The Investigation of Quantization based on Yolov5

基於Yolov5的量化研究與軟體模擬

組別 ： A184　組員 ： 沈軒泓、周茂鈞　指導教授 ： 呂仁碩

摘要

隨著科技的發展，終端計算(edge AI)成為目前發展的主流，但受限於終端裝置的硬體設備效能十分有限，因此各式各樣的小型模型、把模型縮小的方法，以及類神經網路硬體加速運算的方法便隨之而生，而本專題要探討的加速方法為「量化 (quantization)」。

由於yolov5在模型僅僅29MB的情況下，可以達到不輸yolov4的效能，非常具有硬體化的潛力，因此本專題要探討對yolov5s的卷積層採用量化的方法進行加速後，對於結果會有怎麼樣的影響。

我們總共用三個版本的量化進行軟體模擬，分別是八位元、六位元以及四位元，並且比較其結果，最後我們發現我們確實可以在卷積層運算精度變小的情況之下，達到差不多的訓練結果，而運算精度變小就可以達到我們一開始設立的目標，也就是減少終端設備的運算資源，實現類神經網路晶片的運算加速。最後我們發現對所有卷積層做八位元量化最具有硬體實作的價值。

期許若未來有繼續研究下去的話，可以在考慮權重的分布後再進行量化，或是用硬體語言實作到FPGA板上。

軟體模擬

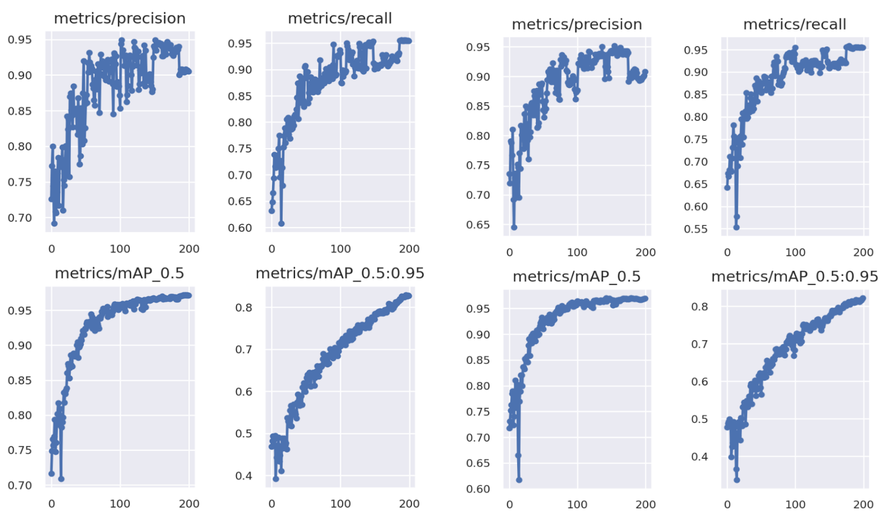
用Netron打開.onnx檔來檢查Yolov5s每個層級的運算大小後，發現模型中的卷積層平均每一層都有100000個以上的矩陣元素需要進行計算，而且每一個元素對應到的權重、偏壓值以及輸入值都是三十二位元的浮點數，初步估計，光是在卷積層就有七百多萬個權重參數，意謂光是考慮卷積層中的乘法浮點數運算，跑一輪訓練或是一次inference就需要作至少七百萬次的三十二位元浮點數乘法，更不用說還有與偏壓的加法、標準化過程，以及矩陣乘法等等運算過程，對於類神經網路的晶片設計來說頗不友善，很難在面積、功耗與效能之間做取捨。

因此我們打算從卷積層的改良下手，查閱了許多相關的期刊論文，研究了許多可以減少卷積層的運算量的方法之後，我們最終決定使用「量化(Quantization)」這個方法。原理十分簡單，根據統計[2]，權重在訓練時，大概都是分布於0附近的小數點，用軟體模擬時使用三十二位元精度的浮點數十分合理，但是在硬體設計上卻會導致極大的面積、功耗，以及走線，若是在不影響結果太多的情況下，將三十二位元精度的浮點數運算，改成八位元，甚至是八位元以下的精度在硬體去做運算，所有卷積層的面積、功耗可以省掉百分之七十五左右，甚至更多。

在本專題中，我們使用PyTorch進行實驗，且總共對Yolov5s用三個版本的量化進行軟體模擬，分別是八位元、六位元以及四位元進行量化去比較其結果。

軟體模擬結果

我們認為使用8位元量化全卷積層會在運算量以及訓練結果達到最佳的平衡，以下是訓練結果對比。



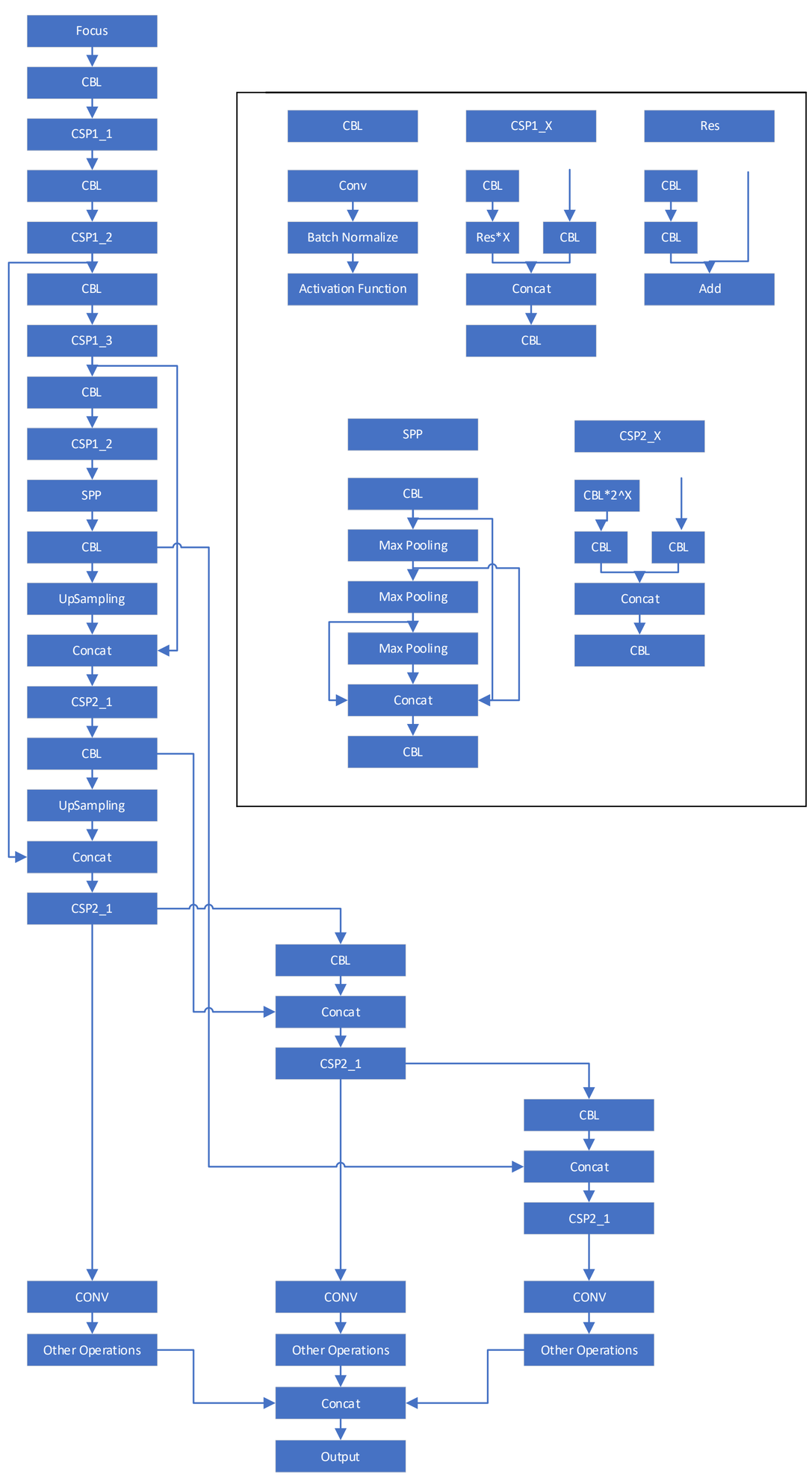
32位元精度模型的mAP

8位元量化全卷積層模型的mAP

32位元精度模型的inference結果

8位元量化全卷積層模型的inference結果

系統設計



Yolov5架構圖

結論

本專題針對Yolov5s的模型進行三種位元數做量化後進行軟體模擬訓練，權衡省下的運算量以及量化後造成的誤差之後，抓出了我們認為具有硬體實作意義的數據，也就是對於全部的卷積層進行八位元量化。

期許若未來有繼續研究下去的話，可以在考慮權重的分布後再進行量化[3] ，或是用硬體語言實作到FPGA板上。

[1] https://github.com/ultralytics/yolov5
[2] Energy-Efficient Neural Network Accelerator Based on Outlier-Aware Low-Precision Computation
[3] Mix and Match A Novel FPGA-Centric Deep Neural Network Quantization Framework