

# 没有最快，只有更快！富士通 74.7 秒在 ImageNet 上训练完 ResNe

人工智能4月3日



大数据文摘出品

**74.7 秒！**

根据日本富士通实验室最新研究。他们应用了一种优化方法，在 ABCI 集群上，实现了 74.7 秒的训练时间。训练吞吐量为 173 万图像/秒，top-1 验证准确率为 75.08%。

当然，如此快的速度也依赖于巨量的算力——使用了 2,048 个 GPU。

目前此论文已经放在了 arxiv 上面供大家免费下载。

LGJ 29 Mar 2019

# Yet Another Accelerated SGD: ResNet-50 Training on ImageNet in 74.7 seconds

Masafumi Yamazaki, Akihiko Kasagi, Akihiro Tabuchi, Takumi Honda, Masahiro Miwa,  
Naoto Fukumoto, Tsuguchika Tabaru, Atsushi Ike, Kohta Nakashima  
*Fujitsu Laboratories Ltd.*  
{m.yamazaki, kasagi.akihiko, tabuchi.akihiro, honda.takumi, masahiro.miwa,  
fukumoto.naoto, tabaru, ike, nakashima.kouta}@fujitsu.com

**Abstract**—There has been a strong demand for algorithms that can execute machine learning as faster as possible and the speed of deep learning has accelerated by 30 times only in the past two years. Distributed deep learning using the large mini-batch is a key technology to address the demand and is a great challenge as it is difficult to achieve high scalability on large clusters without compromising accuracy. In this paper, we introduce optimization methods which we applied to this challenge. We achieved the training time of 74.7 seconds using 2,048 GPUs on ABCI cluster applying these methods. The training throughput is over 1.73 million images/sec and the top-1 validation accuracy is 75.08%.

use convolutional layers for 2D and 3D image data. Ioffe et al. [9] introduced the batch normalization technique, in which the feature values in hidden layers are normalized to avoid vanishing gradients. In addition, this technique enables training of models with a large number of layers, such as ResNet.

Generally, the mini-batch size should be large for distributed deep learning on large clusters. Goyal et al. [2] proposed the warm-up technique to keep the validation accuracy with 8,192 mini-batch size. Google [3] and Sony [4] use the large mini-batch size which becomes larger and achieved highly parallel processing.

## I. INTRODUCTION

论文下载地址：

<https://arxiv.org/abs/1903.12650>

论文中提到，深度学习在过去的两年内迅速发展，发展速度加快了三十倍，人们对于能够高效执行机器学习的算法的需求越来越大。使用大规模 mini batch 的分布式深度学习是解决这个需求的关键技术，但很难准确地在大型集群上实现很高的可扩展性。

在大型数据集上训练的神经网络（DNN）模型在各领域（如物体检测，语言翻译等）都取得了令人瞩目的成果。然而，随着 DNN 模型和数据集的增多，DNN 训练的计算成本也增加了。

众所周知，具有数据并行性的分布式深度学习能快速进行群集训练。此方法中，在群集上进行的所有进程都具有相同的 DNN 模型和权重。每个进程使用不同的 mini batch 训练模型，但是所有进程的权重梯度通过组合来更新。

这种通信开销 ( communication overhead ) 成为大型集群的重要问题。为了减少大型集群的开销 , 研究团队增加了 DNN 的 mini batch 并且同时进行 DNN 训练。然而 , 使用大型 mini-batch 训练通常会导致 DNN 模型的验证准确性更差。因此 , 他们使用几种方法来增加 mini-batch 的大小 , 这代表在迭代中计算的输入图像的数量 , 而且不会影响验证准确性。

关于使用的计算资源 , 他们在 ABCI 集群和优化的 MXNet 深度学习框架上使用 2,048 个 GPU。在 ImageNet 上使用 81,920 mini-batch , 74.7 秒内就训练好 ResNet-50 , 达到了 75.08% 验证准确度。

## 具体方法

根据论文介绍 , 具体的研究方法分为 3 个部分 , 准确率提升、框架优化以及通信优化 ( Communication Optimizations )

关于准确率提升 , 其使用了通常用于深度学习优化的随机梯度下降 ( SGD )。当使用 large mini-batch 训练时 , SGD 更新数量会随着小批量的增加而减少。所以 , 提高 large mini-batch 上最终验证准确率是一个很大的挑战 , 他们采用了学习速率控制 : 由于更新次数少 , 需要使用较高的学习速度来加速训练。然而 , 高的学习率使模型的训练在早期阶段变得不稳定。因此 , 我们通过使用热身 ( warmup ) 来稳定 SGD , 从而逐渐提高学习率。此外 , 对于某些层次来说 , 所有层的学习速度都太高 , 使用分层自适应率来稳定训练。

|                   | Batch Size     | Processor                 | DL Library   | Time            | Accuracy      |
|-------------------|----------------|---------------------------|--------------|-----------------|---------------|
| He et al. [1]     | 256            | Tesla P100 × 8            | Caffe        | 29 hours        | 75.3 %        |
| Goyal et al. [2]  | 8,192          | Tesla P100 × 256          | Caffe2       | 1 hour          | 76.3 %        |
| Smith et al. [3]  | 8,192 → 16,384 | full TPU Pod              | TensorFlow   | 30 mins         | 76.1 %        |
| Akiba et al. [4]  | 32,768         | Tesla P100 × 1,024        | Chainer      | 15 mins         | 74.9 %        |
| Jia et al. [5]    | 65,536         | Tesla P40 × 2,048         | TensorFlow   | 6.6 mins        | 75.8 %        |
| Ying et al. [6]   | 65,536         | TPU v3 × 1,024            | TensorFlow   | 1.8 mins        | 75.2 %        |
| Mikami et al. [7] | 55,296         | Tesla V100 × 3,456        | NNL          | 2.0 mins        | 75.29%        |
| <b>This work</b>  | <b>81,920</b>  | <b>Tesla V100 × 2,048</b> | <b>MXNet</b> | <b>1.2 mins</b> | <b>75.08%</b> |

ImageNet 上 ResNet-50 的训练时间和 top-1 验证准确率。

关于框架。其使用了 MXNet，这是一个用 C++ 和 CUDA C 语言编写的开源深度学习框架。MXNet 具有灵活性和可扩展性，可以在集群上高效地训练模型。然而，在中小型集群环境中只占总时间的一小部分的计算可能成为大规模集群环境中的瓶颈。他们分析了 CPU 和 GPU 的性能，并找出了瓶颈。优化了瓶颈，提高训练吞吐量如下。采用的方法包括：并行 DNN 模型初始化 ( Parallel DNN model initialization )，在 GPU 上的批量规范计算 ( Batched norm computations on the GPU )

分布式并行深度学习需要所有减少通信以在所有进程之间交换每层的梯度，由于每个 GPU 的 batch 规模小，因此通信时间变长，而计算时间变短，所以减少通信开销在大型集群环境中是不可忽略的。为了克服这些问题，他们采用的方法包括：调节通信的数据大小 ( Adjusting data size of communication )，即研究人员收集了各层的梯度，然后将 allreduce 的数据大小调整为几兆字节；最佳的通信调度 ( Optimal scheduling of communications ) 即预先以统计的方式将所有层分为好几组。一旦每个进程完成一组中所有层的反向处理，就计划开始 allreduce 操作。

最后，训练准确度以及验证准确度结果如下

