

Neural Network

Procesadores de Alto Rendimiento

22/01/2023

Índice

[Índice](#)

[Introducción](#)

[Alojamiento](#)

[Aprendizaje](#)

[Paralelización](#)

[Reparto](#)

[Validación y Comparación](#)

[Conclusión](#)

[Bibliografía](#)

Introducción

Para llevar a cabo la paralelización del aprendizaje y validación de la red neuronal, he seguido una serie de pasos, que bien no creo que sean los mejores, pero debido a las complicaciones respecto a este proyecto son las que me he visto obligado a tomar.

Alojamiento

Para alojar la memoria de la GPU (Graphics Processing Unit), mi primera idea ha sido hacer uso de una o varias funciones que realicen el alojamiento y retornen su dirección para posteriormente hacer uso de estos direccionamientos a lo largo de todo el código. Esta idea ha sido descartada por varios motivos^[1] y tras muchas vueltas, he decidido clonar el archivo *matrix.c* renombrándolo como *matrixGPU.cu* y amoldar las funciones ya existentes alojando la memoria necesaria para cada una de ellas. De la misma forma he copiado las definiciones de las funciones del archivo *matrix.h* y las he modificado en su respectivo archivo *matrixGPU.cuh*.

Aprendizaje

Paralelización

En este apartado, he hecho uso de las funciones modificadas en *matrixGPU.cu* para invocar los *kernels* necesarios en cada función. Pese a haber creado los *kernels* para varias de las funciones que realizan alguna operación con matrices, he de decir que no he hecho uso de todos ellos puesto que por una parte me he quedado corto de tiempo y por otra he determinado que ciertas operaciones eran mejor realizarlas con la CPU (Central Processing Unit).

Los *kernels* implementados están verificados en el servidor de *dif-cluster*, pero no me ha dado tiempo a completar las funciones que invocan a estos.

Reparto

Para la estrategia de reparto en mi caso he usado una dimensión tanto para el tamaño del bloque como para el de la rejilla. Aunque me he informado respecto al reparto, que ambos sean de una dimensión no es debido a ninguna mejora de rendimiento, si no a la comodidad para realizar las operaciones.

Validación y Comparación

Por desgracia, tanto para la validación como para la comparación entre la versión CPU contra la de GPU, no tengo mucho que aportar ya que no he logrado terminar la versión GPU.

En la parte de validación si que he completado las 3 funciones que se requerían pero no tengo la certeza de que funcionen siquiera.

Conclusión

El tiempo que he podido dedicarle tanto a los ejercicios como al proyecto ha sido alrededor de 40 horas repartidas en 6 días. Considero que no ha sido suficiente y que he perdido la mayor parte del tiempo interpretando el proyecto y sobre todo planeando el alojamiento de las matrices.

Bibliografía

Todos los enlaces disponibles a fecha 22/01/2023

Traspuesta:

<https://stackoverflow.com/questions/10163948/multiplying-two-matrices-with-different-dimensions>

cudaMalloc():

<https://stackoverflow.com/questions/5885195/using-cudamalloc-to-allocate-a-matrix>

Bloque y rejilla^[2]:

http://www2.izt.uam.mx/newpage/contactos/anterior/n80ne/cuda_cts.pdf