Supplementary Materials for "Structured Pruning of Neural Networks with Budget-Aware Regularization"

1. Sigmoidal Transition Function

The sigmoidal transition function of Section 3.4 is defined as follows:

$$T(t, d) = \frac{\text{Sigmoid}(d(t - 0.5)) - \delta}{1 - 2\delta}$$
$$\delta = \text{Sigmoid}(-0.5d) \tag{1}$$

where d controls the "hardness" of the sigmoid curve. $d \to 0$ gives a linear transition, and $d \to \infty$ gives a hard sigmoid. $d = 10$ was found empirically to be effective. The sigmoid must be shifted by δ and scaled by $(1 - 2\delta)$ so that $T(0, d) = 0$ and $T(1, d) = 1$.

2. Mixed Block Implementation

In Algorithm 1 below, we present an implementation for the atypical connectivity presented in Figure 4 of the paper. While the input and delta tensors have the same number of prunable features, they can have a different pruning mask. We use indexing operations to exclude pruned features from the computations.

Note that, in the case where the block is responsible of performing a downsampling of the input tensor (or augmenting the number of features, or both), the residual connection is not an identity function; rather, it is a convolution $f_{\text{res}}(\cdot)$ that can have pruned features.

Algorithm 1: Forward Pass for the Mixed Block

Data:
x_{in} : Input tensor of size $(\text{batch\_size}, \text{n\_features}, \text{height}, \text{width})$
$f_{\Delta}(\cdot)$: Delta branch function
i_{in} : Indices of alive features in input tensor
i_{Δ} : Indices of alive features in delta tensor
$f_{\text{res}}(\cdot)$: Residual branch function (can be $\varnothing$ if identity connection)
i_{res} : Indices of alive features in output of residual function (if $f_{\text{res}}(\cdot) \neq \varnothing$)
N_{res} : Number of output features of unpruned residual function (if $f_{\text{res}}(\cdot) \neq \varnothing$)
Result: x_{out} : Output tensor

$x_{\text{alive}} \Leftarrow x_{\text{in}}[:, i_{\text{in}}, :, :]$
if $f_{\text{res}}(\cdot) = \varnothing$ **then**
  $x_{\text{out}} \Leftarrow i_{\text{in}}$
else
  $x_{\text{res}} \Leftarrow f_{\text{res}}(x_{\text{alive}})$
  $\text{height}, \text{width} \Leftarrow \text{get\_height}(x_{\text{res}}), \text{get\_width}(x_{\text{res}})$
  $x_{\text{out}} \Leftarrow \text{zeros}(\text{batch\_size}, N_{\text{res}}, \text{height}, \text{width})$
  $x_{\text{out}}[:, i_{\text{res}}, :, :] \Leftarrow x_{\text{res}}$
$\Delta \Leftarrow f_{\Delta}(x_{\text{alive}})$
$x_{\text{out}}[:, i_{\Delta}, :, :] \Leftarrow x_{\text{out}}[:, i_{\Delta}, :, :] + \Delta$

3. Training Schedules

Here, we give details about the number of epochs used for each training phase and for each method.

3.1. Training Schedule for LZR, IB, MorphNet and Our Method

LZR[22], IB[1] and Ours For CIFAR-10 and CIFAR-100, we first train for 80 epochs. Then, the network is "hard-pruned" using the algorithm associated to the pruning method. Next, we freeze Φ, and we train for 40 additional epochs. Finally, we reduce the learning rate from 10^{-3} to 10^{-4} and train for 10 more epochs. For Mio-TCD, we shorten the three stages to $(40, 20, 10)$ epochs (this dataset is $\sim 10\times$ larger). For Mio-TCD only, we initialize the weights to those of the full (unpruned) network (c.f. Section 3.3).

MorphNet[9] For CIFAR-10 and CIFAR-100, we first train for 60 epochs. Then, the network is "hard-pruned" using the algorithm proposed by the original paper. Next, we train for 50 additional epochs. Finally, we reduce the learning rate from 10^{-3} to 10^{-4} and train for 20 more epochs. For Mio-TCD, we shorten the three stages to $(40, 20, 10)$ epochs.

3.2. Pruning Scheme for Random, VM, VQ and ID

For *Random*, *WM*[10], *VQ*[8], and *ID*[6,16], we implemented the following pruning scheme which revealed to be effective and efficient:

1. Perform initial training of the full network for 40 epochs;
2. Reduce learning rate and continue training for 10 epochs;
3. Prune 50% of the network's volume V;
4. Train the network again with high learning rate, for 40 epochs;
5. Reduce learning rate and continue training for 10 epochs;
6. Record the network performance for the current pruning factor;
7. Return to 3 for the remaining pruning steps.

This scheme leads to four pruning factors: 2, 4, 8, 16.

3.3. Training Schedule for the Full (Unpruned) Networks

For CIFAR-10 and CIFAR-100, we trained for 80 epochs at learning rate 10^{-3}, and 10 more epochs at learning rate 10^{-4}. For Mio-TCD, a dataset $\sim 10\times$ larger, we trained for 40 epochs at high learning rate and 10 epochs at low learning rate.

4. Initialization of Dropout Sparsity Parameters Φ

For LZR, where $\Phi := \{\alpha_l\}$, we initialized all α from a uniform distribution $\mathcal{U}(0, 0.01)$. We did not observe a significant change between using $\mathcal{U}(0, 0.01)$ or $\mathcal{N}(0, 0.01^2)$ (the initialization distribution suggested by [22]). Our method has the same Φ and initialization scheme than LZR[22]. For IB, where $\Phi := \{\mu_l, \sigma_l\}$, we initialized the parameters from Gaussian distributions: $\mu \sim \mathcal{N}(1, 0.01^2), \log \sigma \sim \mathcal{N}(-9, 0.01^2)$ (values obtained by personal communication with the authors [1]).

5. Impact of Mixed-Connectivity Block on Metrics

Here we compare the objective metrics (i.e. Activation Volume V and FLOP) when the regular Resblock is used and when our Mixed-Connectivity block is used. These are the results obtained from our method (BAR). Δ is the relative difference to the results for the Regular block. The following three tables provide further details to Table 2 in the paper.

5.1. CIFAR-10

	Activation Volume (V)			FLOP		
Pruning Factor	Regular	Mixed	Δ	Regular	Mixed	Δ
2	1.79E+06	1.58E+06	-12%	2.71E+09	2.35E+09	-13%
4	1.37E+06	7.88E+05	-43%	1.51E+09	9.49E+08	-37%
8	8.43E+05	3.93E+05	-53%	8.06E+08	4.46E+08	-45%
16	4.67E+05	1.97E+05	-58%	4.20E+08	2.20E+08	-48%

5.2. CIFAR-100

	Activation Volume (V)			FLOP		
Pruning Factor	Regular	Mixed	Δ	Regular	Mixed	Δ
2	1.84E+06	1.58E+06	-14%	2.68E+09	2.29E+09	-15%
4	1.53E+06	7.88E+05	-49%	1.85E+09	1.10E+09	-40%
8	8.76E+05	3.93E+05	-55%	1.08E+09	6.03E+08	-44%
16	4.59E+05	1.97E+05	-57%	6.21E+08	3.27E+08	-47%

5.3. Mio-TCD

	Activation Volume (V)			FLOP		
Pruning Factor	Regular	Mixed	Δ	Regular	Mixed	Δ
2	2.66E+06	1.81E+06	-32%	1.15E+09	8.33E+08	-28%
4	1.45E+06	9.07E+05	-37%	2.50E+08	1.56E+08	-38%
8	7.51E+05	4.54E+05	-40%	6.87E+07	3.98E+07	-42%
16	4.70E+05	2.26E+05	-52%	3.31E+07	1.66E+07	-50%

6. Pruning Results

Here we give the numbers that we used to plot the curves of Figures 6 and 8. All methods prune according to the Activation Volume V, except for "Ours (F-trained)", which prunes with a FLOP reduction objective. Please note that results with a volume factor greater than 16 have not been included.

6.1. CIFAR-10

Method	V factor	F factor	Test Accu.	Method	V factor	F factor	Test Accu.
Random	2.0	4.0	0.8963	LZR	1.1	1.2	0.9182
	4.0	16.0	0.8802		2.1	5.3	0.9204
	8.0	64.0	0.8462		4.3	38.1	0.9210
	16.0	255.8	0.8136		7.3	62.0	0.9171
VQ	3.1	9.4	0.9124		15.8	167.4	0.8970
	4.6	21.0	0.9089	IB	1.3	1.8	0.9085
	8.1	64.9	0.8862		1.3	1.8	0.9128
	16.0	256.7	0.8459		2.3	5.2	0.9097
WM	2.0	4.0	0.9111		5.8	38.6	0.9048
	4.0	16.0	0.9120		7.4	79.9	0.9014
	8.0	64.0	0.8968		9.5	113.7	0.8965
	16.0	255.8	0.8633		15.2	196.4	0.8795
ID	2.0	4.0	0.9109	MorphNet	2.0	4.0	0.9325
	4.0	16.0	0.9144		4.0	16.0	0.9289
	8.0	64.0	0.9037		† 7.9	64.0	0.9066
	16.0	255.8	0.8692		† 15.8	255.8	0.8643
Ours	2.0	6.4	0.9270	Ours	1.7	4.0	0.9395
	4.0	15.9	0.9278	(F-trained)	2.9	16.0	0.9350
	† 7.9	33.8	0.9280		4.4	† 63.9	0.9217
	† 15.8	68.5	0.9162		6.1	256.0	0.9010

The results marked with a dagger (†) are slightly below target because of a slight miscalculation of the total volume and of the budget; this has since been fixed in our implementation.

6.2. CIFAR-100

Method	V factor	F factor	Test Accu.	Method	V factor	F factor	Test Accu.
Random	2.0	4.0	0.6751	LZR	1.0	1.1	0.7201
	4.0	16.0	0.6364		2.0	6.0	0.7039
	8.0	64.0	0.5922		3.0	31.9	0.7152
	16.0	255.8	0.4888		6.6	57.5	0.7075
VQ	2.5	9.4	0.6963		13.0	110.6	0.6728
	4.3	21.0	0.6927	IB	1.2	1.5	0.7127
	8.0	64.9	0.6703		1.3	1.8	0.7093
	16.0	256.7	0.5899		2.0	3.1	0.7079
WM	2.0	4.0	0.6898		4.8	24.9	0.6508
	4.0	16.0	0.6910		6.3	58.5	0.6905
	8.0	64.0	0.6542		8.8	94.3	0.5709
	16.0	255.8	0.5899		12.7	148.5	0.5671
ID	2.0	4.0	0.6929	MorphNet	2.0	4.0	0.7359
	4.0	16.0	0.6975		4.0	16.0	0.7042
	8.0	64.0	0.6603		† 7.9	64.0	0.6494
	16.0	255.8	0.5913		† 15.8	255.8	0.5549
Ours	2.0	6.6	0.7408				
	4.0	13.7	0.7359				
	† 7.9	25.0	0.7259				
	† 15.8	46.2	0.7053				

The dagger (†) has the same meaning as in the previous table.

6.3. Mio-TCD

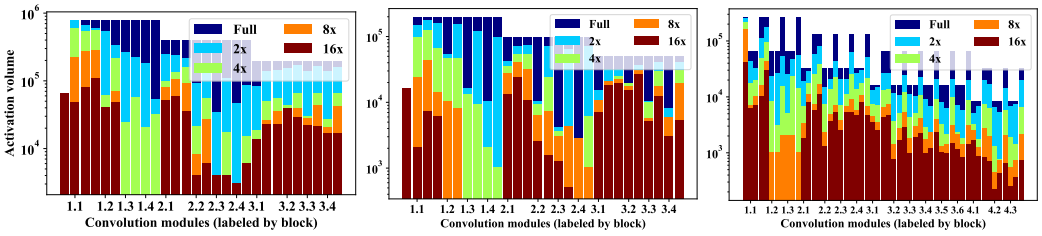
Method	V factor	F factor	Test Accu.	Method	V factor	F factor	Test Accu.
Random	2.0	4.0	0.9417	LZR	1.0	1.0	0.9521
	4.0	16.0	0.9330		1.0	1.0	0.9509
	8.0	63.7	0.9191		1.0	1.0	0.9524
	16.0	253.7	0.9119		1.6	2.9	0.9524
VQ	3.6	12.6	0.9504		3.8	14.8	0.9507
	5.5	30.0	0.9472		11.7	201.6	0.9510
	8.7	75.8	0.9448	IB	1.4	2.3	0.9509
	16.3	266.3	0.9343		1.9	3.1	0.9516
WM	2.0	4.0	0.9497		7.5	57.3	0.9478
	4.0	16.0	0.9507		11.5	145.0	0.9211
	8.0	63.7	0.9488		14.2	192.6	0.9127
	16.0	253.7	0.9406		15.2	232.6	0.9070
ID	2.0	4.0	0.9512	MorphNet	2.0	4.0	0.9709
	4.0	16.0	0.9493		4.0	16.0	0.9681
	8.0	63.7	0.9461		8.0	63.7	0.9590
	16.0	253.7	0.9376		16.0	253.7	0.9396
Ours	2.0	3.1	0.9536				
	4.0	16.7	0.9543				
	8.0	65.1	0.9567				
	16.1	156.7	0.9534				

6.4. TinyImageNet

Method	V factor	F factor	Test Accu.	Method	V factor	F factor	Test Accu.
Random	2.0	4.0	0.4825	Ours	2.0	4.0	0.5235
	4.0	16.0	0.4608		4.0	16.0	0.5198
	8.0	63.7	0.3941		8.0	63.7	0.5140
	16.0	253.7	0.2953		16.0	253.7	0.5196
VQ	2.5	6.2	0.4896	LZR	3.4	38.8	0.5034
	4.2	18.0	0.4993		4.0	55.3	0.5001
	8.0	64.3	0.4842		5.5	73.8	0.4971
	16.0	255.9	0.3926		9.6	108.3	0.4903
WM	2.0	4.0	0.4901	IB	2.0	3.9	0.4445
	4.0	16.0	0.4967		4.2	18.0	0.4282
	8.0	63.7	0.4772		6.4	41.3	0.3757
	16.0	253.7	0.4019		12.7	161.7	0.3513
ID	2.0	4.0	0.4996	MorphNet	2.0	4.0	0.5815
	4.0	16.0	0.4955		4.0	16.0	0.5577
	8.0	63.7	0.4577		8.0	63.7	0.5169
	16.0	253.7	0.3972		16.0	253.7	0.3919

7. Pruning Results

In Fig. 1, we show pruning results of our method on TinyImageNet, CIFAR-100, and Mio-TCD (result on CIFAR-10 is in the paper).



(a) WideResNet-TinyImageNet (b) WideResNet-CIFAR-100 (c) ResNet-Mio-TCD

Figure 1: **Result of pruning with our method**. Total number of active neurons in the full networks and with four different pruning rates. Sections without an orange (8x) or red (16x) bar are those for which a res-Block has been eliminated.

8. Sensitivity Analysis

Here we show sensitivity analysis results for our method on all 4 datasets. For each of the 4 pruning factors, we run 10 experiments, where we sample the value of 7 hyperparameters from a uniform distribution centered around their tuned value. Depending on the scale of the parameter, the width of the interval is either 10^{-5}, 0.2, or 1.0. We have plotted the results of this analysis in Fig. 2, using the data shown in Table 1. As one can see, our method is not over sensitive to changes of its hyperparameters.

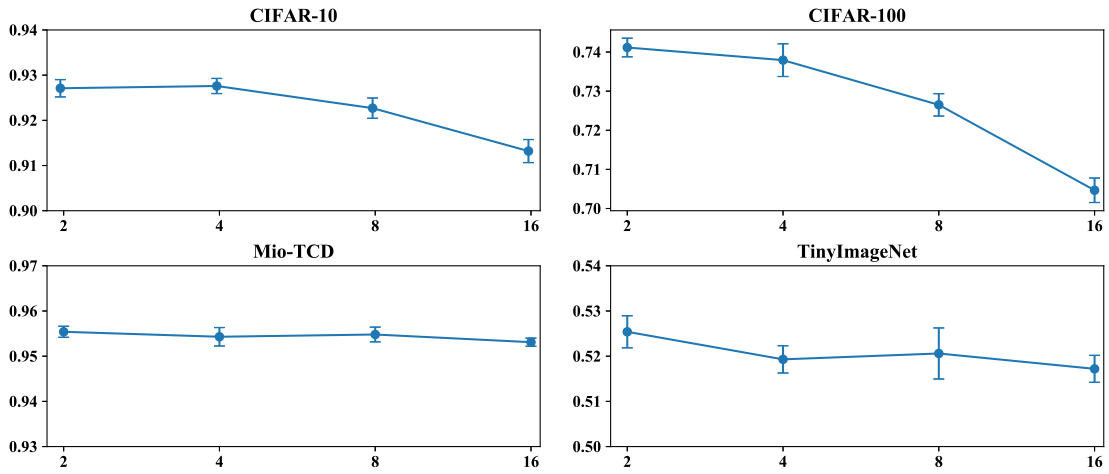


Figure 2: Sensitivity analysis with error bars

V Factor	Test Accuracy	Std dev.
2	0.9271	0.001915
4	0.9276	0.001674
8	0.9227	0.002244
16	0.9132	0.002552

(a) CIFAR-10

V Factor	Test Accuracy	Std dev.
2	0.7412	0.002384
4	0.7379	0.004185
8	0.7265	0.002860
16	0.7047	0.003129

(b) CIFAR-100

V Factor	Test Accuracy	Std dev.
2	0.9554	0.001222
4	0.9543	0.002039
8	0.9548	0.001631
16	0.9531	0.000903

(c) Mio-TCD

V Factor	Test Accuracy	Std dev.
2	0.5254	0.003548
4	0.5193	0.003017
8	0.5206	0.005651
16	0.5172	0.002984

(d) TinyImageNet

Table 1: Sensitivity analysis data

9. Dropout Sparsity Learning with the Hard Concrete Distribution

We describe the Hard Concrete distribution, and how it can be used for Dropout Sparsity Learning.

9.1. The Hard Concrete Distribution

The Hard Concrete distribution [22] (HC) is a modification of the Binary Concrete distribution (BC), which in turn is a special case of the Concrete distribution from Maddison et al. (2017). The BC is a continuous relaxation of the Bernouilli distribution. The HC has the advantage of allowing to put significant mass on $P(z = 0)$, where $z \in [0, 1]$ is the relevance of a neuron (or group of neurons). We are interested in drawing samples z from the HC, which can be done by using $z = Q^{-1}_{\text{HC}}(\epsilon|\phi)$, where Q^{-1} is the inverse cumulative distribution function (ICDF) and $\epsilon \sim \mathcal{U}(0, 1)$. Fig. 3 compares the probability density functions (PDF) and ICDF of the BC and the HC distributions.

The PDF $q_{\text{BC}}(s|\phi)$ and CDF $Q_{\text{BC}}(s|\phi)$ of the BC have parameters $\phi := (\alpha, \beta)$ and are defined as follows:

$$q_{\text{BC}}(s|\phi) = \frac{\beta\alpha s^{-\beta-1}(1-s)^{-\beta-1}}{(\alpha s^{-\beta} + (1-s)^{-\beta})^2} \tag{2}$$

$$Q_{\text{BC}}(s|\phi) = \text{Sigmoid}((\log s - \log(1-s))\beta - \log\alpha). \tag{3}$$

To obtain the HC, we stretch the domain of q_{BC} to the (γ, ζ) interval, with $\gamma < 0$ and $\zeta > 1$. Since this stretching operation results in some probability mass being outside $[0, 1]$, we assign the mass of $[\gamma, 0]$ and $[1, \zeta]$ to $P(z = 0)$ and $P(z = 1)$, respectively. For example, with $\log\alpha = 0$, $P(z = 0) =$

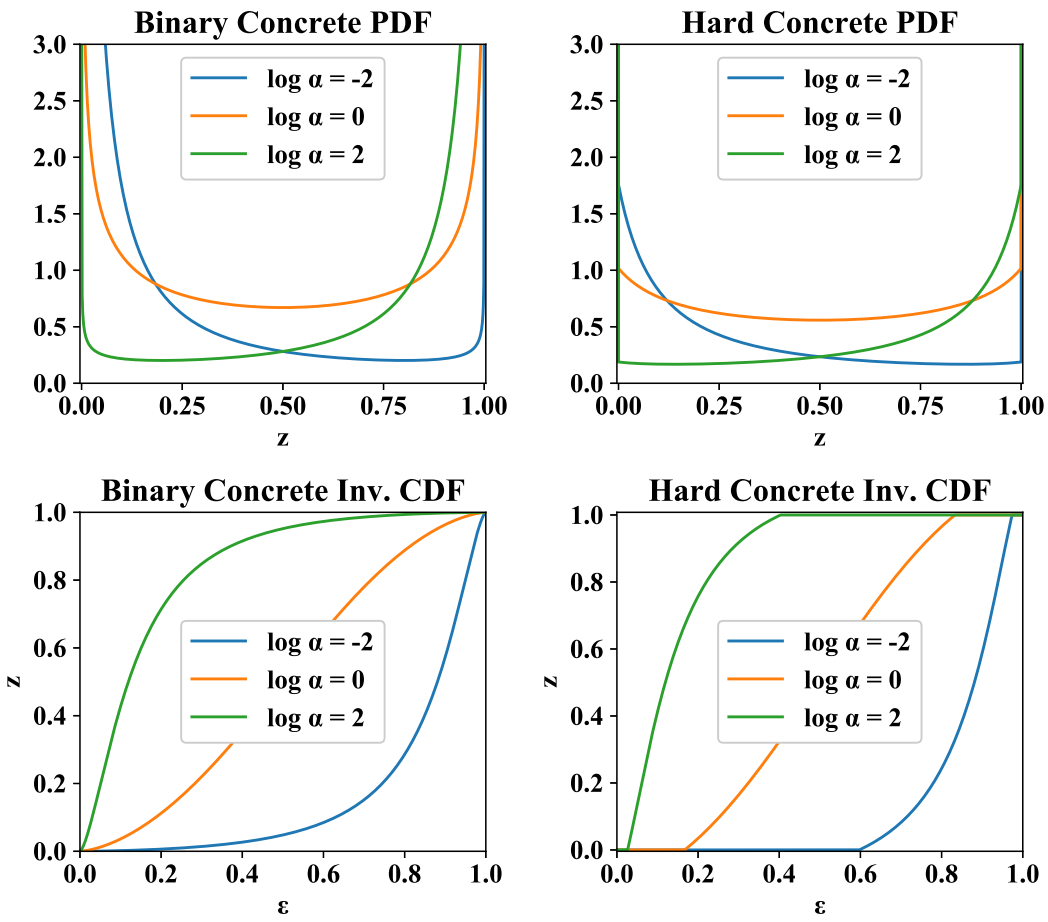


Figure 3: **Comparison of the BC and the HC distributions**. The parameters β, γ, ζ are set to $2/3, -0.1, 1.1$.

$P(z = 1) \approx 0.23$. The resulting PDF is better understood visually (c.f. Fig. 3). The HC distribution has parameters $\phi := (\alpha, \beta, \gamma, \zeta)$, and we set β, γ, ζ to $2/3, -0.1, 1.1$ for all our experiments, as per [22].

For our purposes, we only need to draw samples for the HC, which can be done with its inverse CDF. In our experiments, we use the following formula, given by [22] :

$$Q^{-1}_{\text{HC}}(\epsilon|\phi) = \text{Clamp}_{0,1}\left[\text{Sigmoid}\left(\frac{\log\epsilon - \log(1-\epsilon) + \log\alpha}{\beta}\right)(\zeta - \gamma) + \gamma\right]. \tag{4}$$

9.2. The Hard Concrete Sparsity Loss

We define our sparsity loss as the expectation of the L_0 norm of the set of all dropout variables z. We replace this discrete norm by a continuous relaxation $L_{\text{HC}}(\Phi)$, where $\Phi := \{\phi_i\}$ and $\phi := (\alpha, \beta, \gamma, \zeta)$:

$$L_{\text{HC}}(\Phi) = \sum_{\phi\in\Phi} P_\phi(z > 0) = \sum_{\phi\in\Phi}(1 - Q_{\text{HC}}(0|\phi)) = \sum_{\phi\in\Phi}\text{Sigmoid}(\log\alpha - \beta\log\frac{-\gamma}{\zeta}) \tag{5}$$