Network Pruning - Extended

1st Brianna Butler
Department of Computer Science
Princeton University
Princeton, New Jersey
bb5943@princeton.edu

***Abstract*—This paper expands on the network pruning option for assignment 1 for Computer Science 598 at Princeton University. The assignment draws from three papers discussed in the class that highlight the SNIP pruning method, picking winning tickets, and pruning neural networks while conserving synaptic flow. The original assignment examines these pruning methods with different compression sizes on the Cifar-10 dataset, but in this final project, I decided to expand this investigation to the Cifar-100 dataset as well. The results of this paper provide insight into the composition of datasets and how they affect the results of pruned neural networks.**

***Index Terms*—network pruning, imagenet, cifar10, cifar100, singleshot**

I. Introduction

A. Datasets

In the original iteration of this experiment, we conducted each trial on the Cifar-10 dataset. As mentioned before, we expand these trials to the Cifar-100 dataset. Both the Cifar-10 and the Cifar-100 are labeled subsets of the "80 Million Tiny Images Dataset" created in 2008 [5]. The Cifar-10 dataset contains 60,000 32x32 images colour images with 10 classes. There a 6,000 images per class. 50,000 of the 60,000 images are training images, and 10,000 are testing images.

The dataset is divided into 5 training batches and 1 test batch - all with 10,000 images each.

The Cifar-100 dataset has exactly the same images as the Cifar-10 dataset. However, they differ in the way they are constructed. The Cifar-100 dataset contains 100 classes with 600 images each. These classes are divided into 20 superclasses (For example: the superclass is insects, and its corresponding classes are bee, beetle, butterfly, caterpillar, cockroach. Each class has 500 training images and 100 testing images per class. Each image has a fine label that describes an object's class and a coarse label that describes its superclass.

The Cifar-100 superclasses and classes are as follows:

- **Aquatic Mammals**: beaver, dolphin, otter, seal, whale
- **Fish**: aquarium fish, flatfish, ray, shark, trout
- **Flowers**: orchids, poppies, roses, sunflowers, tulips
- **Food**: containers bottles, bowls, cans, cups, plates
- **Fruit and Vegetables**: apples, mushrooms, oranges, pears, sweet peppers
- **Household Electrical Devices**: clock, computer keyboard, lamp, telephone, television
- **Household Furniture**: bed, chair, couch, table, wardrobe
- **Insects**: bee, beetle, butterfly, caterpillar, cockroach

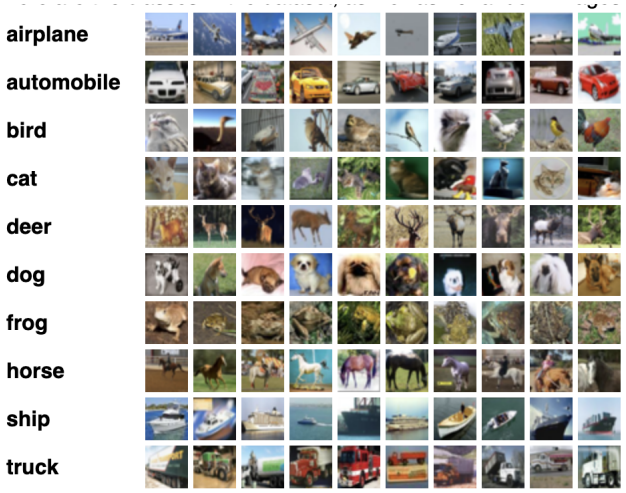

Fig. 1. The CIFAR-10 contains 10 different classes. The classes are mutually exclusive. Here are those 10 classes and 10 random images from each class. Notice there is on overlap between automobiles and trucks. "Automobile" includes sedans, SUVs, things of that sort. "Truck" includes only big trucks. Neither includes pickup trucks. [2]

- **Large Carnivores**: bear, leopard, lion, tiger, wolf
- **Large Man-made Outdoor Things**: bridge, castle, house, road, skyscraper
- **Large Natural Outdoor Scenes**: cloud, forest, mountain, plain, sea
- **Large Omnivores and Herbivores**: camel, cattle, chimpanzee, elephant, kangaroo
- **Medium-sized Mammals**: fox, porcupine, possum, raccoon, skunk
- **Non-Insect Invertebrates**: crab, lobster, snail, spider, worm
- **People**: baby, boy, girl, man, woman
- **Reptiles**: crocodile, dinosaur, lizard, snake, turtle
- **Small Mammals**: hamster, mouse, rabbit, shrew, squirrel
- **Trees**: maple, oak, palm, pine, willow
- **Vehicles 1**: bicycle, bus, motorcycle, pickup truck, train
- **Vehicles 2**: lawn-mower, rocket, streetcar, tank, tractor

[2].

II. Results

A. Hyper-parameter Tuning

a) Testing Accuracy (top 1) with different Datasets: There are a few interesting patterns/results from this first

TABLE I
TESTING ACCURACY (TOP 1) WITH DIFFERENT DATASETS - IN %

Dataset	Arch	Synflow	Mag	Snip	Grasp	Rand
Cifar10	VGG16	81.39	80.39	76.98	11.18	10.00
MNIST	FC	70.59	97.53	77.99	95.55	99.87
Cifar100	VGG16	72.61	70.49	63.63	5.00	5.00

Table 1: Comparison of testing accuracy between the Cifar10 dataset, Cifar100 and the MNIST dataset with different pruning methods. The Cifar10 and Cifar100 datasets utilizes the VGG16 arch, and the MNIST dataset utilizes the FC arch.
Note: Because my experiments were run on Google Colab, I used the default –post-epoch 10 for the Cifar10 and Cifar100 experiments. For the magnitude based pruning experiments, I set –pre-epochs to 100 to save time, as hinted by the README in the provided source code.

TABLE II
TESTING ACCURACY (TOP 1) WITH DIFFERENT COMPRESSION SIZES - IN % FOR CIFAR-10 VS. CIFAR-100

Data	Compression	Synflow	Mag	Snip	Grasp	Rand
Cifar10	0.05	78.82	78.21	77.60	29.97	80.80
Cifar100	0.05	64.34	62.84	n/a	5.02	58.62
Cifar10	0.1	79.25	78.98	76.59	30.16	79.48
Cifar100	0.1	63.68	67.02	5.00	5.00	55.90
Cifar10	0.2	79.85	80.12	77.95	25.06	81.36
Cifar100	0.2	68.41	64.25	60.97	5.00	65.04
Cifar10	0.5	80.24	80.31	78.46	23.18	78.57
Cifar100	0.5	72.66	72.54	62.14	5.00	61.19
Cifar10	1.0	81.39	80.39	76.98	11.18	10.00
Cifar100	1.0	72.61	70.49	63.63	5.00	5.00
Cifar10	2.0	10.00	48.21	57.25	17.17	10.00
Cifar100	2.0	5.00	5.00	5.00	4.99	5.00

Table 2: Comparison of testing accuracy between different pruning methods and different compression sizes on the Cifar10 dataset compared to the Cifar100 dataset. Note: Because my experiments were run on Google Colab, I used the default –post-epoch 10 for the Cifar10 and Cifar100 experiments.

experiment that were very interesting to me. First, I found it very interesting that there was an extreme drop off in testing accuracy for the grasp and rand pruning methods with the Cifar-10 dataset and vgg16 model. This drop off is also apparent for the Cifar-100 dataset on the same model. In fact, the ranking of highest to lowest accuracy amongst the pruning methods was consistent for the Cifar-10 and Cifar-100 datasets.

The rand pruning method performed exceptionally well on the MNIST dataset and fc model. The worst performing pruning method for the MNIST dataset was the same pruning method that resulted in the best accuracy amongst pruning methods for the Cifar10 and Cifar100 datasets. The only pruning method with a similar testing accuracy amongst differing datasets and arches was the SNIP pruning method. Based on the provided sources on SNIP, the success and versatility of SNIP for varying datasets and models is due to its ability to "prune irrelevant connections for a given task at single-shot prior training" [3]. The highest performing combination was the mnist dataset with the fc mode and the mag pruning method.

b) Testing Accuracy (top 1) with different Compression Sizes: The largest experiment conducted involved examining the top 1 accuracy of each dataset with different compression sizes for both the Cifar-10 and Cifar-100 datasets.

Amongst all the pruning methods except grasp and rand, there seems to be a huge dropoff in top 1 accuracy once the compression size is 2 with Synflow showing the largest drop off of all of the methods. This is consistent amongst the Cifar-10 and the Cifar-100 dataset. The grasp pruning method was the worst-performing pruning method overall, never reaching a top 1 accuracy over 31% for Cifar-10 and 5.02% for Cifar-100. The pruning method that resulted in the least worst drop off for Cifar-10 was SNIP. While SNIP did not result in the highest top 1 accuracy, it did result in the most consistent top 1 accuracies in the trials for Cifar-10. This consistency could be reflective of the versatility of SNIP mentioned in the discussion for the dataset comparison trial table.

This was not the same for the Cifar-100 dataset. For every pruning method, there was a big drop off in accuracy for a compression size of 2 (the grasp method performing poorly no matter the compression size, and rand showing a drop off as soon as the compression reached 1). Interestingly, for the Cifar-100 dataset, the SNIP pruning method encountered an error with a compression size of 0.05, so those results were inconclusive. Another result of note is that the Cifar-100 top 5 accuracy for any of the pruning methods never outperforms Cifar-10 on these same pruning methods for their respective compression size trials. The smallest advantage that Cifar-10 has over Cifar-100 is approximately 8.0%, which is still a significant increase in accuracy. This difference in accuracies for these datasets may provide some insight into which dataset/tasks would be best for this neural network and its pruning methods.

c) Testing time (inference on testing dataset): For this experiment, we also tracked runtimes for each trial. In my opinion, the most consistently faster runtimes originated from the synflow and snip pruning methods. For smaller compression sizes, trial timing nearly increased 3 times except for 0.05 compression and for the snip and synflow pruning methods. For compression levels 0.2 and above trial times became much more similar across pruning methods. The speed efficiency of SNIP could be due to its "filtering" process of removing irrelevant connections during the training process, resulting in less connections to read/iterate over later in the process. I admit, the synflow paper was a bit confusing for my current level of understanding of network pruning, but, in the synflow paper, they mention that synflow "avoids layer-collapse and reaches Maximal Critical Compression", which would make sense in the context of trial speed and why it was more successful than other pruning methods [4].

Except for the experiment error when testing the neural network using the SNIP pruning method and a compression size of 0.05, there were no clear trends in testing time for the time to run the experiment for the Cifar-100 dataset. Neither the chosen compression size and the chosen pruning method did not inspire a convincing trend of growing or shrinking

TABLE III
TESTING TIME (INFERENCE ON TESTING DATASET) - IN SECONDS FOR CIFAR-10 VS. CIFAR-100

Data	Compression	Synflow	Mag	Snip	Grasp	Rand
Cifar10	0.05	297.3	633.7	284.8	635.7	666.1
Cifar100	0.05	290.8	290.1	15.0	287.5	289.8
Cifar10	0.1	636.2	631.9	632.2	635.3	631.0
Cifar100	0.1	290.8	291.43	283.5	286.2	289.4
Cifar10	0.2	284.3	284.9	282.5	282.7	295.1
Cifar100	0.2	291.6	290.4	290.9	286.8	291.4
Cifar10	0.5	283.0	288.5	287.4	281.4	285.0
Cifar100	0.5	291.9	293.8	292.1	290.4	292.8
Cifar10	1.0	287.8	289.1	286.8	286.0	274.2
Cifar100	1.0	291.0	294.8	291.0	292.1	282.2
Cifar10	2.0	270.4	287.0	283.0	282.1	268.9
Cifar100	2.0	278.0	280.9	278.5	287.7	278.7

Table 3: Comparison of testing time between different pruning methods and different compression sizes on the Cifar10 dataset compared to the Cifar100 dataset. Note: Because my experiments were run on Google Colab, I used the default –post-epoch 10 for the Cifar10 and Cifar100 experiments.

TABLE IV
NUMBER OF FLOATING POINT OPERATIONS (FLOPS) WITH DIFFERENT COMPRESSION SIZES - IN % FOR CIFAR-10 VS. CIFAR-100 (IN MILLIONS)

Data	Compression	Synflow	Mag	Snip	Grasp	Rand
Cifar10	0.05	297.5	287.9	301.1	257.1	279.5
Cifar100	0.05	297.3	287.8	n/a	253.8	279.5
Cifar10	0.1	283.0	265.0	290.6	228.7	249.1
Cifar100	0.1	282.7	264.8	286.6	227.0	249.1
Cifar10	0.2	257.6	225.9	245.5	179.3	197.9
Cifar100	0.2	9.3	225.7	242.5	182.1	198.0
Cifar10	0.5	201.4	145.4	146.1	118.2	99.3
Cifar100	0.5	201.0	145.0	146.3	117.8	99.3
Cifar10	1.0	143.3	76.9	63.9	55.1	31.6
Cifar100	1.0	142.3	76.3	69.1	58.6	31.6
Cifar10	2.0	57.8	24.5	14.0	18.7	3.4
Cifar100	2.0	51.0	23.5	18.8	20.1	3.4

Table 4: Comparison of the number of flops between different pruning methods and different compression sizes on the Cifar10 dataset compared to the Cifar100 dataset. Note: Because my experiments were run on Google Colab, I used the default –post-epoch 10 for the Cifar10 and Cifar100 experiments.

runtime for the neural network with the Cifar-100 dataset.

Another interesting trend from the experiments is that the trial time for Cifar-10 was consistently (except for one circumstance) shorter than that of the trials ran on the Cifar-100 dataset. The difference in time, however, can be considered miniscule, but it is still there: each trial for the Cifar-10 dataset was always approximately 3 to 8 seconds shorter than that of their counterpart trial for the Cifar-100 dataset. However, with a compression size of smaller than 0.1, the trial time for Cifar-10 skyrockets for every pruning method. As mentioned before, this rise in trial time is prevented when running Cifar-100, so the Cifafr-10 time rises and Cifar-100 trial time stays the same.

d) Number of Floating Point Operations:

$$FLOPS = cores \times \frac{cycles}{second} \times \frac{FLOPs}{cycle}$$

For each pruning method, there is a decrease in FLOP sparsity as compression size increases. In this study, FLOP stands for number of floating point operations. Similar to the pattern of decrease in top 1 accuracy, the largest drop off in FLOPs was between the trials ran with a compression of 1 and the trials ran with a compression of 2. This was consistent for both the Cifar-10 dataset and the Cifar-100 dataset.

Another important observation from this trial is that the number of FLOPs were relatively consistent amongst both datasets for any compression value or pruning method. For many of the trials the number of FLOPS differed down to the hundred thousands place, which is miniscule considering the number of FLOPs was regularly in the hundred millions. Usually, the trials ran on the Cifar-100 dataset resulted in the smaller number of FLOPs for each pruning method-compression size combination. The only time it did not was when running the SNIP pruning methods for compression sizes 0.5, 1.0, and 2.0.

There were a few FLOP outliers that stemmed from the trials. One included the aforementioned erroneous result from running the SNIP pruning method with a compression size 0.05 on the Cifar-100 dataset. The next, and last, outlier is the resulting FLOPs from the Cifar-100 trial involving the Synflow pruning method with a compression size of 0.2. This value was inconsistently low, and thus, the only result from these trials that resulted in a significant difference of FLOPs between the two datasets and their respective experiment parameters.

B. The compression ratio of each layer

TABLE V
COMPRESSION RATIO OF EACH LAYER - IN % FOR CIFAR-10 VS. CIFAR-100

Data	Compression	Synflow	Mag	Snip	Grasp	Rand
Cifar10	0.5	31.64	31.64	31.64	31.64	31.64
Cifar100	0.5	31.64	31.64	31.64	31.64	31.64

Table 5: Comparison of the compression ratio between different pruning methods and different compression sizes on the Cifar10 dataset compared to the Cifar100 dataset. Note: Because my experiments were run on Google Colab, I used the default –post-epoch 10 for the Cifar10 and Cifar100 experiments.

Interestingly, the compression ratio for each pruning method was exactly the same for both datasets. Based on [1], SNIP and GRasp are similar pruning methods with GRasp using SNIP as a basis framework with the intention of GRasp being an improvement of SNIP. My understanding of network pruning is still on a novel level, but my interpretation of course material suggests that all of these pruning methods go through the similar process of removing parameters during the training process, which could be contributing to these same parameter sparsity values.

Admittedly, I have minimal understanding of the histograms, but one observation I made was that the histogram for the rand pruning method was that the weight values are very similar amongst layers in the model. The most widely distributed pruning method (weight value-wise) seems

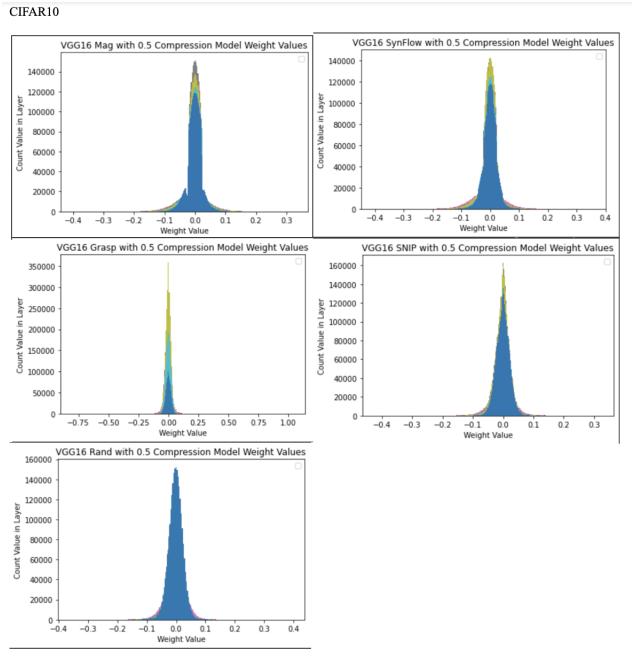

Fig. 2. Histograms representing the distribution of weight values for each pruning method discussed in the assignment. All pruning method models utilized the VGG16 model and **Cifar10** dataset

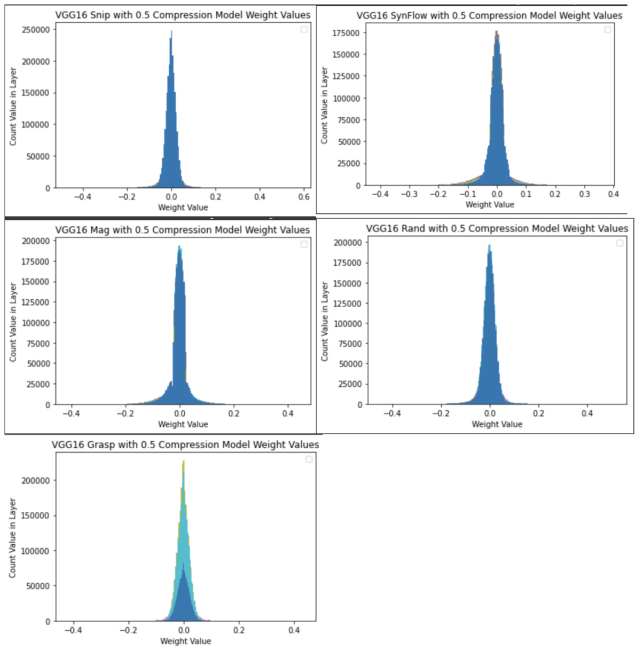

Fig. 3. Histograms representing the distribution of weight values for each pruning method discussed in the assignment. All pruning method models utilized the VGG16 model and **Cifar100** dataset

to be Synflow. Grasp is the pruning method with the highest frequency count out of all the pruning methods tested. These trends were consistent for both the histograms describing the distribution for Cifar-10 and Cifar-100.

III. Conclusion

Based on this comparison experiment, we can hypothesize a few things about network pruning methods and what types of datasets they perform the best on. From our trials, we have seen that the selected pruning methods, overall, perform better on the Cifar-10 than the Cifar-100 dataset without a significant enough difference in training time to consider the latter over the former. There is also a slight descrease in FLOPs when utilizing the Cifar-100, but this is also not a significant difference either.

One big difference between the Cifar-10 and Cifar-100 datasets is the diversity of labels applied to their images. Cifar-10 has less labels, but the same images as the Cifar-100 dataset. Perhaps, there is an optimal number of labels these pruning methods can utilize before experiencing the drop off in accuracy observed in this experiment. This would be an excellent direction for future research to observe this aspect of network pruning: their optimal datasets and their optimal number of labels.

References

[1] Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N., and Peste, A. (2021). Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks. J. Mach. Learn. Res., 22, 241:1-241:124.

[2] Krizhevsky, A. (n.d.). CIFAR-10 and CIFAR-100 datasets. Retrieved April 24, 2022, from https://www.cs.toronto.edu/ kriz/cifar.html

[3] Lee, N., Ajanthan, T. and Torr, P.H., 2018. Snip: Single-shot network pruning based on connection sensitivity. arXiv preprint arXiv:1810.02340.

[4] Tanaka, H., Kunin, D., Yamins, D.L. and Ganguli, S., 2020. Pruning neural networks without any data by iteratively conserving synaptic flow. arXiv preprint arXiv:2006.05467

[5] Wikimedia Foundation. (2022, January 29). 80 million tiny images. Wikipedia. Retrieved April 24, 2022, from https://en.wikipedia.org/wiki/80_Million_Tiny_Images