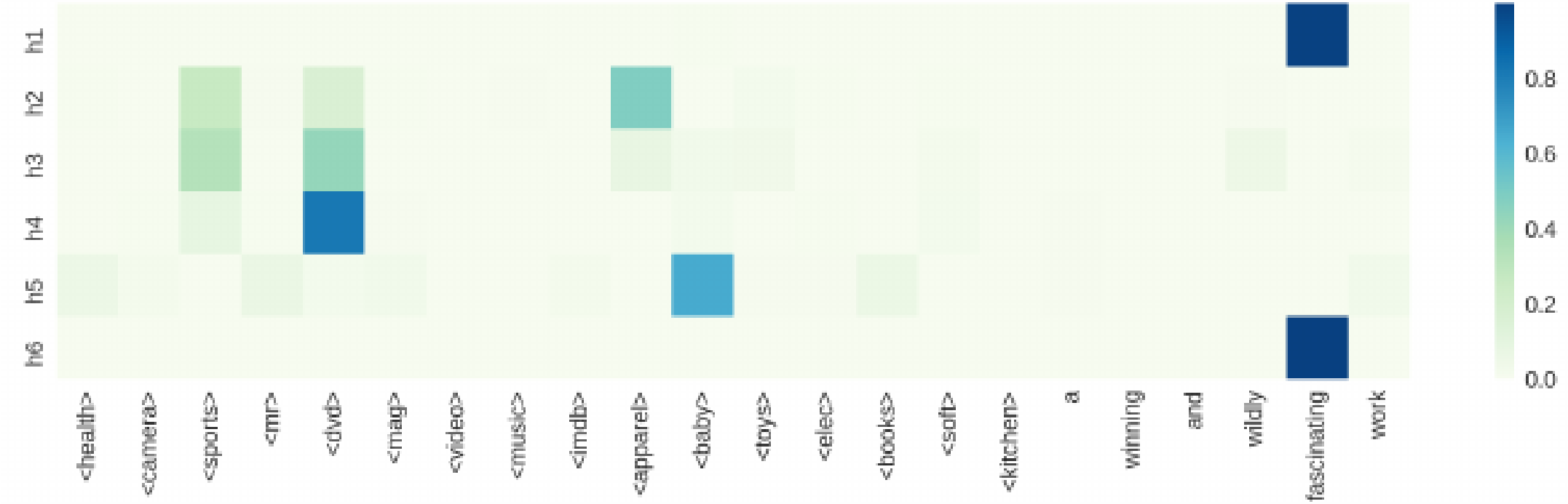
实验结果

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%



实验结果

- 注意力矩阵可视化



(b) 电影评论（MR）

目录 – 总结

- 研究意义
- 相关研究进展
- 模型与实验
- 总结
  - 优点
  - 不足

总结

- 优点
  - To my best knowledge，本文是首次较为系统地研究Transformer上的多任务共享架构
  - 提出了一种新型的逐层共享结构，同时具备硬共享模式参数量少的特点以及软共享模式的灵活性
- 不足
  - 目前只能处理句子级NLP任务
  - 未在跨度更大的任务上进行实验

谢 谢 ！