

具有快速收敛和

高效沟通



                    费晨蜜罗振华董∗∗

华为诺亚方舟实验室华为诺亚方舟实验室华为诺亚方舟实验室陈。邮箱：huawei.comrosemarluo@outlook.com电子邮箱：dongzhenhua@huawei.com

                                        李秀强何振国

华为诺亚方舟实验室华为诺亚方舟实验室李振国@华为网hexiuqiang1@huawei.com

# 摘要

跨移动设备分布式网络协作训练机器学习模型的统计和系统挑战一直是联合学习在现实世界中应用的瓶颈。在这项工作中，我们展示了元学习是处理这些问题的自然选择，并提出了一个联邦元学习框架FedMeta，其中一个参数化的算法（或metalearner）被共享，而不是以前方法中的全局模型。我们对LEAF数据集和实际生产数据集进行了广泛的实证评估，并证明FedMeta在更快收敛的情况下，所需的通信成本降低了2.82-4.33倍，与联邦学习中领先的优化算法FedAvg相比，精度提高了3.23%-14.84%。此外，FedMeta保护了用户隐私，因为只有参数化的算法在移动设备和中央服务器之间传输，而没有原始数据被收集到服务器上。

# 1        介绍

深度学习的成功在很大程度上依赖于大量的标记数据。在许多情况下，数据分布在不同的客户机之间，并且对隐私敏感，因此将原始数据收集到中央服务器上进行模型训练是不现实的。同时，随着移动设备的存储量和计算能力的增长，将机器学习模型的训练等计算从云计算转移到边缘设备越来越具有吸引力。这些问题激发了联合学习[14,2,8,10]，其目的是通过在中央服务器上维护共享模型并以分布式方式利用所有客户端的数据来协同训练模型。这种设置在不收集任何原始数据的情况下保留了用户隐私，而统计挑战和系统挑战则成为算法设计的重要问题。对于统计挑战，分散数据是非IID、高度个性化和异构的，这导致模型精度显著降低[27]。对于系统性的挑战，设备的数量通常比传统分布式设置中的数量级大。此外，在存储、计算和通信容量方面，每个设备可能具有重要的约束。为了解决这两个挑战，[14]提出了联邦平均（FedAvg）算法，该算法可以灵活地确定SGD局部训练的周期数和批量大小，从而达到高精度以及计算和通信开销之间的权衡。

基于初始化的元学习算法（如MAML[5]）以快速适应新任务和良好的泛化能力而闻名，这使得它特别适合于联邦设置



∗

同等贡献。这项工作是米洛在华为诺亚方舟实验室实习时完成的。

分散训练数据是非IID和高度个性化的。受此启发，我们提出了一个联合元学习框架，它与以前的联合学习有很大的不同。我们的工作将元学习方法和联合学习联系起来。在元学习中，一个参数化的算法（或元学习者）是通过元训练过程从大量的任务中缓慢地学习出来的，在每个任务中由该算法快速地训练出一个特定的模型。任务通常由彼此不相交的支持集和查询集组成。在支持集上训练一个任务特定的模型，然后在查询集上进行测试，测试结果用于更新算法。相比之下，在联邦元学习中，在服务器上维护一个算法，并分发给客户机进行模型训练。在每一集的元训练中，一批样本客户接收算法的参数并进行模型训练。然后将查询集上的测试结果上载到服务器以进行算法更新。我们框架的工作流程如图1所示。

联合元学习与联合学习的比较。除了在联邦学习模型和全局学习算法之间传输的是一个类似的全局学习模型。然而，我们注意到元学习在概念上不同于分布式模型训练，并且联邦元学习中的共享算法可以更灵活地应用于联邦学习中的共享模型。例如，在图像分类中，类别的图像可以不均匀地分布在客户端之间，其中每个客户端最多拥有*nk*分类与。联合学习需要培养大量*n*-way分类器利用来自所有客户端的数据，而a-way分类器足够了，因为它每次都为一个客户端进行预测。大型模型增加了通信和计算成本。可以只向客户端发送模型的一部分来更新相关参数，但这需要事先了解客户端的私有数据才能决定该部分。另一方面，在元学习中，该算法可以训练包含不同类别的任务。例如，模型不可知元学习（MAML）算法[5]将在联邦元学习部分详细描述，它可以通过元训练的途中任务来为a-way分类器提供初始化，而不考虑特定的类别。因此，在联邦元学习框架中，我们可以使用MAML来对所有类别进行元训练a-way分类器初始化。通过这种方式，联合元学习大大降低了通信和计算成本。*kkkkn*

贡献。我们专注于联邦设置的算法设计方面，为此我们提出了一个新的框架以及大量的实验结果。我们的贡献是三方面的。首先，我们证明了元学习是联邦环境下的自然选择，并提出了一个新的联邦元学习框架FedMeta，它将元学习算法与联合学习相结合。该框架允许以更灵活的方式共享参数化算法，同时保留客户机隐私，而不将数据收集到服务器上。我们将基于梯度的元学习算法MAML和元SGD集成到框架中进行说明。其次，我们在叶数据集上进行了实验，以比较FedMeta框架中包含的运行实例与基准FedAvg在精确度、计算成本和通信成本方面的差异。结果表明，FedMeta可以以较少或相当的系统开销获得较高的精度。第三，我们将FedMeta应用于每个客户都有高度个性化记录的工业推荐任务中，实验表明元学习算法比联合或独立推荐方法获得更高的推荐精度。

# 2        相关工作

基于初始化的元学习。在元学习中，目标是学习一组任务的模型，这样它就可以用少量的样本来解决新的任务[4]。作为元学习的一个有前途的方向，基于初始化的方法最近通过“学习到微调”来证明其有效性。在各种方法中，有些侧重于学习优化器，例如基于LSTM的元学习器[19]和具有外部存储器的元网络[15]。另一种方法是学习一个好的模型初始化[5,12,17,16]，这样在少量梯度下降后，模型在有限样本的新任务上具有最大的性能。上面提到的所有工作都只探索任务具有统一形式的设置（例如，用于图像分类的5路5拍）。在这项工作中，我们通过研究真实世界联邦数据集上的元学习算法来填补这一空白。我们将注意力集中在模型初始化方法上，其中的算法与模型和任务无关，并且可以作为任务和模型来部署

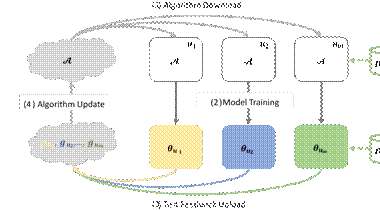


图1：联邦元学习框架的工作流程。

在联邦设置中有所不同。据我们所知，我们提出的框架是第一个从元学习的角度探索联邦环境的。

联合学习。为了解决狂热环境下的统计和系统挑战，人们提出了许多优化方法[14,20,13]，这些方法显著提高了收敛性能和系统开销。例如，[14]为每个客户机增加了更多的计算量，[20]在局部目标函数中加入了一个最近项，该函数对当前全局模型中的较大变化进行了惩罚。然而，上述工作都是为了学习一个跨分散数据的大型全局模型，这增加了通信和计算成本，限制了他们处理非IID数据和客户间异构结构的能力。相比之下，[23]考虑了联邦环境下的多任务学习，并提出了一种通信效率高的原始-对偶优化方法，该方法为每个客户机学习独立但相关的模型。然而，这种方法不适用于非凸的深度学习模型，因为强对偶性不再得到保证。

与[23]相似，我们提出的联邦元学习框架将每个客户端视为一个任务。我们的目标不是训练一个包含所有任务的全局模型，而是训练一个能够快速适应新任务的初始化良好的模型。元学习算法背后的直觉是提取和传播先前任务的内部可转移表示。因此，它们可以防止过度拟合并提高新任务的泛化能力，这显示了在处理联邦设置的统计和系统挑战方面的潜力。此外，我们考虑了模型无关的元学习算法，包括MAML、一阶MAML（FOMAMAL）和meta-SGD，因此我们的框架能够很好地处理非凸问题。

# 3        联合元学习

在这一部分中，我们详细介绍了联邦元学习框架。我们首先讨论了元学习方法，并给出了模型不可知元学习（MAML）[5]和MetaSGD[12]算法作为运行示例。然后我们描述了元学习算法是如何在联邦环境下实现的。

3.1元学习法

元学习的目标是元训练一个算法A，它可以快速地训练模型，例如深层神经网络，以适应新的任务。算法Aψ通常是参数化的，其中它的参数在元训练过程中使用任务集合进行更新。任务*ϕT*元训练由一个支持集和一个查询集组成，这两个集都包含标记的数据点。算法A训练模特*f*关于支持集与输出参数*θT*，我们称之为内部更新。然后在查询集上对模型进行评估，并计算一些测试损失LD（θT）来反映一个ψ的训练能力。最后，为了最小化测试损失，我们称之为外部更新。注意，支持集和查询集是不相交的算法1:FedMeta与MAML和Meta-SGD*fθTD问T夸脱*

1              //在服务器上运行

2              算法更新：

3              为MAML初始化，或为元SGD初始化（θ，α）。*θ*

4              每集=1,2，。。。做*t*

5              样品一套*美国犹他州*属于*米*客户，并分发*θ*（用于MAML）或(θ,α)（对于Meta-SGD）给抽样客户。

6              对于每个客户机∈Ut并行do*美国*

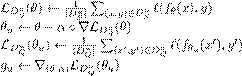
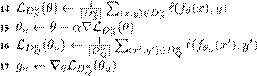
7得到测试损失←ModelTrainingMAML（θ）或*顾*

*顾*←模型训练metasgd（θ，α）

8              结束

9              更新MAML或

|  |  |
| --- | --- |
| 11 //在客户端上运行*美国* |  |
| 12    ModelTrainingMAML（）：*θ*  13    样本支持集和查询集*配电装置D问美国* | ModelTrainingMetaSGD（）：*θ,α*  样本支持集和查询集*配电装置DQu公司* |

元新加坡元。10结束

18 返回服务器*顾顾*

最大限度地提高了θ的泛化能力。元训练以一种情节式的方式进行，在每一集中，从一个元训练集中的任务分布T中抽取一批任务。因此，算法Aψ优化目标如下：

*.* （一）

MAML算法[5]是一种典型的基于梯度的元学习方法，用梯度更新步骤训练模型。MAML的算法A只是用来为模型提供初始化。特别是，对于每个任务，算法保持=θ作为模型参数的初始值。那么*Tϕffθ*对支持进行了更新*Tθ*在训练损失的情况下使用一个（或多个）梯度下降步骤，其中*`*是损失函数，例如图像分类任务的交叉熵。最后在查询集上进行了测试*fθTDQT*并计算了试验损耗。式（1）中的优化目标例示如下：

*,* （二）

其中是内部梯度更新的学习率。*α*

Meta-SGD[12]在MAML的基础上进一步学习了初始化和内部学习率。注意，测试损耗LD（θT）可以被视为和的函数，这两者都可以通过在LD（θT）上取梯度来使用SGD在外环中更新。此外，学习速率是一个与的维数相同的向量，这样对应于坐标方向。*θα夸脱θα夸脱αθαθ*

Meta-SGD的优化目标可以写为

*.* （三）

表1：所选数据集的统计数据。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 数据集 | 客户 | 样品 | 班级 | 每个客户的样本数 | | 每个客户端的类 | |
| 意思是 | 标准偏差 | 最小 | 最大值 |
| 女性主义者 | 1,068 | 235,683 | 62 | 220 | 90 | 9 | 62 |
| 莎士比亚 | 528 | 625,127 | 70 | 1183 | 1218 | 2 | 70 |
| Sent140型 | 3,790 | 171,809 | 2 | 45 | 28 | 1 | 2 |
| 生产数据集 | 9,369 | 6,430,120 | 2,400 | 686 | 374 | 2 | 36 |

3.2联邦元学习框架

在联邦学习[14]的设置下，训练数据分布在一组客户机之间，其中一个目标是协同训练一个模型，而不需要将数据收集到服务器上。模型在客户机上分布和训练，服务器通过平均从客户机收集的更新模型来维护共享模型。在许多实际应用中，例如为移动电话用户提供建议，该模型反过来又用于对同一组客户进行预测。

我们将元学习纳入联合学习框架。目标是使用分布在客户机之间的数据协作地元训练算法。以MAML为例，我们的目标是通过使用所有客户机的数据来训练模型的初始化。回想一下，MAML包含两个层次的优化：内部循环使用维护的初始化来训练特定于任务的模型，外循环则用任务的测试损失来更新初始化。在联邦设置中，每个客户机从服务器检索初始化，使用设备上的一组支持数据训练模型，并在一个单独的查询集中向服务器发送测试丢失LD（θ）。服务器维护初始化，并通过从小批量客户机收集测试损失来更新它。*美国θ配电装置问美国DQu公司*

在这个过程中传输的信息包括模型参数初始化（从服务器到客户端）和测试丢失（从客户端到服务器），不需要向服务器收集数据。对于Meta-SGD，矢量也作为算法参数的一部分进行传输，并用于内环模型训练。*α*

算法1说明了联邦元学习框架FedMeta与MAML和MetaSGD，其中通信轮对应于元学习术语中的情节。算法在AlgorithmUpdate过程中进行维护。在每一轮更新中，服务器在一组抽样的客户机上调用ModelTrainingMAML或ModelTrainingMeta SGD来收集测试损失。为了在元训练后将模型部署到客户机上，使用的训练集更新初始化，并使用获得的数据进行预测。*美国θ美国θu*

# 4        实验

在本节中，我们将评估FedMeta在不同任务、模型和实际联邦数据集上的经验性能。首先，我们在LEAF[3]数据集上进行了实验，结果表明FedMeta与传统的联邦学习方法相比，具有更快的收敛速度、更高的精度和更低的系统开销。其次，我们在一个更现实的场景——一个工业推荐任务中对FedMeta进行了评估，并证明了FedMeta可以在保持较高容量的同时保持较小规模的算法和模型。表1总结了所选数据集的统计数据。

4.1评价方案

在所有的实验中，我们随机选择80%的客户作为培训客户，10%的客户作为验证客户，剩下的作为测试客户，因为我们认为将新客户推广到新客户的能力是联合学习的一个重要属性。对于每个客户机，本地数据分为支持集和查询集。为了研究FedMeta如何有效地适应数据有限的新用户，我们改变了每个客户的支持集的数据部分。在本节剩下的部分中，我们用“支持”来表示这个设置。*pp*

表2：叶数据集的精度结果。对于女性主义者、莎士比亚和Sent140，模特们分别接受2000、400和400轮的训练。



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | 20%支持 | 50%支持 | 90%支持 |
| 女性主义者 | 费达夫  FedAvg（Meta）  费德梅塔（MAML） | 76.79%±0.45%83.58%±0.13%  88.46%±0.25% | 75.44%±0.73%87.84%±0.11%  89.77%±0.08% | 77.05%±1.43%88.76%±0.78%  89.31%±0.15% |
|  | FedMeta（元新加坡元） | 89.26%±0.12% | 90.28%±0.02% | 89.31%±0.09% |
| 莎士比亚 | 费达夫  FedAvg（Meta）  费德梅塔（MAML） | 40.76%±0.62%  38.71%±0.51%  46.06%±0.85% | 42.01%±0.43%  42.97%±0.97%  46.29%±0.84% | 40.58%±0.55%  43.48%±0.64%  46.49%±0.77% |
|  | FedMeta（元新加坡元） | 44.72%±0.72% | 45.24%±0.53% | 46.25%±0.63% |
| Sent140型 | 费达夫  FedAvg（Meta）  费德梅塔（MAML） | 71.53%±0.18%70.10%±0.66%  76.37%±0.06% | 72.29%±0.49%73.88%±0.06%  78.63%±0.19% | 73.38%±0.38%75.86%±0.46%  79.53%±0.25% |
|  | FedMeta（元新加坡元） | 77.24%±0.32% | 79.38%±0.09% | 80.94%±0.29% |



对于传统的联邦学习，我们考虑了联邦平均算法（FedAvg）[14]，它是一种基于平均局部随机梯度下降（SGD）更新的启发式优化方法，并且在非凸环境下具有良好的经验效果。为了公平比较，我们还实现了FedAvg的元学习版本，由FedAvg（meta）表示。与直观的FedAvg不同，FedAvg（Meta）利用测试客户端的支持集，在测试前对服务器接收到的模型初始化进行微调，体现了元学习的精髓——“学习微调”。在培训过程中，FedAvg和FedAvg（Meta）都使用培训客户机上的所有数据。

至于联邦元学习，我们包括三个面向优化的算法：MAML，MAML的一阶近似（用FOMAML表示）[5]和meta SGD[12]，所有这些都是模型无关的方法，可以在我们的FedMeta框架内很容易地实现。FOMAML是MAML的一个简化版本，其中省略了二阶导数，据报道，它具有与MAML相似的性能，同时导致大约33%的计算速度加快[5]。因此，在比较系统开销时，我们还考虑了FOMAML。关于实现的更多细节在附录中提供。

4.2叶数据集

我们首先研究LEAF[3]，它是联邦设置的基准。LEAF由三个数据集组成：（1）FEMNIST用于62类图像分类，它是流行的MNIST数据集的一个更复杂的版本[9]。根据数字/字符的写入者对数据进行分区。（2） 莎士比亚为下一个人物预测，这是建立在威廉莎士比亚全集[21]。每出戏中的每个演戏角色都被视为不同的客户。（3） 情感140[6]用于2级情感分类，它是根据tweet中的表情符号对tweet进行注释而自动生成的。每个twitter用户都被视为一个客户端。我们为FEMNIST使用CNN模型，为莎士比亚使用堆叠字符级LSTM模型，为Sent140使用LSTM分类器。我们过滤记录数少于记录的非活动客户机，FEMNIST、Shakespeare和Sent140分别设置为10、20和25。关于数据集和我们采用的模型的全部细节在附录中提供。*k*

精度和收敛性比较。我们研究了FedAvg和FedMeta框架在叶数据集上的性能。考虑到边缘设备的计算能力有限，我们将所有方法的本地epoch数设为1。

如图2所示，FedMeta框架内的所有方法都能以更快和更稳定的收敛性来提高最终精度。我们可以看到，在SGL上，和SENTAR140相比，Meta和SENTRANCE的收敛速度要明显好于MAMD。

表2显示了经过几轮沟通后，四种方法的最终精度。首先，通过比较不同的方法，我们发现FedAvg的性能明显低于fedmata，尤其是在图像分类任务上。相比之下，MAML和Meta-SGD达到最高

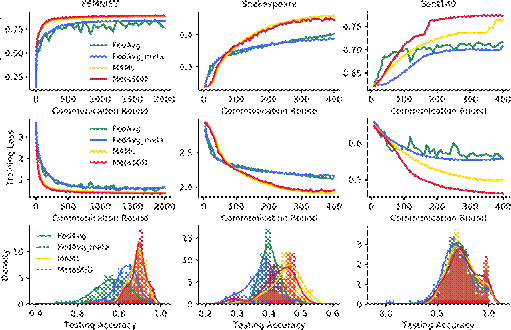


图2:FedAvg的叶数据集和FedMeta的三个运行示例的性能。所有数据集的支持率设置为20%。与直观的FedAvg相比，FedMeta框架内的所有运行示例都提供了更快的收敛速度和更高的精确度。

不同情况下的准确度，最终精度提高3.23%-14.84%。我们还发现，在大多数情况下，FedAvg（Meta）比FedAvg获得了更高的精确度。然而，两个特例是莎士比亚和Sent140，其中支持率为20%。出乎意料的是，FedAvg（Meta）导致精度略有下降。这可能是因为在对少量数据进行微调后，模型与全局最优值的偏差过大。其次，当我们增加支持度时，FedAvg（Meta）和FedMeta的准确度几乎在所有情况下都会提高。然而，FedAvg（Meta）的生长速率大于FedMeta。以莎士比亚为例，当支持度在20%-90%之间变化时，FedAvg（Meta）的准确度提高了4.77%，而MAML的准确度仅提高了0.43%。这表明FedMeta框架具有较好的泛化能力，能够有效地适应数据量有限的新客户。*p*

系统开销。我们用跨越所有设备的flop总数，以及从服务器上传和下载的总字节数来描述系统预算。图3显示了实现不同方法的目标测试集精度所需的系统开销。比较通信成本，我们发现FedMeta在所有情况下都能将所需通信成本降低2.82-4.33倍。在计算成本方面，由于显著的快速收敛，FOMAML为FEMNIST和Sent140提供了最低的计算成本。对于莎士比亚来说，FedAvg的成本最低，大约是MAML和Meta SGD的5倍。这是因为元学习方法在反向传播元梯度时使用二阶导数会产生很大的计算开销。与预期的一样，通过比较MAML和FOMAML，FOMAML降低了所有数据集的计算成本。对于两种语言的建模任务，与MAML相比，FOMAML降低了通信开销。但是对于图像分类任务，FOMAML增加了通信开销，这说明FOMAML中后向通路的下降对卷积网络的影响比LSTM大。一般来说，在实际应用中，我们可以灵活地选择不同的方法来权衡通信和计算开销。

公平比较。在最近的工作[11]之后，我们通过比较多次运行的最终精度分布来研究FedAvg和FedMeta框架的公平性。在图2的最后一行，我们展示了不同方法的核密度估计。对于女性主义者，我们观察到MAML和Meta-SGD不仅导致更高的平均精度，而且以较低的方差实现更集中的精度分布。对莎士比亚来说，尽管费德梅塔

                               女性主义者莎士比亚Sent140

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6  4  2  0 |  |  | | |  | | | | | | | | 20  10  0 |  |  | |  | | | | | | | | | | 30  20  10  0 |  | | | |  | | | | | | | | | | |
|  |  | |  | | | | | | | | | |
|  |  |  | |  |  | | | | | | |  |  | |  | | | | |  |  | |  |  |  |  | | |  |  | | | | |  |  | | | |
|  |  |  | |  |  | |  | |  | |  |  | |  |  | |  |  |  | |  |  | |  |  | |  |  | |  |  |
| 费达夫 | | | FedAvg（Meta） | | | FOMAML公司 | | 奶妈 | | 梅塔斯格德 | | 费达夫 | | FedAvg（Meta） | | FOMAML公司 | | | 奶妈 | | | 梅塔斯格德 | | | 费达夫 | | | FedAvg（Meta） | | | FOMAML公司 | | | 奶妈 | | | 梅塔斯格德 | | |
| 字节（1e10） | 6.15 | | | 5.37 | | | 1.5 | | 1.42 | | 1.44 | | 字节  （1e9） | 21.9 | | 23.83 | | 8.44 | | | 10.28 | | | 10.45 | | | 字节  （九） | 17.49 | | | 33.57 | | | 9.74 | | | 14.91 | | | 10.91 | | |
| 扑克牌（1e12） | 3.65 | | | 3.16 | | | 0.59 | | 0.72 | | 0.73 | | 扑克牌（1e13） | 2.03 | | 2.20 | | 5.61 | | | 11.31 | | | 11.48 | | | 扑克牌（1e11） | 8.52 | | | 16.39 | | | 7.85 | | | 17.70 | | | 13.10 | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

图3：在不同的方法女性主义者、莎士比亚和Sent140的目标准确率分别为74%、38%和70%。

表3：生产数据集的精度。在混合环境下，神经网络统一训练

训练前80000步（梯度）。在自设置中，LR和NN都在本地训练100到10000步。在元设置中，算法以100步进行局部训练。



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | | 80%支持 | | 5%支持 | |
| 前1名 | 前4名 | 前1名 | 前4名 |
| 混合的 | NN统一（918452参数） | | 76.72% | 89.13% | 66.47% | 79,88% |
| 自我 | 多功能单元  MRU公司  NB公司 | | 42.92%70.44%  百分之十八 | 81.49%81.43%  92.57% | 42.18%70.44%  59.16% | 72.87%  81.43%  72.83% |
| LR（4160参数） | 100步  10000步 | 58.30%  78.31% | 86.52%  93.70% | 52.53%  65.35% | 75.25%  77.11% |
| NN（9256参数） | 100步  10000步 | 57.20%  83.79% | 88.37%  94.56% | 49.89%  68.87% | 75.26%  77.66% |
| 元 | MAML+LR（8320参数）  元SGD+LR（12480个参数）  MAML+NN（18512参数） | | 47.69%81.70%  83.87% | 71.60%93.56%  94.88% | 46.75%  72.32%  73.08% | 66.26%77.94%  78.02% |
|  | 元SGD+NN（27768个参数） | | 86.23% | 96.46% | 72.98% | 78.17% |

方差越大，平均准确度也越高。对于Sent140，所有方法的最高峰值附近的精度分布几乎相同。然而，我们发现MAML和Meta SGD导致更多的客户的准确率在100%左右。总的来说，FedMeta鼓励在图像分类任务的各个设备之间进行更公平的精度分配。而对于语言建模任务，FedMeta仍然具有可比性，或者为了更高的平均精度而牺牲公平性。

4.3实际行业推荐任务

为了证明我们的FedMeta框架在实际应用中的有效性，我们还对一个工业推荐任务中的大型生产数据集进行了FedMeta评估。我们的目标是根据每个客户过去的记录，为他们推荐最好的移动服务。如表1所示，这个数据集中有2400个不同的服务，9369个客户端和大约640万条使用记录。除此之外，每个客户还有36到5000条记录。*k*

设置。我们把这个推荐任务看作一个分类问题，并考虑三个设置：META、MIXED和SELF。后两种算法被视为基线，因为我们希望包含一些经典的独立推荐算法来进行公平比较。（1） 元设置对应于联邦元学习方法，我们采用40个类分类器，而不是混合设置中采用的2420个类分类器。与联邦学习中的元学习模型相比，联邦学习允许小规模的元学习。该分类器考虑了两种结构：logistic回归和神经网络，分别用LR和NN表示。（2） 混合类型表示联邦学习方法，其中统一的2420类分类器首先在训练客户端上训练，然后使用相应的支持集对每个测试客户端进行微调。对于分类器，我们考虑一个隐层为64个神经元的神经网络，输出层由2420个神经元组成。我们用NN统一表示。我们避免使用深度神经网络，因为我们专注于研究元学习算法在训练推荐模型方面的优势，而不是寻找最佳模型。此外，在实践中，模型将在计算资源有限的用户设备上进行训练，其中简单模型更可取。（3） SELF设置表示一种独立的方法，在这种方法中，为每个客户机使用其本地数据训练不同的模型。我们选择了以下几种分类方法：最常用（MFU）、最近使用（MRU）、朴素贝叶斯（NB）和元设置中采用的两种架构（LR和NN）。关于输入特征向量构造和实现的完整细节可以在附录中找到。

精度比较。从表3可以看出：（1）在元设置下比较元学习算法，在其他模块和设置相同的情况下，meta-SGD优于MAML，NN优于LR。最简单的组合MAML+LR的性能最差，这意味着无论是算法还是模型都应该具有一定的复杂性，以保证FedMeta框架的性能。（2） 比较混合学习法、自学习法和元学习法，MAML+NN和META-SGD+NN的学习效果一般优于所有的基线，尽管这两种方法只训练了100个梯度步。另一个有趣的观察是，MRU在“5%支持率”的情况下具有最高的4个准确度。这可能是因为用户在短时间内使用了少量的服务，MRU不受低支持率的影响。然而，随着支持集的扩展（这在实践中经常发生），元学习方法将优于MRU。

收敛性比较。我们进一步研究了元设置和自设置的收敛性能。元学习方法训练的模型与使用（非参数）优化方法从头训练的模型相比，收敛速度更快，这意味着个性化模型的训练得益于良好的初始化。完整的实验结果见附录。

# 5        结论和今后的工作

在这项工作中，我们展示了元学习是处理联合学习中统计和系统挑战的自然选择，并提出了一个新的联邦元学习框架FedMeta。我们对一组联邦数据集的实证评估表明，FedMeta框架在准确性、收敛速度和通信成本方面取得了显著的改进。我们进一步验证了FedMeta在一个工业推荐场景中的有效性，其中FedMeta的性能优于独立模型和由联合学习方法训练的统一模型。

未来，我们将探讨以下几个方向：（1）我们希望从模型攻击的角度来研究FedMeta框架在保护用户隐私方面是否具有额外的优势[26,24,25,22]，因为当前联合学习方法共享的全局模型仍然隐含了所有用户的隐私，而在FedMeta中元学习者是共享的。（2） 我们将部署我们的FedMeta框架在线应用推荐，以评估其在线性能，涉及到许多工程工作尚未完成。

# 工具书类

[1] Martín Abadi、Paul Barham、Jianmin Chen、Zhifeng Chen、Andy Davis、Jeffrey Dean、Matthieu Devin、Sanjay Ghemawat、Geoffrey Irving、Michael Isard等。Tensorflow:大规模机器学习系统。第12页，第265-283页，2016年。*乌塞尼克斯操作系统设计与实现专题讨论会(奥斯迪16条）*

[2] Keith Bonawitz、Hubert Eichner、Wolfgang Grieskamp、Dzmitry Huba、Alex Ingerman、Vladimir Ivanov、Chloe Kiddon、Jakub Konecny、Stefano Mazzocchi、H Brendan McMahan等。面向大规模联合学习：系统设计。arXiv预印本arXiv:1902.010462019年。

[3] 塞巴斯蒂安·卡尔达斯、彼得·吴、田莉、雅库布·科尼肯、H·布伦丹·麦克马汉、弗吉尼亚·史密斯和阿梅特·塔尔沃卡尔。Leaf：联邦设置的基准。arXiv预印本arXiv:1812.010972018年。

[4] 陈伟宇，刘延成，奇拉，王玉强，黄佳斌。对少数镜头分类的进一步研究。arXiv预印本arXiv:1904.042322019年。

[5] 切尔西·芬恩，彼得·阿贝尔和谢尔盖·莱文。模型不可知元学习快速适应深层网络。2017-1125页，共1135页。

[6] Alec Go，Richa Bhayani和Lei Huang。基于远程监控的Twitter情感分类。CS224N项目报告，斯坦福，1（12）：2009，2009。

[7] 迪德里克·金马和吉米·巴。亚当：一种随机优化方法。arXiv预印本阿拉十四世：1412.69802014年。

[8] Jakub Konecnˇy、H Brendan McMahan、Felix X Yu、Peter Richtárik、Ananda Theertha Suresh和Dave Bacon。联合学习：提高沟通效率的策略。arXiv预印本arXiv:1610.054922016年。

[9] 杨乐坤。手写数字的mnist数据库。http://yann。乐昆。com/exdb/mnist/，1998年。

[10] 田莉，阿尼特·库马尔·萨胡，阿梅特·塔尔瓦卡和维吉尼亚·史密斯。联合学习：挑战、方法和未来方向。arXiv预印本arXiv:1908.078732019年。

[11] 田莉、马齐亚尔·桑贾比和维吉尼亚·史密斯。公平学习资源分配。arXiv预印本arXiv:1905.104972019年。

[12] 李振国，周凤伟，陈飞，李杭。元SGD：学习快速学习为少数镜头学习。arXiv预印本arXiv:1707.098352017年。

[13] 陶林，塞巴斯蒂安·斯蒂奇和马丁·贾吉。不要使用大的小批量，使用当地的sgd。arXiv预印本arXiv:1808.072172018年。

[14] 布伦丹·麦克马汉、艾德·摩尔、丹尼尔·拉马奇、塞思·汉普森和布莱斯·阿奎拉·阿卡斯。从分散数据中有效地学习深层网络。AISTATS，第1273-1282页，2017年。

[15] 曾素伦·蒙克达赖和洪宇。元网络。第34届机器学习国际会议论文集第70卷，第2554-2563页。吉姆勒。org，2017年。

[16] 亚历克斯·尼克尔和约翰·舒尔曼。爬行动物：一种可伸缩的金属学习算法。arXiv预印本arXiv:1803.029992018年2月2日。

[17] 亚历克斯·尼科尔、约书亚·阿奇姆和约翰·舒尔曼。关于一阶元学习算法。arXiv预印本arXiv:1803.029992018年。

[18] 杰弗里·彭宁顿、理查德·索彻和克里斯托弗·曼宁。glow：单词表示的全局向量。《2014年自然语言处理经验方法会议论文集》，第1532-1543页，2014年。

[19] 萨钦·拉维和雨果·拉罗谢尔。优化作为少镜头学习的模型。2017年ICLR。

[20] 阿尼特·库马尔·萨胡、田莉、马齐亚尔·桑贾比、曼齐尔·扎希尔、阿梅特·塔尔沃卡尔和维吉尼亚·史密斯。关于异构网络中联邦优化的收敛性。arXiv预印本arXiv：1812.061272018年。

[21]威廉·莎士比亚。莎士比亚全集。华兹华斯版，2007年。[22]Reza Shokri，Marco Stronati，Congzheng Song和Vitaly Shmatikov。针对机器学习模型的成员推断攻击。2017年IEEE安全与隐私研讨会（SP），第3-18页。IEEE，2017年。

[23]维吉尼亚·史密斯、蒋超楷、马齐亚尔·桑贾比和阿梅特·S·塔尔瓦卡。联合多任务学习。神经信息处理系统的进展，第4424-4434页，2017年。

[24]宋聪正，托马斯·里斯滕帕特，维塔利·什马蒂科夫。记忆太多的机器学习模型。2017年ACM SIGSAC计算机和通信安全会议记录，第587-601页。ACM，2017年。

[25]克里斯蒂安·塞格迪、沃伊切赫·扎勒巴、伊利亚·萨茨克弗、琼·布鲁纳、杜米特鲁·厄汉、伊恩·古德费罗和罗布·弗格斯。神经网络有趣的特性。arXiv预印本arXiv：1312.61992013年。

[26]Florian Tramèr、Fan Zhang、Ari Juels、Michael K Reiter和Thomas Ristenpart。通过预测API窃取机器学习模型。第25页，第601-618页，2016年。*乌塞尼克斯安全专题讨论会(乌塞尼克斯安全16）*

[27]赵悦，李萌，赖良珍，娜文素达，达蒙·茨文，维卡斯·钱德拉。非iid数据的联合学习。arXiv预印本arXiv:1806.005822018年。

# 实验细节

A、 1数据集和模型

FEMNIST：我们在FEMNIST上研究了一个62类的图像分类任务，它是流行的MNIST数据集的一个更复杂的版本[9]。根据数字/字符的写入者对数据进行分区。我们考虑具有两个5x5卷积层的CNN（第一个有32个通道，第二个有64个通道，每个通道后面都有2×2个最大池），一个有2048个单元和ReLU激活的完全连接层，以及一个最终的softmax输出层。CNN模型的输入是一个28×28的平坦图像，输出是一个介于0和61之间的数字。

莎士比亚：我们研究下一个关于莎士比亚的人物预测任务，它是根据

*莎士比亚全集*[21]。在数据集中，每个剧本中的每个角色都被视为不同的客户。这个语言建模任务可以被建模为一个53类分类问题。我们使用一个包含256个隐藏单元和一个8D嵌入层的两层LSTM分类器。嵌入层以80个字符的序列作为输入，输出是一个介于0到52之间的类标签。

我们在140个句子的基础上自动生成了140个句子表情。我们使用两个预埋手套和100M的隐藏层进行分类。输入是一个25个单词的序列，每个单词随后通过查找glow嵌入到300维空间中。最后一个密集连接层的输出是0或1的类标签。

生产数据集：我们研究了一个基于生产数据集的行业推荐任务。有2400个不同的服务和9369个客户机，其中每个客户机有100到5000个记录和2到36个服务。在每个使用记录中，标签是用户已经使用过的服务，该特征包含服务特征（如服务ID等）、用户特征（如上次使用的服务等）和上下文特征（如电池电量、时间等）。我们用三种模型结构进行了实验：NN-unified、NN和LR。实验部分提供了这些模型的详细信息。

A、 2实施细则

库：我们在TensorFlow[1]中实现了FedAvg、FedAvg（Meta）、FedMeta和MAML、FOMAML和MetaSGD，允许在元学习过程中通过梯度更新自动区分。我们使用Adam[7]作为所有方法的局部优化器。

评估在联邦设置中定义测试精度有两种常见方法，即相对于所有数据点的准确性和针对所有设备的准确性。在这部作品中，我们选择前者。对于采样方案，在服务器端更新算法时，我们在每个通信轮对客户端进行统一采样，并对局部模型进行加权平均，权重与本地数据点的个数成正比。为了公平比较，客户机的划分和支持/查询集的划分对于所有方法都是相同的。

为每个活跃客户机调整每轮叶参数。对于FEMNIST、Shakepseare和Sent140，活跃客户数量分别为4、50和60。我们还对学习率进行了网格搜索，表4提供了最好的学习率。

Table 4: Learning rates setup for LEAF experiments.

|  |  |  |
| --- | --- | --- |
|  | FedAvg | FedMeta(, )*αβ* |
| FEMNIST | 10e-5 | (0.001,0.0001) |
| Shakespeare | 0.001 | (0.1, 0.01) |
| Sent140 | 0.001 | (0.001, 0.0001) |

A.3 Additional Experiments on LEAF

As shown in Figure 4 and Figure 5, we provide the convergences curves when the support fractions are 50% and 90%. We notice that the gap between FedAvg(Meta) and FedMeta is shrinking when we increase the support fraction. An example is that the convergence curves of FedAvg(Meta) and FedMeta almost coincide when the support fraction is 90%. This is consistent with previous industrial experience that meta-learning methods do have a greater advantage in the low-data regime than in medium- or large- data regime.

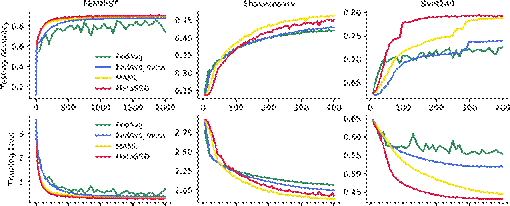
A.4 Additional Experiments on Industrial Task

Input Feature Vector Construction: The input feature vectors are constructed differently for different approaches. For both META and SELF, user-specific models are trained, and the feature vector encodes each usage record with dimension 103. For MIXED, a unified model is trained across all users. To improve the prediction accuracy, the feature vector further encodes user ID and service ID, which has dimension 11892.

Setting: We randomly sample 7000 users as training users, some of them are used in pre-training (for baselines) or meta-training (for meta-learning methods), and the remaining ones as testing users. For each testing user, the last 20% records in chronological order are used as query set, and for “Support” case, the fraction of records are used as support set. In each step in MIXED setting, a batch of 500 data points (or half of the user records, whichever is smaller) are sampled from a single user. In the SELF setting, the batch size is 500 or half size of the query set. In the META setting, the algorithms are trained with 20000 episodes, each of them consists one task from a single training user during meta-training. In each meta-training task, 500 data points (or half of the user records) are sampled, among which the first 80% are used as support set and the remaining as query set.*p p*

Convergence Comparison: We further study the convergence of meta-learning methods in terms of the number of episodes during meta-training. as shown in Figure 6. The performance of MAML + LR surprisingly becomes worse as meta-training proceeds, especially for the Top 4 recommendation. The other three methods converge within 20000 episodes, while Meta-SGD converges faster than MAML. In particular, Meta-SGD + NN already outperforms the best baseline NN after 4000 episodes.

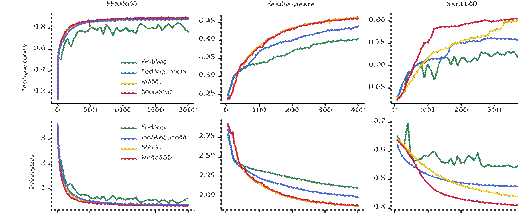
In Table 3, the SELF baselines — LR and NN perform significantly better when the number of training gradient steps increases from 100 to 10000. It remains to see how many steps are sufficient to train the models. Figure 7 shows that LR and NN indeed converge in 10000 steps, where the accuracy of LR is below or marginally above Meta-SGD + LR, while the accuracy of NN is below Meta-SGD + NN. We stress that Meta-SGD trains the models with only 100 steps, which is much more efficient than training the models from scratch by using (non-parametric) optimization algorithms.



0 500 1000 1500 2000 0 100 200 300 400 0 100 200 300 400

                         Communication Round Communication Round Communication Round

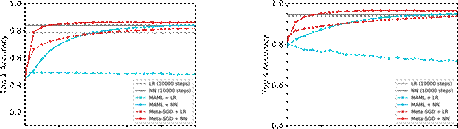
Figure 4: Performance on LEAF datasets for FedAvg and three running examples of FedMeta. The support fraction setting for all datasets is 50%



0 500 1000 1500 2000 0 100 200 300 400 0 100 200 300 400

                         Communication Round Communication Round Communication Round

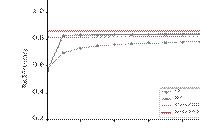
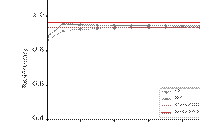
Figure 5: Performance on LEAF datasets for FedAvg and three running examples of FedMeta. The support fraction setting for all datasets is 90%



                                 0 4000 8000 12000 16000 20000                                                                                  0 4000 8000 12000 16000 20000

                                            Episode Episode

Figure 6: Convergence of Top 1 (left) and Top 4 (right) accuracies of meta-learning methods (“80% Support” case).

100 2000 4000 6000 8000 10000 100 2000 4000 6000 8000 10000

                                             Step Step

Figure 7: Convergence of Top 1 (left) and Top 4 (right) accuracies of SELF baselines LR and NN (“80% Support” case).