Get Started With Deep Learning Performance

Getting Started | NVIDIA Docs

DU-09794-001_v001 | February 2023

Table of Contents

Chapter 1. Overview

GPUs accelerate machine learning operations by performing calculations in parallel. Many operations, especially those representable as matrix multiplies, will see good acceleration right out of the box. Even better performance can be achieved by tweaking operation parameters to efficiently use GPU resources.

This document presents the tips that we think are most widely useful. We link to each of the other pages, with more in-depth information, where appropriate. If you want to jump straight to optimizing a network, read our Checklists!

Chapter 2. Recommendations

2.1. Operating In Math-Limited Regime Where Possible

GPUs excel at performing calculations in parallel, but data also needs to be loaded and stored around those calculations, and thus data movement speed can also limit achievable performance. If the speed of a routine is limited by calculation rate (**math-limited** or **math-bound**), performance can be improved by enabling Tensor Cores and following our other recommendations.

On the other hand, if a routine is limited by the time taken to load inputs and write outputs (**bandwidth-limited** or **memory-bound**), speeding up calculation does not improve performance. For fully-connected and convolutional layers, this occurs mostly when one or more parameters of a layer are small.

In other words, if an operation is memory-bound, tweaking parameters to more efficiently utilize the GPU is ineffective. Operations not representable as matrix multiplies, including activation functions, pooling, and batch normalization, are nearly always memory-bound. Those with an equivalent matrix multiply, including fully-connected, convolutional, and recurrent layers, may be memory-bound or math-bound depending on their sizes. Larger layers tend to have more calculations relative to the number of memory accesses, a ratio that we refer to as **arithmetic intensity**. If arithmetic intensity exceeds a particular threshold (dependent on the GPU type and the type of calculation being done), the operation is math-bound and can be optimized effectively with our tips. See Understanding Performanceand Math and Memory Bounds for background and details.

2.2. Using Tensor Cores Efficiently With Alignment

Tensor Cores are most efficient when key parameters of the operation are multiples of 4 if using TF32, 8 if using FP16, or 16 if using INT8 (equivalently, when key dimensions of the operation are aligned to multiples of 16 bytes in memory). For fully-connected layers, the relevant parameters are the batch size and the number of inputs and outputs; for convolutional layers, the number of input and output channels; and for recurrent layers, the

minibatch size and hidden sizes. With NVIDIA® cuBLAS 11.0 or higher and NVIDIA CUDA® Deep Neural Network library (cuDNN) 7.6.3 or higher, Tensor Cores can be used even if this requirement is not met, though performance is better if it is. In earlier versions, Tensor Cores may not be enabled if one or more dimensions aren't aligned. This requirement is based on how data is stored and accessed in memory. Further details can be found in Tensor Core Requirements.

TF32, a datatype introduced with the NVIDIA Ampere Architecture, works with existing FP32 code to leverage Tensor Cores. More detail on TF32 can be found at this link. Mixed precision is another option for networks that currently use FP32, and works with both the NVIDIA Ampere Architecture and NVIDIA Volta™ and NVIDIA Turing™ GPUs. The NVIDIA Training with Mixed Precision Guide explains how to use mixed precision with Tensor Cores, including instructions for getting started quickly in a number of frameworks.

2.3. Choosing Parameters To Maximize Execution Efficiency

GPUs perform operations efficiently by dividing the work between many parallel processes. Consequently, using parameters that make it easier to break up the operation evenly will lead to the best efficiency. This means choosing parameters (including batch size, input size, output size, and channel counts) to be divisible by larger powers of two, at least 64, and up to 256.

There is no downside to using values divisible by 512 and higher powers of two, but there is less additional benefit. Divisibility by powers of two is most important for parameters that are small; choosing 512 over 520 has more impact than choosing 5120 over 5128. Additionally, for these tweaks to improve efficiency, the operation *must already be math-bound*, which usually requires at least one parameter to be substantially larger than 256. See Operating In Math-Limited Regime Where Possible and other linked sections about calculating arithmetic intensity.

More specific requirements for different routines can be found in the corresponding checklist and guide. Background on why this matters can be found in GPU Architecture Fundamentals and Typical Tile Dimensions in CUBLAS and Performance.

Chapter 3. Checklists

We provide the following quick start checklists with tips specific to each type of operation.

- Checklist for Fully-Connected Layers
- Checklist for Convolutional Layers
- Checklist for Recurrent Layers
- Checklist for Memory-Limited Layers

Chapter 4. How This Guide Fits In

NVIDIA's GPU deep learning platform comes with a rich set of other resources you can use to learn more about NVIDIA's Tensor Core GPU architectures as well as the fundamentals of mixed-precision training and how to enable it in your favorite framework.

The NVIDIA V100 GPU architecture whitepaper provides an introduction to NVIDIA Volta, the first NVIDIA GPU architecture to introduce Tensor Cores to accelerate Deep Learning operations. The equivalent whitepaper for the NVIDIA Turing architecture expands on this by introducing NVIDIA Turing Tensor Cores, which add additional low-precision modes. The whitepaper for the NVIDIA Ampere architecture introduces Tensor Core support for additional precisions (including TF32, which works with existing FP32 workloads to leverage Tensor Cores), up to 2x throughput with the Sparsity feature, and virtual partitioning of GPUs with the Multi-Instance GPU feature.

The NVIDIA Training With Mixed Precision User's Guide describes the basics of training neural networks with reduced precision such as algorithmic considerations following from the numerical formats used. It also details how to enable mixed precision training in your framework of choice, including TensorFlow, PyTorch, and MXNet. The easiest and safest way to turn on mixed precision training and use Tensor Cores is through Automatic Mixed Precision, which is supported in PyTorch, TensorFlow, and MxNet.

Additional documentation is provided to help explain how to:

- Tweak parameters of individual operations by type, with examples:
  - NVIDIA Optimizing Linear/Fully-Connected Layers User's Guide
  - NVIDIA Optimizing Convolutional Layers User's Guide
  - NVIDIA Optimizing Recurrent Layers User's Guide
  - NVIDIA Optimizing Memory-Limited Layers User's Guide
- Understand the ideas behind these recommendations:
  - NVIDIA GPU Performance Background User's Guide
  - NVIDIA Matrix Multiplication Background User's Guide

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Google

Android, Android TV, Google Play and the Google Play logo are trademarks of Google, Inc.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
www.nvidia.com

Trademarks

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
www.nvidia.com