AdaEnsemble：用于点击率预测的自适应稀疏结构化集成网络学习

Yachen Yan
yachen.yan@creditkarma.com
Credit Karma 美国加利福尼亚州旧金山

Liubo Li
liubo.li@creditkarma.com
Credit Karma 美国加利福尼亚州旧金山

arXiv:2301.08353v1 [cs.IR] 6 Jan 2023

摘要

特征交互学习对于推荐系统和广告排序中的大规模点击率预测至关重要。研究人员和从业者广泛提出了各种神经网络架构，用于搜索和建模特征交互。然而，我们观察到，不同数据集偏好不同的神经网络架构和特征交互类型，这表明不同的特征交互学习方法可能各有独特优势。受这一观察启发，我们提出AdaEnsemble：一种稀疏门控混合专家（SparseMoE）架构，可利用异构特征交互专家的优势，并针对每个样本自适应地学习路由，将其分配到稀疏的专家组合中，从而构建由不同类型和阶数的特征交互组成的动态层次结构。为进一步提高预测准确率和推理效率，我们引入动态提前退出机制，以选择特征交互深度。AdaEnsemble 可以自适应地选择特征交互深度，并找到相应的 SparseMoE 堆叠层，在该层退出并据此计算预测结果。因此，我们提出的架构继承了 SparseMoE 层中稀疏门控专家的指数级组合所带来的优势，并进一步动态选择最优特征交互深度，而无需执行更深层。我们实现了所提出的AdaEnsemble，并在真实世界数据集上评估其性能。大量实验结果证明，与最先进模型相比，AdaEnsemble 高效且有效。

CCS概念

• **计算方法**；• **机器学习**；• **机器学习方法**；• **神经网络**；

关键词

点击率预测、推荐系统、特征交互、混合专家模型、动态推理、提前退出、自动化机器学习、深度神经网络

特此授权免费为个人或课堂用途制作本作品全部或部分内容的数字或纸质副本，前提是这些副本不得用于营利或商业利益，也不得为此进行制作或分发，并且副本须保留本声明及首页上的完整引用信息。对于本作品中非 ACM 所有的部分，必须尊重其版权。注明出处后允许摘录。若要以其他方式复制、再版、发布到服务器或分发至列表，则须事先获得明确许可和/或支付费用。请向 permissions@acm.org 申请许可。*Conference'17*，*2017*年7月，美国华盛顿特区 © 2022 计算机协会。ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/XXXXXXX.XXXXXXX

ACM 引用格式：
Yachen Yan 和 Liubo Li. 2022. AdaEnsemble：用于点击率预测的自适应稀疏结构化集成网络。载于*ACM*会议论文集（*Conference'17*）。ACM，纽约，NY，美国，9页。https://doi.org/XXXXXXX.XXXXXXX

1 引言

点击率（CTR）预测模型 [25] 是大规模搜索排序、在线广告和推荐系统 [4, 9, 20, 37]的重要组成部分。

工业界已提出许多基于深度学习的模型来解决点击率预测问题。这些模型已成为以端到端方式学习混合类型输入中有用特征交互的主流方法[37]。尽管大多数现有方法都能有效捕获高阶特征交互，但我们观察到，它们在不同数据集上的表现各不相同。我们认为这是由其归纳偏置导致的：不同方法学习不同类型的特征交互，因此适用于不同的数据集。

尽管现有方法都专注于自动建模不同类型的特征交互，但鲜有研究尝试联合、动态地建模不同类型的交互。我们认为，将不同的交互模块集成为异构特征交互，可以互补各类交互学习方法所学到的不同知识；与此相反，同质交互建模方法会限制所能学习的特征交互类型。为了利用多种交互模块学习不同类型的特征交互，我们采用稀疏门控混合专家（SpasrseMoE）架构来增强模型容量，同时通过条件计算提高计算效率。

我们提出 AdaEnsemble：一种稀疏门控混合专家（SparseMoE）层次化架构，用于集成不同的交互学习模块，并动态选择最优特征交互深度。在 AdaEnsemble 的每个 SparseMoE 层中，都包含一组交互学习专家，可训练门控网络会为每个样本确定要使用的专家稀疏组合。在深度选择控制器中，可训练门控网络会为每个样本选择特征交互深度，并递归地将特征交互表示沿 SparseMoE 层传播至相应深度，以计算预测结果。借助这些条件计算机制，我们在不增加推理成本的情况下，将模型容量呈指数级扩大。本文的主要贡献可总结如下：

• 我们设计了一种名为 AdaEnsemble 的新型模型架构，通过稀疏门控混合专家（SparsseMoE）集成各种类型的特征交互学习模块

通过递归使用带有残差连接和归一化的混合专家层，AdaEnsemble 可以联合且动态地对不同类型的交互进行建模。

- 我们设计了一种高效且有效的深度选择控制器，以自适应地选择最优的特征交互深度。借助该控制器，AdaEnsemble 能够动态确定提前退出的层，从而提高预测准确率和推理效率。
- 我们设计了一种双层优化算法，用于迭代训练建模网络和门控网络。
- 我们在真实世界数据集上进行了广泛实验，并研究了 AdaEnsemble 的学习模式。

2 相关工作

2.1 特征交互建模

学习特征交互是点击率预测问题中的关键课题，已在文献中得到广泛探讨。各种混合网络架构 [4, 8, 16, 22, 23, 31, 32] 以带有非线性激活函数的前馈神经网络为核心组件，用于学习隐式交互。对隐式交互建模的补充，能够提升仅对显式交互建模的网络性能 [3]。

另一类模型专注于探索按位/按向量的特征交互。深度交叉网络（DCN） [31]及其改进版本 DCN V2 [32] 以递归方式显式探索按位层面的特征交互。深度因子分解机（DeepFM） [8] 利用因子分解机层对成对向量交互进行建模。产品神经网络（PNN） [22, 23] 引入内积层和外积层，分别用于学习向量级交互和按位交互。xDeepFM [16] 使用压缩交互网络（CIN）学习显式的向量级交互；CIN 采用类似循环神经网络的架构，并利用哈达玛积学习向量级交互。FiBiNET [11] 利用挤压与激励网络动态学习特征的重要性，并通过双线性函数对特征交互建模。AutoInt [28] 采用 Transformer [30]架构，学习输入特征的不同阶特征组合。xDeepInt [35] 引入多项式交互层，以递归方式学习更高阶的向量级

以及受控阶数的按位交互，无需联合训练的深度神经网络和非线性激活函数。

2.2 稀疏专家混合网络 ork

稀疏门控混合专家模型 [2, 27]结合了多个专家和一个可训练门控网络，后者会为每个样本选择一部分专家。这种网络架构可视为一种动态稀疏结构：它保留所有权重，同时通过条件计算在模型中引入稀疏性。SparseMoE 广泛应用于自然语言处理研究领域。大多数讨论都聚焦于改进专家的路由机制。SwitchTransformer [7] 简化了 top-1 路由算法。GShard [14] 采用组级 top-2 路由。两者均通过负载均衡损失进行训练，并以更低的通信和计算成本提升语言模型。BASE Layers [15] 将路由视为线性指派

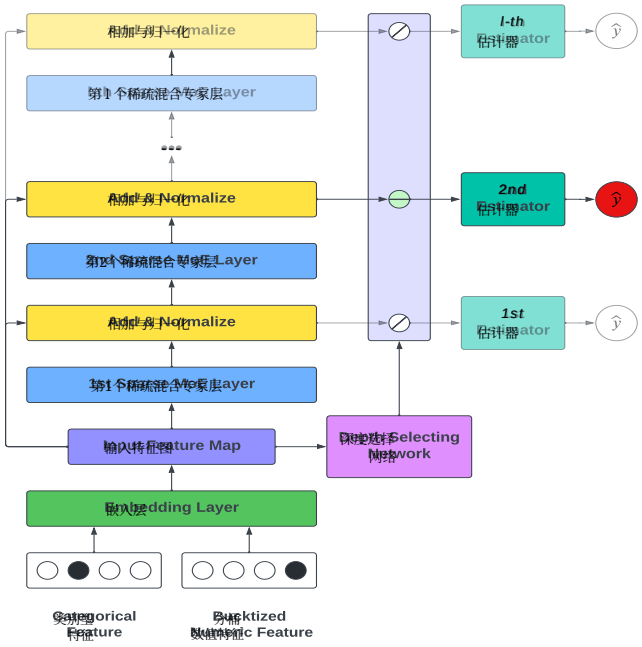


图 1：AdaEnsemble 的架构

在此样本中，深度选择网络选择在第 2 层退出并计算最终预测，因此更深的层不会被激活，并在图中以半透明方式绘制。

问题，并免去了负载均衡辅助损失。M6-T [36] 将专家划分为不同组，并采用 k 个 top-1 路由过程。一些研究探索了稀疏门控混合专家模型的训练。EvoMoE [21] 通过先训练所有专家，再逐步、自适应地变得更加稀疏，并在学习稀疏门控的过程中将样本路由到更少的专家，从而将专家训练与稀疏门控解耦。ST-MoE [40]进一步研究了 MoE 模型的训练不稳定性和质量不确定问题。X-MoE [5] 提出了降维和 L2 归一化方法，以解决 MoE 模型训练中的表示坍塌问题。

2.3 早退网络

神经网络早退的概念最初由 BranchyNet [29] 为计算机视觉提出。这项技术也应用于自然语言处理任务，随后推出 DeeBERT [33], FastBERT [18],和 PABEE [39]，以提高基于 Transformer 的 BERT 模型的推理效率。

对于早退机制，BranchyNet [29],DeeBERT [33], FastBERT [18] 和 SDN [12] 采用基于熵或基于置信度的标准。虽然使用基于熵或基于置信度的标准简单有效，但这种方法利用了模型输出在多分类任务中为概率分布这一特点。通常，这项技术无法应用于二分类和回归任务。另一方面，BERxiT [34] 和 EPNet [6] 则使用学习模块进行早退。

3 提出的模型：AdaEnsemble

本节概述 AdaEnsemble 的架构。首先，我们介绍特征处理和嵌入层，该层将连续特征和高维类别特征映射为稠密嵌入向量。其次，我们介绍用于联合学习深层特征表示层级结构的特征交互专家。第三，我们介绍稀疏专家混合（SparseMoE）层，该层可动态集成多个交互专家，以及与每个 SparseMoE 层关联的估计器。第四，我们讨论如何基于深度选择控制器自动、动态地选择特征交互深度。最后，我们将介绍用于训练的双层优化算法。

3.1 嵌入层

在大规模点击率预测任务中，输入包括连续特征和类别特征。类别特征通常直接通过独热编码进行编码，这会导致维度过高且稀疏的特征空间。

假设我们有 F 个字段。在特征处理步骤中，我们将所有连续特征划分到等频区间中，然后对分桶后的连续特征进行嵌入，并将类别特征中的每个特征嵌入为维度相同的稠密嵌入向量 e_i D。

$$\mathbf{e}_i = \mathbf{x}_i \mathbf{V}_i,$$

其中，$e_i \in R^D$，$\mathbf{V}_i$ 是第 i 个字段的嵌入矩阵，而 $\mathbf{x}_i$ 是对应的独热向量。最后，我们将 F 个嵌入向量拼接起来，并将嵌入层 $X_0 \in R^{F \times D}$ 的输出记为输入特征图：

$$X_0 = [e_1, e_2, \cdots, e_F]^\intercal. \tag{1}$$

3.2 特征交互专家

我们在模型中考虑了几种类型的特征交互专家：全连接层、卷积层、多头自注意力层、多项式交互层和交叉层。本质上，任何特征交互学习层都可以纳入我们的框架，并将残差连接和归一化应用于其集成。现在，我们介绍框架中包含的这些特征交互专家。请注意，我们提出的框架具有通用性，可以使用任意特征交互模块。可用的潜在特征交互专家并不限于以下几种。

3.2.1 全连接层。全连接层也称为密集层，是用于建模隐式特征交互的最常用模块。本文使用带有非线性激活函数的全连接层来学习深层特征表示。给定一个嵌入 X_{l-1}作为输入，嵌入 X_l 的输出可由下式得到：

$$X_l = \sigma(W_l \cdot X_{l-1}) \tag{2}$$

其中，σ 表示激活函数，W_l 表示第 l个全连接层的权重。

3.2.2 卷积层。卷积层广泛应用于计算机视觉问题。本文将一维卷积用作一种交互专家。由于卷积层是局部连接的，我们先在卷积层前使用全连接层融合输入嵌入。给定嵌入X_{l-1} 作为输入，可得到嵌入 X_l 的输出：

$$X_l = \text{Dense}(\text{Pooling}(\text{Conv1D}(\text{Reshape}(X_{l-1})))) \tag{3}$$

这里，我们首先重塑输入嵌入，然后进行一维卷积，接着使用池化层。最后，我们使用全连接层将输出投影到所需维度。

3.2.3 多头自注意力层。 多头自注意力层 [30] 因其在自然语言处理领域的卓越表现而广泛应用于 Transformer 网络，并且已开始在计算机视觉研究领域流行起来。我们考虑使用多头自注意力层来建模特征之间的依赖关系，并形成有意义的高阶特征。给定一个嵌入输入 X_{l-1}，嵌入输出 X_l 由下式得到：

$$X_l = \text{Dense}(\text{MultiHeadSelfAttention}(\text{Reshape}(X_{l-1})) \tag{4}$$

这里，我们首先重塑输入嵌入，然后应用多头自注意力层，接着使用全连接层将输出投影到所需维度。

3.2.4 多项式交互层。 多项式交互网络 [35] 旨在显式捕获有界阶数的特征交互。本文采用 PIN 层作为特征交互学习专家之一。给定一个嵌入输入X_{l-1}，第 l个 PIN 层输出的数学表征为：

$$X_l = X_{l-1} \circ (W_l \cdot X_0) \tag{5}$$

其中，$\circ$ 表示哈达玛积，W表示 PIN 层的核权重。我们在上述公式中省略了原论文中的残差连接，因为残差连接将在混合专家层之间使用。

3.2.5 交叉层。深度交叉网络 [32] 随后被提出，用于以递归方式探索特征交互。给定嵌入 X_{l-1} 作为输入，嵌入 X_l 的输出如下：

$$X_l = X_0 \circ (W_l \cdot X_{l-1}) + b_l \tag{6}$$

其中，W和 b分别表示第 l 个 DCN 层中的权重矩阵和偏置向量。我们同样在上述公式中省略了原始实现中的残差连接，因为我们将在混合专家层之间使用残差连接。

3.3 稀疏混合专家层

稀疏混合专家层将上述异构特征交互专家集成起来，并由其他几个关键部分组成，以确保整个模型能够稳定训练。

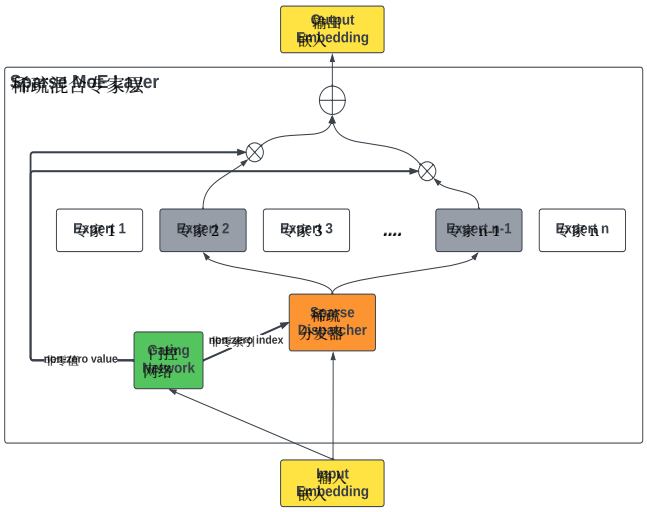


图 2：稀疏混合专家层的架构

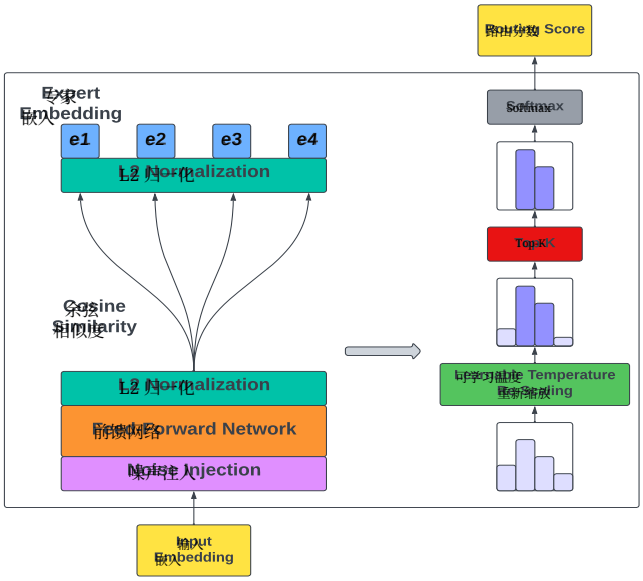


图 3：稀疏混合专家层中的噪声门控网络

3.3.1 噪声门控网络。门控网络本质上会计算门控值，用于选择各个专家的输出嵌入并为其赋权。

对于门控网络的输入嵌入 X_0，首先由门控网络进行处理：该网络是一个两层前馈网络，即包含一个缩减率为 r [10]的降维层、一个非线性激活函数，以及一个将结果投影到隐状态 $h \in R^d$的全连接层。此外，我们还应用了乘性抖动噪声，以引入探索并促进不同专家之间的负载均衡。

$$h = \text{FFN}(X_0 \circ \text{RandomUniform}(1.0 - \text{eps}, 1.0 + \text{eps})) \tag{7}$$

将输入嵌入投影到隐状态 $h \in R^d$ 后，我们对隐状态 $h \in R^d$ 和可学习的专家嵌入 $e_j \in R^d$应用 L_2 归一化，其中 j 是专家的索引。随后，我们计算隐状态与专家嵌入之间的余弦相似度，作为初始路由分数。这里我们鼓励表示均匀性，以避免专家主导问题。

$$s_j = \frac{h \cdot e_j}{\|h\| \|e_j\|} \tag{8}$$

最后，我们使用可学习温度标量 τ 将路由分数重新缩放到范围 $[-1, +1]$。

$$g_j = s_j / \tau \tag{9}$$

对于计算得到的路由分数 g，我们只保留其中最大的 k 个值，并将其余值设为 $-\infty$，从而使相应的 softmax 门控值等于 0。门控网络输出的第 i个元素为

$$ExpertG(x)_i = \frac{\exp\left(\text{TopK}(g,k)_i\right)}{\sum_{j=1}^{N} \exp\left(\text{TopK}(g,k)_j\right)}, \tag{10}$$

其中

$$\text{TopK}(g,k)_j = \begin{cases} g_j & \text{if } g_j \text{ is in the top } k \text{ elements of } g \\ -\infty & \text{otherwise.} \end{cases} \tag{11}$$

这些门控值将由稀疏分发器用于将样本路由到不同的专家。这是实现稀疏混合专家层稀疏性的关键步骤。请注意，无论 k[7]的值如何， G（x）都是可微的。

3.3.2 退火 *Top-K* 门控。我们还在 Top-K 操作中引入了退火机制。我们从 k 值等于专家数量的状态开始，这意味着初始门控是完全密集的，会将样本路由到所有专家。随后，我们逐渐减小 k ，将样本路由到更少的专家，从而自适应地使结构更加稀疏，并持续提升计算效率。

通过对 k 值进行退火，我们以密集结构开始训练架构，使其能够充分学习所有专家，并在训练初期朝正确方向调整门控网络。因此，我们可以在训练过程中控制架构的稀疏性，这不仅能加快门控网络的收敛，还能让专家更好地学习特定类型的特征交互，发挥其专长。

3.3.3 稀疏分发器。稀疏分发器以样本的门控值和专家作为输入。它首先将样本分发给对应于非零门控值的专家，并让专家生成输出嵌入。稀疏混合专家层的输出 y 是由非零门控值对专家输出嵌入进行线性加权组合得到的。

$$y = \sum_{j \in \phi} ExpertG_j(x) E_j(x) \tag{12}$$

其中，ϕ表示选出的非零索引。我们利用 G（x）的稀疏性来减少计算量。对于 G（x）$_j = 0$，我们不会将该专家传递给对应的专家，也无需计算专家嵌入 E_j（x）。

3.3.4 负载分布正则化。如前文研究所述 [5, 7, 27, 40], ，如果不应用正则化，门控网络往往只会选择少数几个专家，尤其是在某些专家比其他专家更容易训练时。这种现象会自我强化，因为被选中的专家会得到更多训练，因此门控网络也会更频繁地选择它们。因此，我们应用负载均衡损失来促使专家路由均匀分布。

$$L_{\text{balance}} = \lambda \cdot N \cdot \sum_{j=1}^{N} f_j \cdot P_j \tag{13}$$

其中，N是专家数量，f_j是分配给专家 j 的样本占比，P_j是分配给专家 j 的路由器概率的平均值，λ是正则化项的系数。

$$f_j = \frac{1}{B} \sum_{x \in \mathcal{B}} \mathbf{1}\{\text{argmax}\, p(x) = j\} \tag{14}$$

$$P_j = \frac{1}{B} \sum_{x \in \mathcal{B}} p_j(x) \tag{15}$$

默认的负载均衡损失适用于专家类型相同的情况，且效果良好；但 AdaEnsemble 使用的是异构特征交互专家，每个专家的最优负载并不均匀。因此，我们应用以下负载分布正则化，以鼓励异构专家形成预期的负载分布。

$$L_{\text{distribution}} = \lambda \cdot \sum_{j=1}^{N} \frac{f_j \cdot P_j}{w_j} \tag{16}$$

其中，w_j是分派给专家 j 的样本的预期负载占比，自然地，$\sum_{j=1}^{N} w_j = 1$。实际上，λ应足够大，以防止在训练初期出现专家选择自我强化现象，同时又不能削弱主要的对数损失目标。

3.4 估计器层

稀疏混合专家层的输出是一个特征图，其中包含不同阶数和类型的特征交互，包括残差连接保留的原始输入特征图，以及由专家共同学习的高阶特征交互。对于最终预测，我们仅使用如下公式：

$$\hat{y} = \sigma(W_l X_l + b_l) \tag{17}$$

其中，σ是 Sigmoid 函数，$W_l \in R^{1 \times F}$是特征图聚合向量，用于对特征图中学习到的所有特征交互进行线性组合；$b \in R$是偏置。

3.5 深度选择控制器

3.5.1 深度选择网络。深度选择网络的配置与前文所述的 SparseMoE层噪声门控网络基本相同。我们将其记为 $DepthG$（x）。$DepthG(x)$的输出为 $[g_1^{depth}, g_2^{depth}, \cdots\ g_L^{depth}]$，表示每个样本的最优前向传播深度。第 l 个单元表示选择第 l 个混合专家层后退出的概率。最优深度会自动选为概率最大的单元所对应的深度。与专家选择不同，在动态推理中为每个样本选择最优深度时，我们仅保留深度选择网络输出单元中的排名第一的深度索引。请注意，我们还可以应用负载分布正则化，以鼓励样本的传播深度分布更加均衡。

3.5.2 动态传播机制。利用深度选择网络计算得到的深度门控 $g_l^{depth} \in [0, 1]$，我们可以获得每个样本的最优深度。如果 $g_l^{depth} = 0$，则递归地将样本前向传播通过混合专家层，并计算更深层的表示，直到 $g_l^{depth} = 1$ 或到达最终层。如果 $g_l^{depth} = 1$，则停止前向传播，并由相应的第l个估计器计算预测结果。为了高效处理最优传播深度各不相同的一批样本，我们使用算法 **1** 进行动态前向传播。

算法 1 动态传播

```
深度门控 ←深度选择网络(x)
ŷ ←动态传播(x, 深度门控                    , 深度=0)
返回 ŷ

函数动态传播(输入, 门控, 深度)
    输出 = MoE(输入)
    深度 += 1
    如果 深度 ==层数 则
        ŷ =估计器(输出)
    else
        g = 门控[:, 深度]
        输出keep,      输出exit =Dispatch(输出, g)
        门控keep, _ = Dispatch(门控, g)

        ŷkeep =DynamicPropagation(输出keep, 门控keep, 深度)
        ŷexit =估计器(输出exit)
        ŷ = 组合(ŷkeep, ŷexit)
    结束条件判断
    返回 ŷ
结束函数
```

3.6 训练

3.6.1 训练目标。我们使用的损失函数是对数损失与辅助负载分布正则化的线性加权组合，

$$Loss = L_{\text{LogLoss}} + \lambda_1 L_{\text{distribution}}^{\text{expert}} + \lambda_2 L_{\text{distribution}}^{\text{depth}} \tag{18}$$

其中，λ_1 和 λ_2 是用于加权负载分布正则化的系数。

3.6.2 双层优化。AdaEnsemble 的训练任务是联合优化参数 W和 α，其中W表示专家层和估计器层，α表示专家门控网络和深度选择网络。受 DARTS [17],的启发，我们采用双层优化算法训练模型，其中 α是上层参数，W是下层参数。我们应用算法2交替迭代地优化W和 α。

算法2 AdaEnsemble 的双层优化

输入：带有相应标签的训练样例，步长 t
输出：已充分学习的参数 $\mathbf{W}^*$ 和 α^*

当未收敛时 **执行**
  采样一个验证数据小批量
  通过下降 $\nabla_{\alpha} \mathrm{L}_{val}(\mathbf{W} - \xi \nabla_{\mathbf{W}} \mathrm{L}_{train}(\mathbf{W}\ \alpha)\ \alpha)$ 来更新 α
  ($\xi = 0$用于一阶近似)
  对于 $i \leftarrow 1,\ t$，**执行**
    采样一个训练数据小批量
    通过下降 $\nabla_{\mathbf{W}} \mathrm{L}_{train}(\mathbf{W}\ \alpha)$ 更新 $\mathbf{W}$
  结束循环
结束循环

3.7 AdaEnsemble 讨论

在每个 SparseMoE 层中采用稀疏专家路由，并结合深度选择控制器，为所提出的模型带来了两项优势。一方面，堆叠的 SparseMoE 层使所提出的模型能够利用稀疏门控专家的指数级组合，从而增强预测能力。另一方面，深度选择控制器使所提出的模型能够学习适用于不同实例的模型深度，从而提高模型服务期间的效率。下一节中，我们将通过一些实验研究来展示所提出模型的有效性。

4 实验

本节重点评估我们提出的模型的有效性，并探寻以下研究问题的答案：

- **Q1**：我们提出的 AdaEnsemble 在点击率预测问题上的表现与各个基线相比如何?
- **Q2**：SparseMoE层与使用所有特征交互专家的密集混合专家相比，表现如何？SparseMoE 层级联是否能够有效捕捉不同类型的特征交互?
- **Q3**：深度选择控制器与全深度网络相比，表现如何？早退机制能否兼顾效果与效率?
- **Q4**：不同的超参数设置会如何影响 AdaEnsemble 的表现?

4.1 实验设置

4.1.1 数据集。我们在三个广泛用于研究的公开真实世界数据集上评估所提出的模型。

1. Criteo。[1] Criteo数据集来自2014年Kaggle竞赛。此后，Criteo AI Lab正式发布了该数据集，供学术研究使用。该数据集包含13个数值特征和26个类别特征。我们通过转换函数 $\lfloor Log\,(V^2) \rfloor$ 将所有数值特征离散化为整数，并将其视为类别特征；Criteo竞赛的获胜团队也采用了这一做法。

2. Avazu。[2] Avazu 数据集来自 2015年 的 Kaggle 竞赛。Avazu 提供了 10 天的点击数据。我们总共使用 21 个特征进行建模。该数据集中的所有特征均为类别特征。

3.iPinYou。[3] 亿玛数据集来自 2013年 iPinYou 全球实时竞价（RTB）竞价算法竞赛。我们遵循 [38] 的数据处理步骤，并考虑全部 16 个类别特征。

对于所有数据集，我们将样本随机分为三部分：70% 用于训练，10% 用于验证，20% 用于测试。我们还会移除每个类别特征中出现次数少于 20 次的低频取值，以缓解稀疏性问题。请注意，我们旨在比较自动学习高阶特征交互的有效性和效率，因此不进行任何特征工程，只进行特征转换，例如数值特征分桶和类别特征频率阈值处理。

4.1.2 评估指标。我们使用曲线下面积和对数损失来评估模型性能。

对数损失对数损失既是我们的损失函数，也是评估指标。它衡量所有样本的预测概率与真实标签之间的平均距离。

曲线下面积ROC曲线下面积（曲线下面积）衡量模型将随机选取的正样本排在随机选取的负样本之前的概率。曲线下面积只考虑正样本与负样本之间的相对排序。曲线下面积越高，表示排序性能越好。

4.1.3 竞争模型。我们将 AdaEnsemble 与以下模型进行比较：逻辑回归（逻辑回归） [19, 20], 因子分解机（因子分解机） [24], 深度神经网络（多层感知机），Wide & Deep [4], Deep-Crossing [26],深度交叉网络（深度交叉网络） [31],产品神经网络（同时采用内积层和外积层） [22, 23],DeepFM [8],xDeepFM [16], AutoInt [28], FiBiNET [11], xDeepInt[35] 以及 DCN V2 [32]。其中一些模型是点击率预测问题中的最先进模型，并且在工业界得到了广泛应用。

4.1.4 可复现性。我们使用 Tensor- flow [1]实现所有模型。所有特征的小批量大小均为 4096，嵌入维度均为 16。在优化方面，对于所有神经网络模型，我们采用 Adam [13]，学习率在 10^{-4} 到 10^{-3} 之间调优；对于逻辑回归和因子分解机，我们采用 FTRL [19, 20]，学习率在 10^{-2} 到 10^{-1} 之间调优。在正则化方面，我们对全连接层采用 L2正则化，其 λ取值范围为 10^{-4} 到 10^{-3}。我们在验证数据集上对每个对比模型的超参数进行网格搜索。全连接层或交互层的数量为 1 到 4 层。神经元数量范围为

[1]https://www.kaggle.com/c/criteo-display-ad-challenge [2] https://www.kaggle.com/c/avazu-ctr-prediction [3] http://contest.ipinyou.com/

128 到 1024。所有模型均采用提前停止进行训练，并每经过 2000 个训练步骤评估一次。

AdaEnsemble 超参数搜索的设置如下ble：递归特征交互层的数量 l 在 1 到 4 之间搜索。对于每个 SparseMoE 层选取的专家数量 k，搜索范围为 1 到 3。对于专家门控网络和深度选择网络的缩减率，我们在 4 到 16 之间搜索。嵌入表使用 G-FTRL 优化器，模型权重使用 Adam。对于 AdaEnsemble，由于使用更多专家或层通常能获得更好的性能，因此如果使用较少专家或层的模型与多使用一个专家或层的模型相比，曲线下面积差异在 0.02% 以内，我们只报告使用较少专家或层的模型结果。

4.2 模型性能比较（Q1）

表1：Criteo、Avazu 和 iPinYou 数据集上不同算法的性能比较。

	Criteo		Avazu		iPinYou	
模型	AUC	对数损失	AUC	对数损失	AUC	对数损失
LR	0.7924	0.4577	0.7533	0.3952	0.7692	0.005605
FM	0.8030	0.4487	0.7652	0.3889	0.7737	0.005576
DNN	0.8051	0.4461	0.7627	0.3895	0.7732	0.005749
Wide&Deep	0.8062	0.4451	0.7637	0.3889	0.7763	0.005589
DeepFM	0.8069	0.4445	0.7665	0.3879	0.7749	0.005609
DeepCrossing	0.8068	0.4456	0.7628	0.3891	0.7706	0.005657
DCN	0.8056	0.4457	0.7661	0.3880	0.7758	0.005682
PNN	0.8083	0.4433	0.7663	0.3882	0.7783	0.005584
xDeepFM	0.8077	0.4439	0.7668	0.3878	0.7772	0.005664
AutoInt	0.8053	0.4462	0.7650	0.3883	0.7732	0.005758
FiBiNET	0.8082	0.4439	0.7652	0.3886	0.7756	0.005679
xDeepInt	0.8111	0.4408	0.7672	0.3876	0.7790	0.005567
DCN V2	0.8086	0.4433	0.7662	0.3882	0.7765	0.005593
AdaEnsemble	**0.8132**	**0.4394**	**0.7687**	**0.3865**	**0.7807**	**0.005550**

不同模型架构的整体性能如表1所示。从模型效果方面，我们有以下观察：

• 与逻辑回归基线相比，随着模型复杂度的增加，因子分解机带来了最显著的相对性能提升。这揭示了学习特征交互的重要性。

• 采用多个特征交互模块的模型通常比仅采用单个特征交互模块的模型表现更好，这表明联合学习特征交互表示的重要性。

• 最佳特征交互深度因特征交互模块类型而异，而且在与不同类型的模块结合时也有所不同，这表明有必要在不同交互深度上动态组合不同的特征交互。

• AdaEnsemble在所有模型中取得了最佳预测效果。我们模型的优越性能可能归因于：AdaEnsemble通过控制器自适应地选择特征交互专家组合，并确定最优特征交互深度，从而联合建模各种类型的特征交互。

4.3 特征交互专家选择分析（Q2）

我们比较了AdaEnsemble架构中密集混合专家层与SparseMoE层的模型性能和浮点运算次数。我们还纳入了不同多层单专家模型及其集成的性能。上述方法的性能均列于表2。我们还绘制了桑基图（图4），以说明各SparseMoE层的专家选择之间的依赖关系。流的颜色按照专家组合的出现频率进行聚类。根据以上观察，我们得出了以下认识：

• 总体而言，使用不同特征交互专家的模型比单专家模型表现更好。SparseMoE层在准确率与计算效率之间实现了更好的权衡。

• 在每个SparseMoE层中仅使用一个专家通常会损害模型性能，因为模型无法集成不同类型的特征交互。

• 在每个SparseMoE层中使用多个专家时，尽管只选择了部分特征交互专家，SparseMoE仍能有效捕获不同深度下最重要的特征交互，并保持与DenseMoE层相近的性能；但增加专家数量也会带来更高的计算成本。

• 图4显示，SparseMoE 层在不同层中动态采用不同的专家组合，从而有效捕捉复杂的特征交互。这也解释了为什么融合不同的特征交互对于预测准确率至关重要。

表2：SparseMoE与密集-MoE在Criteo数据集上的性能比较。

	AUC	对数损失	浮点运算次数
SparseMoE(k=1)	0.8096	0.4423	2.26M
SparseMoE(k=2)	0.8121	0.4400	4.14M
SparseMoE(k=3)	0.8132	0.4394	6.02M
SparseMoE(k=4)	0.8133	0.4393	7.09M
DenseMoE	0.8133	0.4392	9.78M
集成	0.8120	0.4401	12.15M
仅密集专家	0.8050	0.4463	3.71M
仅交叉专家	0.8086	0.4433	3.36M
仅多项式专家	0.8111	0.4408	3.32M
仅卷积神经网络专家	0.8022	0.4501	1.11M
仅多头自注意力专家	0.8051	0.4465	2.17M

4.4 深度选择分析（Q3）

我们比较了采用和未采用深度选择控制器的 AdaEnsemble 模型性能，以探究模型在深度选择方面是否实现了预测准确率与推理效率之间的平衡。不同类型的混合专家层及其集成结果列于表3。

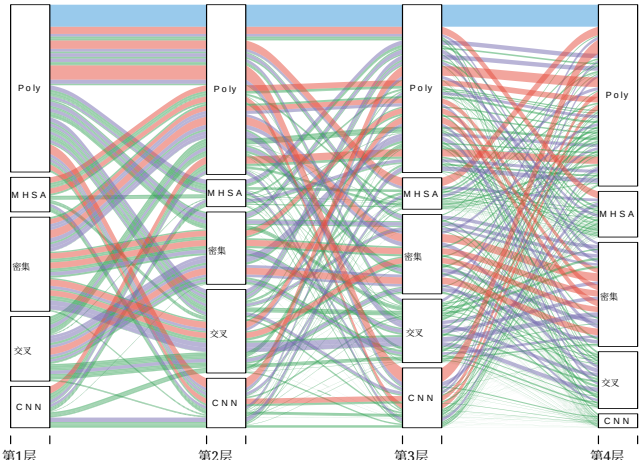


图4：用于说明每个 SparseMoE层专家选择依赖关系的桑基图

每条纵轴代表一个 SparseMoE层及其专家的使用比例。横向流线表示各 SparseMoE层之间专家选择的依赖关系和关联。专家组合的比例以流线宽度表示，并进一步用不同颜色进行区分。

引入深度选择控制器后，我们可以观察到，与全深度模型相比，我们的模型在取得略好准确率表现的同时，能够显著降低训练复杂度并提升推理效率（以浮点运算次数衡量）。我们认为，与 AdaEnsemble 相比，全深度模型更容易过拟合，因此准确率表现略差。带有深度选择控制器的 AdaEnsemble 会针对每个样本自适应地选择特征交互深度，从而在预测准确率与推理效率之间实现更好的权衡。每个样本的前向传播深度分布列于表4中。

表3：Criteo数据集上AdaEnsemble有无控制器的性能比较。

	曲线下面积	对数损失	浮点运算次数
使用控制器	0.8132	0.4394	6.02M
不使用控制器	0.8128	0.4396	8.58M

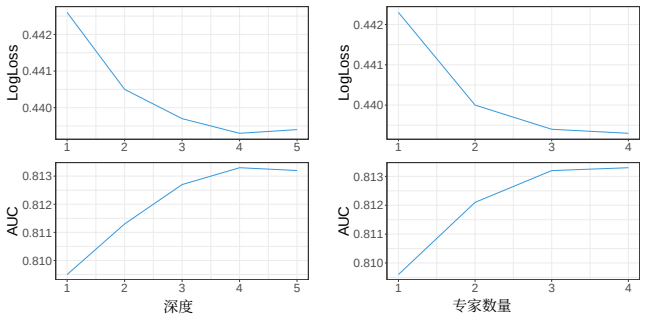
表4： AdaEnsemble 在 Criteo 数据集上的传播深度。集。

	第1层	第2层	第3层	第4层
占比	6.53%	19.36%	66.43%	7.68%

4.5 超参数研究（4）

为了深入了解所提出的模型，我们在 Criteo数据集上进行实验，并比较模型在不同超参数设置下的性能。本节评估模型性能随超参数变化的情况。

包括：1）SparseMoE 层的深度；2）SparseMoE 层中选定的专家数量；



（a）层深度 （b）专家数量

图 5：对数损失和曲线下面积随特征交互深度及专家数量的变化。

4.5.1 深度。 SparsMoE层的深度 l 决定了所学习的特征交互的最高阶数。在本实验中，我们将选定的专家数量 k 设为 3，这通常是 Criteo数据集上的一个不错选择。

图 5a 展示了 AdaEnsemble 在 Criteo 数据集上的性能随深度 l 变化的情况。我们观察到，随着深度增加，性能持续提升，直到深度达到 4。这与我们对性能随模型复杂度变化的理解一致。请注意，我们仍让控制器为每个样本确定交互深度；此处的深度用于控制最大深度和模型复杂度。

4.5.2 专家数量。SparsMoE 层中选取的专家数量 k 决定了每个 SparseMoE 层中选取的特征交互专家数量。在本实验中，我们将 AdaEnsemble l 的深度设为 4，这一设置在 Criteo 数据集上表现最佳。

图 5b 展示了 AdaEnsemble 在 Criteo 数据集上的性能随专家数量 k 变化的情况。我们观察到，性能持续提升，直到 k 等于 3。这表明，随着 SparseMoE 层中选取的专家数量增加，增益会逐渐减小。

5 结论

本文提出了一种新的点击率模型，该模型采用稀疏门控混合专家（SparseMoE）层次化架构，将不同的交互学习专家集成起来。我们还引入了深度选择控制器，为每个样本选择最优深度。基于这两种条件计算机制，我们的模型会为每个样本选择一部分专家和一个最优深度。该模型在不增加推理成本的情况下，使模型容量呈指数级扩展。全面的实验结果证明了我们方法的有效性和高效性。

在后续工作中，我们希望研究如何有效地将我们的方法扩展到用户行为序列。通过学习不同模型的稀疏集成，我们期望该方法能够针对用户行为序列数据中的不同行为，动态选择最优专家。

参考文献

[1] Martín Abadi、Paul Barham、Jianmin Chen、Zhifeng Chen、Andy Davis、Jeffrey Dean、Matthieu Devin、Sanjay Ghemawat、Geoffrey Irving、Michael Isard 等。2016。TensorFlow：一种用于大规模机器学习的系统。载于第 12 届 {USENIX} 操作系统设计与实现研讨会（{OSDI} 16）。265–283。[2] Yoshua Bengio、Nicholas Léonard 和 Aaron Courville。2013。针对条件计算，通过随机神经元估计或传播梯度。arXiv 预印本 arXiv:1308.3432（2013）。[3] Alex Beutel、Paul Covington、Sagar Jain、Can Xu、Jia Li、Vince Gatto 和 Ed H Chi。2018。潜在交叉：利用循环推荐系统中的上下文。载于第十一届 ACM 国际网络搜索与数据挖掘会议论文集。ACM，46–54。[4] Heng-Tze Cheng、Levent Koc、Jeremiah Harmsen、Tal Shaked、Tushar Chandra、Hrishi Aradhye、Glen Anderson、Greg Corrado、Wei Chai、Mustafa Ispir 等。2016。面向推荐系统的宽深学习。载于第一届推荐系统深度学习研讨会论文集。ACM，7–10。[5] Zewen Chi、Li Dong、Shaohan Huang、Damai Dai、Shuming Ma、Barun Patra、Sak- sham Singhal、Payal Bajaj、Xia Song 和 Furu Wei。2022。稀疏专家混合模型的表示坍塌。arXiv 预印本 arXiv:2204.09179（2022）。[6] Xin Dai、Xiangnan Kong 和 Tian Guo。2020。EPNet：通过灵活的多分支网络学习退出。载于第 29 届 ACM 信息与知识管理国际会议论文集。235–244。[7] William Fedus、Barret Zoph 和 Noam Shazeer。2021。Switch Transformer：通过简单高效的稀疏性扩展至万亿参数模型。[8] Huifeng Guo、Ruiming Tang、Yunming Ye、Zhenguo Li 和 Xiuqiang He。2017。DeepFM：一种用于点击率预测的、基于因子分解机的神经网络。arXiv 预印本 arXiv:1703.04247（2017）。[9] Xinran He、Junfeng Pan、Ou Jin、Tianbing Xu、Bo Liu、Tao Xu、Yanxin Shi、Antoine Atallah、Ralf Herbrich、Stuart Bowers 等。2014。Facebook 广告点击预测的实践经验。载于第八届在线广告数据挖掘国际研讨会论文集。ACM，1–9。[10] Jie Hu、Li Shen 和 Gang Sun。2018。挤压与激励网络。载于 IEEE 计算机视觉与模式识别会议论文集。7132–7141。[11] Tongwen Huang、Zhiqi Zhang 和 Junlin Zhang。2019。FiBiNET：结合特征重要性与双线性特征交互进行点击率预测。arXiv 预印本 arXiv:1905.09433（2019）。[12] Yigitcan Kaya、Sanghyun Hong 和 Tudor Dumitras。2019。浅深网络：理解并缓解网络过度思考。载于机器学习国际会议。PMLR，3301–3310。[13] Diederik P Kingma 和 Jimmy Ba。2014。Adam：一种随机优化方法。arXiv 预印本 arXiv:1412.6980（2014）。[14] Dmitry Lepikhin、HyoukJoong Lee、Yuanzhong Xu、Dehao Chen、Orhan Firat、Yanping Huang、Maxim Krikun、Noam Shazeer 和 Zhifeng Chen。2020。GShard：通过条件计算和自动分片扩展巨型模型。arXiv 预印本 arXiv:2006.16668（2020）。[15] Mike Lewis、Shruti Bhosale、Tim Dettmers、Naman Goyal 和 Luke Zettlemoyer。2021。BASE Layers：简化大型稀疏模型的训练。载于机器学习国际会议。PMLR，6265–6274。[16] Jianxun Lian、Xiaohuan Zhou、Fuzheng Zhang、Zhongxia Chen、Xing Xie 和 Guangzhong Sun。2018。xDeepFM：结合显式与隐式特征交互的推荐系统。载于第 24 届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集。ACM，1754–1763。[17] Hanxiao Liu、Karen Simonyan 和 Yiming Yang。2018。DARTS：可微架构搜索。arXiv 预印本 arXiv:1806.09055（2018）。[18] Weijie Liu、Peng Zhou、Zhe Zhao、Zhiruo Wang、Haotang Deng 和 Qi Ju。2020。FastBERT：具有自适应推理时间的自蒸馏 BERT。arXiv 预印本 arXiv:2004.02178（2020）。[19] H Brendan McMahan。2011。跟随正则化领导者与镜像下降：等价定理及 L1 正则化。（2011）。[20] H Brendan McMahan、Gary Holt、David Sculley、Michael Young、Dietmar Ebner、Julian Grady、Lan Nie、Todd Phillips、Eugene Davydov、Daniel Golovin 等。2013。广告点击预测：一线实践视角。载于第 19 届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集。ACM，1222–1230。[21] Xiaonan Nie、Shijie Cao、Xupeng Miao、Lingxiao Ma、Jilong Xue、Youshan Miao、Zichao Yang、Zhi Yang 和 Bin Cui。2021。用于混合专家模型的稠密到稀疏门控。arXiv 预印本 arXiv:2112.14397（2021）。[22] Yanru Qu、Han Cai、Kan Ren、Weinan Zhang、Yong Yu、Ying Wen 和 Jun Wang。2016。用于用户响应预测的基于乘积的神经网络。载于 2016 年 IEEE 第 16 届国际数据挖掘会议（ICDM）。IEEE，1149–1154。[23] Yanru Qu、Bohui Fang、Weinan Zhang、Ruiming Tang、Minzhe Niu、Huifeng Guo、Yong Yu 和 Xiuqiang He。2018。基于乘积的神经网络：面向多字段类别数据的用户响应预测。ACM 信息系统汇刊（TOIS）37，1（2018），5。

[24] Steffen Rendle。2010。因子分解机。载于*2010 IEEE*国际数据挖掘会议。IEEE，995–1000。[25] Matthew Richardson、Ewa Dominowska和Robert Ragno。2007。预测点击：估算新广告的点击率。载于第*16*届国际万维网会议论文集。ACM，521–530。[26] Ying Shan、T Ryan Hoens、Jian Jiao、Haijing Wang、Dong Yu和JC Mao。2016。Deep Crossing：无需手工构造组合特征的网络规模建模。载于第*22*届*ACM SIGKDD*知识发现与数据挖掘国际会议论文集。ACM，255–262。[27] Noam Shazeer、Azalia Mirhoseini、Krzysztof Maziarz、Andy Davis、Quoc Le、Geoffrey Hinton和Jeff Dean。2017。规模惊人的神经网络：稀疏门控混合专家层。*arXiv*预印本 *arXiv:1701.06538*（2017）。[28] Weiping Song、Chence Shi、Zhiping Xiao、Zhijian Duan、Yewen Xu、Ming Zhang和Jian Tang。2018。AutoInt：通过自注意力神经网络实现自动特征交互学习。*arXiv*预印本 *arXiv:1810.11921*（2018）。[29] Surat Teerapittayanon、Bradley McDanel和Hsiang-Tsung Kung。2016。Branchynet：通过深度神经网络提前退出实现快速推理。载于*2016*年第*23*届2464 2469国际模式识别会议（*ICPR*）。IEEE，–。[30] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser和Illia Polosukhin。2017。注意力就是你所需要的一切。载于神经信息处理系统进展。5998–6008。[31] Ruoxi Wang、Bin Fu、Gang Fu和Mingliang Wang。2017。用于广告点击预测的深度交叉网络。载于*ADKDD'17*论文集。ACM，12。[32] Ruoxi Wang、Rakesh Shivanna、Derek Cheng、Sagar Jain、Dong Lin、Lichan Hong和Ed Chi。2021。DCN V2：改进的深度交叉网络及面向网络规模学习排序系统的实践经验。载于*2021*年万维网会议论文集。1785–1797。[33] Ji Xin、Raphael Tang、Jaejun Lee、Yaoliang Yu和Jimmy Lin。2020。Dee-BERT：通过动态提前退出加速BERT推理。*arXiv*预印本 *arXiv:2004.12993*（2020）。[34] Ji Xin、Raphael Tang、Yaoliang Yu和Jimmy Lin。2021。BERxiT：通过更好的微调实现BERT提前退出，并将其扩展至回归任务。载于欧洲计算语言学协会分会第*16*届会议论文集：主会场。91–104。[35] Yachen Yan和Liubo Li。2020。xDeepInt：用于建模向量级与位级特征交互的混合架构。（2020）。[36] An Yang、Junyang Lin、Rui Men、Chang Zhou、Le Jiang、Xianyan Jia、Ang Wang、Jie Zhang、Jiamang Wang、Yong Li等。2021。M6-T：探索稀疏专家模型及其拓展。*arXiv*预印本 *arXiv:2105.15082*（2021）。[37] Shuai Zhang、Lina Yao、Aixin Sun和Yi Tay。2019。基于深度学习的推荐系统：综述与新视角。*ACM*计算综述（*CSUR*）52卷，1期（2019），第5篇。[38] Weinan Zhang、Shuai Yuan、Jun Wang和Xuehua Shen。2014。基于亿玛数据集的实时竞价基准测试。*arXiv*预印本 *arXiv:1407.7073*（2014）。[39] Wangchunshu Zhou、Canwen Xu、Tao Ge、Julian McAuley、Ke Xu和Furu Wei。2020。BERT失去耐心：利用提前退出实现快速、稳健的推理。神经信息处理系统进展33（2020），18330–18341。[40] Barret Zoph、Irwan Bello、Sameer Kumar、Nan Du、Yanping Huang、Jeff Dean、Noam Shazeer和William Fedus。2022。设计有效的稀疏专家模型。*arXiv*预印本 *arXiv:2202.08906*（2022）。