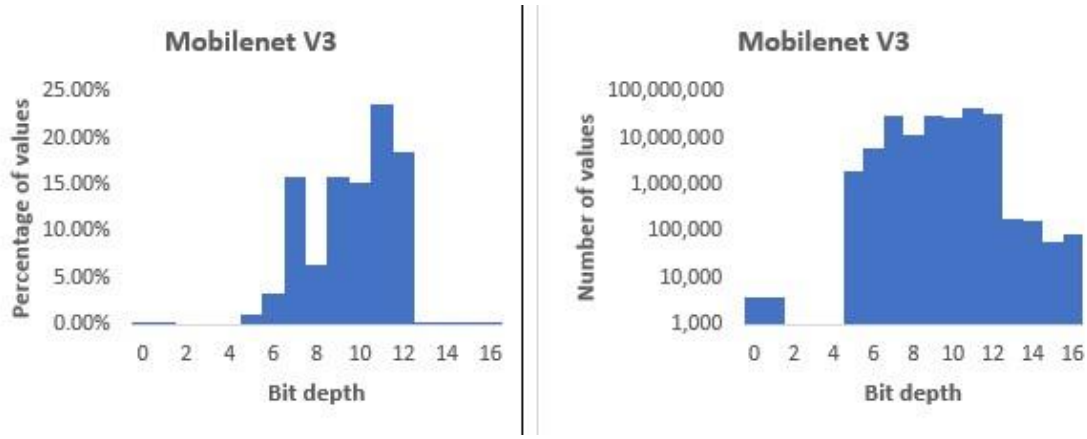
使用蒸餾法對低精度推論之浮點網路進行高傳真轉換

作者：Szabolcs Cséfalvay

[原文網址]

使神經網路加速器能夠進行快速且低功耗推論的一個主要挑戰，來自於模型的大小。 近年來，隨著模型尺寸的增加，運算時間和每次運算能耗也相應增加，神經網路邁向更深的神經網路發展，其啟動和係數也逐漸增加。 這在資源受限的行動裝置和汽車應用中尤為重要。低精度推論有助於透過降低 DRAM 頻寬（這是影響裝置能耗的一個重要因素）、運算邏輯成本和功耗來降低運算推論成本。

在這種情況下，以下的問題自然而然就出現了：編碼神經網路權重和啟動的最佳位元深度為何？ 有幾個建議的數位格式可以減少位元深度，包括 Nvidia 的 TensorFloat、Google 的 8 位元非對稱定點（Q8A）和 bfloat16。 雖然這些格式是朝著正確方向邁進，但不代表它們是最佳的，例如：大多數格式都是為了表示的每個值儲存一個指數，當多個值在同一區間時，這些指數可能是多餘的。更重要的是，他們並未考慮到神經網路的不同部分，一般具有不同的位元深度要求，有些圖層可以用較低的位元深度編碼，而其他圖層（如輸入和輸出層）則需要更高的位元深度。 MobileNet v3 就是一個例子，它可以從 32 位元浮點轉換為大多數在 5-12 位元區間內的位元深度（參考圖 1）。



我們稱使用不同位元深度來編碼原始浮點網路的不同部分，為可變位元深度（VBD） 壓縮。

當然，權重的位元深度編碼和網路準確度，也是需要權衡的。較低的位元深度會導致更有效的推論，但刪除過多的資訊則會降低準確性，這意味著，需要找到一個最佳的折衷方案。VBD 壓縮的目標是在壓縮和精度之間進行平衡。原則上，可以將其視為最佳化問題：我們希望盡可能用最少的位元數，達到最佳精度的網路。這是

透過在損失函數中添加新條件來實現的，該條件表示網路的大小，以便可以沿原始損失函數最小化，該函數可以大致表示為：

總損失 = 網路錯誤 + γ（網路大小）

其中 γ 是一個權重因數，用於控制網路大小和誤差之間的目標權衡。

因此，要以這種方式壓縮網路，我們需要兩個要件：精度的可微度量（錯誤）和網路大小的可微度量（壓縮位元深度）。網路大小項和網路誤差項，相對於位元數的可微分性非常重要，因為它使我們能夠最佳化（學習）位元深度。

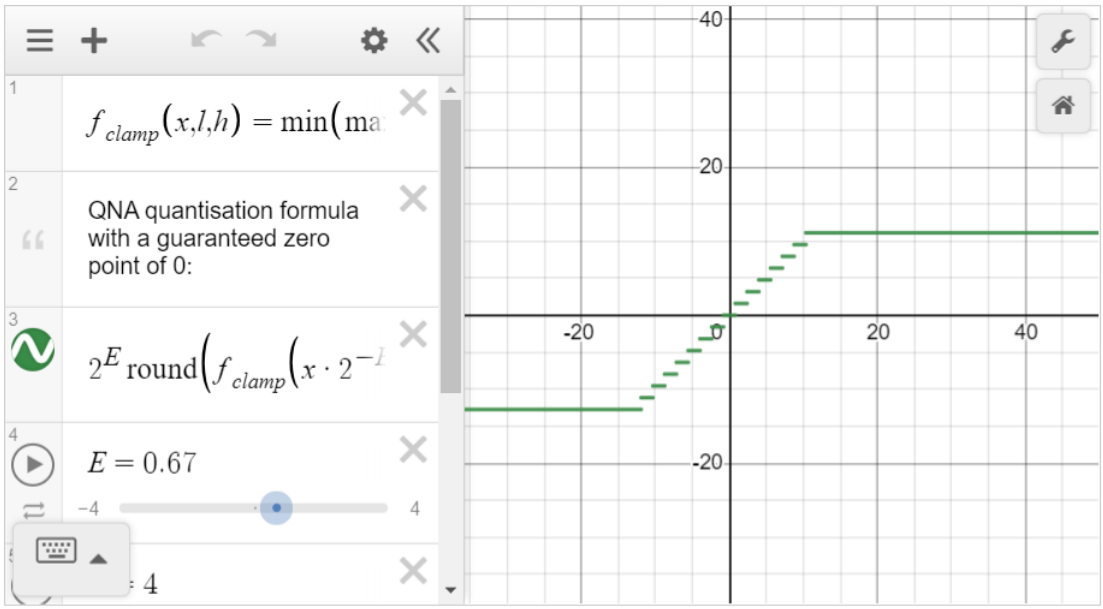
可微分的網路規模

可微分量化可透過任何具有可微分位元深度參數的函數進行，該參數將一個或多個浮點值，映射為硬體可表示的壓縮數位元格式。例如：可以使用 Google Q8A 量化的可變位元深度版本（其中可表示範圍不是以零為中心）：

$$f_q(x,E,B,\alpha) = 2^E \left\lfloor \min\left(\max\left(2^{-E}x, 2^{B-1}(\alpha-1)\right), 2^{B-1}(\alpha+1)-1\right)\right\rceil$$

其中：

- ⌊x⌉是 x 四捨五入至最近的整數（使用目標硬體平台的捨入模式）
- B 是位元深度。
- E 是浮點表示的指數。
- α 是不對稱參數。



在實踐中，量化參數 B、E 和 α 用於壓縮多個權重，例如，單個參數用於壓縮整個神經網路層（啟動或權重張量）或層內的通道。

我們還可以透過將不對稱參數設定為 0，來使用對稱量化（有效地將其轉換為縮放的 B 位元無符號整數格式）：

$$f_q(x, E, B) = 2^E \lfloor \min(\max(2^{-E}x, -2^{B-1}), 2^{B-1} - 1) \rceil$$

為了實現對所有參數的反向傳播，我們使用直通估計器(Straight-Through Estimator)，將圓形函數的梯度作為 1，這使得公式中的所有操作都是可微分的，使所有參數（包括位元深度）都可學習！

此時，我們可以選擇要訓練的參數：1.權重和最大壓縮位元深度。2.只有權重和指數（對於固定的位元深度）。3.只有量化參數，它有幾個優點（如下所述），成本可能更低的壓縮比。

在進行本文的結果時選擇了選項 3。

另一個需要考慮的方面是優化位元深度參數 B（對於某些格式的指數 E）：任何硬體都需要 B 是整數，而要找到整數解有多種選擇：

- 將四捨五入與直通估計器一起應用於 B 參數（例如，使用公式），但這給優化表面帶來了不連續性，雖然可處理但超出了本文的範圍。

$$2^E \left\lfloor \min\left(\max\left(x, -2^{\lfloor B \rceil - 1}\right), 2^{\lfloor B \rceil - 1} - 1\right)\right\rceil$$

- 這裡選擇的替代方案是在訓練的第一階段優化浮點數 B，「保守地」將其四捨五入到最近的整數[B]（否則啟動和權重張量的重要部分可能會被鉗制），將其固定為常量並繼續訓練。這樣平均會損失大約 0.5 位元的潛在壓縮，但保證不會發生不適當的裁剪。

可微分的精度測度

為了測量網路的準確性，我們可以簡單地使用網路最初訓練的相同損失函數。 然而，在許多應用中，目標是壓縮一個已經用 32 位元浮點訓練的網路，這意味著我們可以用蒸餾損失代替。意指壓縮網路的精度是根據原始網路的輸出，來衡量的。在這項操作中，選擇（輸出）logits 之間的絕對差異作為蒸餾損失，但也可以採用其他措施。

使用的蒸餾損失定義為：

$$\left|f(x) - f_q(x, E, B, \alpha)\right|$$

(等式 1)

請參閱圖 2 以說明這一點。

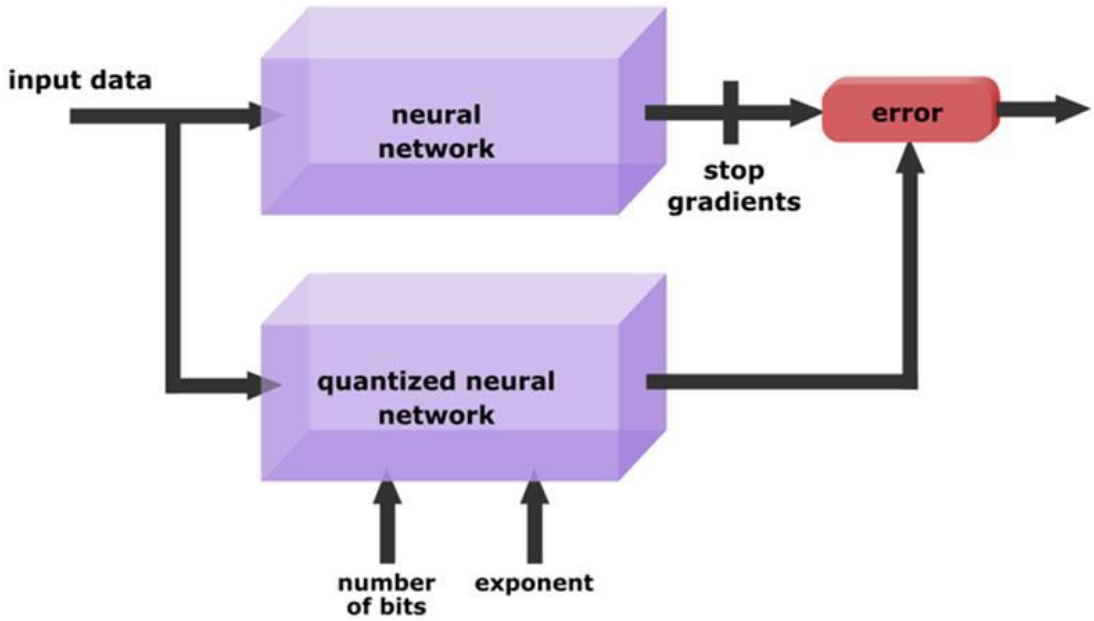


圖 2：用於壓縮的蒸餾損失的視覺化。

蒸餾損失有許多優點：

- 無標籤訓練：我們不需要標籤數據來壓縮網路，因為我們用原始網路的輸出取代了標籤。這意味著我們可以壓縮網路，而無需存取 (可能是專有的) 原始數據集；我們只需要具有代表性的輸入和原始網路。
- 通用性：神經網路壓縮機具有通用性，它不需要用於特定網路。
- 更少的訓練數據：由於我們只是訓練量化參數，過度擬合的範圍大大縮小，所以我們可以使用較少的訓練數據，有時候一張圖片就足夠了！
- 軟目標：由於從輸入得到的蒸餾損失，比標籤包含更多的量化誤差資訊，因此它能夠更快、更準確的收斂。
- 我們可以使用蒸餾損失將權重與量化參數一起訓練。然而，在這種情況下，我們需要更多的訓練數據來防止過度擬合，兩全其美的方法是以很小的學習率來訓練權重，並提前停止。如此，權重可以抵消量化誤差，而不會過度擬合數據集。

可微分的壓縮

將通用、可微分的精度度量與可微量化相結合，將得到可微壓縮的損失函數：

$$\left|f(x) - f_q(x, E, B, \alpha)\right|$$

(等式 2)

第一個項是誤差，第二項是網路大小的成本。B 是網路的平均位元深度，可根據整個網路的深度參數來計算：

$$B = \frac{\sum_i c_i B_i}{\sum_i c_i}$$

(等式 3)

其中 c_i 是使用位元深度參數，B_i 量化的網路參數（權重或啟動）的數量。請注意，此測量方式取決於批次大小。例如，如果對 32 個批次網路進行評估，則啟動張量的大小實際上比使用 1 批次大小的高出 32 倍。如果目標是將網路權重存儲在盡可能小的空間中，則此指標中也可以忽略啟動。

選擇量化粒度

一些神經網路表示，如 Google 的 Q8A 格式，允許將不同的比例係數（與上面的指數 E 的 2 次冪相關）應用於權重張量（篩檢程式）的不同通道。這種更精細的粒度可提高指定壓縮級別的網路精度。透過對每個通道應用單獨的 E 和 α 參數，同時對整個張量使用相同的 B 參數，可以透過可變位元深度壓縮實現相同的目標。然而，每個通道的量化會導致更慢的收斂，因此，根據我們的經驗，使用學習前張量參數的訓練階段更快，然後將這些參數分解為每通道參數，並讓它們在另一個訓練階段收斂。

這最終導致三個階段的訓練計劃：

- 根據張量訓練所有量化參數
- 切換到每通道指數和移位參數
- 將位元深度捨入到整數，並將其固定到常量，然後訓練指數和移位參數 α。此外，權重也以較小的學習率進行訓練。

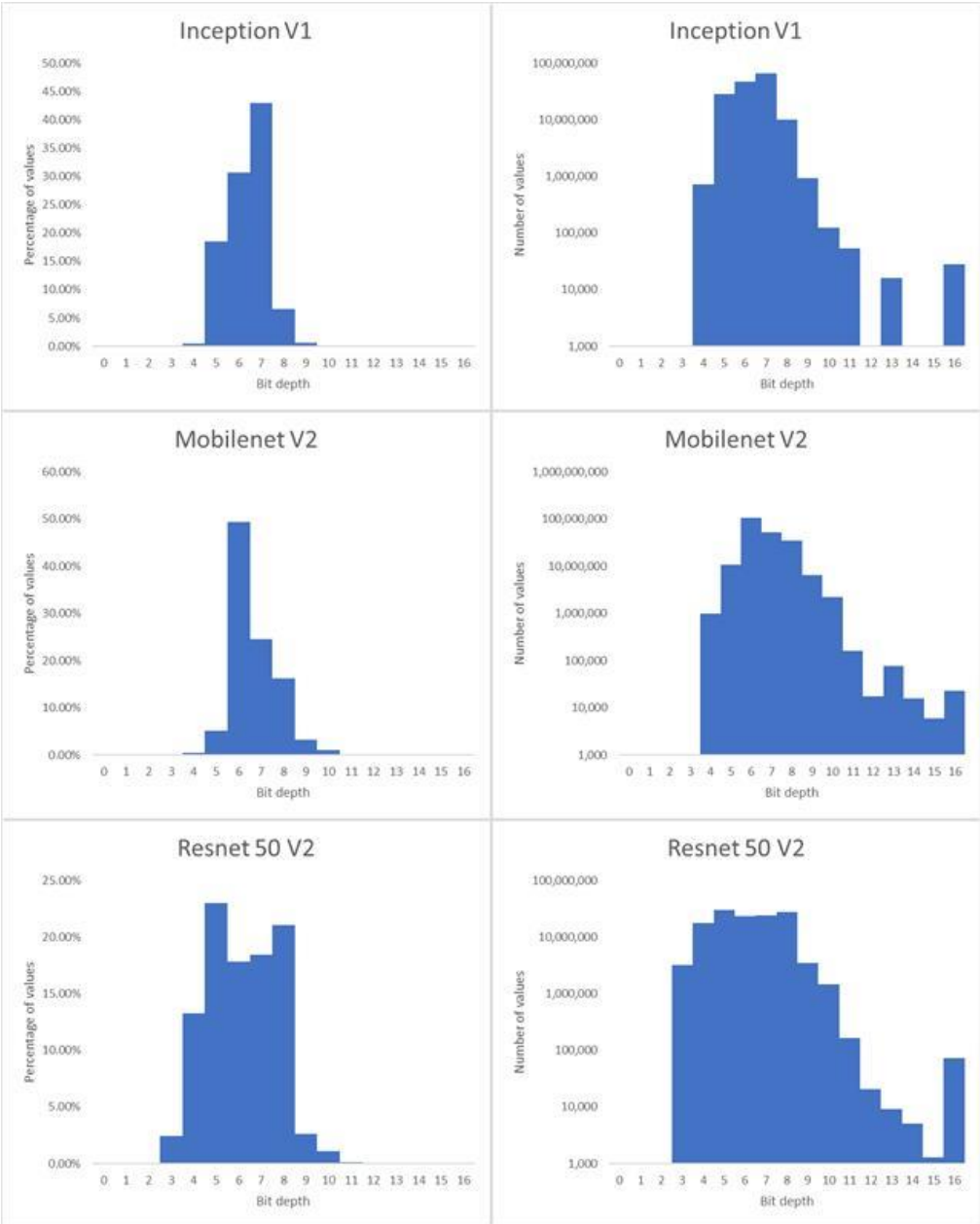
結果

物件分類

	Original		After Conversion			
	top1	top5	top1	top5	bits/t	Batch size
Mobilenet v2	71.58	90.49	71.46	90.36	**6.80**	**16**
Mobilenet v3	67.55	87.38	67.13	87.24	**8.43**	**32**
Inception v1	69.76	89.54	69.09	89.18	**6.34**	**16**
Resnet v2 (50)	69.73	89.41	67.88	88.19	**6.42**	**4**

壓縮分類網路的精度。圖 1 中的 Mobilenet v3 直方圖是不同（較短）訓練的結果，與此表中的網路相比，壓縮程度較低。

下面顯示了此表中選定網路的位元深度直方圖。

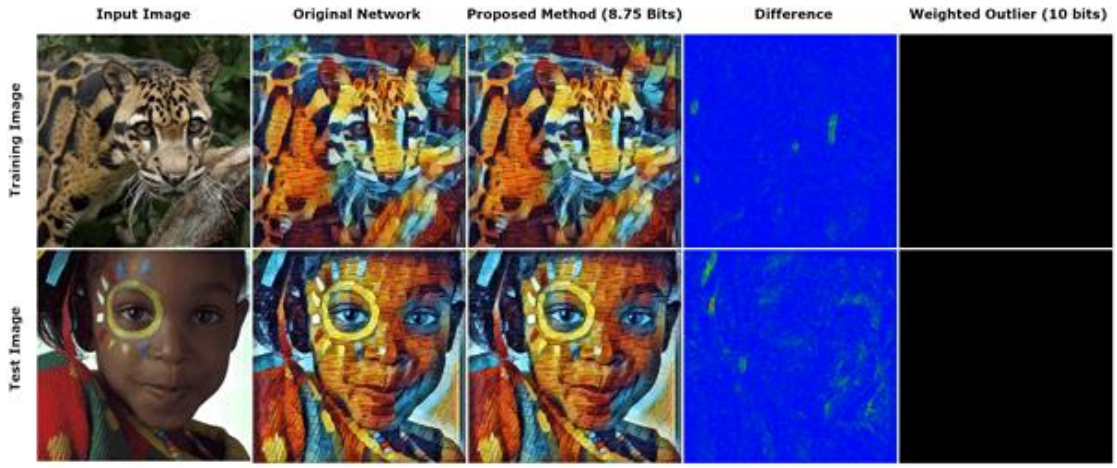


圖像分割

	Original	8 bits weighted outlier	8.72 bits (learned formats per tensor)	7.18 bits (learned formats per channel, weight tuning)
	mIOU: 0.669	0.571	0.670	0.668
				
				

應用於分割網路的不同壓縮方法之間的比較。 第二個列基於啟發式演算法，該演算法試圖可在不使用反向傳播的情況下，確定固定位深度的最佳指數。

風格轉換



最後一列使用上述啟發式給出 10 位元時的完全空白輸出。

結論

用於編碼神經網路權重和啟動的位元深度，對推理性能具有顯著的影響。 將大小精度權衡作為損失函數的一部分，可以在神經網路訓練過程中，學習任意粒度的最佳比特深度。 此外，當最佳化將 0 位元分配給網路的一部分時，它會有效地從架構中刪除該部分，作為一種架構搜索，進而降低運算成本和頻寬成本。 今後的工作便將進一步探索此方面的可微網路壓縮。

我們提出了一種基於微分量化和蒸餾的通用靈活方法，允許在不影響精度的情況下為各種任務最佳化位元數。 我們的方法有幾個優點，包括訓練時間短，重複使

用訓練過的網路，不需要標籤，可調整的大小精度權衡和問題無關的損失功能。 藉由這種方式，我們可以將網路壓縮為有效的可變位元深度表示法，而不犧牲對原始浮點網路的傳真度。

[原文網址]