2.8.12. cublasGemmEx()

```
cublasStatus_t cublasGemmEx(cublasHandle_t handle,
                           cublasOperation_t transa,
                           cublasOperation_t transb,
                           int m,
                           int n,
                           int k,
                           const void    *alpha,
                           const void     *A,
                           cudaDataType_t Atype,
                           int lda,
                           const void     *B,
                           cudaDataType_t Btype,
                           int ldb,
                           const void    *beta,
                           void           *C,
                           cudaDataType_t Ctype,
                           int ldc,
                           cudaDataType_t computeType,
                           cublasGemmAlgo_t algo)
```

This function is an extension of `cublas<t>gemm` that allows the user to individually specify the data types for each of the A, B and C matrices, the precision of computation and the GEMM algorithm to be run. Currently supported combinations of arguments are listed further down in this section.

$C = \alpha \text{ op } (A) \text{ op } (B) + \beta C$

where α and β are scalars, and A , B and C are matrices stored in column-major format with dimensions $\text{op } (A)$ $m \times k$, $\text{op } (B)$ $k \times n$ and C $m \times n$, respectively. Also, for matrix A

$\text{op } (A) = A$ if transa == CUBLAS_OP_N A^T if transa == CUBLAS_OP_T A^H if transa == CUBLAS_OP_C

and $\text{op } (B)$ is defined similarly for matrix B .

Param.	Memory	In/out	Meaning
handle		input	handle to the cuBLAS library context.
transa		input	operation op(`A`) that is non- or (conj.) transpose.
transb		input	operation op(`B`) that is non- or (conj.) transpose.
m		input	number of rows of matrix op(`A`) and `C`.
n		input	number of columns of matrix op(`B`) and `C`.

Param.	Memory	In/out	Meaning
k		input	number of columns of op(A) and rows of op(B).
alpha	host or device	input	scalar scaling factor for A*B; of same type as computeType.
A	device	input	<type> array of dimensions lda x k with lda>=max(1,m) if transa == CUBLAS_OP_N and lda x m with lda>=max(1,k) otherwise.
Atype		input	enumerant specifying the datatype of matrix A.
lda		input	leading dimension of two-dimensional array used to store the matrix A.
B	device	input	<type> array of dimension ldb x n with ldb>=max(1,k) if transa == CUBLAS_OP_N and ldb x k with ldb>=max(1,n) otherwise.
Btype		input	enumerant specifying the datatype of matrix B.
ldb		input	leading dimension of two-dimensional array used to store matrix B.
beta	host or device	input	scalar scaling factor for C; of same type as computeType. If beta==0, C does not have to be a valid input.
C	device	in/out	<type> array of dimensions ldc x n with ldc>=max(1,m).
Ctype		input	enumerant specifying the datatype of matrix C.
ldc		input	leading dimension of a two-dimensional array used to store the matrix C.
computeType		input	enumerant specifying the computation type for cublasGemmEx.
algo		input	enumerant specifying the algorithm for cublasGemmEx.

Computation type supported by cublasGemmEx are listed below :

computeType
CUDA_R_16F

computeType
CUDA_R_32F
CUDA_R_32I
CUDA_R_64F
CUDA_C_32F
CUDA_C_64F

For CUDA_R_16F computation type the matrix types combinations supported by `cublasGemmEx` are listed below :

A	B	C
CUDA_R_16F	CUDA_R_16F	CUDA_R_16F

For CUDA_R_32I computation type the matrix types combinations supported by `cublasGemmEx` are listed below. This path is only supported with alpha, beta being either 1 or 0; A, B being 32-bit aligned; and lda, ldb being multiples of 4.

A	B	C
CUDA_R_8I	CUDA_R_8I	CUDA_R_32I

For CUDA_R_32F computation type the matrix types combinations supported by `cublasGemmEx` are listed below

A	B	C
CUDA_R_16F	CUDA_R_16F	CUDA_R_16F
CUDA_R_16F	CUDA_R_16F	CUDA_R_32F
CUDA_R_8I	CUDA_R_8I	CUDA_R_32F
CUDA_R_32F	CUDA_R_32F	CUDA_R_32F

For CUDA_R_64F computation type the matrix types combinations supported by `cublasGemmEx` are listed below :

A	B	C
CUDA_R_64F	CUDA_R_64F	CUDA_R_64F

For CUDA_C_32F computation type the matrix types combinations supported for cublasGemmEx are listed below :

A	B	C
CUDA_C_8I	CUDA_C_8I	CUDA_C_32F
CUDA_C_32F	CUDA_C_32F	CUDA_C_32F

For CUDA_C_64F computaion type the matrix types combinations supported by cublasGemmEx are listed below :

A	B	C
CUDA_C_64F	CUDA_C_64F	CUDA_C_64F

cublasGemmEx routine is run for the following algorithm.

CublasGemmAlgo_t	Meaning
CUBLAS_GEMM_DFALT	Apply Heuristics to select the GEMM algorithm
CUBLAS_GEMM_ALGO0 to CUBLAS_GEMM_ALGO13	Explicitly choose an algorithm
CUBLAS_GEMM_DEFAULT_MATRIX_MATH	Apply Heuristics to select the GEMM algorithm, while allowing the library to use multi-element dot product math operations if supported by hardware

The possible error values returned by this function and their meanings are listed below.

Error Value	Meaning
CUBLAS_STATUS_SUCCESS	the operation completed successfully
CUBLAS_STATUS_NOT_INITIALIZED	the library was not initialized
CUBLAS_STATUS_ARCH_MISMATCH	cublasCgemmEx is only supported for GPU with architecture capabilities equal or greater than 5.0
CUBLAS_STATUS_NOT_SUPPORTED	the combination of the parameters Atype, Btype and Ctype and the algorithm type, algo is not supported
CUBLAS_STATUS_INVALID_VALUE	the parameters m,n,k<0

Error Value	Meaning
`CUBLAS_STATUS_EXECUTION_FAILED`	the function failed to launch on the GPU

For references please refer to:

Read more at: http://docs.nvidia.com/cuda/cublas/index.html#ixzz4hgK5yKE2
Follow us: @GPUComputing on Twitter | NVIDIA on Facebook