Supplementary material

Jinhyuk Park and Albert No

1 Student Network Design

Instead of applying optimized network architecture search (NAS), we use the most naive approach to construct the student network f_s. Given the pruned network $f_t(\cdot; w_p)$ (via unstructured pruning), we count the number of nonzero parameters for each layer. Then, the student network f_s is constructed to have the same number of layers as $f_t(\cdot; w_p)$, but each layer has reduced number of neurons (or channels). The number of neurons is specifically chosen to (approximately) match the number of parameters per layer of the pruned network $f_t(\cdot; w_p)$.

Consider the case where the original network is a convolutional neural network (CNN), the most common scenario. Recall that the number of parameters of a convolutional layer is

$$x \times y \times c_{in} \times c_{out} \tag{1}$$

where $x \times y$ corresponds to the size of the filter, c_{in} is the number of input channels, and c_{out} is the number of output channels. Note that we ignore the bias for simplicity. Thus, we can sequentially adjust the number of channels per layer to match the number of parameters.

More precisely, suppose $f_t(\cdot; w_p)$ be a pruned CNN with L layers, and $n_1, \ldots, n_L$ be the number of nonzero parameters in each layer of $f_t(\cdot; w_p)$. We construct a new student CNN f_s with L layers where the number of channels at each layer is $c_0, c_1, \ldots, c_L$ (c_0 is the number of channels of input, which is 3 for an RGB image). In each i-th layer, the size of filter $x_i \times y_i$ is the same as the pruned CNN f_P. Then, we iteratively match the number of parameters using

$$c_i = \left[\frac{n_i}{x_i \times y_i \times c_{i-1}} \right] \tag{2}$$

where $[\cdot]$ is a rounding operator.

2 Training Details

In this section, we describe the detailed experimental setting. Table 1 provide hyperparameters for regular training, pruning (LR rewinding), and knowledge distillation (vanilla KD), respectively. Most hyperparameters are common choices in practice. However, note that we use Nesterov stochastic gradient descent (SGD) as an optimizer since it is a default optimizer for LR rewinding. This optimizer may not be an optimal choice, however, our goal is not achieving state-of-the-art test accuracy but having fair comparison between pruned teacher and unpruned teacher. For MoblineNetV2, some hyperparameters related to learning rate are modified to ensure accuracy. Since MobileNetV2 is a student network in our experiment, we do not prune MobileNetV2.

Table 1: Hyperparameters for training, pruning, and KD.

Training	VGG	ResNet	MobileNetV2
Optimizer	nesterov SGD (0.9)	nesterov SGD (0.9)	nesterov SGD (0.9)
Trainig epochs	200	100	100
Batch size	128	128	256
Learning rate	0.1	0.01	0.05
Learning rate drops	[60, 120, 160]	[30, 60, 80]	[60, 80]
Drop factor	0.2	0.1	0.1
Weight decay	0.0005	0.0001	0.0005
Pruning	VGG	ResNet	-
Pruner	LR rewinding	LR rewinding	-
Iterative pruning rate	0.2	0.2	-
Optimizer	nesterov SGD (0.9)	nesterov SGD (0.9)	-
Post trainig epochs	130	50	-
Batch size	128	512	-
Learning rate	0.1	0.04	-
Learning rate drops	[39, 84]	[10, 30]	-
Drop factor	0.1	0.1	-
Weight decay	0.0002	0.0001	-
Distillation	VGG	ResNet	MobileNetV2
KD	vanilla	vanilla	vanilla
Optimizer	nesterov SGD (0.9)	nesterov SGD (0.9)	nesterov SGD (0.9)
KD epochs	200	100	100
KD batch Size	128	128	256
KD learning Rate	0.1	0.01	0.05
Learning rate drops	[60, 120, 160]	[30, 60, 80]	[60, 80]
Drop factor	0.2	0.2	0.1
Weight decay	0.0005	0.0005	0.0005
Alpha	0.95	0.95	0.95
Temprature	10	10	10

3 Agreement between Teacher and Student

In this section, we investigate the agreement between the teacher and student's prediction in various settings. Table 2 presents the agreement as well as students' accuracy. As we discussed, increment in agreement does not always guarantee the accuracy. This implies that the teacher may not "teach" the student, but "help" the student with regularization.

Table 2: Agreement between the teacher and the student.

Teacher	Pruning Ratio	Student	Student Accuracy	Agreement
VGG19	None	VGG19	73.74 ± 0.20	76.67 ± 0.12
	36%	VGG19	74.10 ± 0.26	77.70 ± 0.12
	59%	VGG19	74.26 ± 0.37	77.05 ± 0.16
	79%	VGG19	74.35 ± 0.10	78.95 ± 0.10
VGG19	36%	VGG19-ST36	73.77 ± 0.16	77.09 ± 0.19
	59%	VGG19-ST59	73.81 ± 0.10	77.42 ± 0.42
	79%	VGG19-ST79	73.39 ± 0.11	77.61 ± 0.26
ResNet18	None	ResNet18	57.97 ± 0.10	73.91 ± 0.31
	36%	ResNet18	59.39 ± 0.21	72.07 ± 0.12
	59%	ResNet18	58.99 ± 0.26	70.79 ± 0.16
	79%	ResNet18	59.33 ± 0.18	70.57 ± 0.60
ResNet18	36%	ResNet18-ST36	58.75 ± 0.19	70.59 ± 0.29
	59%	ResNet18-ST59	57.76 ± 0.31	68.03 ± 0.10
	79%	ResNet18-ST79	56.23 ± 0.16	64.68 ± 0.27

4 Number of Parameters

Table 3 shows the number of parameters in networks. As we intended, we can see that the number of parameters coincides with the target sparsity of pruned teachers. For example, the number of parameters in VGG19-ST79 is roughly 21%, matching 79% sparsity. We also count FLOPs using ptflops [1]. Note that the model with fewer parameters may have more FLOPs. For example, VGG19-ST79 has fewer weights than VGG19-CL1 but has more FLOPs. However, VGG19-ST79 shows higher test accuracies, indicating the effectiveness of the student network architecture learned from the pruned teacher.

Table 3: Number of parameters and FLOPs of various models in our expriements.

Datasets	Model	# of param	FLOPs
CIFAR100	VGG19	20.1M	399M
	VGG19-CL1	11.0M	158M
	VGG19-CL2	9.9M	264M
	VGG19DBL	75.4M	1495M
	VGG19DBL-ST36	48.2M	1187M
	VGG19DBL-ST59	30.8M	916M
	VGG19DBL-ST79	15.7M	677M
	VGG19-ST36	12.8M	321M
	VGG19-ST59	8.2M	248M
	VGG19-ST79	4.2M	174M
TinyImageNet	ResNet18	11.3M	149M
	ResNet18-ST36	7.3M	114M
	ResNet18-ST59	4.7M	91M
	ResNet18-ST79	2.4M	66M
	VGG16	18.1M	1381M
	MobileNetV2	2.5M	27M

4.1 VGG-ST

Table 4 summarizes the number of weights in each layer for VGG19, pruned VGG19 (79% sparsity), and VGG19-ST79. As we described in SND, We set the number of filters based on the number of weights per layer of the pruned teacher. Note that we have modified VGG which has a single fully-connected (FC) layer. We do not control the number of parameters of FC, which is deterministic based on the number of filters in the previous layer. Thus, the weight ratio of the fc layer does not match the pruned network. Other student networks, VGG-ST36 and VGG-ST59, were constructed similarly.

Table 4: Number of parameters in each layer of unpruned VGG19, pruned VGG19 (79%), and VGG19-ST79.

	VGG19	Pruned VGG19 (79%)		VGG19-ST79	
	# of weight	# of weight	ratio(%)	# of weight	ratio(%)
conv-0	1728	1087	62.91	1080	62.50
conv-1	36864	18102	49.10	17640	47.85
conv-2	73728	50134	68.00	48951	66.39
conv-3	147456	97936	66.42	96903	65.72
conv-4	294912	198189	67.20	196425	66.60
conv-5	589824	381144	64.62	378675	64.20
conv-6	589824	379358	64.32	376992	63.92
conv-7	589824	344924	58.48	342720	58.11
conv-8	1179648	548035	46.46	544680	46.17
conv-9	2359296	749074	31.75	746532	31.64
conv-10	2359296	461873	19.58	461340	19.55
conv-11	2359296	196359	8.32	196020	8.31
conv-12	2359296	99450	4.22	98901	4.19
conv-13	2359296	84433	3.58	83916	3.56
conv-14	2359296	225496	9.56	224532	9.52
conv-15	2359296	328861	13.94	326106	13.82
fc	51200	44546	87.00	12200	23.83
total	20070088	4209001	20.97	4153613	20.70

4.2 VGG-CL

We design VGG19-CL1 and VGG19-CL2 so that the number of parameters of the model is roughly half of the original unpruned model. For VGG-CL1, we remove half of filters for each layer except conv-0, conv-1, conv-13, conv-14, and conv-15 The role of those layers (that are close to either input or output) are crucial, we keep the whole filters for VGG19-CL1. VGG-CL1 was designed to check the importance of each layer by remove the channels uniformly across the layers.

For VGG-CL2, we design a network somewhere between pruned VGG19 (59%) and VGG19-CL1. Similar to VGG19-CL1, another customized network VGG19-CL2 has the same number of filters in crucial layers (the first and the last). Thus, conv-15 has 512 filters and an the fully-connected (FC) layer has 51200 weights. On the other hand, the number of channels in other layers are chosen to match the number of parameters per layer of pruned VGG19 (59%). The number of filters for each remaining layer was set to approximate the number of parameters of pruned VGG19 (79%). Table 5 shows the number of parameters in each layer of VGG19-CL1 and VGG19-CL2.

Table 5: Number of parameters in each layer of pruned VGG19 (59%), VGG19-CL1, and VGG19-CL2.

	VGG19	Pruned VGG19 (59%)		VGG19-CL1		VGG19-CL2	
	# of weight	# of weight	ratio	# of weight	ratio	# of weight	ratio
conv-0	1728	1210	70.02	1728	100	1728	100
conv-1	36864	22885	62.08	36864	100	22464	60.94
conv-2	73728	59344	80.49	36864	50	62829	85.22
conv-3	147456	118013	80.03	36864	25	127269	86.31
conv-4	294912	242091	82.09	73728	25	251694	85.35
conv-5	589824	487123	82.59	147456	25	493830	83.72
conv-6	589824	490757	83.2	147456	25	504990	85.62
conv-7	589824	452699	76.75	147456	25	475668	80.65
conv-8	1179648	769861	65.26	294912	25	806796	68.39
conv-9	2359296	1281396	54.31	589824	25	1364922	57.85
conv-10	2359296	1064558	45.12	589824	25	1111500	47.11
conv-11	2359296	751546	31.85	589824	25	711000	30.14
conv-12	2359296	435158	18.44	1179648	50	385362	16.33
conv-13	2359296	380092	16.11	2359296	100	339021	14.37
conv-14	2359296	711337	30.15	2359296	100	684297	29.00
conv-15	2359296	903232	38.28	2359296	100	2520576	106.84
fc	51200	49403	96.49	51200	100	51200	100
total	20070088	8220705	40.96	11001536	54.82	9915146	49.40

5 Mismatched Pair of Networks

We apply KD to mixed pair of teacher and student networks. For example, VGG19-ST36 is a student network that corresponds to pruned VGG19 teacher with sparsity 36%. In this section, we transfer knowledge from a teacher to a mismatched student, for example, the pruned VGG19 teacher with 59% sparsity when the student is VGG19-ST36.

Table 6: Distillation between mismatched pair of teacher and student networks (VGG19).

Teacher	Pruning Ratio	Teacher Accuracy	Student	Student Accuracy
None	-	-	VGG19-ST36	72.32 ± 0.12
VGG19	None	73.13	VGG19-ST36	73.52 ± 0.20
	36%	73.30	VGG19-ST36	73.77 ± 0.16
	59%	72.25	VGG19-ST36	73.91 ± 0.15
	79%	73.43	VGG19-ST36	74.00 ± 0.20
None	-	-	VGG19-ST59	71.80 ± 0.18
VGG19	None	73.13	VGG19-ST59	73.18 ± 0.10
	36%	73.30	VGG19-ST59	73.42 ± 0.24
	59%	72.25	VGG19-ST59	73.81 ± 0.10
	79%	73.43	VGG19-ST59	73.69 ± 0.27
None	-	-	VGG19-ST79	70.89 ± 0.14
VGG19	None	73.13	VGG19-ST79	72.42 ± 0.16
	36%	73.30	VGG19-ST79	72.97 ± 0.17
	59%	72.25	VGG19-ST79	73.13 ± 0.09
	79%	73.43	VGG19-ST79	73.39 ± 0.11
None	-	-	ResNet18-ST36	56.44 ± 0.26
ResNet18	None	57.75	ResNet18-ST36	57.74 ± 0.22
	36%	57.66	ResNet18-ST36	58.75 ± 0.19
	59%	57.58	ResNet18-ST36	58.57 ± 0.22
	79%	57.32	ResNet18-ST36	58.46 ± 0.18
None	-	-	ResNet18-ST59	55.93 ± 0.32
ResNet18	None	57.75	ResNet18-ST59	56.70 ± 0.35
	36%	57.66	ResNet18-ST59	58.20 ± 0.06
	59%	57.58	ResNet18-ST59	57.76 ± 0.29
	79%	57.32	ResNet18-ST59	57.94 ± 0.20
None	-	-	ResNet18-ST79	54.48 ± 0.53
ResNet18	None	57.75	ResNet18-ST79	55.65 ± 0.24
	36%	57.66	ResNet18-ST79	56.66 ± 0.15
	59%	57.58	ResNet18-ST79	56.19 ± 0.12
	79%	57.32	ResNet18-ST79	56.23 ± 0.16

Table 6 shows the result when we mix the teacher and student pair. Although the student is not designed for the teacher, we can see that the pruned teacher teaches better than the unpruned teacher.

6 Large Scale Experiments

We conduct a large scale experiment to further justify the proposed algorithm. Table 7 shows the self distillation result of ResNet50 with and without pruned teacher. It is clear that distillation from pruned teacher is better than the distillation from unpruned teacher.

Table 7: Self distillation of ResNet50 with teacher pruning. Teacher "None" indicates the student is trained without a teacher, while the pruning ratio "None" means the distillation from the unpruned teacher.

Teacher	Pruning Ratio	Teacher Accuracy	Student	Student Accuracy
None	-	-	ResNet50	62.88 ± 0.25
ResNet50	None	62.88	ResNet50	64.54 ± 0.35
	36%	62.72	ResNet50	64.86 ± 0.06
	59%	62.85	ResNet50	65.21 ± 0.21
	79%	63.46	ResNet50	64.97 ± 0.10

Table 8 shows the performance of the proposed compression algorithm on ResNet50 with TinyImageNet. Similar to our main experiment with ResNet18 models, we also observe the effectiveness of our scheme in the larger model.

Table 8: Performance of the proposed compression algorithm on ResNet50 with TinyImageNet. ResNet50-ST(X) is the constructed student network based on the proposed algorithm from X% pruned teacher. Teacher "None" indicates the student is trained without a teacher, while the pruning ratio "None" means the distillation from the unpruned teacher.

Teacher	Pruning Ratio	Teacher Accuracy	Student	Student Accuracy
None	-	-	ResNet50-ST36	62.24 ± 0.14
ResNet50	None	62.88	ResNet50-ST36	64.11 ± 0.26
	36%	62.72	ResNet50-ST36	64.12 ± 0.30
None	-	-	ResNet50-ST59	60.04 ± 0.29
ResNet50	None	62.88	ResNet50-ST59	63.84 ± 0.17
	59%	62.85	ResNet50-ST59	63.58 ± 0.29
None	-	-	ResNet50-ST79	58.74 ± 0.06
ResNet50	None	62.88	ResNet50-ST79	62.25 ± 0.46
	79%	63.46	ResNet50-ST79	62.84 ± 0.26

We also run an experiment with ImageNet, which is larger and realistic dataset. Table 9 shows the performance of the proposed compression algorithm on ResNet18 with ImageNet. For the pruning ratio of 36%, the pruned teacher performs better than the unpruned teacher as we observed in the previous experiments. However, for the pruning ratio of 79%, the pruned teacher is not effective, mainly because ResNet18 is not sufficiently large for the ImageNet dataset. This result emphasizes the importance of finding the right pruning ratio for the teacher.

Table 9: Performance of the proposed compression algorithm on ResNet18 with ImageNet. The pruning ratio "None" means the distillation from the unpruned teacher.

Teacher	Pruning Ratio	Teacher Accuracy	Student	Student Accuracy
ResNet18	None	64.90	ResNet18-ST36	60.93
	36%	65.41	ResNet18-ST36	61.10
ResNet18	None	64.90	ResNet18-ST79	50.24
	79%	64.70	ResNet18-ST79	50.14

References

1. Sovrasov, V.: Flops counter for convolutional networks in pytorch framework (2019), `https://github.com/sovrasov/flops-counter.pytorch/`