基於非揮發性記憶體特性優化大型深度神經網路模型之運算效能

專題編號：111-CSIE-S017
執行期限：110 年第 1 學期至 111 年第 1 學期
指導教授：陳碩漢
專題參與人員： 108590045 廖永誠

一、摘要

本研究計畫主要著墨在如何優化深度學習框架 Tensorflow[1]內部運算的部分，由於觀察到在模型運算時，會有大量 Tensor [2]張量運算，在過程中會消耗到大量的時間以及內存，且在過去的研究中基本是朝模型架構、分散運算等等的方式來優化，研究目的為改善運算本身的並不多，也因此本專題研究期望在運算時透過非揮發性記憶體 Skyrmions Racetrack Memory(SK-RM)[3]來協助運算，由於其讀取快，存儲密度高等等的特性，以此來取代或同時兼容傳統的顯存或 DRAM，這樣做不僅可以在運算時加快了資料 IO 的進程也可以降低內存的負擔。

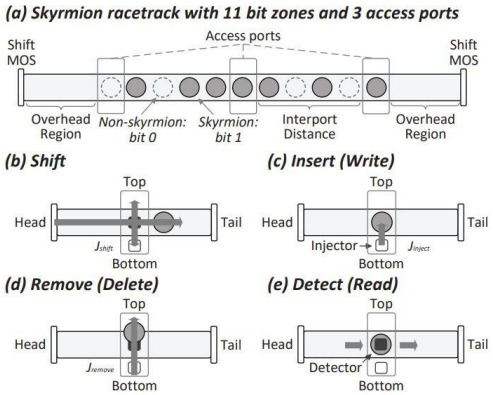
關鍵詞:NLP、DeepLearning、NVM

二、緣由與目的

目前語言模型主流的深度模型架構都是使用 Transformers 的架構來實現，而該架構的實現方法是透過 self-attention [4]的機制來去找到序列資料裡面的關聯關係，該架構表現都強大，但其需要大量的參數去紀錄，這樣產生的第一個問題為內存占用過大，再來就是推理的過程以及梯度下降的過程運算量很大，將長時間佔用運算能力。

Skrmion Racetrack Memory(Sk-RM)具有非揮發性的特性，且跟 SRAM 與 DRAM 相比有相似的訪問性能和更高的存儲密度，SK-RM 是透過兩端或者在 access port[3]施加電流來操作 Skyrmions 粒子，並以此來實現偵測、移動、刪除、新增等等的動作。

SK-RM 有一特性即是要讀取到粒子的狀態一定得將其 shift 移位到 Access port 上，所以資料的擺放將會很重要，不然即會有多餘的 shift operations。另一點需要特別注意的是 insert operation，其的能耗跟耗時相比其他的 operation 有明顯的差距，因此如何解決其能耗跟耗時也是很重要的課題之一。



圖(1). SK-RM 的架構以及 operations

Operations	Read	Shift	Remove	Insert
Latency	0.1 ns	0.5 ns	0.8 ns	1 ns

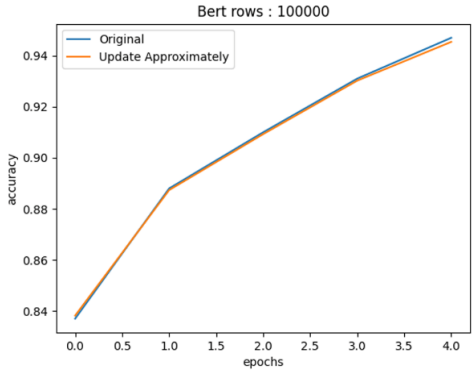
圖(2). SK-RM 各個 operations 的操作延遲

三、基於神經網路容錯性的改善

基於 ieee754 的浮點數儲存特性以及神經網路一定的容錯性，觀察到了 bits 越靠近 LSB 對於結果的影響力越小。

SK-RM 的更新寫入機制可以保留固有的 bits 來進行再利用，但是如果有多餘或不夠的仍需要透過 insert&remove 這兩個較耗能的 operation 來實現，而本專題固定 1 的數量來表示 float，雖然不能保證 100%準確，但可以從圖(3)看到對於神經網路模型的 Accuracy 影響很小。而由於 SK-RM 可以重複運用，因此固定 1 即是

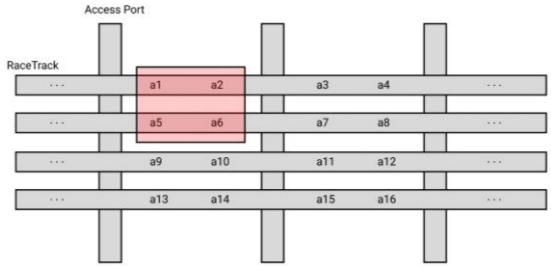
完全不需要去進行 insert & remove 的操作，因此可以最大程度的減少能耗。



圖(3). Bert 套用近似演算法後的精度比較

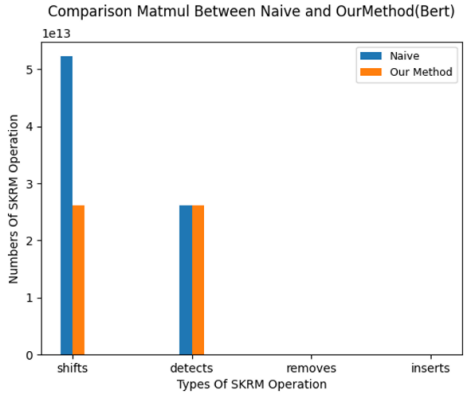
四、 基於 Multiply 讀取特性來減少操作

本專題觀察到 Tensorflow 在實現 Matrix Multiply 時會反覆重複讀取 bits，而對於 SK-RM 的讀取機制來說非常不利，因為需要將 Skyrmions 讀出再放回。



圖(4). 將矩陣的資料在 SK-RM 的 Layout

為了改善此問題本專題設計了一個符合矩陣資料結構並能放到 SK-RM 裡的 Layout，並以此為基礎提出一整套讀取機制，主要運用到的概念是透過新增一 Buffer Track 來在讀取的過程移動到別處，省略掉了放回去的步驟，經過實際的實驗驗證後證實可以減少將近 50%的 Operations。



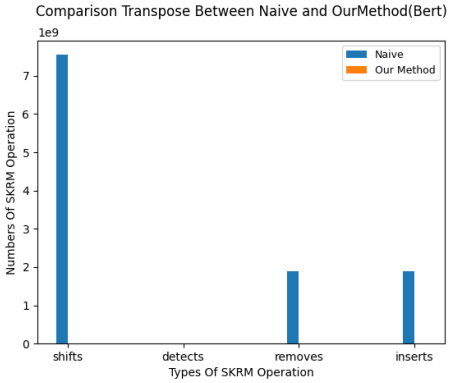
圖(4). Bert 計算矩陣相乘需要的操作比較

五、 基於 Transpose 轉換機制來跳過操作

基於本專題提出的 Data Layout，當矩陣發生 Transpose 後，邏輯上的行列互換實際上會對應到 Racetrack、Offset 互換。

而 Tensorflow 的實現機制會計算出新的 index 並以此為基礎來重新寫入一個新的 Storage，而本專題觀察到這樣的過程會產生很多 SK-RM 的 Operation，因此提出了能跳過 Transpose 的辦法。

透過考慮二維、多維的置換特性，以及資料的擺放方式，透過設計演算法來回推原有的 index，實際上可以跳過資料的更新，也因此能節省大量的運算請求。



圖(5). Bert 跳過 Transpose 能減少的操作數

六、 結論

本專題基於 Tensorflow、SK-RM 等得的特性，探討了多個核心使用場景會出現的問題，透過一系列的解決方案，本專題顯著地改善了這些問題。

參考文獻

[1] M, Abadi, P. Barham, Tensorflow: A System For Large-Scale Machine Learning, 2016
[2] Y. Panagakis, Tensor Methods in Computer Vision and Deep Learning, 2017
[3] T. Yang, M. Yang, J. Li, W. Kang, Permutation-Write: Optimizing Write Performance and Energy for Skyrmion Racetrack Memory, 2020
[4] A. Vaswani, N. Shazeer, Attention Is All You Need, 2017