Why Can GPT Learn In-Context?
Language Models Implicitly Perform Gradient Descent as Meta-Optimizers

Damai Dai[†]*, **Yutao Sun**[‖]*, **Li Dong**[‡], **Yaru Hao**[‡], **Shuming Ma**[‡], **Zhifang Sui**[†], **Furu Wei**[‡]

[†] MOE Key Lab of Computational Linguistics, Peking University

[‖] Tsinghua University [‡] Microsoft Research

{daidamai,szf}@pku.edu.cn

{lidong1,fuwei}@microsoft.com

Abstract

大型预训练语言模型展示了出人意料的上下文学习能力（ICL）。凭借少量示范输入-标签对，它们可以在没有参数更新的情况下预测未见输入的标签。尽管在性能上取得了巨大的成功，它的工作机制仍然是一个悬而未决的问题。在本文中，我们将语言模型解释为元优化器，并将上下文学习理解为隐式微调。从理论上我们弄清楚了Transformer注意力有一个双重的梯度下降形式。在此基础上，我们理解ICL如下：GPT首先根据示范样例产生元梯度，然后这些元梯度被应用于原始GPT以构建一个ICL模型。我们通过全面比较ICL与显式微调在真实任务中的行为，提供了支持我们理解的经验性证据。实验结果表明，从多个角度来看，上下文学习的行为与显式微调类似。受到Transformer注意力和梯度下降之间的双重形式的启发，我们设计了一种基于动量的注意力，类比于带动量的梯度下降。香草注意力表现的改进进一步支持了我们的另一方面理解，并更重要的是展示了利用我们理解进行未来模型设计的潜力。

代码可在https://aka.ms/icl获取。

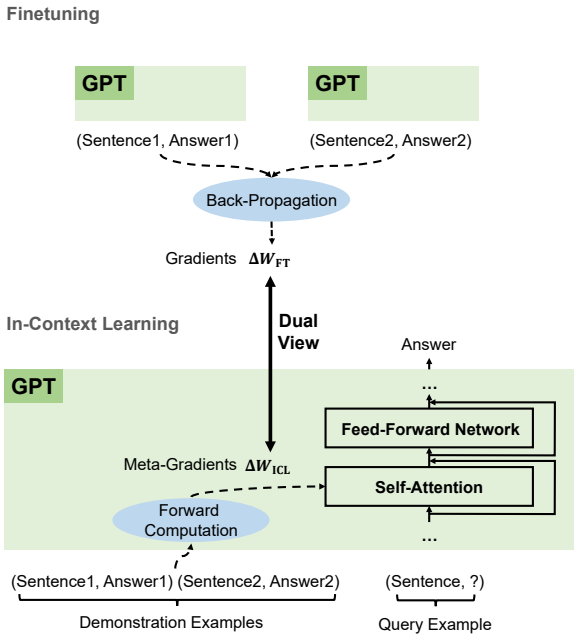

图 1: 根据演示例子，GPT通过前向计算生成用于上下文学习（ICL）的元梯度。 ICL通过将这些元梯度应用于模型来发挥作用。 ICL的元优化过程与微调共享一个双重视角，微调通过反向传播的梯度显式地更新模型参数。

1 Introduction

近年来，大型预训练语言模型，特别是基于Transformer架构的模型（例如GPT; Brown et al. 2020），展现出了强大的上下文学习（in-context learning, ICL）能力 (Wei et al., 2022; Dong et al., 2023)。与需要额外参数更新的微调（finetuning）不同，ICL只需要在查询输入之前添加一些示例，然后模型就可以为未知输入预测标签。在许多下游任务中，大型GPT模型可以达到令人惊讶的性能，甚至超过了经过监督微调的较小模型。然而，尽管ICL取得了

*Contribution during internship at Microsoft Research.

很好的性能，其工作机制仍然是一个有待探究的问题。

本文将ICL解释为元优化的过程，并分析了基于GPT的ICL和微调之间的联系。我们专注于注意力模块，并发现Transformer注意力具有梯度下降的双重形式。在此基础上，我们提出了一种新颖的观点来解释ICL：（1）预训练的GPT作为元优化器；（2）通过正向计算，它根据示例生成元梯度；（3）元梯度通过注意力应用于原始语言模型，构建了一个ICL模型。正如图 1所示，在上下文学习和显式微调中，存在梯度下降的双重视图，ICL通过正向计算产生元梯度，而微调通过反向传播计算梯度。因此，将上下文学习理解为隐式微调是合理的。

为了提供实证证据来支持我们对ICL的理解，我们基于真实任务进行了全面的实验。在六个分类任务中，我们比较了上下文学习和微调之间的模型预测、注意力输出、对查询标记的注意力权重以及对训练标记的注意力权重。实验结果验证了从多个角度来看，上下文学习的行为类似于显式微调。这些结果是证明我们将上下文学习理解为隐式微调的理性依据。

此外，受Transformer注意力和梯度下降之间的双重形式的启发，我们设计了一种基于动量的注意力，将注意力值视为元梯度，并将动量机制 (Polyak, 1964; Sutskever et al., 2013) 应用于这些梯度。语言建模和上下文学习的实验表明，我们的基于动量的注意力始终优于普通注意力，再次从另一个角度支持了我们对元优化的理解。值得探究的是，在此初步尝试之外，我们的理解可能还有更多潜力来启示模型设计，这值得在未来进行研究。

以下是我们的主要贡献：

- 我们找到了Transformer注意力和梯度下降之间的对偶形式，并将ICL解释为元优化过程。
- 我们分析了在上下文学习和显式微调之间的关联，并提出将ICL理解为隐式微调。
- 我们提供了几条实证证据来证明从多个角度来看，ICL和显式微调表现出相似的行为。
- 我们设计了一种基于动量的注意力，并验证了其有效性，这再次支持了我们对元优化的理解，并展示了我们理解的潜力，可以启发未来模型设计。

2 Background

2.1 In-Context Learning with GPT

本文主要研究使用GPT（GPT-3）的ICL分类任务。GPT模型由L个相同的Transformer解码器层堆叠而成，每层包含一个注意力模块和一个前馈网络。对于一个分类任务，给定一个查询输入文本x和一个候选答案集合$Y = \{y_1, y_2, \ldots, y_m\}$，我们需要在$n$个演示示例$C = \{(x'_1, y'_1), (x'_2, y'_2), \ldots, (x'_n, y'_n)\}$的条件下预测一个标签$\hat{y}$，其中$(x'_i, y'_i)$是与查询不同的输入-标签对。形式上，给定GPT模型$\mathcal{M}$，我们首先计算每个答案y_j的概率：

$$P_{\mathcal{M}}(y_j \mid C, x). \tag{1}$$

由于分类的标签空间受限，我们通过从候选答案集合 Y 中选择具有最高概率的答案来预测最终答案 $\hat{y}$：

$$\hat{y} = \arg\max_{y_j} P_{\mathcal{M}}(y_j \mid C, x). \tag{2}$$

在实践中，我们通常使用预定义的模板来格式化演示，并将其置于查询输入之前。设$\mathcal{T}(\cdot)$是格式化示例的函数，例如：

$$\mathcal{T}(x, y) = \text{Sentence: } x. \text{ Sentiment: } y. \tag{3}$$

上下文模型输入I的组织方式如下

$$\mathcal{T}(x'_1, y'_1)\ \mathcal{T}(x'_2, y'_2)\ \ldots\ \mathcal{T}(x'_n, y'_n)\ \mathcal{T}(x, \_). \tag{4}$$

将这个上下文输入馈送到$\mathcal{M}$中，答案y_j的概率被计算为

$$l_j = \mathcal{M}(I) \cdot \mathbf{e}_{y_j}, \tag{5}$$

$$P_{\mathcal{M}}(y_j \mid C, x) = \text{softmax}(l_j), \tag{6}$$

其中，$\mathcal{M}(I)$ 表示最后一个标记位置处的输出隐藏状态； $\mathbf{e}_{y_j}$ 表示 y_j 的输出词嵌入；而 l_j 是与第 j 个答案对应的逻辑值。

2.2 Dual Form Between Attention and Linear Layers Optimized by Gradient Descent

这篇论文中将语言模型解释为元优化器的思路受到Aizerman et al. (1964); Irie et al. (2022)的启发。他们提出，经由梯度下降优化的线性层具有线性注意力的双重形式。设$W_0, \Delta W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$为初始化的参数矩阵和更新矩阵，$\mathbf{x} \in \mathbb{R}^{d_{\text{in}}}$为输入表示。通过梯度下降优化的线性层可以表示为

$$\mathcal{F}(\mathbf{x}) = (W_0 + \Delta W)\,\mathbf{x}. \tag{7}$$

在反向传播算法中，ΔW的计算是通过累加历史输入表示$\mathbf{x}_i'^T \in \mathbb{R}^{d_{\text{in}}}$的外积和相应输出的误差信号$\mathbf{e}_i \in \mathbb{R}^{d_{\text{out}}}$来实现的：

$$\Delta W = \sum_i \mathbf{e}_i \otimes \mathbf{x}_i', \tag{8}$$

其中，$\mathbf{e}_i$ 是通过将负学习率$-\gamma$乘以历史输出梯度得到的。结合方程（7）和方程（8），我们可以推导使用梯度下降优化的线性层的对偶形式：

$$\begin{aligned}
\mathcal{F}(\mathbf{x}) =& (W_0 + \Delta W)\,\mathbf{x} \\
=& W_0\mathbf{x} + \Delta W\mathbf{x} \\
=& W_0\mathbf{x} + \sum_i \left(\mathbf{e}_i \otimes \mathbf{x}_i'\right)\mathbf{x} \\
=& W_0\mathbf{x} + \sum_i \mathbf{e}_i \left(\mathbf{x}_i'^T\mathbf{x}\right) \\
=& W_0\mathbf{x} + \text{LinearAttn}\left(E, X', \mathbf{x}\right),
\end{aligned} \tag{9}$$

其中，$\text{LinearAttn}(V, K, \mathbf{q})$ 表示线性注意力操作，其中我们将历史输出误差信号 E 视为值，历史输入 X' 视为键，当前输入 $\mathbf{x}$ 视为查询。

3 Understanding In-Context Learning (ICL) as Implicit Finetuning

我们首先采用一种放松的线性注意力形式对Transformer的注意力进行定性分析，以找出它与梯度下降之间的双重形式。然后，我们将上下文学习与显式微调进行比较，以分析这两种优化形式之间的关系。基于这些理论发现，我们提出将上下文学习理解为隐式微调。

3.1 Understanding Transformer Attention as Meta-Optimization

令 $\mathbf{x} \in \mathbb{R}^d$ 表示查询标记 t 的输入表示，$\mathbf{q} = W_Q\mathbf{x} \in \mathbb{R}^{d'}$ 表示注意力查询向量。在 ICL 设置中，一个头的注意力结果可以表示为

$$\begin{aligned}
\mathcal{F}_{\text{ICL}}(\mathbf{q}) =& \text{Attn}(V, K, \mathbf{q}) \\
=& W_V[X'; X]\,\text{softmax}\left(\frac{(W_K[X'; X])^T\,\mathbf{q}}{\sqrt{d}}\right),
\end{aligned} \tag{10}$$

其中，$W_Q, W_K, W_V \in \mathbb{R}^{d' \times d}$ 是用于计算注意力查询、关键字和值的投影矩阵； $\sqrt{d}$ 表示缩放因子； X 表示在 t 之前的查询标记的输入表示； X' 表示演示标记的输入表示； $[X'; X]$ 表示矩阵拼接。为了方便定性分析，我们通过移除 softmax 操作和缩放因子，将标准注意力近似为放松线性注意力。

$$\begin{aligned}
\mathcal{F}_{\text{ICL}}(\mathbf{q}) \approx& W_V[X'; X]\,(W_K[X'; X])^T\,\mathbf{q} \\
=& W_V X\,(W_K X)^T\,\mathbf{q} + W_V X'\,(W_K X')^T\,\mathbf{q} \\
=& \widetilde{\mathcal{F}}_{\text{ICL}}(\mathbf{q}).
\end{aligned} \tag{11}$$

我们定义$W_{\text{ZSL}} = W_V X\,(W_K X)^T$作为需要更新的初始化参数，因为$W_{\text{ZSL}}\mathbf{q}$是零样本学习（ZSL）设置中的注意力结果，其中没有给出演示。按照方程（9）的反向方向，我们推导出Transformer注意力的对偶形式：

$$\begin{aligned}
\widetilde{\mathcal{F}}_{\text{ICL}}(\mathbf{q}) =& W_{\text{ZSL}}\mathbf{q} + W_V X'\,(W_K X')^T\,\mathbf{q} \\
=& W_{\text{ZSL}}\mathbf{q} + \text{LinearAttn}\left(W_V X', W_K X', \mathbf{q}\right) \\
=& W_{\text{ZSL}}\mathbf{q} + \sum_i W_V\mathbf{x}_i' \left((W_K\mathbf{x}_i')^T\,\mathbf{q}\right) \\
=& W_{\text{ZSL}}\mathbf{q} + \sum_i \left((W_V\mathbf{x}_i') \otimes (W_K\mathbf{x}_i')\right)\mathbf{q} \\
=& W_{\text{ZSL}}\mathbf{q} + \Delta W_{\text{ICL}}\mathbf{q} \\
=& \left(W_{\text{ZSL}} + \Delta W_{\text{ICL}}\right)\mathbf{q}.
\end{aligned} \tag{12}$$

如上所示的方程表明，对于演示标记的关注等效于对W_{ZSL}产生实际影响的参数更新ΔW_{ICL}。此外，类比于方程（9）中的E，我们将$W_V X'$视为元梯度，用于计算更新矩阵ΔW_{ICL}。

总之，我们解释了上下文学习的过程是一种元优化的过程： (1) 预训练的GPT模型充当元优化器； (2) 通过前向计算，它根据演示示例产生元梯度； (3) 通过注意力，将元梯度应用于原始语言模型以构建ICL模型。

3.2 Comparing ICL with Finetuning

基于对上下文学习的上述理解，我们进一步比较了上下文学习的元优化与显式优化细调之间的联系进行分析。考虑到上下文学习仅直接影响注意力键和值，我们设计了一个特定的细调设置作为比较基线，该设置仅更新键和值投影的参数。此外，在放松的线性注意力形式中，细调后的头部的注意力结果被表述为

$$\begin{aligned}\tilde{\mathcal{F}}_{\mathrm{FT}}(\mathbf{q}) &= (W_V + \Delta W_V)XX^T(W_K + \Delta W_K)^T\mathbf{q} \\ &= (W_{\mathrm{ZSL}} + \Delta W_{\mathrm{FT}})\,\mathbf{q},\end{aligned} \tag{13}$$

ΔW_K和ΔW_V分别表示从任务特定的训练目标中通过反向传播得到的W_K和W_V的参数更新；ΔW_{FT}是通过微调引入的W_{ZSL}的更新。

为了与上下文学习进行更公平的比较，我们进一步限制了微调设置，具体如下：（1）我们将训练示例指定为上下文学习的演示示例；（2）我们按照上下文学习的演示顺序，仅对每个示例进行一步训练；（3）我们使用用于上下文学习的相同模板格式化每个训练示例$\mathcal{T}(x_i', y_i')$，并对微调使用因果语言建模目标。

通过比较上下文学习和这种微调设置，我们发现上下文学习与微调具有许多共同的性质。我们将这些共同性质归为以下四个方面。

两者均进行梯度下降 通过比较公式 (12)和公式 (13)，我们发现上下文学习和微调都引入了对W_{ZSL}的更新（ΔW_{ICL}和ΔW_{FT}），分别来自于显式梯度下降和隐式梯度下降。主要区别在于上下文学习通过正向计算产生元梯度，而微调通过反向传播获取真正的梯度。

相同的训练信息 上下文学习的元梯度根据演示示例产生。微调的梯度也是从相同的训练示例派生出来的。也就是说，上下文学习和微调共享相同的训练信息来源。

相同的训练示例的因果顺序 上下文学习和我们的微调设置共享相同的训练示例的因果顺序。上下文学习使用仅编码器的Transformer模型，因此演示中的后续令牌不会影响前面的令牌。对于我们的微调设置，我们使用相同的训练示例顺序进行仅一轮的训练，因此我们也可以保证后续示例不会影响前面的示例。

两者的目标都是关注机制 与零样本学习相比，上下文学习和我们的微调的直接影响都仅限于注意力键和值的计算。对于上下文学习，模型参数保持不变，它将演示信息编码为额外的键和值来改变注意力行为。对于微调，由于我们的限制，训练信息只能被引入到注意力键和值的投影矩阵中。

考虑到上下文学习和微调之间的上述共同性质，我们认为将上下文学习理解为隐式微调是合理的。在本文的剩余部分，我们将从多个角度对上下文学习和显式微调进行实证比较，以提供定量结果来支持这一理解。

4 Experiments

4.1 Experimental Settings

我们分析了两个现成的预训练GPT模型，分别拥有13亿和27亿个模型参数，这些模型是由fairseq发布的[1]。在本文的剩余部分，我们将它们简称为GPT 1.3B和GPT 2.7B。所有实验都在NVIDIA V100 GPU上进行，内存为32GB。

对于每个任务，我们使用相同的模板来为零-shot学习（ZSL）、微调（FT）和上下文学习（ICL）格式化示例。每个任务使用的模板的详细信息在附录 A中提供。ZSL和微调的答案预测过程与第 2.1 节中描述的ICL相同，只是它们没有演示示例。

对于上下文学习，我们将演示示例的最大数量固定为32，并为每个任务调整随机种子，以找到一组能够达到最佳验证性能的演示示例。对于显式微调，我们使用与上下文学习相同的演示示例作为训练示例，并使用SGD作为优化器。为了进行公平比较，我们只对模型进行一次微调，训练示例的提供顺序与上下文学习演示的顺序相同。我们调整微调的学习率，并选择达到最佳验证性能的学习率。有关随机

[1] https://github.com/facebookresearch/fairseq

	SST2	SST5	MR	Subj	AGNews	CB
# Validation Examples	872	1101	1066	2000	7600	56
# Label Types	2	5	2	2	4	3
ZSL Accuracy (GPT 1.3B)	70.5	39.3	65.9	72.6	46.3	37.5
FT Accuracy (GPT 1.3B)	73.9	39.5	73.0	77.8	65.3	55.4
ICL Accuracy (GPT 1.3B)	92.7	45.0	89.0	90.0	79.2	57.1
ZSL Accuracy (GPT 2.7B)	71.4	35.9	60.9	75.2	39.8	42.9
FT Accuracy (GPT 2.7B)	76.9	39.1	80.0	86.1	65.7	57.1
ICL Accuracy (GPT 2.7B)	95.0	46.5	91.3	90.3	80.3	55.4

表 1: 六个分类数据集的统计数据（行1-2），以及在这些数据集上零样本学习（ZSL）、微调（FT）和上下文学习（ICL）设置下的验证准确率（行3-8）。

Model	SST2	SST5	MR	Subj	AGNews	CB	Average
GPT 1.3B	91.84	66.67	97.08	87.17	83.08	87.50	85.56
GPT 2.7B	96.83	71.60	95.83	87.63	84.44	100.00	89.39

表 2: 对于六个数据集的两个GPT模型，Rec2FTP得到以下结果。从模型预测的角度来看，ICL可以覆盖大部分fine-tuning的正确行为。

种子和学习率的搜索范围和选择的值的详细信息，请参见附录 B。

4.2 Evaluation Datasets

我们比较了基于上下文学习和微调的方法，并使用涵盖三种分类任务的六个数据集进行了比较。 **SST2**(Socher et al., 2013)、**SST5**(Socher et al., 2013)、**MR**(Pang and Lee, 2005)和**Subj**(Pang and Lee, 2004)是情感分类的四个数据集； **AGNews**(Zhang et al., 2015)是一个主题分类数据集；而**CB**(De Marneffe et al., 2019)则用于自然语言推理。验证集示例数量和标签类型的统计结果见表1。

为了参考，我们在表1中展示了零样本学习（ZSL）、微调和上下文学习（ICL）三种设置下在六个分类数据集上的验证准确率。与ZSL相比，ICL和微调都取得了显著的改进，这表明它们对于这些下游任务的优化都是有帮助的。

4.3 ICL Covers Most of Correct Predictions of Finetuning

我们计算一个回忆到微调预测（Rec2FTP）来衡量ICL可以从模型预测的角度覆盖多少微调行为。首先，我们计算finetuning可以正确预测但ZSL不能的查询样例数量$N_{\text{FT>ZSL}}$。然后，在这些样例中，我们计算ICL也可以正确预测的数量$N_{(\text{FT>ZSL})\wedge(\text{ICL>ZSL})}$。最后，我们计算Rec2FTP得分为$\frac{N_{(\text{FT>ZSL})\wedge(\text{ICL>ZSL})}}{N_{\text{FT>ZSL}}}$。较高的Rec2FTP得分说明ICL从模型预测的角度覆盖了更多正确的微调行为。

我们在表 2中展示了两个GPT模型在六个数据集上的Rec2FTP得分。如表所示，平均而言，ICL可以正确预测超过85%的finetuning可以从ZSL中纠正的样例。这些结果表明，从模型预测的角度来看，ICL可以覆盖大部分的正确微调行为。

Model	Metric	SST2	SST5	MR	Subj	AGNews	CB	Average
GPT 1.3B	SimAOU (Random Δ)	0.002	0.003	0.001	0.002	0.002	0.003	0.002
	SimAOU (ΔFT)	**0.110**	**0.080**	**0.222**	**0.191**	**0.281**	**0.234**	**0.186**
GPT 2.7B	SimAOU (Random Δ)	0.000	-0.002	0.000	0.001	-0.002	0.000	-0.001
	SimAOU (ΔFT)	**0.195**	**0.323**	**0.157**	**0.212**	**0.333**	**0.130**	**0.225**

表 3: 对于六个数据集上两个GPT模型的模拟结果。ICL更新更类似于微调更新，而不是随机更新。从表示的角度看，ICL倾向于以与微调相同的方向改变注意力输出表示。

Model	Metric	SST2	SST5	MR	Subj	AGNews	CB	Average
GPT 1.3B	SimAM (Before Finetuning)	0.555	0.391	0.398	0.378	0.152	0.152	0.338
	SimAM (After Finetuning)	**0.585**	**0.404**	**0.498**	**0.490**	**0.496**	**0.177**	**0.442**
GPT 2.7B	SimAM (Before Finetuning)	**0.687**	0.380	0.314	0.346	0.172	**0.228**	0.355
	SimAM (After Finetuning)	**0.687**	**0.492**	**0.347**	**0.374**	**0.485**	0.217	**0.434**

表 4: SimAM for two models on six datasets. From the perspective of attention behavior, compared with attention weights before finetuning, ICL is more inclined to generate similar attention weights to those after finetuning.

4.4 ICL Tends to Change Attention Outputs in the Same Direction as Finetuning

从表示的角度来看，我们计算一个称为**注意力输出更新的相似度（SimAOU）**来衡量ICL和微调所做的更新之间的相似性。对于一个查询示例，让$\mathbf{h}_{\mathrm{X}}^{(l)}$表示X设置中第$l$个注意力层中最后一个标记的标准化输出表示。ICL和微调相对于ZSL的更新分别为$\mathbf{h}_{\mathrm{ICL}}^{(l)} - \mathbf{h}_{\mathrm{ZSL}}^{(l)}$和$\mathbf{h}_{\mathrm{FT}}^{(l)} - \mathbf{h}_{\mathrm{ZSL}}^{(l)}$。我们计算这两个更新之间的余弦相似度，以得到第$l$个层的**SimAOU (ΔFT)**。更高的SimAOU (ΔFT)意味着ICL更倾向于以与微调相同的方向更新注意力输出。为了比较，我们还计算了一个基准指标，称为**SimAOU（随机Δ）**，用于计算ICL更新和随机生成的更新之间的相似性。

我们在表 3中呈现了两个GPT模型在六个数据集上示例和层次平均的SimAOU得分。从表中可以看出，SimAOU（随机Δ）始终接近零，而SimAOU（ΔFT）保持更为正向的值。这些结果表明，ICL的更新与微调的更新比与随机更新更相似。从表示的角度来看，我们证明了ICL倾向于以与微调相同的方向改变注意力输出。

4.5 ICL Is Inclined to Generate Similar Attention Weights to Finetuning

从注意力行为的角度出发，我们计算一个**注意力图相似度（SimAM）**来衡量注意力图与查询标记的相似度，以评估ICL和微调。对于一个查询样本，设$\mathbf{m}_{\mathrm{X}}^{(l,h)}$为在X设置中第$l$层第$h$个注意头中最后一个标记的softmax之前的注意力权重。对于ICL，我们省略了对演示标记的注意力，只监控对查询标记的注意力权重。首先，在微调之前，我们计算$\mathbf{m}_{\mathrm{ICL}}^{(l,h)}$和$\mathbf{m}_{\mathrm{ZSL}}^{(l,h)}$之间的余弦值，然后对各个注意头的相似度求平均得到每层的**SimAM（微调之前）**。类似地，在微调之后，我们计算$\mathbf{m}_{\mathrm{ICL}}^{(l,h)}$和$\mathbf{m}_{\mathrm{FT}}^{(l,h)}$之间的余弦值，得到**SimAM（微调之后）**。SimAM（微调之后）相对于SimAM（微调之前）更高，表明ICL的注意力行为更类似于微调模型而不是非微调模型。

表 4展示了两个GPT模型在六个数据集上

Model	Metric	SST2	SST5	MR	Subj	AGNews	CB	Average
GPT 1.3B	Kendall (ICL, Random)	0.000	-0.001	0.000	0.001	-0.001	0.000	0.000
	Kendall (ICL, FT)	**0.192**	**0.151**	**0.173**	**0.181**	**0.190**	**0.274**	**0.193**
GPT 2.7B	Kendall (ICL, Random)	-0.001	0.000	0.000	0.000	0.000	-0.001	0.000
	Kendall (ICL, FT)	**0.213**	**0.177**	**0.264**	**0.203**	**0.201**	**0.225**	**0.214**

表 5: 两个GPT模型在六个数据集上的Kendall等级相关系数。与随机注意力权重相比，ICL注意力权重对训练的符号更接近fine-tuning注意力权重。

的示例和层之间平均的SimAM分数。我们观察到，与微调之前的注意权重相比，ICL更倾向于生成与微调之后的注意权重类似的注意权重。再次从注意力行为的角度出发，我们证明了ICL的行为与微调类似。

4.6 ICL and Finetuning Tend to Pay Similar Attention to Training Tokens

由于我们将ICL视为一个元优化的过程，我们还比较了ICL和fine-tuning对训练标记的注意力，使用**Kendall等级相关系数**(Kendall, 1948)进行比较。对于一个查询示例，令$\mathbf{m}_{\text{ICL}}^{(l)}$表示对于第$l$个注意力层中最后一个查询标记的演示标记的ICL注意力权重，这些权重在注意头之间求和。对于fine-tuning，我们首先记录所有训练标记的注意力查询$Q'^{(l,h)} \in \mathbb{R}^{d' \times N}$，然后使用它们与查询示例中最后一个标记的注意力查询$\mathbf{q}^{(l,h)} \in \mathbb{R}^{d'}$之间的内积作为fine-tuning对训练标记的注意力权重：$\mathbf{m}_{\text{FT}}^{(l)} = \sum_h Q'^{(l,h)^T}\mathbf{q}^{(l,h)}$，这些权重也在注意头之间求和。$\mathbf{m}_{\text{ICL}}^{(l)}$和$\mathbf{m}_{\text{FT}}^{(l)}$之间的Kendall系数被计算为**Kendall (ICL, FT)** $= \frac{P_c - P_d}{N(N-1)/2}$，其中$N$表示训练标记的数量，$P_c$表示一致对的数量，$P_d$表示不一致对的数量。较高的Kendall系数意味着ICL和fine-tuning对训练标记的注意力权重的顺序更相似。为了对比，我们还计算了$\mathbf{m}_{\text{ICL}}^{(l)}$和随机生成的注意力权重$\mathbf{m}_{\text{Random}}^{(l)}$之间的Kendall系数，我们称之为**Kendall (ICL, Random)**。

表 5显示了两个GPT模型在六个数据集上的平均示例和层之间的Kendall相关系数。我们发现，Kendall (ICL, Random)始终接近零，而Kendall (ICL, FT)始终保持明显的正值。这些结果表明，ICL和fine-tuning倾向于对训练标记付予类似的注意力。

5 Momentum-Based Attention Inspired by Dual Form of Transformer Attention

我们已经找到了Transformer注意力机制和梯度下降之间的对偶形式。如图 2 所示，在这种对偶视角的启发下，我们研究是否可以利用动量 (Polyak, 1964; Sutskever et al., 2013)，一种被广泛用于优化算法的技术，来改进Transformer的注意力机制。

动量梯度下降算法对时间戳进行梯度平均处理：

$$\Theta_t = \Theta_{t-1} - \gamma \sum_{i=1}^{t-1} \eta^{t-i} \nabla f_{\Theta_i}, \tag{14}$$

其中γ是学习率，η是介于0和1之间的标量。如3.1小节所述，注意力值被视为元梯度（meta-gradients）。类似于动量法（momentum）的梯度下降法，我们尝试使用指数移动平均（Exponential Moving Average，简称EMA；Hunter 1986）对注意力值进行平均，以建立基于动量的注意力：

$$\begin{aligned}\text{MoAttn}(V, K, \mathbf{q}_t) &= \text{Attn}(V, K, \mathbf{q}_t) + \text{EMA}(V) \\ &= V\,\text{softmax}(\frac{K^T\mathbf{q}_t}{\sqrt{d}}) + \sum_{i=1}^{t-1} \eta^{t-i}\mathbf{v}_i,\end{aligned}$$

其中，$\mathbf{v}_i$是第i个注意力值向量。注意力值向量的动量明确加强了注意力的最近偏好，这已经被证明对语言建模是有帮助的 (Press et al.,

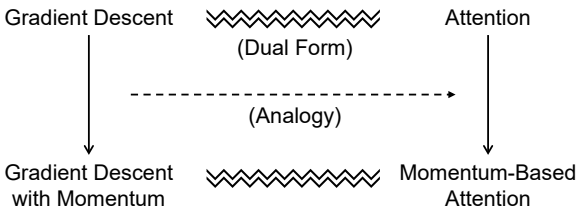

图 2: 受到注意力和梯度下降之间的双重形式的启发，我们通过类比带有动量的梯度下降，将动量机制引入Transformer注意力中。

2022)。因此，我们假设将动量引入注意力将有助于更快的收敛和更好的性能。

语言建模实验 首先，我们评估基于动量的注意力对语言建模的影响。我们从零开始训练了两个具有350M参数的GPT模型，其中一个是传统的Transformer模型，另一个应用了动量注意力。有关更多训练细节，请参见附录C。我们分别在训练集和输入长度分别为256、512和1024的三个验证集上评估了这两个模型的困惑度。结果如表 6所示。在所有的验证集上，相比传统的Transformer模型，应用动量注意力均能显著提高困惑度。

上下文学习实验 我们还评估了上述语言模型的上下文学习能力，以验证基于动量的注意力在下游任务中的有效性。我们考虑了六个数据集，用于情感分析（SST5 (Socher et al., 2013)，IMDB (Maas et al., 2011)和MR (Pang and Lee, 2005)）、自然语言推理（CB (De Marneffe et al., 2019)）和多项选择（ARC-E (Clark et al., 2018)和PIQA (Bisk et al., 2020)）。对于所有这些数据集，我们最多使用32个示例作为演示。如表 7所示，在所有这些数据集上，相比于传统的Transformer模型，使用基于动量的注意力能够获得更高的准确度。

无论是在语言建模还是上下文学习方面的性能改进，都证明了我们的推论，即引入动量将改善Transformer的注意力。从另一个角度来看，这些结果进一步支持我们将Transformer的注意力视为元优化的理解。

6 Related Work

最近，一些工作试图理解上下文学习的推理机制。Bayesian（2007）将上下文学习解释为隐性的贝叶斯推理。他们指出，当语言模型能够在预训练期间学习到的共享潜在概念中进行推断时，上下文学习才会出现。Olsson等人（2022）着重研究了Transformer中的特定模块。他们发现Transformer中的一些感知模块参考了前序序列中的抽象模式，以帮助预测下一个符号。他们指出，感知模块推动了上下文学习的能力。与他们不同的是，我们聚焦于ICL的学习算法，并将其解释为元优化的过程。

其他一些工作也研究了ICL的学习算法。以一个案例研究为例，ICL案例研究（年份）展示了Transformer可以通过上下文学习来学习一类线性函数，并且性能可与最小二乘估计量相媲美。基于线性回归，ICL学习算法（年份）证明他们可以构建Transformer的参数以实现基于梯度下降的学习算法。此外，他们还表明，通过上下文学习目标训练的模型往往与通过显式学习算法计算的模型的行为相匹配。基于回归任务，ICL-GD（年份）展示了仅基于线性注意力的Transformer与通过构建参数实现梯度下降和通过上下文学习目标学习的模型之间高度相关性。与他们相比，我们是第一批在真实场景中解释上下文学习的研究者。具体而言，（1）我们分析了现成的GPT模型的上下文学习，而不是通过ICL目标从头开始训练的模型；（2）我们的实验基于真实的自然语言处理任务，而不是像线性回归这样的玩具问题。

7 Conclusion

本文旨在解释基于GPT的ICL（Implicit Continuous Learning）的工作机制。从理论上讲，我们找到了Transformer attention和梯度下降之间的双重形式，并提出将ICL理解为元优化的过程。此外，我们分析了ICL与显式微调之间的联系，并显示将ICL视为隐式微调是合理的。从经验上看，我们全面比较了ICL和基

Model	Train_{1024}	Valid_{256}	Valid_{512}	Valid_{1024}
Transformer	17.61	19.50	16.87	15.14
$\text{Transformer}_{\text{MoAttn}}$	**17.55**	**19.37**	**16.73**	**15.02**

表 6: 在语言建模中，使用不同输入长度的训练集和验证集计算困惑度。基于动量的注意力机制相较于普通Transformer稳定地提升了困惑度。

Model	SST5	IMDB	MR	CB	ARC-E	PIQA	Average
Transformer	25.3	64.0	61.2	43.9	48.2	68.7	51.9
$\text{Transformer}_{\text{MoAttn}}$	**27.4**	**70.3**	**64.8**	**46.8**	**50.0**	**69.0**	**54.7**

表 7: 六个具有上下文学习数据集上的准确性。将动量引入注意力机制，平均可使原始Transformer的准确性提高2.8。

于六个真实NLP任务的微调。结果证明，从多个角度来看，ICL的行为类似于显式微调。受到对元优化的理解的启发，我们设计了一种基于动量的attention，能够在普通attention的基础上实现持续的性能改进。我们相信我们的理解将对未来的ICL应用和模型设计提供更多潜力启示。

Limitations

尽管已经发现了不同体系结构（如Transformer和LSTM）的上下文学习能力，但本文仅考虑基于Transformer的上下文学习，因为Transformer是当前NLP的主流体系结构。然而，对于上下文学习本身而言，弄清楚它在其他体系结构中的工作原理也是一个有意义的问题，我们鼓励在将来进行研究。

关于我们指出的Transformer注意力和梯度下降之间的双重形式，我们考虑了一种放松的线性注意力形式进行定性分析。尽管实验结果很好地支持了我们的理解，但标准Transformer注意力机制（不进行近似）可能更加复杂，需要更清楚地研究。

对于经验实验，我们的分析需要记录大量中间结果（例如注意力输出表示、对查询标记和演示标记的注意权重等）来处理数千个验证示例。考虑到分析的存储空间和计算成本，我们仅分析具有2.7B参数的GPT模型，将更大的模型（如GPT 13B）留待将来研究。此外，为了问题定义的清晰度和实验的方便性，我们的分析仅基于分类任务。尽管分类是上下文学习的代表性应用，但本文未考虑其他任务，如多项选择和开放式生成，这些任务可以在将来进行研究。

Acknowledgement

戴大麦和随志芳得到了中国2020AAA0106700国家重点研发计划和NSFC项目U19A2065的支持。

References

Mark A Aizerman, Emmanuil M Braverman, and Lev I Rozonoer. 1964. Theoretical foundation of potential functions method in pattern recognition. *Avtomatika i Telemekhanika*, 25(6):917–936.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. *CoRR*, abs/2211.15661.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 7432–7439. AAAI Press.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal,

Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey for in-context learning. *CoRR*, abs/2301.00234.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. 2022. What can transformers learn in-context? A case study of simple function classes. *CoRR*, abs/2208.01066.

J Stuart Hunter. 1986. The exponentially weighted moving average. *Journal of quality technology*, 18(4):203–210.

Kazuki Irie, Róbert Csordás, and Jürgen Schmidhuber. 2022. The dual form of neural networks revisited: Connecting test time predictions to training patterns via spotlights of attention. In *International Conference on Machine Learning, ICML 2022*, volume 162 of *Proceedings of Machine Learning Research*, pages 9639–9659. PMLR.

Maurice George Kendall. 1948. Rank correlation methods.

Louis Kirsch, James Harrison, Jascha Sohl-Dickstein, and Luke Metz. 2022. General-purpose in-context learning by meta-learning transformers. *CoRR*, abs/2212.04458.

Louis Kirsch and Jürgen Schmidhuber. 2021. Meta learning backpropagation and improving it. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, pages 14122–14134.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *CoRR*, abs/2209.11895.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 115–124. The Association for Computer Linguistics.

Boris T Polyak. 1964. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *The Tenth International Conference on Learning Representations, ICLR 2022*. OpenReview.net.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. 2013. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1139–1147. JMLR.org.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information*

Processing Systems, pages 5998–6008. Curran Associates, Inc.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2022. Transformers learn in-context by gradient descent. *ArXiv preprint*, abs/2212.07677.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *CoRR*, abs/2206.07682.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR 2022*. OpenReview.net.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pages 649–657.

Appendix

A Templates for In-Context Learning

我们在表 8中展示了在我们的实验中使用的用于格式化示例和候选答案集的模板。

B Hyper-Parameters for In-Context Learning and Finetuning

我们使用网格搜索来寻找ICL和微调的最佳随机种子和学习率。所有数据集的搜索范围相同。对于随机种子，我们在$\{1,2,3,4,5,6,7\}$中进行搜索。对于学习率，搜索的基础值为$\{1,2,3,4,5,6,7,8,9\}$，然后将它们按比例缩小为0.1、0.01、0.001和0.0001，即我们有$9 \times 4 = 36$个值进行搜索。唯一的例外是，在对SST5进行微调的GPT 1.3B模型中，我们进行了更精细的搜索，最终将其学习率设置为0.00016，因为使用上述36个学习率进行微调无法超过零-shot学习的效果。

在表 9中，我们展示了两个GPT模型在六个分类数据集上选择的随机种子和学习率的详细信息。

C Hyper-Parameters for Training Language Models from Scratch

训练两个语言模型的初始超参数总结如表 10。

Dataset	Template	Candidate Answer Set
SST2	Sentence: {Sentence} Label: {Label}	{ Negative, Positive }
SST5	Sentence: {Sentence} Label: {Label}	{ terrible, bad, neutral, good, great }
MR	Review: {Sentence} Sentiment: {Label}	{ Negative, Positive }
Subj	Input: {Sentence} Type: {Label}	{ objective, subjective }
AGNews	Classify the news articles into the categories of World, Sports, Business, and Technology. News: {Sentence} Type: {Label}	{ World, Sports, Business, Technology }
CB	{Premise} Question: {Hypothesis} True, False, or Neither? Answer: {Label}	{ True, False, Neither }

表 8: 六个分类数据集的格式模板和候选答案集。

Hyper-Parameter	Dataset	GPT 1.3B	GPT 2.7B
Random Seed	SST2	2	7
	SST5	5	5
	MR	5	1
	Subj	4	4
	AGNews	3	3
	CB	3	3
Learning Rate	SST2	0.0005	0.007
	SST5	0.00016	0.04
	MR	0.003	0.001
	Subj	0.003	0.002
	AGNews	0.2	0.2
	CB	0.08	0.01

表 9: 在六个分类数据集上，选择了两个GPT模型的随机种子和学习率。

Hyper-parameter	**Value**
Embedding & Hidden Dimension	1024
FFN Inner Hidden Dimension	4096
Number of Attention Heads	16
Number of Transformer Layers	24
Number of Parameters	350M
Sequence Length	1024
Batch Size	512K Tokens
Optimizer	Adam
Adam Betas	(0.9, 0.98)
Adam Epsilon	1e-6
Maximum Learning Rate	3e-4
Learning Rate Scheduler	Polynomial Decay
Total Training Steps	500K
Warm-up Steps	20K
Gradient Clip Norm	2.0

表 10: 从头训练两个语言模型的超参数。