Analysis of Deep Learning Platform
深度學習平台分析

組別：A36　　指導教授：劉靖家　　組員：盧奕呈、莊喻捷、陳奕仁

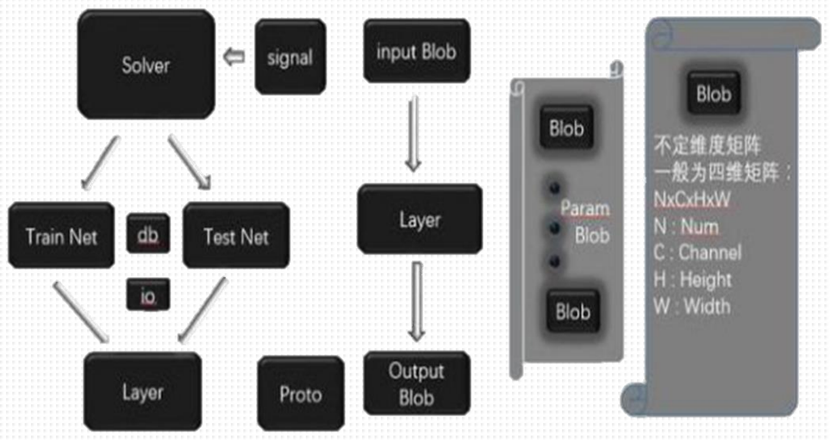
簡介

隨著深度學習的崛起，許多行動裝置已經想要把這項技術套用在裝置上，可惜的是現今的深度學習需要許多資料來跑出訓練的模型，而且為了處理這些海量的資料，又得耗上許多的時間，這些過程對於硬體資源有限的行動裝置而言是非常難應用的。

為了實現其應用性，我們決定先從改善訓練的效率著手，並且挑caffe這個深度學習的架構進行效率的改善，我們改變的效率有兩方面，一方面是改變caffe內最常用到的線性代數函式庫(以下都簡稱 library)的效率，另一方面則是改變precision來增加效率。

caffe的架構

caffe是用C++所寫成的深度學習架構，其基本可以分成四層的架構，由小到大分別為Blob、Layer、Net、Solver，Blob是用來儲存數據的四維數組，Layer是運用一個或多個Blob做輸入以及輸出，Net則是根據Layer的計算給出像對應的任務，最後Solver是協調Net使整個訓練出來的架構做一些參數及執行的模式。



過程&想法

為了要使的caffe運作得更加有效率一些，我們一開始就先去了解整個caffe的運作過程，然後得知了不管是哪一種training，都需要做許多的convolution，於是我們就去找跟這個相關的library，並且發現caffe預設跟convolution有關的library是單核的ATLAS，看到caffe是用單核的系統之後，我們很直觀的就想到，如果用多核心的設計，是否可以將training的速度簡短，增加其效率。

有了這初步的想法後，我們就開始找有沒有跟ATLAS類似的library，但是是可以適用於多核，最後找到兩個，一個是Intel MKL，另一個是OpenBLAS，由於前者是需要收費的，因此我們就使用OpenBLAS來做測試，

由於library已經換成OpenBLAS，因此 我們可以從原本的單核運算變成用八核心來做運算，實際操作過後的結果，雖然達成我們想要的效率提高，但是幅度並沒有太大，這是因為即使計算核心變多了，但傳遞資料的路徑卻沒有增加，所以才導致這樣的成果。

再來是改變precision的作法，因為caffe內預設的是32bit FP(floating point)，但有些需要的precision其實並不需要那麼高，於是就想到能不能把32bit FP改成16bit FP，如此一來便可以使用比較少的硬體資源，傳遞資訊的時候也會更加快速，使得整體的效率提升，但很可惜的caffe只有從32bit FP改成64bit FP的方式，因此只能作罷，但在此同時caffe2被釋出了，我們便轉而使用caffe2，結果發現它可以更改成16bit FP，但是目前這方面還只留給內部開發實驗用，所以無法調整，另外caffe2目前使用上依舊還有許多bug需要克服，如果要將我們在caffe中更改的東西套用進去的話，將會遇到更多的bug襲來，因此就被迫在這止步不前。

實驗結果

	caffe	caffe with OpenBLAAS	caffe with OpenBLAS_opt	caffe2
模式	單核	多核(八核)	多核(八核)	單核
運作時間	14m 6.718s	12m 16.968s	12m 30.222s	9m 44.930s
引用的線性代數函式庫	ATLAS	OpenBLAS	Optimized OpenBLAS	EIGENBLAS

備註：以上training都是用caffe內的例子「MNIST」

圖片來源：http://blog.csdn.net/frd2009041510/article/details/53706735