NVIDIA cuDNN

Best Practices | NVIDIA Docs

BPG-09678-001_v8.3.1 | November 2021

Table of Contents

List of Tables

Chapter 1. Introduction

> **ATTENTION:** These guidelines are applicable to 3D convolution and deconvolution functions starting in NVIDIA® NVIDIA® CUDA® Deep Neural Network library (cuDNN) v7.6.3.

This document provides guidelines for setting the cuDNN library parameters to enhance the performance of 3D convolutions. Specifically, these guidelines are focused on settings such as filter sizes, padding and dilation settings. Additionally, an application-specific use-case, namely, medical imaging, is presented to demonstrate the performance enhancement of 3D convolutions with these recommended settings.

Specifically, these guidelines are applicable to the following functions and their associated data types:

- cudnnConvolutionForward()
- cudnnConvolutionBackwardData()
- cudnnConvolutionBackwardFilter()

For more information, refer to the *NVIDIA cuDNN Developer Guide* and the *NVIDIA cuDNN API Reference*.

Chapter 2. Best Practices For Medical Imaging

To optimize your performance in your model, ensure you meet the following general guidelines:

Layout

The layout is in NCHW format.

Filter size

The filter size is `Tx1x1`, `Tx2x2`, `Tx3x3`, `Tx5x5`, where `T` is a positive integer. There are additional limits for the value of `T` in `wgrad` and strided `dgrad`.

Stride

Arbitrary for forward and backward filter; `dgrad`/`deconv`: 1x1x1 or 2x2x2 with 2x2x2 filter.

Dilation

The dilation is 1x1x1.

Platform

The platform is NVIDIA Volta™, NVIDIA Turing™, and NVIDIA Ampere Architecture with input/output channels divisible by 8.

Batch/image size

cuDNN will fallback to non-Tensor Core kernel if it determines that the workspace required is larger than 256MB of GPU memory. The workspace required depends on many factors. For the Tensor Core kernels, the workspace size generally scales linearly with output tensor size. Therefore, this can be mitigated by using smaller image sizes or mini-batch sizes.

2.1. Recommended Settings In cuDNN While Performing 3D Convolutions

The following tables show the recommended settings in each release.

2.1.1. cuDNN 8.x.x Recommended Settings

Recommended settings while performing 3D convolutions for cuDNN 8.x.x.

		8.3.0 - 8.3.1	8.0.3 - 8.2.4	8.0.0 and 8.0.1 Preview - 8.0.2
Platform		NVIDIA Ampere Architecture NVIDIA Turing Architecture NVIDIA Volta Architecture		
Convolution (3D or 2D)		3D and 2D		

Convolution or deconvolution (`fprop`, `dgrad`, or `wgrad`)		`fprop`
		`dgrad`
		`wgrad`		

Grouped convolution size		`C_per_group == K_per_group == {1,4,8,16,32,64,128,256}` Not supported for INT8	`C_per_group == K_per_group == {4,8,16,32}`	`C_per_group == K_per_group == {4,8,16,32}`
Data layout format (`NHWC`/`NCHW`)[1]		NDHWC		
Input/output precision (FP16, FP32, INT8, or FP64)		FP16, FP32[2], INT8[3]	FP16, FP32	
Accumulator (compute) precision (FP16, FP32, INT32 or FP64)		FP32, INT32	FP32	
Filter (kernel) sizes		No limitation		
Padding		No limitation		
Image sizes		2 GB limitation for a tensor		
Number of channels	C	`0 mod 8` `0 mod 16` (for INT8)	`0 mod 8`	
	K	`0 mod 8` `0 mod 16` (for INT8)	`0 mod 8`	
Convolution mode		Cross-correlation and convolution		
Strides		No limitation	`dgrad`: 1x1x1 or 2x2x2	
Dilation		No limitation		
Data pointer alignment		All data pointers are 16-bytes aligned.		

2.1.2. cuDNN 7.6.x Recommended Settings

Recommended settings while performing 3D convolutions for cuDNN 7.6.x.

1 `NHWC`/`NCHW` corresponds to `NDHWC`/`NCDHW` in 3D convolution.
2 With `CUDNN_TENSOROP_MATH_ALLOW_CONVERSION` pre-Ampere. Default TF32 math in NVIDIA Ampere Architecture.
3 INT8 does not support `dgrad` and `wgrad`.

		7.6.5	7.6.4	7.6.2	7.6.1
Platform		NVIDIA Turing NVIDIA Volta	NVIDIA Volta		
Convolution (3D or 2D)		3D and 2D	3D		

Convolution or deconvolution (`fprop`, `dgrad`, or `wgrad`)		`fprop`
		`dgrad`
		`wgrad`
				`fprop`
				`dgrad`
					`fprop`
					`dgrad`
					`wgrad`

Grouped convolution	Yes or No	Yes		No	
	Group size	`C_per_group == K_per_group == {4,8,16,32}`		NA	
Data layout format (`NHWC`/`NCHW`)[4]		NCDHW			NCDHW[5]
Input/output precision (FP16, FP32, or FP64)		FP16		FP16 or FP32	FP16[6] or FP32[7]
Accumulator (compute) precision (FP16, FP32, or FP64)		FP32		Better to be the same with input/output precision.	FP32
Filter (kernel) sizes		2x2x2 T[8]x1x1 Tx2x2 Tx3x3 Tx5x5			1x1x1 2x2x2 3x3x3 5x5x5 Tx1x1 Tx2x2 Tx3x3 Tx5x5 Tx1x1 Tx2x2 Tx3x3 Tx5x5
Padding		No limitation			Filter // 2[9]
Image sizes		256 MB WS limit		No limitation	256 MB WS limit

4 `NHWC`/`NCHW` corresponds to `NDHWC`/`NCDHW` in 3D convolution.
5 With NCHW <> NHWC format transformation.
6 FP16: `CUDNN_TENSOROP_MATH`
7 FP32: `CUDNN_TENSOROP_MATH_ALLOW_CONVERSION`
8 An arbitrary positive value.
9 `padding = filter // 2`

		7.6.5	7.6.4	7.6.2	7.6.1
Number of channels	C	Arbitrary			0 mod 8
	K	Arbitrary			0 mod 8
Convolution mode		Cross-correlation for `dgrad`; otherwise, both modes		No limitation Cross-correlation	
Strides		1x1x1 and 2x2x2 strides for `dgrad`		2x2x2 Arbitrary stride	1x1x1
Dilation		1x1x1			

Chapter 3. Medical Imaging Performance

The following table shows the average speed-up of **unique cuDNN 3D convolution calls** for each network on V100 and A100 GPUs that satisfies the conditions in Best Practices For Medical Imaging. The end-to-end training performance will depend on a number of factors, such as framework overhead, kernel run time, and model architecture type.

3.1. Average Speedup Of Unique cuDNN 3D Convolutions API Calls

3.1.1. cuDNN 8.x.x Average Speedup

cuDNN version 8.3.1 compared to 7.6.5

Table 1. Average speed-up of unique cuDNN (version 8.3.1 compared to 7.6.5) 3D convolution API calls on V100 and A100 for both FP16 and FP32

		A100 8.3.1 vs V100 7.6.5		V100 8.3.1 vs V100 7.6.5	
Model	**Batchsize**	**FP16**	**FP32**	**FP16**	**FP32**
V-Net (3D-Image segmentation)	2	2.5x	8.1x	2.3x	2.7x
	8	3.8x	6.6x	2.7x	1.9x
	16	4.7x	7.7x	2.8x	2.0x
	32	6.8x	5.9x	3.8x	1.5x
3D-UNet (3D-Image Segmentation)	2	8.6x	7.6x	4.1x	1.2x
	4	13.2x	6.7x	6.1x	1.1x

cuDNN version 8.3.0 compared to 7.6.5

Table 2. Average speed-up of unique cuDNN (version 8.3.0 compared to 7.6.5) 3D convolution API calls on V100 and A100 for both FP16 and FP32

Model	Batchsize	A100 8.3.0 vs V100 7.6.5		V100 8.3.0 vs V100 7.6.5	
		FP16	FP32	FP16	FP32
V-Net (3D-Image segmentation)	2	2.53x	8.0x	2.3x	2.7x
	8	3.8x	6.5x	2.7x	1.9x
	16	4.6x	7.7x	2.8x	2.0x
	32	6.8x	5.9x	3.8x	1.5x
3D-UNet (3D-Image Segmentation)	2	8.5x	7.7x	4.1x	1.2x
	4	13.2x	6.8x	6.1x	1.1x

cuDNN version 8.2.4 compared to 7.6.5

Table 3. Average speed-up of unique cuDNN (version 8.2.4 compared to 7.6.5) 3D convolution API calls on V100 and A100 for both FP16 and FP32

Model	Batchsize	A100 8.2.4 vs V100 7.6.5		V100 8.2.4 vs V100 7.6.5	
		FP16	FP32	FP16	FP32
V-Net (3D-Image segmentation)	2	2.4x	7.4x	2.3x	2.5x
	8	3.6x	6.3x	2.6x	1.7x
	16	4.4x	7.5x	2.7x	2.1x
	32	6.5x	5.7x	3.5x	1.6x
3D-UNet (3D-Image Segmentation)	2	8.0x	7.1x	3.9x	1.5x
	4	12.6x	6.3x	5.9x	1.5x

cuDNN version 8.2.2 compared to 7.6.5

Table 4. Average speed-up of unique cuDNN (version 8.2.2 compared to 7.6.5) 3D convolution API calls on V100 and A100 for both FP16 and FP32.

Model	Batchsize	A100 8.2.2 vs V100 7.6.5		V100 8.2.2 vs V100 7.6.5	
		FP16	FP32	FP16	FP32
V-Net (3D-Image segmentation)	2	2.4x	7.5x	2.2x	2.5x
	8	3.6x	6.3x	2.6x	1.7x
	16	4.4x	7.5x	2.7x	2.1x
	32	6.5x	5.7x	3.5x	1.6x
3D-UNet (3D-Image Segmentation)	2	8.0x	7.1x	3.9x	1.5x
	4	12.6x	6.3x	5.9x	1.5x

cuDNN version 8.2.1 compared to 7.6.5

Table 5. Average speed-up of unique cuDNN (version 8.2.1 compared to 7.6.5) 3D convolution API calls on V100 and A100 for both FP16 and FP32.

Model	Batchsize	A100 8.2.1 vs V100 7.6.5		V100 8.2.1 vs V100 7.6.5	
		FP16	FP32	FP16	FP32
V-Net (3D-Image segmentation)	2	2.5x	7.7x	2.2x	2.5x
	8	3.7x	6.4x	2.6x	1.7x
	16	4.5x	7.5x	2.7x	2.1x
	32	6.5x	5.7x	3.6x	1.6x
3D-UNet (3D-Image Segmentation)	2	8.3x	7.3x	3.8x	1.5x
	4	12.7x	6.4x	5.8x	1.5x

cuDNN version 8.2.0 compared to 7.6.5

Table 6. Average speed-up of unique cuDNN (version 8.2.0 compared to 7.6.5) 3D convolution API calls on V100 and A100 for both FP16 and FP32.

Model	Batchsize	A100 8.2.0 vs V100 7.6.5		V100 8.2.0 vs V100 7.6.5	
		FP16	FP32	FP16	FP32
V-Net (3D-Image segmentation)	2	2.3x	7.3x	2.2x	2.5x
	8	3.4x	5.9x	2.4x	1.8x
	16	4.1x	6.8x	2.5x	2.1x
	32	5.8x	5.1x	3.3x	1.6x
3D-UNet (3D-Image Segmentation)	2	6.8x	5.9x	3.4x	1.5x
	4	10.5x	2.6x	5.1x	1.6x

cuDNN version 8.1.1 compared to 7.6.5

Table 7. Average speed-up of unique cuDNN (version 8.1.1 compared to 7.6.5) 3D convolution API calls on V100 and A100 for both FP16 and FP32.

Model	Batchsize	A100 8.1.1 vs V100 7.6.5		V100 8.1.1 vs V100 7.6.5	
		FP16	FP32	FP16	FP32
V-Net (3D-Image segmentation)	2	2.3x	6.8x	2.1x	2.4x
	8	3.2x	5.1x	2.3x	1.8x
	16	3.8x	5.9x	2.3x	2.1x
	32	5.4x	4.4x	3.1x	1.6x
3D-UNet (3D-Image Segmentation)	2	7.2x	6.3x	3.4x	1.5x
	4	11x	2.6x	4.9x	1.6x

cuDNN version 8.1.0 compared to 7.6.5

Table 8. Average speed-up of unique cuDNN (version 8.1.0 compared to 7.6.5) 3D convolution API calls on V100 and A100 for both FP16 and FP32.

Model	Batchsize	A100 8.1.0 vs V100 7.6.5		V100 8.1.0 vs V100 7.6.5	
		FP16	FP32	FP16	FP32
V-Net (3D-Image segmentation)	2	2.4x	7.3x	2.2x	2.4x
	8	3.4x	5.3x	2.3x	1.8x
	16	3.9x	6x	2.3x	2.1x
	32	5.5x	4.4x	3.1x	1.6x
3D-UNet (3D-Image Segmentation)	2	7.3x	6.4x	3.5x	1.5x
	4	11.2x	2.6x	5x	1.6x

Chapter 4. Medical Imaging Limitations

Your application will be functional but could be less performant if the model has channel counts lower than 32 (gets worse the lower it is).

If the above is in the network, use `cuDNNFind` to get the best option.

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.

ARM

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