本文档中有两个例子，第一个为通过CUDA实现hello world,该例子将详细进行讲解。第二个为通过CUDA实现逻辑斯特回归。

例一：

第一步：以下为先通过C语言实现hello world；

|  |
| --- |
| #include<stdio.h>  int main(void){  printf("hello world!\n");  } |

以上是C语言代码，编译，然后执行；

|  |
| --- |
| gcc hello.c -o hello  ./hello |

可以看到屏幕上会输出

|  |
| --- |
| hello world! |

第二步：以下是通过通过cuda实现；

|  |
| --- |
| #include <stdio.h>  \_\_global\_\_ void helloCUDA(void)  {  printf("hello cuda!\n");  }  int main(void)  {  helloCUDA <<<1,10>>>();  cudaDeviceReset();  return 0;  } |

可以看出，以上代码和C语言无太大区别，只是限定符\_\_global\_\_定义了一个核函数helloCUDA(void)，告诉编译器该函数将在GPU中执行。

helloCUDA <<<1,10>>>();

括号中的1和10表示该函数将有10个进程；

cudaDeviceReset()用来显式的摧毁清理CUDA程序占用的资源。

以下是编译和执行指令；

|  |
| --- |
| nvcc -arch sm\_20 hellocuda.cu -o hellocuda  ./hellocuda |

-arch sm\_20是用来指定编译器使用Fermi架构产生device代码。

一个典型的CUDA程序会包括五个主要步骤：

1. 分配GPU空间；
2. 将数据从CPU端拷贝到GPU端；
3. 调用CUDA kernel从GPU执行计算；
4. GPU完成计算后，将数据拷回CPU；
5. 清理GPU内存空间；

例二：通过CUDA实现逻辑斯特回归

以上两个程序的代码查看当前目录下hello.tar.gz和logistic.tar.gz。