联合学习：提高沟通效率的策略

Jakub Konecnˇy´，H.Brendan McMahan，Felix X.Yu，Ananda Theertha Suresh&Dave Bacon谷歌[[1]](" \l "_ftn1" \o ")

{康基，麦克马汉，费利索，瑟塔，达巴科}@谷歌

彼得·里奇塔里克

沙特阿拉伯图瓦尔阿卜杜拉国王科技大学

苏格兰爱丁堡爱丁堡大学

彼得。里奇塔里克@考斯特教育大学

摘要

联合学习是一种机器学习设置，目标是训练一个高质量的集中式模型，同时训练数据仍然分布在大量客户端上，每个客户端都有不可靠和相对较慢的网络连接。我们考虑这种设置的学习算法，在每一轮中，每个客户机根据其本地数据独立计算对当前模型的更新，并将此更新传递到中央服务器，在中央服务器上，客户端更新被聚合以计算新的全局模型。在这种情况下，典型的客户是移动电话，通信效率是最重要的。

在本文中，我们提出两种降低上行链路通信成本的方法：结构化更新，即直接从使用较少变量（如低秩或随机掩码）参数化的受限空间中学习更新；以及草图更新，在那里我们学习完整的模型更新，然后在发送到服务器之前使用量化、随机旋转和子采样的组合对其进行压缩。在卷积网络和递归网络上的实验表明，所提出的方法可以将通信开销降低两个数量级。

# 1         简介

随着数据集越来越大，模型越来越复杂，训练机器学习模型越来越需要将模型参数的优化分布到多台机器上。现有的机器学习算法是为高度控制的环境（如数据中心）而设计的，在这些环境中，数据以均衡的i.i.d.方式分布在机器之间，并且可以使用高吞吐量网络。

最近，联合学习（和相关的分散方法）（McMahan&Ramage，2017；Konecnˇy et al.，2016；McMahan et al.，2017；Shokri&Shmatikov，2015）被提议作为“另一种设置：在中央服务器的协调下，从参与设备的联盟中训练共享的全球模型。参与的设备（客户端）通常数量庞大，并且具有缓慢或不稳定的互联网连接。当培训数据来自于用户与移动应用程序的交互时，联合学习的一个主要激励示例就出现了。联合学习使移动电话能够协作学习一个共享的预测模型，同时将所有的训练数据保存在设备上，将机器学习的能力与将数据存储在云中的需求分离开来。训练数据被本地保存在用户的移动设备上，这些设备作为节点对其本地数据进行计算以更新全局模型。这超越了使用本地模型在移动设备上进行预测，同时也为设备提供了模型训练。上述框架不同于传统的分布式机器学习（Reddi et al.，2016；Ma et al.，2017；Shamir et al.，2014；Zhang&Lin，2015；Dean et al.，2012；Chilimbi et al.，2014），原因是客户端数量非常多，每个客户端上可用的非i.i.d.数据非常不平衡，网络连接相对较差。联合学习是一个不可靠的挑战，因为在最后一项实践工作中，不对称的学习是不可靠的。

为了简单起见，我们考虑联邦学习的同步算法，其中典型的一轮包括以下步骤：

1.    选择现有客户机的子集，每个客户机都下载当前模型。

2.    子集中的每个客户机根据其本地数据计算更新的模型。

3.    模型更新从选定的客户端发送到服务器。

4.    服务器聚合这些模型（通常通过平均）来构建一个改进的全局模型。

上述框架的简单实现需要每个客户端在每轮发送一个完整的模型（或完整的模型更新）到服务器。对于大型模型，由于多种因素，这一步很可能成为联合学习的瓶颈。一个因素是互联网连接速度的不对称性：上行链路通常比下行链路慢得多。美国的平均宽带速度为55.0Mbps下载，而18.9Mbps的上传速度，一些互联网服务提供商的不对称程度明显更高，例如，Xfinity的下载速率为125Mbps，而上行速率为15Mbps(速度测试.net，2016年）。此外，现有的模型压缩方案，如Han等人。（2015）可以减少下载当前模型所需的带宽，并制定加密协议，以确保在平均数百或数千次其他更新之前无法检查单个客户端的更新（Bonawitz等人，2017），进一步增加了需要上传的比特数。

因此，研究降低上行链路通信成本的方法非常重要。在本文中，我们研究两种一般方法：

•，我们直接从受限空间学习更新，可以使用较少的变量参数化。*结构化更新*

•，在那里我们学习完整的模型更新，然后在发送到服务器之前压缩它。*草图更新*

这些方法，在第2节和第3节中有详细的解释，可以结合起来，例如，首先学习一个结构化的更新，然后画出草图；不过，我们在这篇文章中并没有尝试这种组合。在下面，我们正式描述这个问题。联合学习的目标是从存储在大量客户机上的数据中学习包含在实矩阵中的参数的模型。我们首先提供了联邦学习的通信天真版本。在round≥0时，服务器将当前模型分发给一个子集的客户机。这些客户机根据其本地数据独立更新模型。让更新的本地模型**世界卫生组织**∈*d*1×天[[3]](" \l "_ftn3" \o ") *ttSt新台币*，所以客户端的更新可以写成：=Wti−Wt，对于∈St。*我我t我*

这些更新可以是在客户端上计算的单个梯度，但通常是更复杂计算的结果，例如，在客户端的本地数据集上执行多个随机梯度下降（SGD）步骤。在任何情况下，每个选定的客户端都会将更新发送回服务器，在服务器上，全局更新是通过聚合2所有客户端更新来计算的：

**什么***t*+1 =W+η*ttt,* *.*

服务器选择学习速率。为了简单起见，我们选择=1。*ηtηt*

在第4节中，我们描述了神经网络的联合学习，其中我们使用一个单独的二维矩阵来表示每个层的参数。我们假设它是右相乘的，即分别表示输出和输入维。注意，完全连通层的参数自然地表示为2D矩阵。然而，卷积层的核心是形状#输入×宽度×高度×输出的4D张量。在这种情况下，将从内核整形为形状（#输入×宽度×高度）×输出。**W W W W***d*1 *d*2

概述和总结。通过网络传输大量数据的有效性是提高学习效率的目标，而通过网络传输数据的有效性是为了降低网络学习的效率。我们提出了两类通用的方法，结构化更新和草图更新。在实验部分，我们评估了这些方法在训练深层神经网络中的效果。**H***我t*

在CIFAR数据的模拟实验中，我们研究了这些技术对联邦平均算法收敛性的影响（McMahan等人，2017）。在收敛速度略有下降的情况下，我们能够将通信的数据总量减少两个数量级。这使得我们可以使用全卷积模型获得良好的预测精度，同时总的来说，所传递的信息少于原始CIFAR数据的大小。在一个对用户分区文本数据进行的更大的实际实验中，我们表明我们能够有效地训练递归神经网络进行下一个单词的预测，甚至在使用每个用户的数据之前。最后，我们得到了包含最优结构的预处理结果。这一步的实用性对我们的设置是独一无二的，因为在典型的SGD并行实现中，应用随机旋转的成本将占主导地位，但与联邦学习中的局部训练相比，可以忽略不计。

# 2         结构化更新

第一种有效通信的更新限制更新具有预先指定的结构。本文考虑了两类结构：低秩结构和随机掩模结构。重要的是要强调的是，我们直接训练这种结构的更新，而不是用一个特定结构的对象来近似/描绘一般的更新——这在第3节中讨论过。**H***我t*

低级的。我们强制执行对本地模型的每次更新**HH公司**至多是一个低秩矩阵，其中是一个固定数。为此，我们将其表示为两个矩阵的乘积：*kk我t*

**哈巴**，其中. 在随后的计算中，我们随机生成并在局部训练过程中考虑一个常数，我们只进行优化*我t我t*. 注意，在实际实现中，可以在这种情况下压缩成随机种子的形式，而客户端只需要发送经过训练的*我t*到服务器。这样的方法立即节省了沟通的一个因素。我们在每一轮重新生成矩阵，并为每个客户独立生成。*d*1*/k我t*

我们还尝试了修复和训练，以及同时训练和；两者都没有表现好。我们的方法似乎表现得和Denil等人认为的最佳技术一样好。（2013），不需要任何手工制作的功能。对这一观察结果的直观解释如下。我们可以解释为投影矩阵和重建矩阵。修复和优化for类似于问“给定一个给定的随机重建，什么样的投影能恢复大部分信息？”？”. 在这种情况下，如果重建是满秩，则存在恢复由顶特征向量跨越的空间的投影。然而，如果我们随机地修复投影并搜索重建，我们可能是不幸的，并且重要的子空间可能已经被投影出去，这意味着没有重建将尽可能好，或将很难找到。**巴巴布***我t我t我t我t我t我t我t我tk*

随机蒙版。我们将更新限制为稀疏矩阵，遵循预定义的随机稀疏模式（即随机掩码）。在每一轮中为每个客户机独立地重新生成模式。与低秩方法类似，稀疏模式可以由随机种子完全指定，因此只需要将非零项的值与种子一起发送。**HH公司***我t我t*

# 3         草图更新

第二种解决通信开销的更新，我们称之为sketched，它首先在不受任何约束的本地训练期间计算完整的更新，然后在发送到服务器之前以（有损）压缩形式近似或编码更新。服务器在进行聚合之前对更新进行解码。这种草图绘制方法在许多领域都有应用（Woodruff，2014）。为了完成草图绘制，我们使用了多种工具进行试验，这些工具相互兼容，可以联合使用：**H***我t*

二次抽样。不是发送，每个客户端只通信矩阵，它是由的（缩放）值的随机子集形成的。然后，服务器计算子采样更新的平均值，生成全局更新。这可以使得采样更新的平均值是真实平均值的无偏估计量：E[H]=Ht。与随机掩码结构化更新类似，掩码在每轮中对每个客户端独立随机，并且掩码本身可以存储为同步种子。**HHH***我t*ˆ*我t我t*ˆ*t*ˆ*t*

概率量化。压缩更新的另一种方法是量化权重。

我们首先描述了将每个标量量化为一位的算法。考虑一下更新，let=（h1，…，hd）=vec（Hit），let=maxj（hj），=minj（hj）。的压缩更新由表示，生成如下：**H***我th*1×天2*h*最大值*h*最小*hh*˜

(

*h*●j=h最大值*,* 有可能*.*

*h*最小*,* 有可能

很容易证明它是的无偏估计量。与4字节浮点相比，此方法提供32倍的压缩。例如，Suresh等人分析了这种压缩方案产生的误差。（2017），是Konecnˇy&Richt´arik´（2016）提出的协议的一个特例。*h*˜*h*

我们还可以将上述结果推广到每个标量的1位以上。对于位量化，我们首先将[hmin，hmax]平均划分为2b个区间。假设落在以和为界的区间内。量子化通过将上述方程的和分别替换为和来操作。参数允许简单的方法来平衡精度和通信成本。*b你好h*0 *h*00*h*最小*h*最大值*h*0 *h*00*b*

Alistarh等人最近提出了另一种量化方法，其动机是在平均向量的同时减少通信量。（2016年）。增量、随机和分布式优化算法可以在量化更新设置中进行类似的分析（Rabbat&Nowak，2005；Golovin等人，2013；Gamal&Lai，2016）。

通过结构化随机旋转改进量化。当尺度在不同维度上近似相等时，上述1位和多比特量化方法效果最好。

例如，当max=1和min=-1且大多数值为0时，1位量化将导致较大误差。我们注意到在量化（乘以随机正交矩阵）之前应用随机旋转可以解决这个问题。Suresh等人从理论上支持了这一主张。（2017年）。在这项工作中，说明了结构化随机旋转可以将量化误差降低一个因子O（d/logd），其中的维数是。我们将在下一节中展示它的实用性。*hhdh*

在解码阶段，服务器需要在聚合所有更新之前执行反向旋转。注意，在实践中，的维数很容易高达=106或更高，并且在计算上禁止生成（O（d3））和应用（O（d2））通用旋转矩阵。与Suresh等人相同。（2017），我们使用一种结构化旋转矩阵，它是Walsh-Hadamard矩阵和二进制对角矩阵的乘积。这降低了生成矩阵并将其应用于O（d）和O（dlogd）的计算复杂度，与联邦学习中的局部训练相比，这是微不足道的。*hd*

# 4         实验

我们用联合学习来训练两个不同任务的深层神经网络。首先，我们使用卷积网络和人工分割数据集对CIFAR-10图像分类任务（Krizhevsky，2009）进行了实验，并详细探讨了我们提出的算法的性质。第二，我们使用更现实的联合学习场景——公共Reddit post数据（googlebigquery）来训练一个用于下一个单词预测的循环网络。

Reddit数据集对于模拟的联邦学习实验特别有用，因为它带有自然的每用户数据分区（由文章作者编写）。这包括在实际实施中预期会出现的许多特点。例如，许多用户的数据相对较少

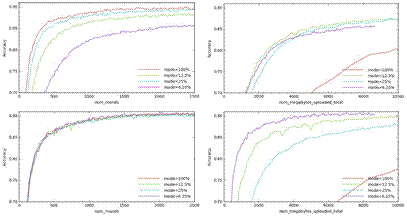


图1:CIFAR数据的结构化更新，用于各种模式的大小缩减。最上面一行是低级更新，下面一行是随机掩码更新。

大多数用户使用的点和词围绕特定用户感兴趣的特定主题进行聚集。

在我们所有的实验中，我们采用了联邦平均算法（McMahan等人，2017），这显著减少了训练一个好模型所需的通信轮数。然而，我们希望我们的技术在应用于同步分布式SGD时也能显示出类似的通信成本降低，例如Alistarh等人。（2016年）。对于联邦平均，我们在每一轮中随机选择多个客户，每个客户在其本地数据集上执行多个SGD阶段，学习率为。对于结构化更新，SGD仅限于受限空间中的更新，即对于低秩更新，SGD仅限于的项，而对于随机掩码技术，则仅限于无掩码项。根据这个更新的模型，我们计算每个层的更新。在所有的情况下，我们用一系列的学习率进行实验，并报告最佳结果。*η***波黑***我t我t*

## 4.1        CIFAR-10数据集上的卷积模型

在本节中，我们使用CIFAR-10数据集来研究我们提出的作为联邦平均算法一部分的方法的属性。

CIFAR-10数据集中有50000个培训示例，我们将其随机划分为100个客户机，每个客户机包含500个培训示例。我们使用的模型体系结构是从Springenberg et al。（2014），共有106个参数。虽然这个模型不是最先进的，但它已经足够满足我们的需要了，因为我们的目标是评估我们的压缩方法，而不是在这个任务上达到尽可能高的精确度。

该模型有9个卷积层，其中第一层和最后一层的参数明显少于其他层。因此，在整个部分中，当我们试图减小单个更新的大小时，我们只压缩内部7层，每个层都具有相同的参数[4]。对于所有方法，我们用关键字“mode”表示。对于低秩更新，“mode=25%”表示更新的秩被设置为全层变换秩的1/4，对于随机遮罩或草图绘制，这是指除25%以外的所有参数被清零。

在第一个实验中，如图1所示，我们比较了第2节中介绍的两种类型的结构化更新——第一行是低秩的，下排是随机掩码。主要的信息是随机掩码的性能明显优于低秩，因为我们减少了更新的大小。特别是，随机掩模的收敛速度基本上不受影响

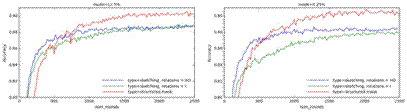


图2：结构化随机掩码更新和不量化CIFAR数据的草图更新的比较。

当以子弹的数量来衡量的时候。因此，如果目标只是最小化上传大小，那么更新大小减小的版本显然是赢家，如右列所示。

在图2中，我们比较了结构化和草图更新的性能，没有任何量化。由于在上面，结构化随机掩码更新的性能更好，因此为了清晰起见，我们省略了低秩更新。我们将这一点与草图更新的性能进行比较，包括使用和不使用随机旋转对更新进行预处理（如第3节所述），以及两种不同的模式。我们用‘HD’表示随机Hadamard旋转，用‘I’表示无旋转。

直观的期望是，直接学习结构化随机掩码更新应该比学习非结构化更新好，然后用相同数量的参数来表示非结构化更新。这是因为通过素描我们丢掉了一些在训练中获得的信息。通过绘制更新，我们应该收敛到略低精度的事实可以从理论上得到支持，使用类似的论点，如（Alistarh等人，2016）中所述，因为绘制更新会增加收敛分析中直接出现的方差。我们看到这种行为时，使用结构化随机掩码更新，我们最终能够收敛到稍高的精度。然而，我们也看到，通过绘制更新草图，我们可以稍微更快地获得适度的准确度（例如85%）。

在对CIFAR数据的最后一个实验中，我们重点讨论了第3节中介绍的所有三个元素之间的相互作用-子采样、量化和随机旋转。请注意，所有这些工具的组合将使压缩比在上述实验中更高。图3中的每对图集中于特定的模式（子采样），在每一对图中，我们用量化中使用的不同比特来绘制性能，有无随机旋转。我们在所有曲线图中都可以看到，随机旋转提高了性能。一般来说，在没有旋转的情况下，算法的性能不太稳定，尤其是在量化比特数较少和模式较小的情况下。

为了突出通信节省的潜力，请注意，通过随机旋转进行预处理，勾画出除6.25%的更新元素外的所有元素，并使用2位进行量化，我们只获得了收敛性的轻微下降，同时在表示各个层的更新所需的位方面节省了256倍。最后，如果我们对最小化上传的数据量感兴趣，我们可以获得适度的准确率，比如85%，而总的来说通信量不到上传原始数据所需的一半。

## 4.2        REDDIT数据的LSTM下一个字预测

我们根据包含Reddit（googlebigquery）上公开的帖子/评论的数据构建了模拟联邦学习的数据集，如Al-Rfou等人所述。（2016年）。关键的是，数据库中的每一篇文章都由作者键入，因此我们可以根据这些键对数据进行分组，假设每个作者都有一个客户端设备。一些作者有大量的帖子，但是在每一轮FedAvg中，我们最多为每个用户处理32000个令牌。我们省略了少于1600个令牌的作者，因为在模拟中每个客户机都有固定的开销，而且数据很少的用户对培训没有多大贡献。这就留下了763430个用户的数据集，平均每个用户有24791个令牌。为了进行评估，我们使用了一个相对较小的测试集，由75122个令牌组成，这些令牌由随机发布的帖子组成。

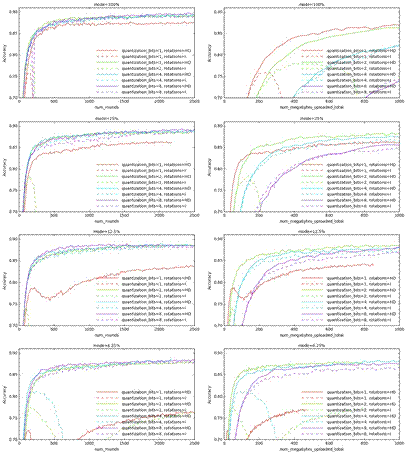


图3：草图更新的比较，将预处理更新与旋转、量化和CIFAR数据的二次采样相结合。

基于这些数据，我们训练了一个LSTM下一个单词预测模型。该模型被训练成给定当前单词和前一时间步传递的状态向量来预测下一个单词。该模型的工作原理如下：通过在10017个单词（标记）的字典中查找单词，将单词映射到嵌入向量∈R。然后用该模型在前一时间步发射的状态∈Rto发射一个新的状态向量和一个“输出嵌入”∈R96。输出嵌入通过内积对词汇表中每个项目的嵌入进行评分，然后通过softmax进行归一化，计算词汇表上的概率分布。与其他标准语言模型一样，我们将每个输入序列视为以隐式“BOS”（序列开始）标记开始，以隐式“EOS”（序列结束）标记结束。与我们学习的嵌入模型和嵌入模型的软件层不同，这两种模型都使用相同的嵌入语言。这使得模型的大小减少了大约40%，而模型质量却有一点下降，这对于移动应用来说是一种有利的权衡。许多标准LSTM-RNN方法的另一个变化是，我们训练这些模型来限制单词嵌入的L2范数为1.0，这是一个改进，可以提高收敛时间。模型共有1.35M的参数。*stet公司*96*et公司st*1 256 *stot公司*

为了减少更新的规模，我们绘制了所有模型变量的草图，除了一些小变量（如偏差）消耗的内存小于0.01%。我们使用AccuracyTop1评估，模型赋予最高概率的单词的概率是正确的。如果真正的下一个单词不在词典中，我们总是认为这是一个错误，即使模型预测“未知”。

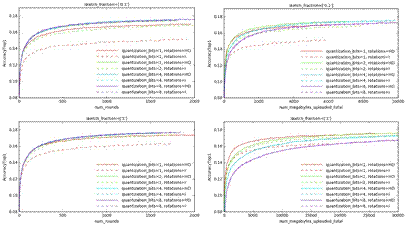


图4：比较草图更新，在Reddit数据上训练一个递归模型，每轮随机抽样50个客户。

在图4中，我们对Reddit数据运行联邦平均算法，并使用指定草图的各种参数。在每一次迭代中，我们随机抽取50个用户，这些用户根据本地可用的数据计算更新，绘制草图，然后计算所有更新的平均值。每轮抽样10、20和100名客户的实验得出了如下类似的结论。

在所有的绘图中，我们结合了三个组件来绘制第3节中介绍的更新。首先，我们应用随机旋转来预处理局部更新。此外，“草图分数”设置为0.1或1，表示正在进行子采样的更新元素的分数。

在左栏中，我们将其与算法的迭代次数进行比较。首先，我们可以看到随机旋转的预处理效果有显著的正效应，特别是在量化比特数较少的情况下。有趣的是，对于所有子采样比的选择，量化为2比特的随机Hadamard变换不会导致任何性能损失。要突出显示的一个重要指标是图中显示的子弹数是2000。因为我们每轮抽样50个用户，所以这个实验不会接触大多数用户的数据一次！这进一步加强了在现实环境中应用联合学习而不会以任何方式影响用户体验的说法。

在右边的一列中，我们根据客户端需要传输回服务器的总兆字节数绘制相同的数据。从这些图中可以清楚地看出，如果一个人需要主要最小化这个指标，我们提出的技术是非常有效的。当然，这两个目标都不是我们在实际应用中要优化的。尽管如此，鉴于目前在大规模部署联合学习过程中固有的问题上缺乏经验，我们相信这些都是实际应用中相关内容的有用代理。

最后，在图5中，我们研究了单轮中使用的客户机数量对收敛的影响。我们对固定数量的轮（500和2500）运行联邦平均算法，每轮有不同数量的客户端，将更新量化为1位，并绘制结果精度图。我们看到，如果每轮有足够数量的客户机（在本例中为1024个），我们可以将子采样元素的分数降低到1%，与10%相比，准确度仅略有下降。这意味着在联邦环境中有一个重要而实际的折衷：可以在每一轮中选择更多的客户端，同时让每个客户端通信更少（例如，更积极的子采样），并获得与使用较少客户端相同的精度，但是让每个客户端通信更多。当有许多客户端可用时，前者可能更可取，但每个客户端的上载带宽非常有限，这是实践中常见的设置。

# 参考文献

拉米·阿勒福、马克·皮克特、哈维尔·斯奈德、尹宣成、布赖恩·斯特罗普和雷·库兹韦尔。会话语境线索：个性化和历史反应排序的案例。arXiv:1606.003722016年。

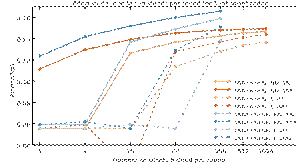


图5：每轮培训中使用的客户数量的影响。

Dan Alistarh，Jerry Li，Ryota Tomioka和Milan Vojnovic。QSGD：通信最优随机梯度下降的随机量化。第16132.14条2016年。

基思·博纳维茨、弗拉基米尔·伊万诺夫、本·克鲁特、安东尼奥·马塞东、H.布伦丹·麦克马汉、萨瓦尔·帕特尔、丹尼尔·拉马奇、亚伦·西格尔和卡恩·塞思。保护隐私机器学习的实用安全聚合。2017年ACM计算机和通信安全会议（ACM CCS）。

Trishul Chilimbi、Yutaka Suzue、Johnson Apacible和Karthik Kalyanaraman。项目亚当：建立一个高效和可扩展的深度学习培训系统。第11届USENIX操作系统设计与实现研讨会（OSDI 14），第571-582页，2014年。

Jeffrey Dean、Greg Corrado、Rajat Monga、Kai Chen、Matthieu Devin、Mark Mao、Andrew Senior、Paul Tucker、Ke Yang、Quoc V Le等。大规模分布式深层网络。NIPS，第1223-12312012页。

Misha Denil、Babak Shakibi、Laurent Dinh、Nando de Freitas等。深度学习中的参数预测。NIPS，第2148-21562013页。

Mostafa El Gamal和Lifeng Lai。带量化更新的随机分布坐标下降问题。arXiv:1609.055392016年。

丹尼尔·戈洛文、D·斯库利、H·布伦丹·麦克马汉和迈克尔·杨。通过随机化，用较少的内存进行大规模学习。在ICML，2013年。

谷歌大查询。Reddit评论数据集。BigQuery，2016年。https://bigcloud.google查询。com/dataset/fh bigquery。

宋寒，毛惠子，戴利。深度压缩：用剪枝、训练量化和哈夫曼编码对深层神经网络进行压缩。arXiv预印本arXiv:1510.00149，2015年。

Jakub Konecnˇy和Peter Richt&apos;arik。随机分布平均值估计：准确性与通讯性arXiv:1611.07555年2016年。

Jakub Konecnˇy、H.Brendan McMahan、Daniel Ramage和Peter Richt&apos;arik。联邦优化：分布式机器学习在设备智能。arXiv预印本arXiv:1610.025272016年。

亚历克斯·克里兹夫斯基。从微小图像中学习多层特征。技术报告，2009年。

Chenxin Ma、Jakub Konecnˇy、Martin Jaggi、Virginia Smith、Michael I Jordan、Peter Richt&apos;arik和Martin&apos;Taka&apos;cˇ。具有任意局部解算器的分布式优化。优化方法与软件，32（4）：813–8482017。

H、 布伦丹·麦克马汉和丹尼尔·拉马奇。联合学习：没有集中训练数据的协作机器学习。https://research.googleblog.com/2017/04/federed-learning-collaborative.html2017年。

H、 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。《第20届国际人工智能与统计会议论文集》，2017年。

M、 拉巴特和诺瓦克。分布式优化的量化增量算法。IEEE通讯选择领域杂志，23（4）：798-8082005。

萨沙克•J•雷迪、雅库布•科尼肯•y、彼得•里希特•阿里克、巴纳布•阿斯•奥斯和亚历克斯•斯莫拉。助手：快速和通信高效的分布式优化。arXiv:1608.068792016年。

Ohad Shamir，Nathan Srebro和Tong Zhang。基于近似牛顿型方法的通信高效分布式优化。ICML，第1000-10082014页。

Reza Shokri和Vitaly Shmatikov。隐私保护深度学习。第22届ACM SIGSAC计算机和通信安全会议记录，CCS’15，2015。

速度测试.net. 快速测试市场报告。http://www.speedtest.net/reports/united-states/2016年8月。

Jost Tobias Springenberg，Alexey Dosovitskiy，Thomas Brox和Martin Riedmiller。力求简单：全卷积网。arXiv:1412.68062014年。

阿南达·瑟塔·苏雷什、费利克斯·尤、桑吉夫·库马尔和H·布伦丹·麦克马汉。通信受限的分布式均值估计。第34届机器学习国际会议论文集，第3329-33372017页。

大卫·P·伍德拉夫。素描作为数值线性代数的工具。理论计算机科学基础与趋势，10（12）：1-1572014。国际标准书号1551-305X.doi:10.1561/040000060。

张雨晨，小林。迪斯科：自洽经验损失的分布优化。ICML，第362-370页，2015年。

[[1]](" \l "_ftnref1" \o ")在爱丁堡大学附属期间完成的工作。

[[2]](" \l "_ftnref2" \o ")为了简单起见，我们只讨论一个矩阵的情况，因为所有的事情都要用多个矩阵来设置，例如对应于深层神经网络中的各个层。

[[3]](" \l "_ftnref3" \o ")加权和可以用来代替基于特定实现的平均值。

[[4]](" \l "_ftnref4" \o ")我们还尝试减小所有9层的大小，但这会在通信方面产生微不足道的节省，同时会稍微降低收敛速度。