가짜연구소 10기 ON DEVICE AI ON THE AIR

Pruning Filter in Filter

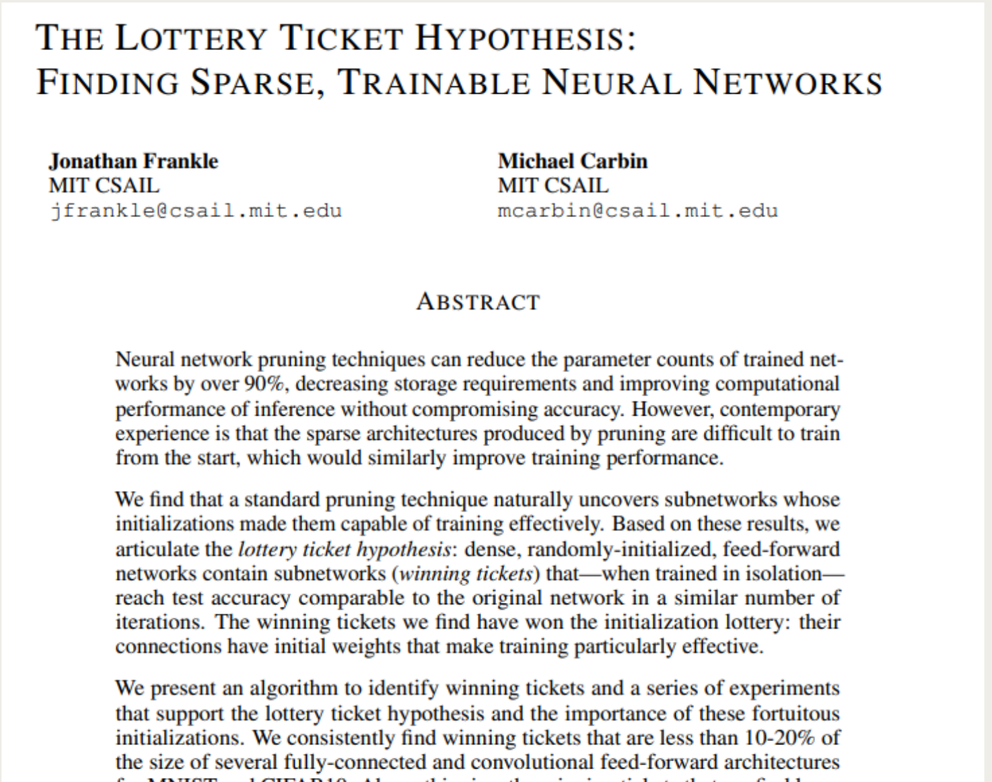
온디에 3주차: Semi-Structured Pruning

발표자: 최유진

UNSTRUCTURED PRUNING

모든 weight를 개별적으로 pruning 할 수 있는 방법

가지치기를 한 상태에서 학습을 하기 위해 파라미터를 초기화 할 때, 가지치기를 하기 전의 초기값으로 초기화하는 방법을 제안



THE LOTTERY TICKET HYPOTHESIS:
FINDING SPARSE, TRAINABLE NEURAL NETWORKS

Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

ABSTRACT

Neural network pruning techniques can reduce the parameter counts of trained networks by over 90%, decreasing storage requirements and improving computational performance of inference without compromising accuracy. However, contemporary experience is that the sparse architectures produced by pruning are difficult to train from the start, which would similarly improve training performance.

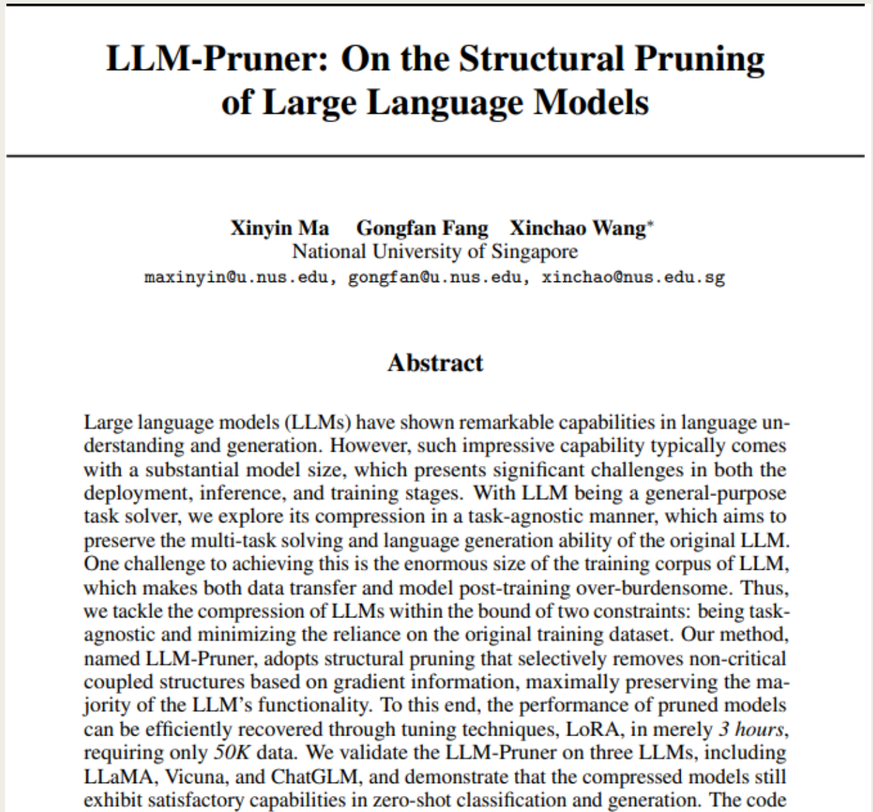
We find that a standard pruning technique naturally uncovers subnetworks whose initializations made them capable of training effectively. Based on these results, we articulate the *lottery ticket hypothesis*: dense, randomly-initialized, feed-forward networks contain subnetworks (*winning tickets*) that—when trained in isolation—reach test accuracy comparable to the original network in a similar number of iterations. The winning tickets we find have won the initialization lottery: their connections have initial weights that make training particularly effective.

We present an algorithm to identify winning tickets and a series of experiments that support the lottery ticket hypothesis and the importance of these fortuitous initializations. We consistently find winning tickets that are less than 10-20% of the size of several fully-connected and convolutional feed-forward architectures

STRUCTURED PRUNING

필터, 채널, 층 단위로 가지치기 하는 방식

각 층에서 중요도가 낮은 뉴런과 헤드를 제거하는 구조적 가지치기 기법을 도입하여 LLM을 경량화



LLM-Pruner: On the Structural Pruning of Large Language Models

Xinyin Ma **Gongfan Fang** **Xinchao Wang***
National University of Singapore
maxinyin@u.nus.edu, gongfan@u.nus.edu, xinchao@nus.edu.sg

Abstract

Large language models (LLMs) have shown remarkable capabilities in language understanding and generation. However, such impressive capability typically comes with a substantial model size, which presents significant challenges in both the deployment, inference, and training stages. With LLM being a general-purpose task solver, we explore its compression in a task-agnostic manner, which aims to preserve the multi-task solving and language generation ability of the original LLM. One challenge to achieving this is the enormous size of the training corpus of LLM, which makes both data transfer and model post-training over-burdensome. Thus, we tackle the compression of LLMs within the bound of two constraints: being task-agnostic and minimizing the reliance on the original training dataset. Our method, named LLM-Pruner, adopts structural pruning that selectively removes non-critical coupled structures based on gradient information, maximally preserving the majority of the LLM's functionality. To this end, the performance of pruned models can be efficiently recovered through tuning techniques, LoRA, in merely *3 hours*, requiring only *50K* data. We validate the LLM-Pruner on three LLMs, including LLaMA, Vicuna, and ChatGLM, and demonstrate that the compressed models still exhibit satisfactory capabilities in zero-shot classification and generation. The code

Weight Pruning (WP)

예측 성능의 저하 없이 희박한 네트워크 생성이 가능한 방법으로 가중치 위치 저장이 필요하고 하드웨어 가속이 어려움.

*희박한 네트워크: 불필요한 가중치를 제거하여 많은 값이 0에 가까운 형태로 변형된 신경망

Filter(Channel) Pruning (FP)

구조적으로 정리된 네트워크를 유지하여 하드웨어 가속이 용이.

필터 스켈레톤(FILTER SKELETON)

- 필터의 가중치뿐만 아니라, 필터 내부의 구조(스트라이프, Stripe) 를 학습하는 행렬.
- 각 값은 필터의 스트라이프(줄무늬)에 해당하며, 필터의 모양(shape) 을 결정.

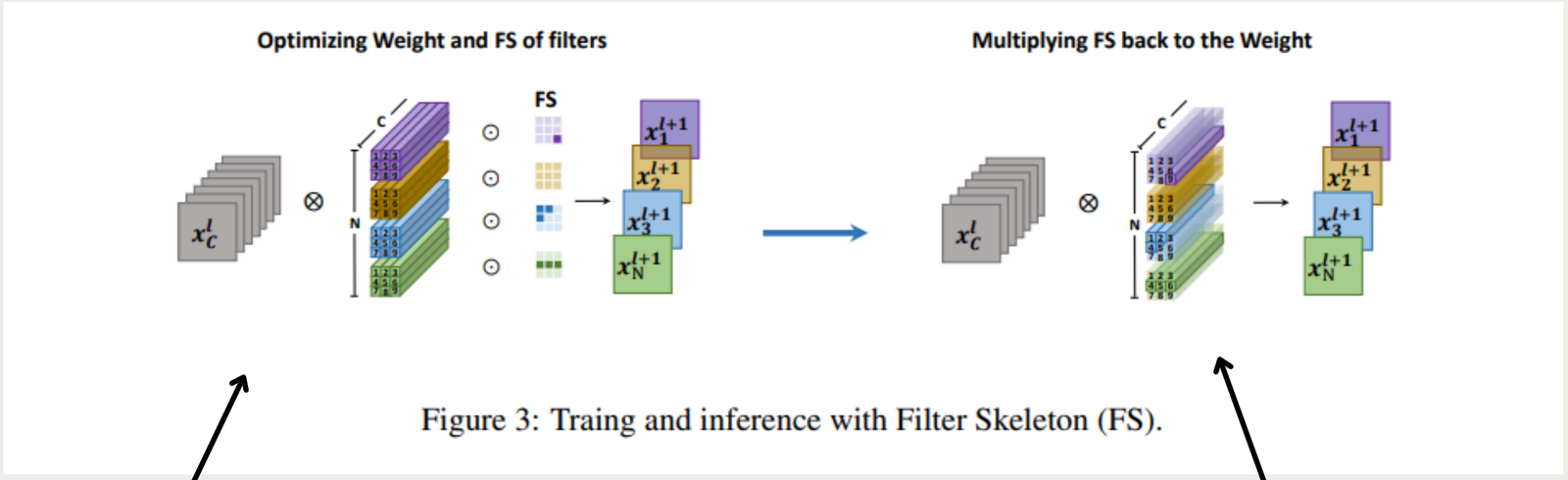


Figure 3: Traing and inference with Filter Skeleton (FS).

필터의 가중치와 FS를 동시에 학습

학습된 FS 값을 다시 필터 가중치에 곱함
이 과정에서 정규화가 된다.

스트라이프 단위 가지치기(SWP)

- 필터 내부의 스트라이프(Stripe)는 네트워크에 동일하게 기여하지 않음 → FS를 희박화(Sparse) 해야 함
- 일부 스트라이프 값을 0에 가깝게 만들기 위해 L1 정규화 적용

- SWP의 가지치기 과정

1️⃣ FS 학습 → 필터의 최적 모양을 암묵적으로 학습

2️⃣ 임계값 설정 → 작은 값(중요하지 않은 스트라이프)은 업데이트되지 않음

3️⃣ 스트라이프 단위 컨볼루션 → 기존 필터 단위가 아닌, 각 스트라이프별로 독립적인 연산 수행 후 결과 합산

스트라이프 단위 가지치기(SWP)

- 장점

✅ 세밀한 가지치기 → 기존 FP보다 더 미세한 가지치기 가능
✅ 성능 유지 → 특정 데이터셋(CIFAR-10 등)에서는 미세 조정 없이도 높은 성능 유지
✅ 추가 연산 부담 없음 → 기존 컨볼루션 연산 방식에서 계산 순서만 변경됨
✅ 효율적인 인덱스 관리 → 가중치 가지치기보다 적은 인덱스 필요

실험

Dataset

CIFAR-10, ImageNet

Models

VGG16. ResNet 56

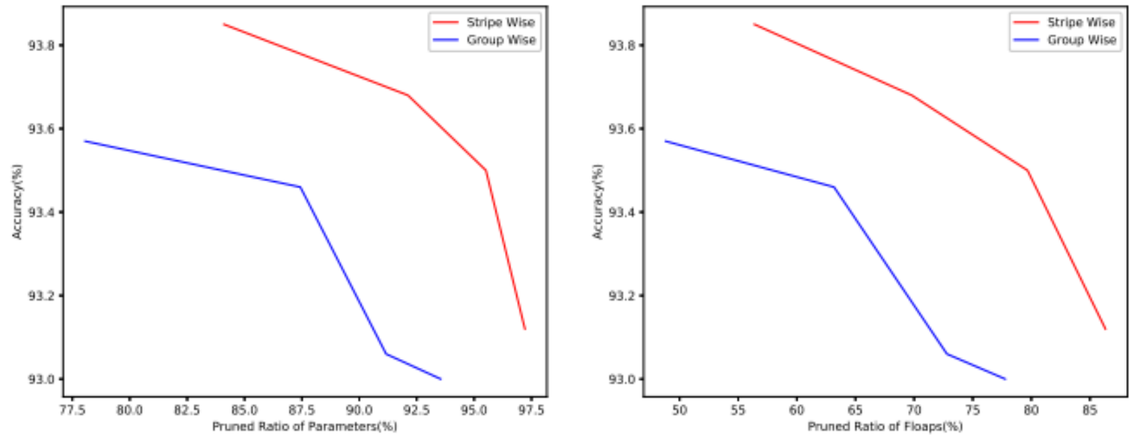


Figure 6: Comparing SWP with group-wise pruning on CIFAR-10. The backbone is VGG16.

실험결과

Table 2: Comparing SWP with state-of-arts FP-based methods on CIFAR-10 dataset. The baseline accuracy of ResNet56 is 93.1% [29], while VGG16's baseline accuracy is 93.25% [7].

Backbone	Metrics	Params(%)↓	FLOPS(%)↓	Accuracy(%)↓
VGG16	L1[7] (ICLR 2017)	64	34.3	-0.15
	ThiNet[24] (ICCV 2017)	63.95	64.02	2.49
	SSS[44] (ECCV 2018)	73.8	41.6	0.23
	SFP[45] (IJCAI 2018)	63.95	63.91	1.17
	GAL[27] (CVPR 2019)	77.6	39.6	1.22
	Hinge[46] (CVPR 2020)	80.05	39.07	-0.34
	HRank[47] (CVPR 2020)	82.9	53.5	-0.18
	Ours	**92.66**	**71.16**	**-0.4**
ResNet56	L1[7] (ICLR 2017)	13.7	27.6	-0.02
	CP[8] (ICCV 2017)	-	50	1.00
	NISP[25] (CVPR 2018)	42.6	43.6	0.03
	DCP[48] (NeurIPS 2018)	70.3	47.1	-0.01
	IR[16] (IJCNN 2019)	-	67.7	0.4
	C-SGD[49] (CVPR 2019)	-	60.8	-0.23
	GBN [29] (NeurIPS 2019)	66.7	70.3	0.03
	HRank[47] (CVPR 2020)	68.1	74.1	2.38
	Ours	**77.7**	**75.6**	**0.12**

Table 3: Comparing SWP with state-of-arts pruning methods on ImageNet dataset. All the methods use ResNet18 as the backbone and the baseline top-1 and top-5 accuracy are 69.76% and 89.08%, respectively.

Backbone	Metrics	FLOPS(%)↓	Top-1(%)↓	Top-5(%)↓
ResNet18	LCCL[50] (CVPR 2017)	35.57	3.43	2.14
	SFP[45] (IJCAI 2018)	42.72	2.66	1.3
	FPGM[51] (CVPR 2019)	42.72	1.35	0.6
	TAS[52] (NeurIPS 2019)	43.47	0.61	-0.11
	DMCP[53] (CVPR 2020)	42.81	0.56	-
	Ours ($\alpha = 5e-6$)	**50.48**	**-0.23**	**-0.22**
	Ours ($\alpha = 2e-5$)	**54.58**	**0.17**	**0.04**

Conclusion

필터 스켈레톤(FS) 도입

가지치기에 최적화된 필터 모양을 효율적으로 학습

새로운 가지치기 패러다임 (SWP) 제안

필터를 여러 스트라이프(Stripe) 단위로 나누어 가지치기 수행

SWP 프레임워크의 효과 입증

기존 방법보다 더 높은 압축률과 성능 유지 가능

THANK YOU

PRUNING OVERVIEW

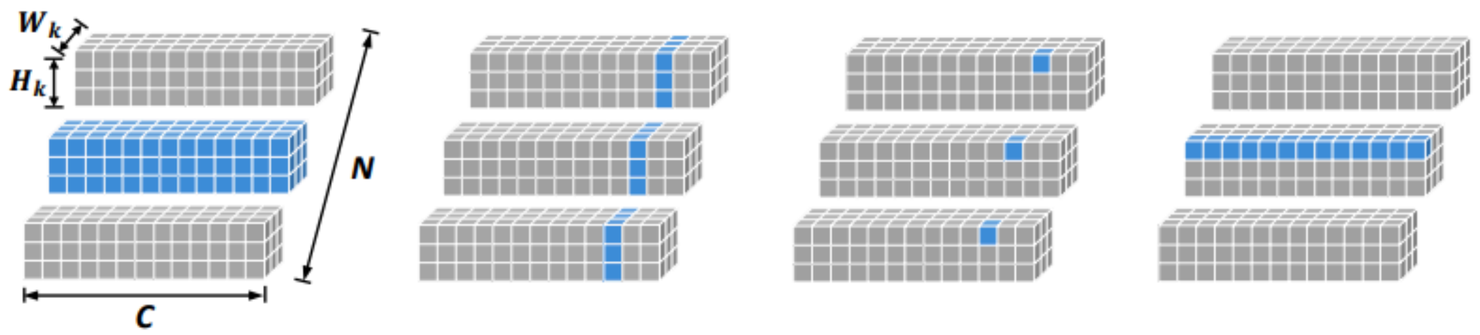


Figure 2: The visualization of different types of pruning.

FILTER SKELETON(FS)

손실

$$L = \sum_{(x,y)} loss(f(x, W \odot I), y) \tag{1}$$

순방향 프로세스

$$X^{l+1}_{n,h,w} = \sum_{c}^{C} \sum_{i}^{K} \sum_{j}^{K} I^{l}_{n,i,j} \times W^{l}_{n,c,i,j} \times X^{l}_{n,h+i-\frac{K+1}{2},w+j-\frac{K+1}{2}} \tag{2}$$

w와 i에 대한 기울기

$$grad(W^{l}_{n,c,i,j}) = I^{l}_{n,i,j} \times \sum_{h}^{M_H} \sum_{w}^{M_W} \frac{\partial L}{\partial X^{l+1}_{n,h,w}} \times X^{l}_{c,h+i-\frac{K+1}{2},w+j-\frac{K+1}{2}} \tag{3}$$

$$grad(I^{l}_{n,i,j}) = \sum_{c}^{C} (W^{l}_{n,c,i,j} \times \sum_{h}^{M_H} \sum_{w}^{M_W} \frac{\partial L}{\partial X^{l+1}_{n,h,w}} \times X^{l}_{c,h+i-\frac{K+1}{2},w+j-\frac{K+1}{2}}) \tag{4}$$

스트라이프 단위 가지치기(SWP)

FS를 희박하게 만들기 위한 정규화 => 각 필터의 최적 모양 암묵적 학습

$$L = \sum_{(x,y)} loss(f(x, W \odot I), y) + \alpha g(I) \tag{5}$$

L1 norm penalty

$$g(I) = \sum_{l=1}^{L} g(I^l) = \sum_{l=1}^{L} (\sum_{n=1}^{N} \sum_{i=1}^{K} \sum_{j=1}^{K} |I^l_{n,i,j}|). \tag{6}$$

SWP의 컨볼루션 프로세스

$$\begin{aligned} X^{l+1}_{n,h,w} &= \sum_{c}^{C} \sum_{i}^{K} \sum_{j}^{K} W^l_{n,c,i,j} \times X^l_{n,h-i+\frac{K+1}{2},w-j+\frac{K+1}{2}} \quad standard\ \ convolution \\ &= \sum_{i}^{K} \sum_{j}^{K} (\sum_{c}^{C} W^l_{n,c,i,j} \times X^l_{n,h-i+\frac{K+1}{2},w-j+\frac{K+1}{2}}) \quad stripe\ wise\ \ convolution \end{aligned} \tag{7}$$

FILTER SKELETHON

FS를 훈련 중에 최적화하는 것에 대한 결과

Table 1: Test accuracy of each network that only learns the ‘shape’ of the filters.

Dataset	Backbone	Test accuracy
CIFAR-10	VGG16	79.83
	ResNet56	83.82
	MobileNetV2	83.52

FILTER SKELETHON

기준 네트워크와 FS로 학습된 네트워크의 가중치 분포
=> 네트워크가 입력 데이터나 특징 변화에 강하다

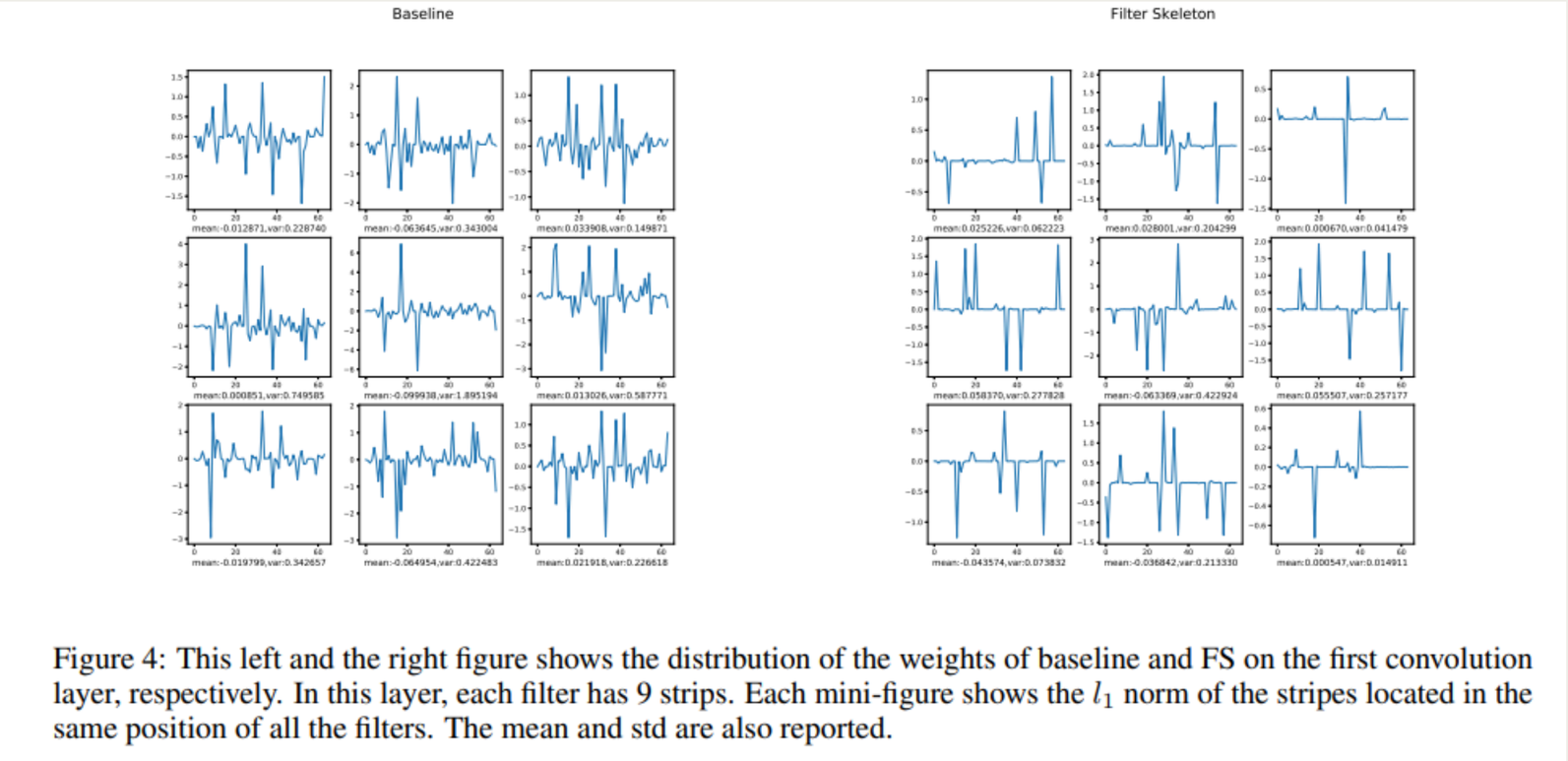


Figure 4: This left and the right figure shows the distribution of the weights of baseline and FS on the first convolution layer, respectively. In this layer, each filter has 9 strips. Each mini-figure shows the l_1 norm of the stripes located in the same position of all the filters. The mean and std are also reported.

VISUALIZE VGG19 FILTERS PRUNED

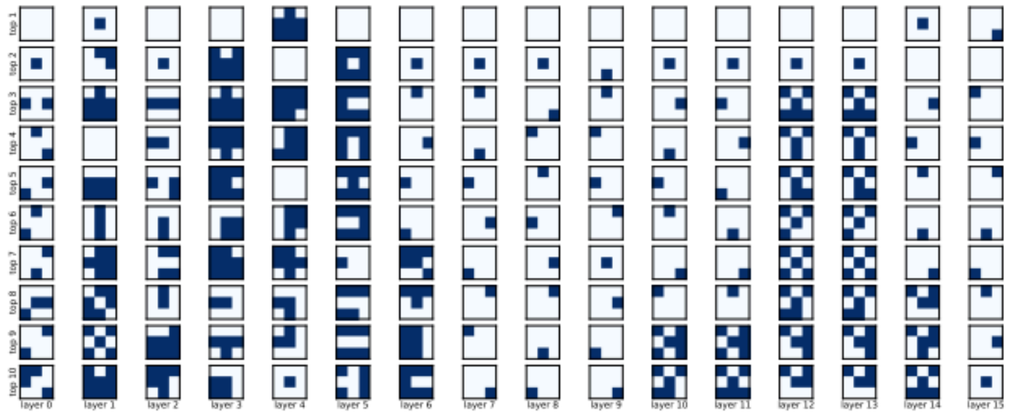


Figure 7: The visualization of the VGG19 filters pruned by SWP. From top to bottom, we display the filters according to their frequency in such layer. White color denotes the corresponding strip in the filter is removed by SWP.

RESULT

We find α = 1e − 5 and δ = 0.05 gives the acceptable pruning ratio and test accuracy.

Table 4: This table shows how α and δ affects SWP results. The experiment is conducted on CIFAR-10. The network is ResNet56.

α	0.8e-5	1.2e-5	1.4e-5	1e-5				
δ	0.05			0.01	0.03	0.05	0.07	0.09
Params (M)	0.25	0.21	0.2	0.45	0.34	0.21	0.16	0.12
Flops (M)	61.16	47.71	41.23	111.68	74.83	56.10	41.59	29.72
Accuracy(%)	92.73	92.43	92.12	93.25	92.82	92.98	92.43	91.83