109613013 廖國評 Homework 2

GTC 2023

演講主題：Running 100,000-Atom Molecular Dynamics with Accurate NN Potential on a GPU using Automatic Recomputation

心得：

會選擇此演講觀看，主要是因為這學期剛好接觸到了分子動力學與 Neural Network Potential 的相關的研究，對於這方面有興趣但所知甚少，因此想要藉由這個機會增進相關知識。其次則是因為我自己在使用個人電腦執行深度學習任務時，經常遇到 GPU 記憶體不足等問題，一般常見的縮小 batch size 或是在迴圈之間清理記憶體等方法效果很有限，而這場演講正提供了一個不必升級硬體就能改善資源用量的方式，對於將來遇到資料量較大的任務時或許會有所幫助。

Neural Network Potential (NNP) 雖然以更快的計算速度取代了以 NFT（密度泛函理論）等理論計算材料勢能的方式，但隨著模型訓練時使用的 input data 增加，不免也會遇到計算資源不足的問題。此演講的主題著重在如何優化神經網路的訓練流程，讓資料量大的模擬能夠在單個 GPU 上執行。

相比其他類型的深度學習任務，NNP 之所以會使用到大量的 GPU 記憶體是因為在計算原子之間的作用力時，需要對能量進行微分，面對如此大量的微分運算就必須使用到和更新神經網路參數時相同的 Backward Propagation 技

術，這也是整個神經網路訓練流程中最佔 GPU 記憶體的環節，因為記憶體必須儲存 Forward pass 時所有節點的完整資訊才能在 Backward pass 時快速的算出每個節點微分後的值。對此，講者提出了一個名為 Recomputation 的方法，這個方法不會讓神經網路儲存所有節點的資訊，而是在 Backward pass 傳遞到某節點時，才去重新計算更新該節點梯度所需節點的值，舉例來說，若是神經網路中有形狀為(1, 1000)和(1000, 1)的兩個張量，並且在 Forward pass 經過一個加法節點，這樣電腦就必須儲存形狀為(1000, 1000)的張量，很佔記憶體空間，使用 Recomputation 則只需儲存原始的張量，只要在 Backward pass 時多計算一次加法即可。不過這個方法也有很明顯的問題，就是會提高計算量，講者提出的觀點是：多等一些時間總比因為記憶體空間不夠而跑出 Error 還要好。最後在講者提供的範例中，他使用了名為 ONNX 的 NN compiler 優化以 Pytorch 寫成的模型，來執行上述的 Recomputaion 技術，並且在以 15625 顆 Pt Bulk 為訓練資料的大型模型當中，僅增加了 11%的計算時間（約 200 ms）就減少了 75%的記憶體用量（約 44GB），相比沒有進行優化的模型，可以在同一個 GPU 上處理 4 倍大的 input 資料。

這場演講篇幅有限沒有提及太多技術細節，這種牽涉到計算資源的技術一定還會有各種軟硬體版本或相容性的問題，因此實際應用上應該不會是表面上看起來這麼容易，但整體演講我仍然覺得十分受用，Recomputation 這項技術讓一般資源有限的使用者只須犧牲一些時間，就帶來讓大量資料可以順利訓練

的可能性，對於學生來說是一大福音，而這種以小幅度犧牲換取更大回報的概念，運用在解決其他問題也是個很好的思考模式。