Weight Fixing Networks - Additional Information

Christopher Subia-Waud & Srinandan Dasmahapatra

University of Southampton, Southampton, SO17 1BJ
{cc2u18, sd}@soton.ac.uk

1 Details of the Powers-of-two Approximation Algorithm

We map our proposal set C^S to a ω-order approximation where each of the clusters $c_k \in C^S$ are written as ω powers-of-two (Eq 1). We do so using Algorithm 1. Figure 1 demonstrates how the values of C^S are rounded given different orders.

$$c_k = \sum_{j=1}^{\omega} r_j,\ \ r_j \in \{-\frac{1}{2^b}, \dots, -\frac{1}{2^{j+1}}, -\frac{1}{2^j}, 0, \frac{1}{2^j}, \frac{1}{2^{j+1}}, \dots \frac{1}{2^b}\} \tag{1}$$

Algorithm 1: Determining possible clusters

Input: The full precision proposal set: C^S, allowable relative distance: δ, pow2 rounding function: $round(x) = sgn(x)2^{\lfloor \log_2(x) \rceil}$
Output: An order ω precision cluster set: $\widetilde{C}^{\omega}$
$\widetilde{C}^{\omega} \leftarrow [\ ]$
for $c_k \in C^S$ **do**
  $c'_k = round(c_k)$
  for $i = 0 \to \omega$ **do**
    $\delta_{c_k} \leftarrow c_k - c'_k$
    if $|\delta_{c_k}| \geq \delta c_k$ **then**
      $c'_k \leftarrow c'_k + round(\delta_{c_k})$
    end
  end
  $\widetilde{C}^{\omega} \leftarrow \widetilde{C}^{\omega} \cup \{c'_k\}$
end

2 Experiment Details

We give a full breakdown of the parameters used across all experiments ran in Table 1.

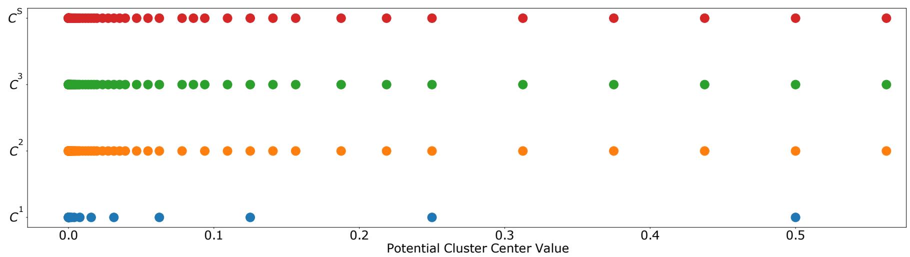

Fig. 1. Approximating clusters in C^S with different orders

Model	Data	Opt	LR	T	Batch size	γ	α
ResNet-18	ImageNet	Adam	2e-4	10	128	{0.05, 0.025, , 0.015, 0.01, 0.0075, 0.005}	{0.2, 0.4}
ResNet-34	ImageNet	Adam	2e-4	10	64	{0.05, 0.025, , 0.015, 0.01, 0.0075, 0.005}	{0.4}
ResNet-50	ImageNet	Adam	2e-4	10	64	{0.05, 0.025, , 0.015, 0.01, 0.0075, 0.005}	{0.4}
GoogLeNet	ImageNet	Adam	2e-4	10	64	{0.01, 0.0075, 0.015}	{0.4}
ResNet-18	CIFAR-10	Adam	3e-4	10	512	{0.01, 0.02, 0.03, 0.04, 0.05}	{0.0, 0.1, 0.2, 0.4, 0.8}
MobileNet	CIFAR-10	Adam	2e-4	10	512	{0.01, 0.02, 0.03, 0.04, 0.05}	{0.0, 0.1, 0.2, 0.4, 0.8}

Table 1. Full set of hyper-parameters explored for each model-dataset combination.

3 Additional Analysis

3.1 Layerwise Breakdown

In Figure 2 we examine how the parameter count and layer-parameter entropy change with each layer for both the WFN and APoT approaches. We find both gains over the unquantised layers of APoT, but also that the entropy and parameter count in the convolutional layers (those quantised by APoT) are similar.

3.2 A Full Metric Comparison

In Table 2 we give the full metric breakdown comparing WFN to the state-of-the-art APoT work. We calculate the unique parameter count and entropy values for subsets of the weights. No BN corresponds to all weights other than those in the batch-norm layers, and No BN-FL is the set of weights not including the first-last and batch-norm layers. It's clear here that WFN outperforms APoT even when we discount the advantage gained of taking on the challenge of quantising all layers.

3.3 Pruning Experiments

To explore how WFN interacts with pruning we conduct a simple set of experiments. Instead of starting the WFN process with all weights un-fixed we randomly select $p\%$ of the weights to be pruned in each layer. We then run WFN as before starting with $p_t = p$, reducing the number of T iterations. The results, shown in Figure 3, are conducted with a ResNet-18 and Cifar-10 combination, painting a mixed picture. On the one hand, WFN and pruning at lower

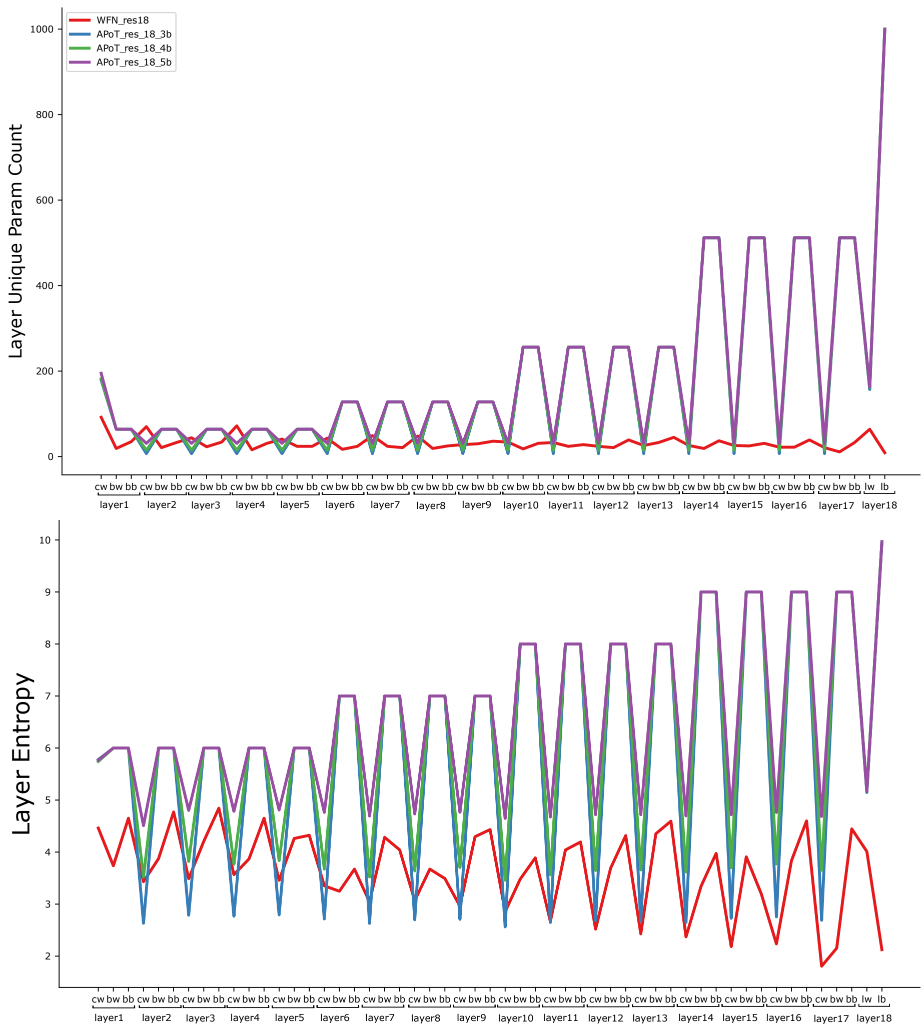

Fig. 2. We compare WFN with a traditional quantisation set-up (APoT) with varying bit-widths applied to a ResNet18 model trained on the ImageNet dataset. The top chart shows the layerwise unique parameter count where WFN has consistently fewer unique parameters per layer.

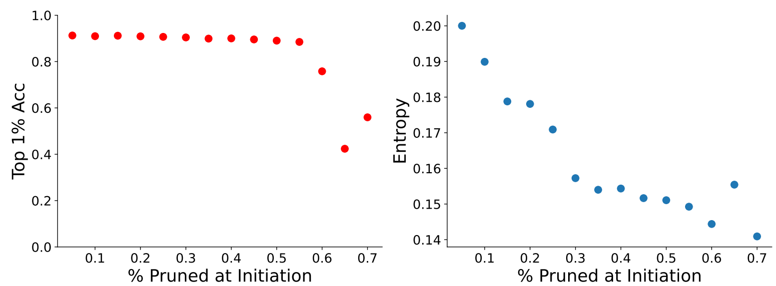

Fig. 3. Here we show that unstructured pruning at initialisation up to 50% can be coupled with the WFN process without degradation of performance and can further reduce the weight-space entropy.

			Full Network		**No BN**		**No BN-FL**	
Model	**Method**	**Top-1**	**Entropy**	**Param Count**	**Entropy**	**Param Count**	**Entropy**	**Param Count**
ResNet-18	Baseline	68.9	23.3	10756029	23.3	10748288	23.3	10276369
	APoT (3bit)	69.9	5.77	9237	5.76	1430	5.47	274
	WFN $\delta = 0.015$)	67.3	2.72	90	2.71	81	2.5	81
	WFN $\delta = 0.01$)	69.7	3.01	164	3.00	153	2.75	142
	WFN $\delta = 0.0075$)	70.3	4.15	193	4.13	176	3.98	162
ResNet-34	Baseline	73.3	24.1	19014310	24.1	18999320	24.10	18551634
	APoT (3bit)	73.4	6.77	16748	6.75	16474	6.62	389
	WFN $\delta = 0.015$)	72.2	2.83	117	2.81	100	2.68	100
	WFN $\delta = 0.01$)	72.6	3.48	164	3.47	132	3.35	130
	WFN $\delta = 0.0075$)	73.0	3.87	233	3.85	191	3.74	187
ResNet-50	Baseline	76.1	24.2	19915744	24.2	19872598	24.20	18255490
	WFN $\delta = 0.015$)	75.1	3.55	125	3.50	105	3.42	102
	WFN $\delta = 0.01$)	75.4	4.00	199	3.97	169	3.88	163
	WFN $\delta = 0.0075$)	76.0	4.11	261	4.09	223	4.00	217

Table 2. A full metric comparison of WFN Vs. APoT. We compare the unique parameter count and entropy of all parameters in the full network, as well as the same measures but not including the batch-norm layers (No BN) and the parameters not including the batch-norm and first and last layers (No BN-FL).

levels ($< 50\%$) is well tolerated and provide two benefits, a lower weight-space entropy and few weight-fixing iterations. On the other hand, full-precision networks can tolerate much higher ranges of pruning so there it would seem that a certain amount of synergy between the two approaches is present but this is tempered compared to full precision networks.

It's important to note that WFN already has a form of pruning built-in with the δ_0 value balancing the emphasis on pruning over quantisation.