# GEMM API Document

## Description

The gemm routines compute a scalar-matrix-matrix product and add the result to a scalar-matrix product, with general matrices. The operation is defined as

where:

A, B and C are matrices:

A is an m-by-k matrix,

B is a k-by-n matrix,

C is an m-by-n matrix.

## High level API

Verisilicon GEMM routines implemented as compatible multi-platform by pure c11 environment, and it depends on ovxlib, openVX driver and hardware driver packages.

To accelerate GEMM routines with NN core for float16 data, you need make sure your VIP hardware include FP16 feature on NN core, if not, GEMM routines won’t executed on NN core.

### API signature



### Input Parameter

|  |  |
| --- | --- |
| *m* | INTEGER. Specifies the number of rows of the matrix op(A) and of the matrix C. The value of *m* must be at least zero. |
| *n* | INTEGER. Specifies the number of columns of the matrix op(B) and the number of columns of the matrix C. The value of *n* must be at least zero. |
| *k* | INTEGER. Specifies the number of columns of the matrix op(A) and the number of rows of the matrix op(B). The value of *k* must be at least zero. |
| *a* | Matrix in m\*k，supported datatype: float32 |
| *b* | Matrix in k\*n， supported datatype: float32 |
| *c* | Matrix in m\*n，supported datatype: float32 |

## Implementation limitation

GEMM base on Neural Network Process involved additional limitations due to tradeoff between performance and flexibilities.

### Matrix B need frozen during runtime

While accelerating GEMM with NN core, Matrix B consumed by NN core as convolution’s weight. Our low-level optimization will compress conv-weight automatically, the advantage is lower bandwidth and higher compute density for convolution.

### Dimension limitation

The row and column of matrix A, B and C MUST not exceed 16\*1024

### Supported Data Type

Only support float32 as input data in current version. Routine will convert it to float16 for internal processing. So, the precision cannot promise once value contained more than 4 digitals in fractional part.

## Performance tips

There are initialize cost for the first GEMM invoking, after that, successor GEMM invoking with same B should be faster than the first call.

## Benchmark:

|  |  |  |  |
| --- | --- | --- | --- |
| matrix A | matrix B | memory footprint (KB) | instruction counts |
| 1024\*1024 | 1024\*1024 |  |  |
| 64\*64 | 64\*64 |  |  |

This table will be filled by first software package release.