

# DeepCore API User Guide

version 1.0

## 简介

deepcore 是一款超轻量级专为 CNN 批量训练量身打造的高度优化核心计算库。  
支持的硬件：计算能力为 5.0, 5.2, 6.0, 6.1, 7.0 的 NVIDIA GPU。  
支持单精度和混合精度（双字节存储，单精度计算）。

deepcore 的数据格式是 CNHW（注意，不同与 cudnn 和其它框架中使用的 NCHW）。  
卷积操作目前支持三种算法：conv, fftconv, cellconv；支持分组卷积；对于 fftconv 和 cellconv, filter\_size\_x 和 filter\_size\_y 必须 > 1 且 input\_channel 和 output\_channel 需要是 16 的倍数（目前内核层面并无此限制，仅仅是为了性能的考虑在接口层面的强行限制）；通过 dc\_gemmOp 支持 1x1 卷积。

目前仅支持 relu 内置激活函数, forward 支持 relu 激活函数, bias 融合; backward 支持 relu 求导融合以及其它任意激活函数的导数相乘融合。支持 reduction 操作; 支持 batch-normalization。

vdeepcore 是专门针对 volta 优化的版本且仅支持 volta GPU, 由于专门针对 tensor-core 进行了优化, 因此数据结构差别很大, 因此为简单以及避免代码过度膨胀, 从 volta 开始会有一个新的分支版本且与之前的版本不兼容。

## 1.0 deepcore 中的数据结构

deepcore 采用 CNHW 数据结构, 其优点是当采用 1x1 卷积核时, 卷积操作直接退化成一个单独的 gemm 操作, 而不是 batched-gemm; 即使对于非 1x1 的卷积核, 其相对于 NCHW 也具有更简单直观的代码和访存优势。

假设 data 和 filter 尺寸都是 2x2, 每个 map 的元素是 {\*x, \*y, \*z, \*w}; 约定 n 为 batch 的编号, c 为 channel 的编号, p 为第 L 层的 channel 数量, q 为第 L+1 层的 channel 数量:

general data layout

```
{
    {a_n0c0x, a_n0c0y, a_n0c0z, a_n0c0w, a_n1c0x, a_n1c0y, a_n1c0z, a_n1c0w, ..., padding... },
    {a_n0c1x, a_n0c1y, a_n0c1z, a_n0c1w, a_n1c1x, a_n1c1y, a_n1c1z, a_n1c1w, ..., padding... },
    .....
}
```

filter data layout

```
{
    {b_p0q0x, b_p0q0_y, b_p0q0_z, b_p0q0_w, b_p1q0x, b_p1q0_y, b_p1q0_z, b_p1q0_w, },
    ...
}
```

```

        {b_p0q1x, b_p0q1_y, b_p0q1_z, b_p0q1_w, b_p1q1x, b_p1q1_y, b_p1q1_z, b_p1q1_w},
        .....
    }

```

## 1.1 vdeepcore 中的数据结构

输入层的数据结构为普通的 CNHW(1.0), 其它层的数据结构为 packed-CNHW 且 channel 数量需为 16 的倍数; 同样以 1.0 中的例子, 则除去输入层外其它层的数据结构如下:

general data layout

```

{
    {a_n0c0x,a_n0c1x,...,a_n0c7x },
    {a_n0c0y,a_n0c1y,...,a_n0c7y },
    {a_n0c0z,a_n0c1z,...,a_n0c7z },
    {a_n0c0w,a_n0c1w,...,a_n0c7w },
    {a_n1c0x,a_n1c1x,...,a_n1c7x },
    {a_n1c0y,a_n1c1y,...,a_n1c7y },
    {a_n1c0z,a_n1c1z,...,a_n1c7z },
    {a_n1c0w,a_n1c1w,...,a_n1c7w },
    {           padding...           },
    .....
    {           padding...           },
    {a_n0c8x,a_n0c9x,...,a_n0c15x},
    {a_n0c8y,a_n0c9y,...,a_n0c15y},

    {a_n0c8z,a_n0c9z,...,a_n0c15z},
    {a_n0c8w,a_n0c9w,...,a_n0c15w},
    {a_n1c8x,a_n1c9x,...,a_n1c15x},
    {a_n1c8y,a_n1c9y,...,a_n1c15y},
    {a_n1c8z,a_n1c9z,...,a_n1c15z},
    {a_n1c8w,a_n1c9w,...,a_n1c15w},
    {           padding...           },
    .....
    {           padding...           }
}

```

filter data layout

```

{
    {b_p0q0x,b_p1q0x,...b_p7q0x },
    {b_p0q0y,b_p1q0y,...b_p7q0y },
    {b_p0q0z,b_p1q0z,...b_p7q0z },
    {b_p0q0w,b_p1q0w,...b_p7q0w },

```

```

{b_p8q0x,b_p9q0x,...b_p15q0x},
{b_p8q0y,b_p9q0y,...b_p15q0y},
{b_p8q0z,b_p9q0z,...b_p15q0z},
{b_p8q0w,b_p9q0w,...b_p15q0w},
{b_p0q1x,b_p1q1x,...b_p7q1x },
{b_p0q1y,b_p1q1y,...b_p7q1y },
{b_p0q1z,b_p1q1z,...b_p7q1z },
{b_p0q1w,b_p1q1w,...b_p7q1w },
{b_p8q1x,b_p9q1x,...b_p15q1x},
{b_p8q1y,b_p9q1y,...b_p15q1y},
{b_p8q1z,b_p9q1z,...b_p15q1z},
{b_p8q1w,b_p9q1w,...b_p15q1w},
.....
}

```

## 1.2 聚合型参数存储

bias,reduce 的计算结果以及 batch-norm 中的 gamm,beta 等参数的数据类型在 fp32 和 mixed 计算模式下均是单精度表示。