有道翻译 输入arXiv论文检索地址，免费翻译全文→ https://fanyi.youdao.com/trans/#/home

BERT是一个跨学科的知识学习者吗?
预训练模型的可迁移性的惊人发现

Wei-Tsung Kao and **Hung-yi Lee**
Graduate Institute of Communication Engineering, National Taiwan University
{r09942067, hungyilee}@ntu.edu.tw

Abstract

本文研究了在文本数据上预训练的模型(如BERT)的能力是否可以转移到一般的token序列分类应用中。为验证预训练模型的可迁移性，在具有标记错误匹配含义的文本分类任务和现实世界的非文本标记序列分类数据上测试了预训练模型，包括氨基酸、DNA和音乐。即使在非文本数据上，在文本上预训练的模型收敛速度更快，表现比随机初始化的模型更好，只比使用特定任务知识的模型略差。文本和非文本预训练模型的表示具有非平凡的相似性。

1 简介

在最近的NLP研究中，预训练的掩码语言模型(mlm)，如BERT (Devlin et al., 2019) 被从业人员广泛使用。在大型语料库(如维基百科)上进行预训练后，模型可以在文本分类和问答等NLP任务上进行快速微调，并在小型数据集(如RTE in GLUE)上具有良好的泛化能力(Wang et al., 2018)。为了在更专业的领域(如科学文章或临床数据)中应用和改进BERT，通过在特定领域的文本数据上进行预训练，提出了几个mlm (Lee et al., 2020; Beltagy et al., 2019)。MLM的概念也可以扩展到其他学科(可能是非语言)，只要输入是离散的。例如，Min et al. (2019) 在氨基酸序列数据上对称为PLUS的mlm进行预训练，并在几个蛋白质分类任务上实现最先进的性能。

本文研究了在大型文本语料库(如BERT)上预训练的模型是否可以有效地适应具有标记数量、标记分布、标签和与自然语言非常不同的结构的数据(目标数据甚至可以是非文本数据)。我们把这种能力称为学科适应性[1]。之前的工作(Papadimitriou and Jurafsky, 2020)只表明在非文本数据上预训练的语言模型(语言模型)可以适应人类语言的语言模型。这项工作首次检查了预训练的mlm是否可以学习在预训练期间从未见过的标签和数据之间的关系。我们的贡献如下。

- 本文提出设置，以检查预训练模型的学科适应性。BERT、BERT- chinese、ALBERT和RoBERTa可以更快地减少训练损失，在非文本数据上比随机初始化的模型泛化得更好，只比在每个学科中使用先验知识的模型略差。
- 分析表明，在微调之前，BERT与在非文本数据上预训练的类mlm模型之间的相似性远远高于与随机初始化模型之间的相似性，这有助于解释BERT在非文本学科中的成功。此外，对关于注意力相似性、非文本数据中的层次结构和训练稳定性的几个假设进行了广泛的调查，表明这些假设并不足以解释BERT的学科适应性。

我们相信，学科适应的发现将引发NLP社区的思考，在预训练过程中学到了什么。这些发现也可以对那些无法获得大规模数据集的学科有所帮助，这对从业人员至关重要。

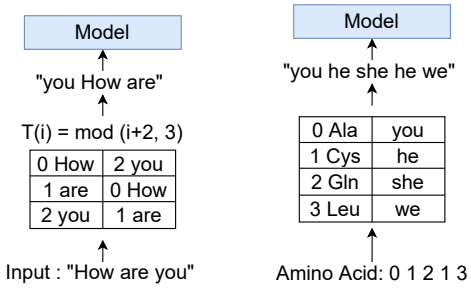
2 方法

为检查在文本语料库上预训练的模型的学科适应性，在两种类型的下游数据上对预训练模型进行微调。第一种类型(2.1节)是合成数据集，通过排列常见NLP数据集中的标记来生成。因此，每个token的含义都发生了变化。第二种类型的数据(2.2部分)是一种更具挑战性的情况，其中下游任务与人类语言无关。

2.1 合成数据

首先在合成文本序列分类数据上对模型进行测试。合成数据生成如下。给定一个文本序列分类数据集，我们定义一个确定的一对一映射T，它将文本序列中的子词标记x_i更改为另一个子词标记$T(x_i)$，如图1a所示。为了阐述合成文本数据的设计，将文本语料库中的标记视为图中的节点，标记之间的关系视为边。合

[1]我们使用术语学科自适应而不是领域自适应。在NLP中，域适应通常是指从通用文本到专业文本数据(如科学论文)的迁移。我们使用术语规程来强调分布和结构非常不同的数据。



(a) 文本数据的排列　　(b) 氨基酸序列

Figure 1: (a) token置换的例子(b)蛋白质分类的氨基酸序列输入。

成数据的图是原始图的同构。因此，合成数据的结构(图的结构)与原始数据的结构相同，并且合成数据集的任务与原始任务一样困难(如果不考虑预训练过程)。假设预训练模型仍然可以优于在这些人工数据上从头训练的模型。这表明预训练模型可以将知识迁移到语义与预训练语料完全不同的下游任务中。因此，在处理真实世界的非文本数据时，我们很可能进一步利用预训练模型。

实验中，首先在普通文本语料库上对模型进行预训练，然后在合成数据上对模型进行微调和测试。我们选择$T(i) = (i + 1000) \bmod D$，其中$D$是模型的词汇量。我们还尝试了随机生成映射。结果相似，见附录。举一个真实的例子，GLUE数据集中的句子"his healthy sense of satire is light and fun..."将被更改为"canadian franzme 1988pia leader watch sports czech at at"[2]。

2.2 真实世界的非文本数据

为了进一步验证预训练模型的*discipline adaptability*，在真实世界的非文本数据上对预训练模型进行微调。在这些下游任务中，token的分布和token的数量都可能与预训练的文本数据有很大不同。因此，这是一个更难评估预训练模型可迁移性的设置。

为了通过BERT处理非文本数据，我们将非文本数据的每个标记映射到一个子词标记，如图1b所示。在接下来的实验中，(确定性)映射表是随机生成的，因为我们发现，只要我们不将非文本标记映射到预训练模型的未使用标记，不同的映射会导致相似的结果。在微调阶段，在预训练模型的顶部添加一个随机初始化的线性分类器，而不随机初始化任何预训练参数，包括嵌入层。然后我们微调整个模型。

[2]标记"."映射到示例中的标记"at"，因此在合成句子中有三个连续的"at"。

3 实验

3.1 设置

我们使用GLUE数据集来生成合成数据。验证集用于测试模型。WNLI被排除在外，如(Devlin et al., 2019);对于现实世界中的非文本数据，我们包括以下任务，它们使用不同的标记数量、标记分布和结构:

蛋白质分类(3个任务):定位(Loc.) (Almagro Armenteros et al., 2017)，稳定性(Stab.) (Rocklin et al., 2017)，荧光(Flu.) (Sarkisyan et al., 2016)在Min et al. (2019)中使用。输入是由20个不同标记组成的氨基酸序列。

DNA分类(4项任务):*H3*，*H4*，*H3K9ac*来自Pokholok et al. (2005), *Splice*来自Asuncion and Newman (2007)用于Yin et al. (2018)。输入是由4个不同标记组成的DNA子序列。

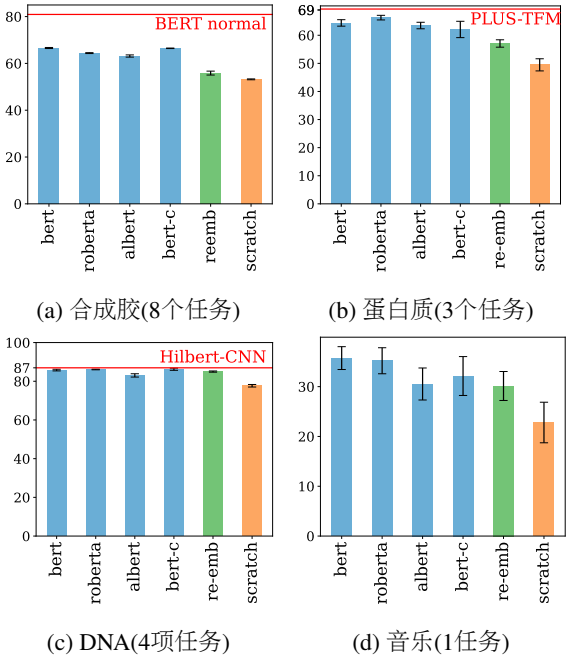
作曲家分类(1个任务):我们使用*MAESTRO-v1*数据集(Hawthorne et al., 2019)。输入是包含128个不同标记的基音序列。

实验中使用的预训练模型包括BERT-base-uncased、BERT-base-Chinese、ALBERT-base-v1和RoBERTa-base。随机初始化(从头开始训练)的模型与BERT-base具有相同的架构。由于篇幅限制，BERT-large的实验留在附录中。模型由广泛用于预训练模型的默认分布初始化(例如，BERT-base的$\mathcal{N}(0, 4 \times 10^{-4})$)。详细的超参数请参见附录。为简单起见，我们使用"预训练模型"来指上述在自然语言上预训练的模型(如果没有指定)。

3.2 结果

图2显示了每个学科中预训练模型(蓝色条)和从头训练模型(橙色条)的平均分数。图2a的均值(条形)和标准差(黑色误差条)是通过三个随机种子计算的，图2b、2c和2d中的平均值是通过六次独立运行(具有不同的token映射)计算的。BERT在普通GLUE上微调的GLUE得分作为学科特定的顶线(红线); 对于蛋白质分类，学科特定模型是PLUS-TFM (Min et al., 2019)，这是一个在蛋白质序列上预训练的12层transformer MLM; 对于DNA分类，学科特定的模型是Hilbert-CNN (Yin et al., 2018); 对于作曲家分类，我们使用数据集中的所有类别进行分类。但是之前的工作(Kim et al., 2020; Spijker, 2020)只使用了部分类，因此没有特定于学科的模型可用。每个学科中每个任务的详细分数留在附录中。

预训练模型在所有学科中都优于从头训练的模型。这种现象是普遍过度预训练模型具有不同的模型结构(ALBERT)、预训练目标、预训



(a) 合成胶(8个任务) (b) 蛋白质(3个任务)

(c) DNA(4项任务) (d) 音乐(1任务)

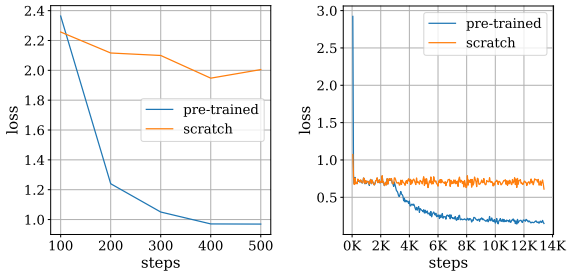
Figure 2: 预训练模型和每个学科中从头训练模型的平均分数(y轴)。分数越高，模型越好。黑色的误差条代表随机种子的标准差。红线表示特定学科模型的性能。“-c”:中文。

练数据量(RoBERTa)和不同的自然语言(BERT-Chinese)。此外，在不使用任何学科特定知识的情况下，预训练模型的表现仅略低于PLUS-TFM和Hilbert-CNN。大多数模型和学科的标准偏差都很小，这意味着不同的token映射的影响是边际的。

乍一看，微调合成GLUE上的预训练模型似乎等同于随机初始化单词嵌入层，然后微调普通GLUE上的预训练模型，我们称之为重新嵌入(re-emb)。如果等价性为真，则预训练模型性能提高的解释只是中间层已经训练好。然而，图2a显示这种等价性并不成立。Re-emb(绿色条)会降低性能。对于非文本数据，re-emb的性能也比图2b、2c和2d中所有预训练参数的模型更差。因此，预训练的词嵌入层有利于非文本下游任务，即使标记的含义与预训练不同。使用预训练模型的未使用标记甚至会使性能退化到从头训练的基线。

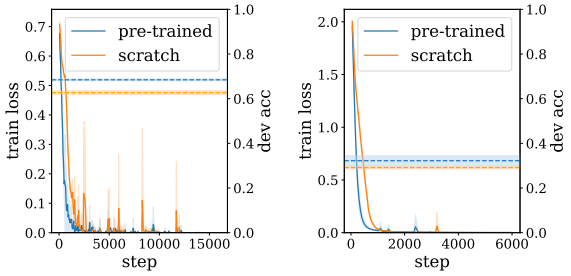
4 讨论

3.2部分的结果验证了预训练模型作为强大的跨学科知识学习者的潜力。预训练模型的成功可能源于更好的泛化能力或更好的训练损失动态。在本节中，我们分析了预训练在优化和泛化方面的贡献。此外，我们尝试通过比较BERT和PLUS在蛋白质数据上的表示来解释预训练模型的成功。



(a) 合成胶(STS-B) (b) 荧光

Figure 3: BERT的训练损失(蓝色线)和从头训练的模型(橙色线)。



(a) H3K9ac (b) 定位

Figure 4: 预训练BERT(蓝色)的训练曲线(实线)和训练结束时的验证结果(虚线)以及从头训练的模型(橙色)。实线表示平均值，阴影区域表示6次独立运行的标准差。

4.1 训练速度

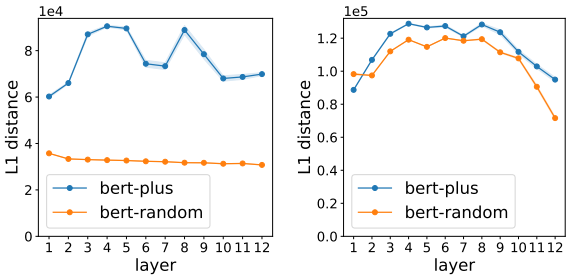
图3显示BERT总是比从头开始训练的模型更快地减少训练损失。对于图3b中的荧光任务，从头训练的模型似乎很快陷入局部最小值，而BERT随着微调的进行而脱离局部最小值。对于图3a中的*STS-B*这样的小数据集，BERT只需数百步就可以减少训练损失，但从头开始训练的模型的训练损失仍然很高。其他任务的结果也类似，请参见附录。

4.2 泛化能力

进一步研究了预训练模型的泛化能力。我们仅在1%的非文本数据上训练所有模型。这样，预训练模型和从头开始训练的模型都可以收敛到几乎零损失。比较它们的验证性能，以了解它们的泛化能力。图4显示了一个DNA数据集和一个蛋白质数据集的结果。其他任务的结果也类似，请参见附录。在1%的训练数据设置下，预训练模型和从零开始训练的模型的训练损失都收敛到零。并且预训练模型仍然超过了在验证集上从头开始训练的模型。因此，模型预训练提高了模型在学科自适应方面的泛化能力。

	Flu.	Stab.	Loc.
BERT - PLUS	**0.729**	**0.634**	**0.504**
BERT - random	0.598	0.545	0.362
PLUS - random	0.461	0.405	0.322
random - random	0.434	0.388	0.387

Table 1: PWCCA蛋白质数据上模型的最后一层表示之间的相似性([−1, 1]中的值)。并非所有的模型都进行了微调。“随机”是指随机初始化具有相同BERT架构的模型。



(a) 无微调 (b) 定位微调

Figure 5: 应用匹配算法后不同模型的注意力图的平均L_1距离。

4.3 表示相似性

为解释文本预训练模型在非文本数据上的成功，在BERT和PLUS-TFM的表示上应用投影加权典型相关分析(PWCCA) (Morcos et al., 2018)。表1中的结果表明，在微调前，BERT与PLUS之间的相似度远高于随机初始化模型与BERT之间的相似度。尽管BERT只在自然语言上进行了预训练，但在处理非文本数据时，BERT的行为与随机初始化的模型不同，这可能是BERT具有学科适应性的原因之一。

4.4 假设

为了阐述预训练传销的学科适应性背后的原因，我们尝试探索了几种可能性。然而，它们不足以解释这种现象。在接下来的小节中，我们将总结这些实验。一些详细的结果在附录中。

4.4.1 注意力相似度

我们检查了BERT、PLUS和随机初始化模型之间的注意图的相似性。对于一个输入数据，我们首先在每一层中提取它们的注意力图。对于同一层的注意力图，我们使用匈牙利算法在来自不同模型的注意力图之间找到最小L1距离匹配。匹配的平均距离代表了每层中注意力模式的相似度。蛋白质数据的结果如图5所示。无论是否微调，几乎在所有层中，BERT与PLUS之间的距离都大

	Flu.	Stab.	Loc.	Avg.
scratch	29.4	59.6	56.6	48.5
uniform	36.6	53.7	57.7	49.3
flat	36.8	56.3	57.8	50.3
nesting	47.9	62.7	60.5	57.0
Kannada	47	71.3	62.8	**60.4**

Table 2: 在人工数据集和人类语言上预训练模型的蛋白质分类结果。

于BERT与random之间的距离。从这个角度来看，我们可能不认为PLUS和BERT有共同的注意力模式。

4.4.2 预训练数据的属性

本文想研究预训练数据的哪些属性会导致学科适应性。我们使用(Chiang and Lee, 2020)中使用的以下数据对mlm进行预训练:

- 均匀:句子中的标记从所有标记的均匀分布中进行*i.i.d*采样。
- 扁平或嵌套括号:句子中的标记是随机、递归生成的，同时是分层匹配的。
- 坎纳达语(Ortiz Suárez et al., 2020):坎纳达语是印度西南部人们使用的语言。

然而，如表2所示，在人工数据上预训练的模型在蛋白质分类上的表现比在自然语言上预训练的模型差。因此，自然语言可能确实与蛋白质有相似之处，而仅仅是层次结构可能不足以解释学科适应性。

4.4.3 梯度稳定性

我们还检查BERT是否满足以下关于训练稳定性的标准:

- Saxe et al. (2014) 如果模型初始化的输出-输入雅可比矩阵的奇异值都等于1(称为动态等距)，则模型可以避免梯度消失或梯度爆炸，训练效果更好。
- Sankararaman et al. (2020) 表明不同数据产生的负相关梯度会减慢收敛速度。
- Liu et al. (2020) 注意，在参数摄动下，transformer输出的大方差会使训练过程不稳定。

在合成的胶水上，BERT不能更好地满足这些条件，甚至比高斯初始化更差。尽管BERT在非文本数据上的优化效果更好，但上述理论未能阐述BERT的优化特性。详细结果见附录。

5 结论

本文研究了BERT作为跨学科知识学习者的潜力。通过在token含义发生变化的合成文本数据和非文本数据上对BERT进行微调，验证了BERT可以有效地适应不同学科的数据，并具有良好的泛化能力。发现在经过PWCCA微调之前，在文本和蛋白质上预训练的模型之间存在着一定的相似性，这有助于解释BERT具有学科适应性背后的原因。希望所提出的设置可以作为研究人员的新的分析工具，并为预训练模型的力量提供新的见解。

更广泛的影响

当大规模预训练数据集不可用时，本文的结果对其他学科的从业人员有帮助。预训练模型的学科适应性也有助于降低计算成本，因为我们可能不需要为每个学科预训练一个模型。我们认为本文的结果不会引起任何伦理问题。

致谢

感谢国家应用研究国家高性能计算中心(NCHC) 提供计算及储存资源之台湾实验室。

References

José Juan Almagro Armenteros, Casper Kaae Sønderby, Søren Kaae Sønderby, Henrik Nielsen, and Ole Winther. 2017. Deeploc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395.

Arthur Asuncion and David Newman. 2007. Uci machine learning repository.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2020. Pre-training a language model without human language. *arXiv preprint arXiv:2012.11995*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. 2019. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *International Conference on Learning Representations*.

Sunghyeon Kim, Hyeyoon Lee, Sunjong Park, Jinho Lee, and Keunwoo Choi. 2020. Deep composer classification using symbolic representation. *arXiv preprint arXiv:2010.00823*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. 2020. Understanding the difficulty of training transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5747–5763, Online. Association for Computational Linguistics.

Seonwoo Min, Seunghyun Park, Siwon Kim, Hyun-Soo Choi, and Sungroh Yoon. 2019. Pre-training of deep bidirectional protein sequence representations with structural information. *arXiv preprint arXiv:1912.05625*.

Ari Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Isabel Papadimitriou and Dan Jurafsky. 2020. Learning Music Helps You Read: Using transfer to study linguistic structure in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6829–6839, Online. Association for Computational Linguistics.

Dmitry K Pokholok, Christopher T Harbison, Stuart Levine, Megan Cole, Nancy M Hannett, Tong Ihn Lee, George W Bell, Kimberly Walker, P Alex Rolfe, Elizabeth Herbolsheimer, et al. 2005. Genome-wide map of nucleosome acetylation and methylation in yeast. *Cell*, 122(4):517–527.

Gabriel J Rocklin, Tamuka M Chidyausiku, Inna Goreshnik, Alex Ford, Scott Houliston, Alexander Lemak, Lauren Carter, Rashmi Ravichandran, Vikram K Mulligan, Aaron Chevalier, et al. 2017. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347):168–175.

Karthik Abinav Sankararaman, Soham De, Zheng Xu, W Ronny Huang, and Tom Goldstein. 2020. The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. In *International Conference on Machine Learning*, pages 8469–8479. PMLR.

Karen S Sarkisyan, Dmitry A Bolotin, Margarita V Meer, Dinara R Usmanova, Alexander S Mishin, George V Sharonov, Dmitry N Ivankov, Nina G Bozhanova, Mikhail S Baranov, Onuralp Soylemez, et al. 2016. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401.

Andrew M Saxe, James L Mcclelland, and Surya Ganguli. 2014. Exact solutions to the nonlinear dynamics of learning in deep linear neural network. In *In International Conference on Learning Representations*. Citeseer.

BC Spijker. 2020. Classifying classical piano music into time period using machine learning. Master's thesis, University of Twente.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Bojian Yin, Marleen Balvert, Davide Zambrano, Alexander Schoenhuth, and Sander Bohte. 2018. An image representation based convolutional network for dna classification. In *International Conference on Learning Representations*.

A 实验超参数

实验中使用的transformer模型为12层、768-hidden、12-attention头模型。参数总数为110M，大小与BERT-base相同。对于BERT-large-uncased (bert-l)和附录中从头训练的large model (scratch-l)，参数总数为340M。PLUS-TFM具有与BERT-base相同的结构，总参数数为110M。对于Hilbert-CNN模型，根据原文，总参数数为961K。

在本文的所有实验中，我们使用Adam优化器，学习率设置为10^{-5}。通过对GLUE数据集的MRPC进行网格搜索来选择优化器，通过对MRPC和荧光蛋白分类任务的验证集进行网格搜索来选择学习率。我们从10^{-4}到10^{-7}搜索学习率。我们在这个范围内均匀采样5个点，并进一步采样10^{-5}和10^{-6}之间的5个点。搜索了包括Adagrad、Adam、Adamax、RAdam和NovoGrad在内的优化器，进行了三次独立运行。选择使随机初始化的12层transformer模型在MRPC训练集上取得最高F1分数和在荧光验证集上取得最高斯皮尔曼相关性的参数(这也是re-emb设置的最佳参数)。我们没有使用梯度裁剪和预热，因此学习率调度与线性学习率衰减相同。所有模型都是在两个RTX 2080-Ti (GLUE数据集)或一个Tesla V100 GPU(蛋白质分类、DNA分类和音乐分类)上以批量大小为32进行训练的。对于GLUE数据集，使用GLUE的验证集作为测试集，并评估最终的检查点。对于所有非文本数据集，在验证集上选择最佳检查点，并在测试集上进行评估。

B 合成胶的全部结果

在合成GLUE数据集上的完整结果如表3所示。预训练模型(包括大型模型)的性能优于从头训练的模型，SST-2和CoLA除外。对于SST-2，预训练模型的泛化能力比从头开始训练的模型差。对于CoLA来说，所有模型都无法训练。但对于其他6个任务，预训练模型的表现优于从头训练的模型。除RTE数据集和大型模型外，大多数模型的标准差都小于2。对于RTE，最大标准差为5.24 (ALBERT)。对于大型模型，标准差要大得多，如表4所示。当我们随机生成token映射时，结果是相似的。这表明不同标记映射的效果是边际的。

对于随机初始化然后微调词嵌入层的BERT (re-emb)，其性能要低于使用整个预训练权重的BERT，这表明即使预训练的词嵌入也是必要的。

C 在非文本数据上的测试和验证性能

表5和7显示了每个非文本分类任务的完整测试和验证结果。表6和8显示各学科的平均成绩。对于大多数任务，文本预训练模型在测试和验证集上的表现都优于从头开始训练的模型和重新emb模型。

D 其他任务的训练损失

图6和7分别显示了BERT和在其他GLUE任务和其他非文本数据集上从头训练的模型的训练损失。BERT可以比从头开始训练的模型更快地减少训练损失，除了SST-2任务，BERT在该任务上的表现更差。结果在学科上是一致的。

E 其他任务的泛化实验

图8和9显示了BERT的结果，以及仅使用1%的GLUE数据集和非文本数据集的训练数据从头训练的模型。由于训练集大小的限制，我们没有在splice和maestro-v1数据集上进行实验。对于大多数任务，BERT比从头开始训练的模型具有更好的泛化能力。

	MNLI m/mm-acc	QQP F1	QNLI acc	SST-2 acc	CoLA mcc	STS-B spr	MRPC F1	RTE acc	avg
Normal data									
BERT	84.0/84.3	87.4	91.2	92.2	55.0	86.7	85.5	62.1	80.9
re-emb	66.1/66.8	78.7	64.5	80.0	0.0	19.0	78.6	49.0	55.9
permutation									
bert	68.6/69.9	81.2	80.0	79.6	0.0	77.8	82.9	60.2	66.6
roberta	66.4/67.5	77.8	79.2	76.2	0.0	73.9	83.0	56.0	64.4
albert	65.9/67.5	79.5	67.5	71.3	0.0	71.6	81.4	53.2	63.2
bert-c	68.9/70.0	81.7	80.2	77.5	0.0	76.1	85.3	58.5	66.4
scratch	61.4/62.1	69.3	61.1	81.0	0.0	8.3	81.3	54.2	53.2
bert-l	44.5/44.7	26.8	60.4	60.7	0.0	73.5	82.3	54.3	49.7
scratch-l	40.6/40.9	21.7	50.2	80.9	0.0	9.2	81.2	50.5	41.7
random mapping									
BERT	68.2/68.6	80.6	79.7	78.7	0.0	75.6	83.4	58.5	65.9
scratch	61.5/62.0	69.0	61.5	79.6	0.0	8.3	81.3	51.0	52.7

Table 3: GLUE验证集的完整结果(在三个随机种子上取平均值)。评估指标列在任务名称下面。正常数据意味着模型在正常胶水上进行了微调。排列意味着模型在合成胶水上进行了微调。随机映射意味着token映射是随机生成的。"avg":平均得分(GLUE得分)。m/mm: MNLI匹配/不匹配的集合。spr:斯皮尔曼相关。“mcc”:马修斯相关系数。“re-emb”:随机初始化BERT的词嵌入层并微调BERT。

	MNLI acc	QQP F1	QNLI acc	SST-2 acc	CoLA mcc	STS-B spr	MRPC F1	RTE acc
permutation								
BERT-l	21.2/21.3	46.5	17.0	16.9	0.0	3.8	0.9	6.7
scratch-l	14.4/14.7	37.6	0.6	0.7	0.0	0.4	0.0	3.1

Table 4: GLUE验证集上大型模型的标准偏差。

F 第4.4节的详细结果

F.1 动力等距

图10显示了bert基、BERT-large和albert基的输出-输入雅可比矩阵的奇异值的分布。雅可比矩阵是通过计算上一层表示相对于输入词嵌入的导数来计算的。输入数据来自普通的GLUE数据集。与随机初始化相比(图10中的scratch)，BERT和ALBERT的奇异值集中在0而不是1，这与动态等距假设相反。因此，很难说BERT和ALBERT的能量来源于动态等距。

F.2 梯度混淆

图11显示了合成GLUE数据集中不同数据点产生的梯度的余弦相似性。尽管BERT的余弦相似度比随机初始化(scratch)的相似度大，但ALBERT显示出了不利的趋势。预训练ALBERT的余弦相似度小于划痕对应的余弦相似度。但预训练ALBERT的性能仍然优于随机初始化，这表明避免梯度混淆可能不是预训练mlm的纪律适应性的关键。

F.3 扰动下的输出方差

我们向模型参数注入零均值高斯噪声，以计算模型输出在噪声下的变化。变差由添加噪声前后模型输出的L2距离表示。我们选择标准差的大小为$10^{-2}, 10^{-4}, 10^{-6}$，和$10^{-8}$。图12和13分别显示了BERT-base和ALBERT-base在三个合成胶水任务上的结果。BERT和ALBERT表现出相反的趋势:BERT的变化量小于随机初始化的对应量，而ALBERT的变化量大于随机初始化的对应量。因此，这个假设不足以解释预训练模型的学科适应性。

	Protein			DNA			
	localization	stability	fluorescence	H3	H4	H3K9ac	Splice
specific	69.0	76.0	63.0	87.3	87.3	79.1	94.1
bert	64.1 (0.4)	70.6 (3.9)	58.4 (3.2)	83.6 (1.3)	85.9 (0.9)	77.2 (0.9)	96.4 (0.8)
roberta	65.4 (1.0)	73.6 (1.7)	59.8 (1.6)	84.2 (0.7)	86.5 (0.4)	78.9 (0.6)	94.7 (0.3)
albert	65.1 (0.6)	70.3 (2.8)	55.0 (2.4)	83.5 (0.3)	86.4 (0.5)	78.2 (0.6)	84.0 (3.6)
bert-c	63.1 (0.8)	69.7 (3.0)	53.4 (8.4)	84.0 (1.3)	86.2 (0.9)	77.9 (0.4)	96.8 (0.8)
re-emb	62.8 (0.5)	71.0 (2.3)	37.4 (4.4)	83.0 (1.2)	83.9 (0.8)	77.5 (0.4)	95.6 (0.4)
scratch	57.8 (0.6)	62.2 (5.3)	28.4 (1.6)	76.1 (0.7)	66.6 (1.3)	72.6 (0.3)	95.3 (1.2)
bert-l	63.0 (0.4)	23.2 (41.8)	44.6 (18.6)	70.5 (14.8)	63.5 (7.3)	65.0 (9.4)	82.6 (18.7)
scratch-l	58.3 (0.5)	59.7 (4.3)	11.6 (3.5)	76.7 (0.5)	58.7 (2.5)	67.1 (7.9)	95.6 (0.8)

Table 5: 蛋白质分类和DNA分类测试结果。该指标为斯皮尔曼相关荧光和稳定性。对于所有其他任务，度量标准是准确性。括号中的数字是标准差(使用不同的标记映射进行6次独立运行计算得到)。“特定”:特定于学科的模型。

	Protein	DNA	Music
bert	64.4 (1.2)	85.8 (0.4)	35.7 (2.3)
roberta	66.3 (0.8)	86.1 (0.2)	35.2 (2.6)
albert	63.5 (1.2)	83.0 (0.9)	30.5 (3.2)
bert-c	62.1 (2.9)	86.2 (0.5)	32.1 (3.9)
re-emb	57.1 (1.3)	85.0 (0.3)	30.1 (2.9)
scratch	49.5 (2.2)	77.7 (0.7)	22.8 (4.1)
bert-l	43.6 (14.7)	70.4 (6.7)	30.8 (4.0)
scratch-l	43.2 (2.6)	74.5 (1.8)	26.0 (5.0)

Table 6: 作曲家分类测试结果、DNA分类测试平均得分、蛋白质分类测试平均得分。括号中的数字是对不同的标记映射进行6次独立运行后计算出的标准差。

G 数据集统计

G.1 胶水

GLUE是一个英文数据集，包含几个任务。表9显示了胶水的统计数据。在实验中，我们使用验证集作为测试集。训练/验证分割可以在下载的数据中找到。

G.2 蛋白质分类

表10显示了蛋白质分类数据集的统计情况。对于预处理，我们将输入序列的长度截断为512。

G.3 DNA分类

表11显示了DNA分类数据集的统计情况。对于训练/验证/测试划分，我们像Hilbert-CNN那样随机选择90%的样本作为训练数据，5%的样本作为验证数据，5%的样本作为测试数据。我们没有对这些数据集进行任何额外的预处理。

G.4 作曲家分类

表12显示了MAESTRO-v1数据集的统计信息。训练集/验证集/测试集可以在下载的文件中找到。我们读取midi数据并将其转换为基音序列。对于大于128的序列，我们将其划分为多个长度为128的片段。对于训练数据，每段是一个训练示例。为了验证和测试，对所有片段进行推理，并通过投票决定最终输出。

	Protein			DNA			
	localization	stability	fluorescence	H3	H4	H3K9ac	Splice
bert	69.6 (0.9)	70.6 (0.8)	57.0 (4.3)	83.5 (0.7)	87.0 (0.6)	78.4 (0.7)	95.9 (0.7)
roberta	71.1 (1.1)	68.8 (0.9)	59.2 (2.7)	86.7 (0.6)	87.6 (0.2)	79.5 (0.6)	95.2 (0.5)
albert	69.0 (0.8)	66.3 (1.9)	50.9 (5.3)	85.6 (0.9)	87.0 (0.3)	79.3 (0.6)	82.5 (3.1)
bert-c	69.0 (0.7)	75.4 (0.5)	50.8 (9.3)	83.6 (1.3)	87.5 (0.7)	79.2 (0.8)	96.5 (0.5)
re-emb	67.9 (0.4)	67.3 (1.7)	36.9 (3.2)	82.8 (0.5)	85.3 (0.7)	78.4 (0.8)	95.4 (0.9)
scratch	59.9 (0.8)	63.6 (0.6)	29.4 (1.3)	75.4 (0.2)	66.5 (0.8)	71.9 (0.2)	96.0 (0.3)
bert-l	69.3 (1.0)	37.5 (27.5)	42.5 (18.4)	70.1 (14.6)	64.0 (8.1)	64.1 (9.4)	82.2 (19.6)
scratch-l	60.9 (0.7)	61.6 (2.1)	13.0 (2.9)	75.7 (0.3)	58.6 (2.1)	66.5 (7.9)	96.0 (0.6)

Table 7: 蛋白质分类和DNA分类验证结果。该指标为斯皮尔曼相关荧光和稳定性。对于所有其他任务，度量标准是准确性。括号中的数字是对不同的标记映射进行6次独立运行后计算出的标准差。

	Protein	DNA	Music
bert	65.7 (1.6)	86.2 (0.4)	43.2 (4.2)
roberta	66.4 (0.8)	87.2 (0.2)	41.1 (3.6)
albert	62.1 (1.9)	83.6 (0.8)	36.3 (3.3)
bert-c	65.1 (3.3)	86.7 (0.6)	42.2 (3.4)
re-emb	57.4 (1.4)	85.5 (0.4)	39.5 (2.8)
scratch	51.0 (0.6)	77.5 (0.2)	31.0 (2.3)
bert-l	49.8 (9.0)	70.1 (7.1)	43.3 (3.5)
scratch-l	45.2 (0.9)	74.2 (1.7)	34.3 (2.6)

Table 8: 作曲家分类、DNA分类平均得分、蛋白质分类平均得分的验证结果。括号中的数字是对不同的标记映射进行6次独立运行后计算出的标准差。

dataset	train	validation
CoLA	8551	1043
SST-2	67349	872
MRPC	3668	408
QQP	363849	40430
STS-B	5749	1500
MNLI	392702	9815/9832
QNLI	104743	5463
RTE	2490	277

Table 9: 训练/验证GLUE数据集的例子。MNLI验证集的个数分别为匹配子集和不匹配子集。数据可在以下网站下载https://gluebenchmark.com

dataset	train	validation	test
fluorescence	21446	5362	27217
stability	53614	2512	12851
localization	9977	1108	2773

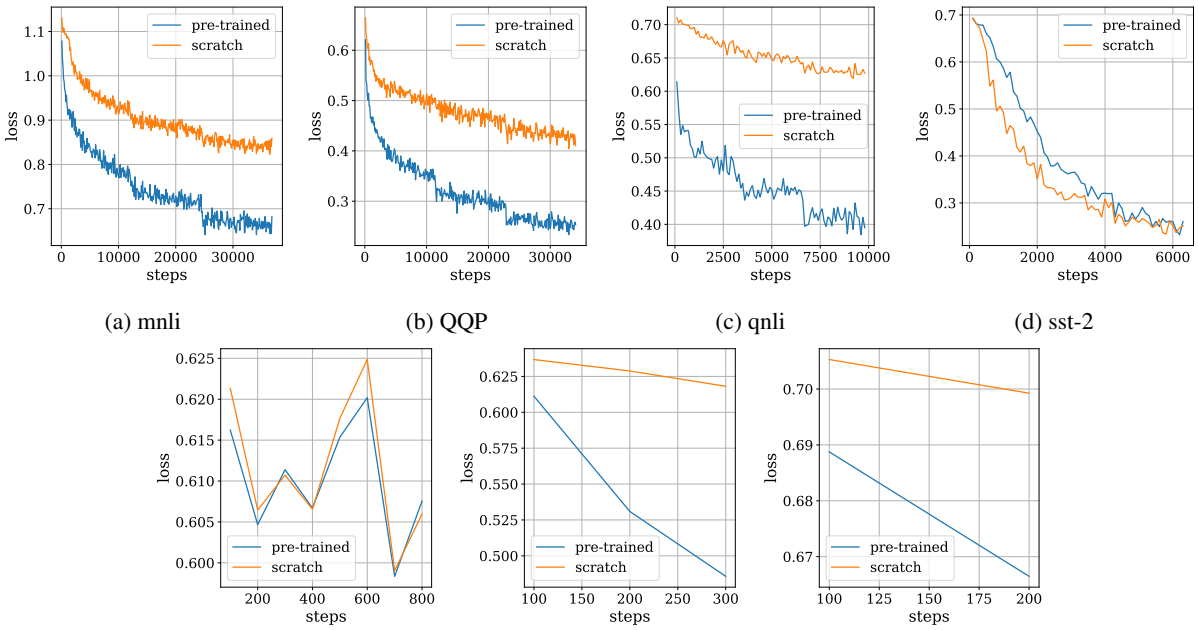
Table 10: 训练/验证/测试蛋白质分类数据集的示例。数据可从http://ailab.snu.ac.kr/PLUS/下载。训练/验证/测试可以在下载的文件中找到。

dataset	#samples
H3	14965
H4	14601
H3K9ac	27782
Splice	3190

Table 11: DNA分类数据集的样本数量。数据可在以下网站下载https://github.com/Doulrs/Hilbert-CNN

dataset	train	validation	test
MAESTRO-v1	954	105	125

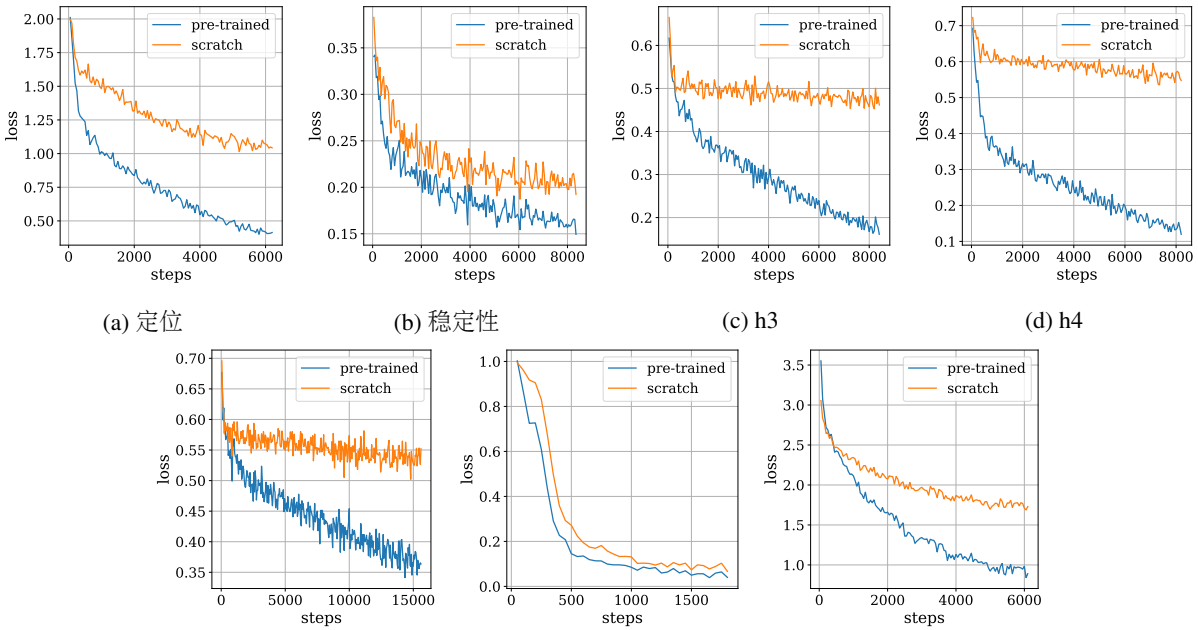
Table 12: 训练/验证MAESTRO-v1数据集的示例。数据可在以下网站下载https://magenta.tensorflow.org/datasets/maestro



(a) mnli (b) QQP (c) qnli (d) sst-2

(e) 可乐 (f) MRPC (g) rte

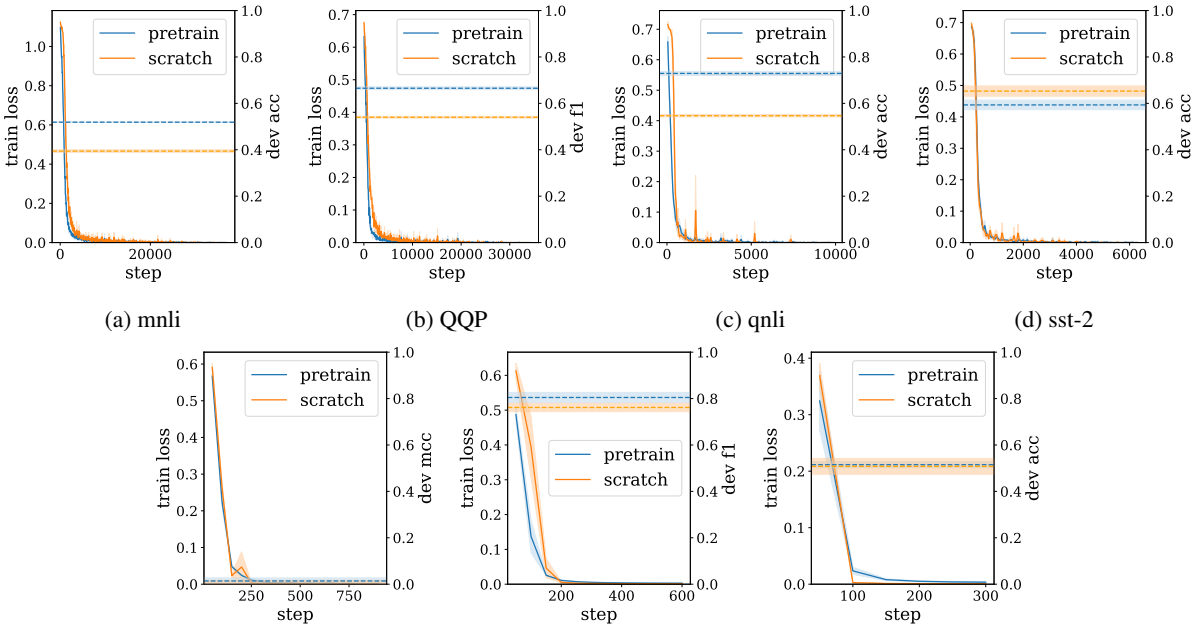
Figure 6: BERT的训练损失(蓝色线)和在其他GLUE任务上从头训练的模型(橙色线)。



(a) 定位 (b) 稳定性 (c) h3 (d) h4

(e) H3K9ac (f) 拼接 (g) maestro-v1

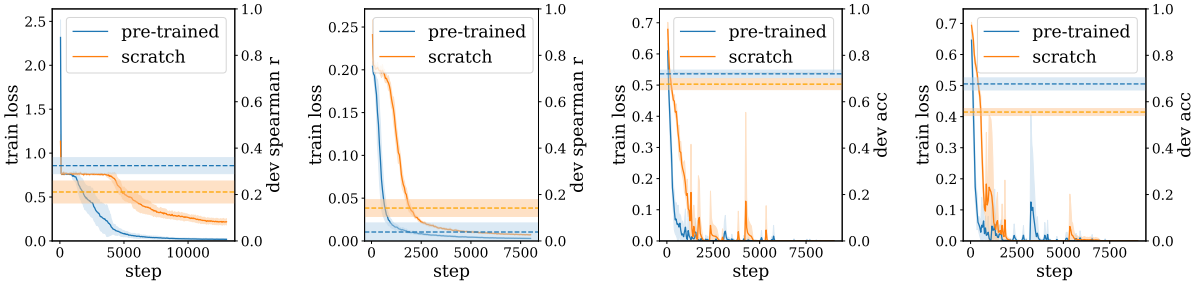
Figure 7: BERT的训练损失(蓝色线)和在其他非文本数据集上从头训练的模型(橙色线)。



(a) mnli (b) QQP (c) qnli (d) sst-2

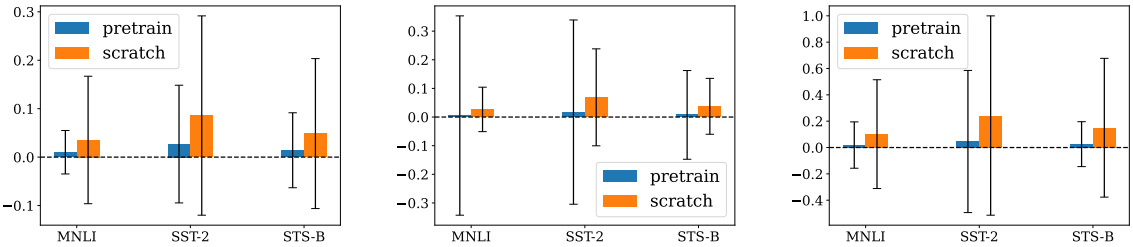
(e) 可乐 (f) MRPC (g) rte

Figure 8: BERT(蓝色线)的训练损失(实线)和验证性能(虚线)，以及在其他GLUE任务上从头训练的模型(橙色线)，只使用1%的训练数据。最后一个检查点用于执行验证。



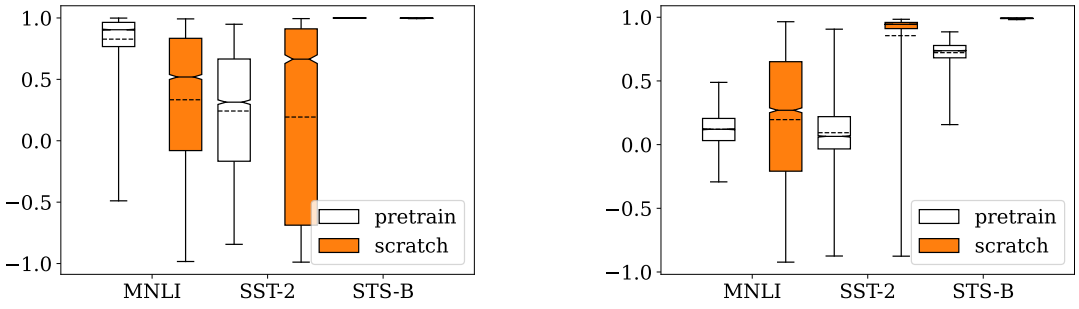
(a) 荧光 (b) 稳定性 (c) h3 (d) h4

Figure 9: BERT(蓝线)的训练损失(实线)和验证性能(虚线)，以及仅使用1%的训练数据在其他非文本数据集上从头训练的模型(橙色线)。由于1%的训练集太小，我们没有在splice和maestro-v1上进行实验。



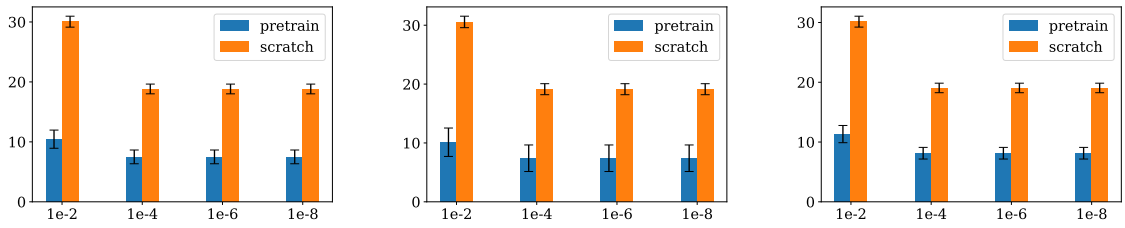
(a) 伯特基 (b) 大伯特 (c) 阿尔伯特基

Figure 10: BERT和ALBERT的输出-输入雅可比矩阵的奇异值分布。在GLUE数据集上计算奇异值。条形代表均值，误差条形代表标准差。



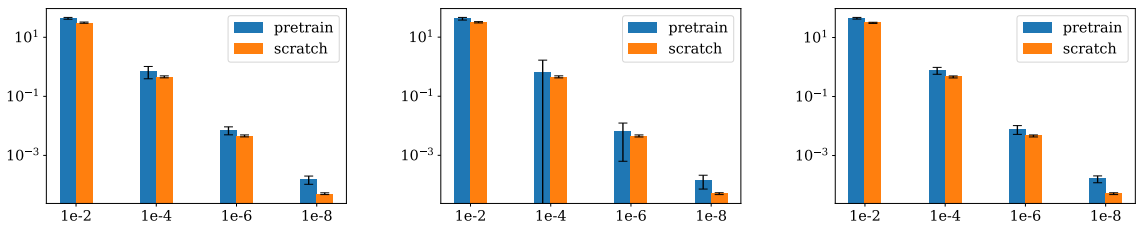
(a) 伯特基　　(b) 阿尔伯特基

Figure 11: 在合成GLUE数据集上(a) BERT和(b) ALBERT的梯度余弦相似度。凹槽表示中位数，没有凹槽的虚线表示平均值。



(a) mnli　　(b) sst-2　　(c) sts-b

Figure 12: BERT输出之间L2距离的均值(条形图)和标准差(误差条形图)，在模型参数中添加噪声和不添加噪声的情况下。“scratch”表示随机初始化的参数。



(a) mnli　　(b) sst-2　　(c) sts-b

Figure 13: 在模型参数添加噪声和不添加噪声的情况下，ALBERT输出之间L2距离的均值(条形图)和标准差(误差条形图)。“scratch”表示随机初始化的参数。