The Unreasonable Effectiveness of Random Pruning: Return of the Most Naive Baseline for Sparse Trainng

Shiwei Liu1, Tianlong Chen2, Xiaohan Chen2, Li Shen3
Decebal Constantin Mocanu1,4, Zhangyang Wang2, Mykola Pechenizkiy1,

발표자: 문규식

Index

1. **Terminology**
2. Overview
3. Introduction
4. Random Pruning
5. Methodology
6. Experiments
7. Conclusion

Terminology (1)

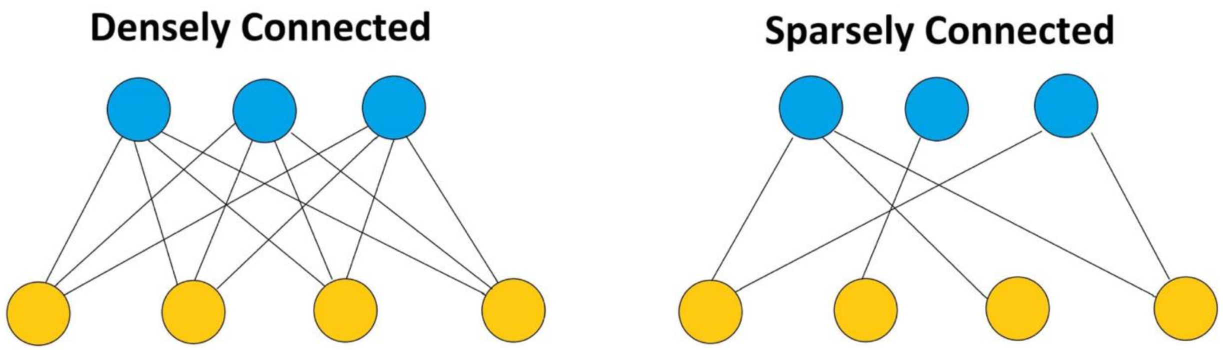
- Dense model
  - 모든 weight가 학습에 참여하는 fully-connected model
  - Pruning이 적용되지 않은 기본 형태의 신경망

- Pruning
  - Model의 weight의 일부를 제거하거나 0으로 만들어 model의 연산량과 메모리 사용량을 줄이는 과정

Terminology (2)

- Sparsity
  - 신경망의 weight 중 0인 값의 비율

- Sparse model
  - 전체 weight 중 상당수가 0으로 설정된 model
  - “상당수”의 기준은 실험 목적과 context에 따라 다름
    - 일반적으로 80%의 sparsity



Index

1. Terminology
2. **Overview**
3. Introduction
4. Random Pruning
5. Methodology
6. Experiments
7. Conclusion

Overview (1)

- Without importance-based pruning, random pruning at initialization enables strong sparse training

- Key findings
  1. Network size matters
     - As networks grow wider and deeper, random pruning becomes more effective
  2. Appropriate layer-wise sparsity ratios
     - Pre-defined layer-wise sparsity ratios boosts performance

Overview (2)

- Outperform dense models in
  - Accuracy
  - Out-of-distribution (OoD) detection
    - Detecting inputs outside the training distribution
  - Uncertainty
    - Reflecting how confident the model is in its prediction
  - Robustness
    - Withstanding noisy or perturbed inputs while maintaining correct predictions

Index

1. Terminology
2. Overview
3. **Introduction**
4. Random Pruning
5. Methodology
6. Experiments
7. Conclusion

Motivation

- Deep learning has led to larger and over-parameterized networks
  - These model achieved high performance
  - But require heavy computational and memory costs
- Sparsity and model compression are now key to efficiency

Limitations of Existing Approaches

- Complex and costly pruning techniques
  - Depend on importance-based criteria
  - Require extra computation and multiple passes over the data
- During or after training pruning
  - Improve inference
  - Does not reduce training cost

Index

1. Terminology
2. Overview
3. Introduction
4. **Random Pruning**
5. Methodology
6. Experiments
7. Conclusion

Categorizing Pruning (1)

- Pruning Granularity
  1. Unstructured pruning
     - 개별 weight 각각을 제거하는 방식
  2. Structured pruning
     - Neuron, channel, filter, layer 등 structure 단위로 제거하는 방식
  3. Semi-structured pruning
     - Weight를 block 단위나 pattern으로 제거하는 방식

Categorizing Pruning (2)

- Pruning Scenario
  1. After training
  2. During training
  3. Before training
     1. Static sparse training (✓)
        - 학습 시작 전 생성된 mask를 사용하여 처음부터 끝까지 동일한 구조로 학습
     2. Dynamic sparse training
        - 학습 도중 mask를 변경하면서 구조를 점진적으로 조정하며 학습

※ Mask: 어떤 weight를 사용할지 제거할지 표시하는 binary 행렬

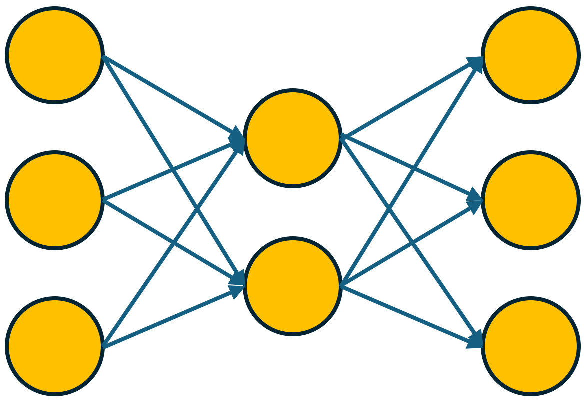
Random Pruning (1)

- Remove weights randomly, without computing importance scores

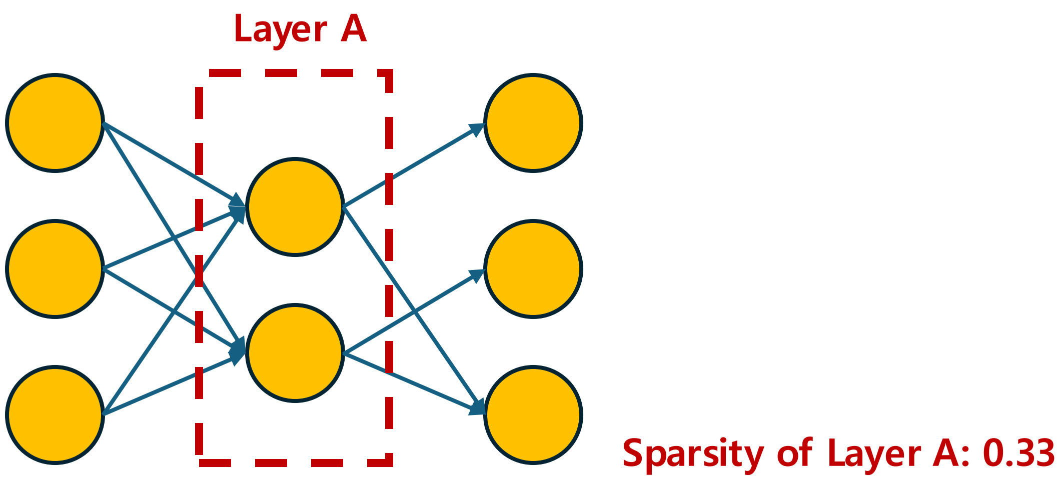
- Key characteristics
  - No importance scores
  - Simple and light weight
    - Does not require additional heuristics or ranking
  - Applicable at various stage
    - Can be applied before, during or after training

Random Pruning (2)

- In this paper
  - Layer-wise sparsity ratios are pre-defined
  - Within each layer, weights are randomly pruned to match the target ratio



Dense Model



Layer-wise Pruning

Index

1. Terminology
2. Overview
3. Introduction
4. Random Pruning
5. **Methodology**
6. Experiments
7. Conclusion

Random Pruning in This Paper

- Random pruning
  - Removes weights or filters in each layer randomly to target sparsity
  - Layer-wise sparsity are pre-defined before training

- Pre-defined layer-wise sparsity ratio
  - Originally designed for:
    - Controlling layer-wise sparsity in importance-based pruning
  - Why it can be used for random pruning:
    - The sparsity ratio is defined independently of importance scores
    - It can be pre-determined based on layer structure

6 Layer-Wise Sparsity Ratios (1)

- 6 layer-wise sparsity ratios
  1. ERK
  2. ERK+
  3. Uniform
  4. Uniform+
  5. SNIP ratio
  6. GraSP ratio

6 Layer-Wise Sparsity Ratio (2)

1. ERK
   - Layer의 크기가 클수록 더 높은 sparsity를 가지도록 설계된 기법

2. ERK+
   - ERK를 수정한 기법
   - Last fully-connected layer를 dense로 유지
   - Last layer 대신 다른 layer에서 더 pruning을 하여 전체 sparsity를 맞춤
   - Why?
     - 일반적으로 마지막 fully-connected layer는 accuracy에 영향을 많이 주는 layer

6 Layer-Wise Sparsity Ratio (3)

3. Uniform
    - 모든 layer에 동일한 sparsity 적용한 기법

4. Uniform+
    - Uniform 기법에 exception 추가
    - First convolutional layer는 dense로 유지 (Sparsity 0%)
    - Last fully-connected layer는 minimum 20%의 parameter를 유지

6 Layer-Wise Sparsity Ratio (4)

5. SNIP ratio
   - Not designed for random pruning
   - Pruning-at-Initialization (PaI) 기법 중 하나
   - $|g \odot w|$의 절대값이 낮은 weight를 제거
     - g: Network weight
     - w: Network gradient
     - $\odot$: 같은 크기의 두 vector나 matrix의 element끼리 곱하는 operation
   - Training 전에 한 번의 iteration만으로 weight를 제거
   - SNIP가 생성한 layer-wise sparsity 비율만을 체택
     - 어떤 weight를 제거할지 나타내는 mask는 버림
     - Random pruning을 위한 mask는 무작위로 생성

6 Layer-Wise Sparsity Ratio (5)

6. GraSP ratio
   - Pruning at Initialization (PaI) 기법 중 하나
   - 각 weight가 gradient norm에 미치는 영향을 평가
     - Gradient norm: 손실 함수의 변화량
   - $-w \odot H_g$: Weight의 중요도를 계산하는 score
     - w: Weight
     - g: gradient
     - H: Hessian
     - H_g: 2차 정보가 반영된 gradient
   - GraSP ratio가 생성한 layer-wise sparsity 비율만을 체택
     - Random pruning을 위한 mask는 무작위로 생성

Index

1. Terminology
2. Overview
3. Introduction
4. Random Pruning
5. Methodology
6. **Experiments**
7. Conclusion

Experiments Settings

Table 1: Summary of the architectures, datasets and hyperparameters used in this paper.

Model	Mode	Data	#Epoch	Batch Size	LR	Momentum	LR Decay, Epoch	Weight Decay
ResNets	Dense	CIFAR-10/100	160	128	0.1	0.9	$10\times$, [80, 120]	0.0005
	Sparse	CIFAR-10/100	160	128	0.1	0.9	$10\times$, [80, 120]	0.0005
Wide ResNets	Dense	ImageNet	90	192*4	0.4	0.9	$10\times$, [30, 60, 80]	0.0001
	Sparse	ImageNet	100	192*4	0.4	0.9	$10\times$, [30, 60, 90]	0.0001

Models

- ResNet
  - Network가 깊어질수록 발생하는 gradient vanishing 문제를 해결한 model
  - Input 값을 skip connection을 통해 직접 다음 layer에 더해줌
  - 100층 이상의 network도 안정적으로 학습 가능
- Wide ResNet
  - ResNet의 너비를 늘린 version
  - Layer의 수를 줄이고, channel의 수를 늘려 연산의 병렬화와 효율 향상
  - 성능은 비슷하거나 더 좋고 학습 속도는 훨씬 빠름

Datasets (1)

- CIFAR-10/100
  - 크기: 32 x 32 RGB image
  - Class 수:
    - CIFAR–10: 10개
    - CIFAR–100: 100개
  - Sample 수: 각 class당 6,000개
  - Computer vision 기초 실험용

Datasets (2)

- ImageNet
  - 크기: 대부분 224 x 224 이상의 image
  - Class. 수: 1,000개
  - Sample 수: 약 140만 장
  - 대규모, 복잡한 data

Settings (1)

- Epoch
  - 전체 학습 dataset의 반복 학습 횟수
- Batch size
  - 한 번의 forward/backward 연산에서 처리하는 data 개수
- Learning rate (LR)
  - Model이 weight를 update할 때 얼만큼 이동할지 결정하는 계수

Table 1: Summary of the architectures, datasets and hyperparameters used in this paper.

Model	Mode	Data	#Epoch	Batch Size	LR	Momentum	LR Decay, Epoch	Weight Decay
ResNets	Dense	CIFAR-10/100	160	128	0.1	0.9	10×, [80, 120]	0.0005
	Sparse	CIFAR-10/100	160	128	0.1	0.9	10×, [80, 120]	0.0005
Wide ResNets	Dense	ImageNet	90	192*4	0.4	0.9	10×, [30, 60, 80]	0.0001
	Sparse	ImageNet	100	192*4	0.4	0.9	10×, [30, 60, 90]	0.0001

Settings (2)

- Momentum
  - 이전 step의 gradient를 일정 비율반영
  - Oscillation을 줄이고 더 빠르게 수렴하도록 돕는 계수
- LR Decay Schedule
  - 학습 중 learning rate를 줄이는 시점과 방식
- Weight Decay
  - Weight가 너무 커지는 것을 방지하는 정규화 기법

Table 1: Summary of the architectures, datasets and hyperparameters used in this paper.

Model	Mode	Data	#Epoch	Batch Size	LR	Momentum	LR Decay, Epoch	Weight Decay
ResNets	Dense	CIFAR-10/100	160	128	0.1	0.9	10×, [80, 120]	0.0005
	Sparse	CIFAR-10/100	160	128	0.1	0.9	10×, [80, 120]	0.0005
Wide ResNets	Dense	ImageNet	90	192*4	0.4	0.9	10×, [30, 60, 80]	0.0001
	Sparse	ImageNet	100	192*4	0.4	0.9	10×, [30, 60, 90]	0.0001

Measurement Metrics

- Out-of-Distribution (OoD)
  - Model이 학습 data와 다른 분포의 input을 감지하는 정도
- Adversarial robustness
  - Noise가 추가되었을 때도 model이 잘 예측하는 정도
- Uncertainty estimation
  - Model이 자신의 예측에 대해 가지고 있는 확신을 측정

Main Findings

- Model의 크기가 커질수록
  1. Random pruning의 성능 향상
  2. Pruning 기법 간의 성능 차이가 줄어듦
- CIFAR-10 실험 (작은 model, data set)
  - ERK 기반 random pruning이 SNIP, GraSP보다 뛰어난 성능을 보일 때도 있음
- ImageNet 실험 (큰 model, data set)
  - SNIP이 ERK 기반 random pruning에 비해 훨씬 좋은 성능을 보임

Main Findings - Depth

- 같은 width이지만 depth를 변경
  - Depth가 커질수록 동일 sparsity에서도 dense model과 유사한 성능을 달성

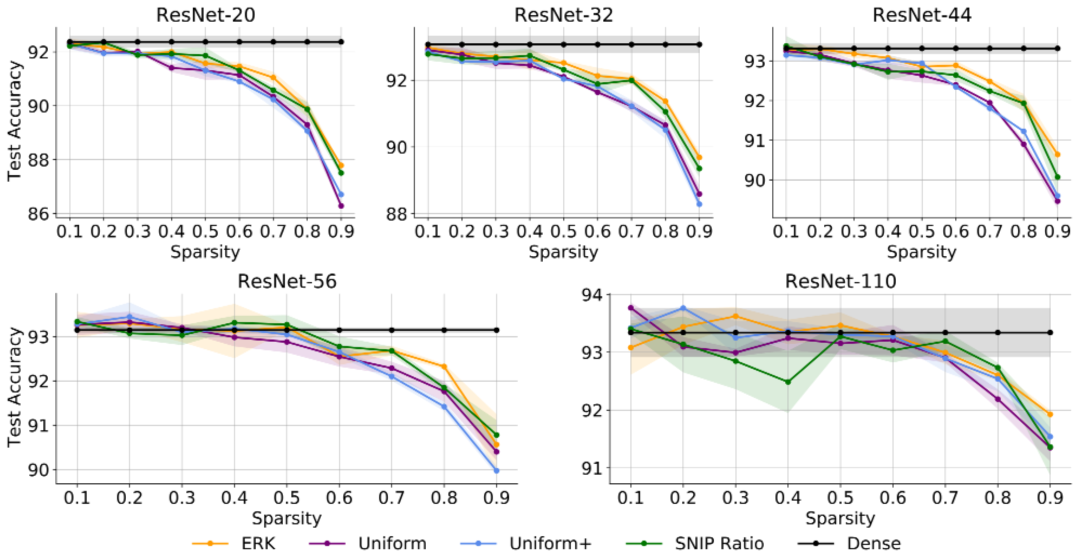


Figure 1: **From shallow to deep.** Test accuracy of training randomly pruned subnetworks from scratch with different depth on CIFAR-10. ResNet-A refers to a ResNet model with A layers in total.

Main Findings - Width

- 같은 depth이지만 width를 변경
  - Width가 커질수록 dense model과 비슷하거나 더 높은 accuracy 유지 가능

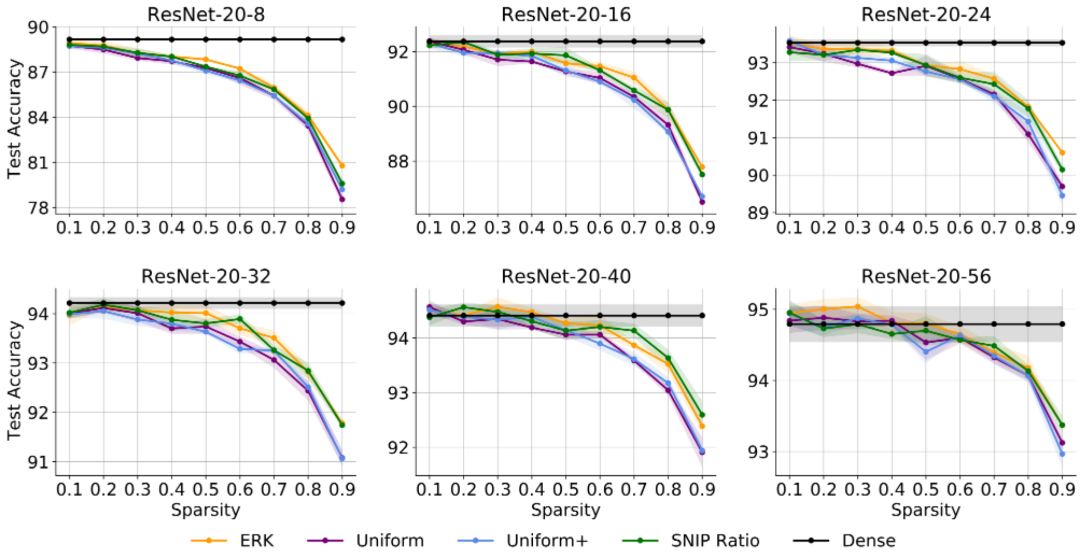


Figure 2: **From narrow to wide.** Test accuracy of training randomly pruned subnetworks from scratch with different width on CIFAR-10. ResNet-A-B refers to a ResNet model with A layers in total and B filters in the first convolutional layer.

Main Findings – Metrics (1)

- Model이 커질수록 향상
  - Uncertainty estimation
  - OoD detection

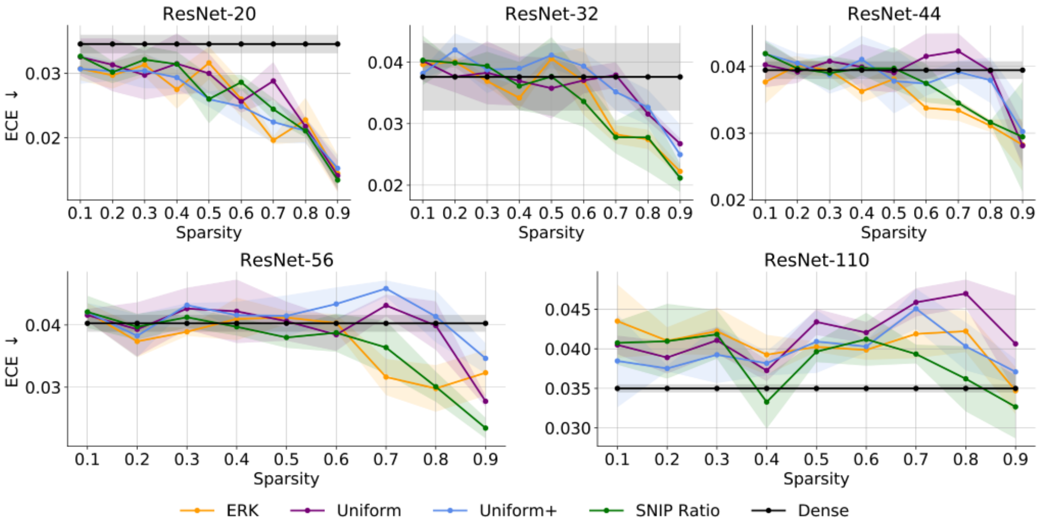


Figure 3: **Uncertainty estimation (ECE).** The experiments are conducted with various models on CIFAR-10. Lower ECE values represent better uncertainty estimation.

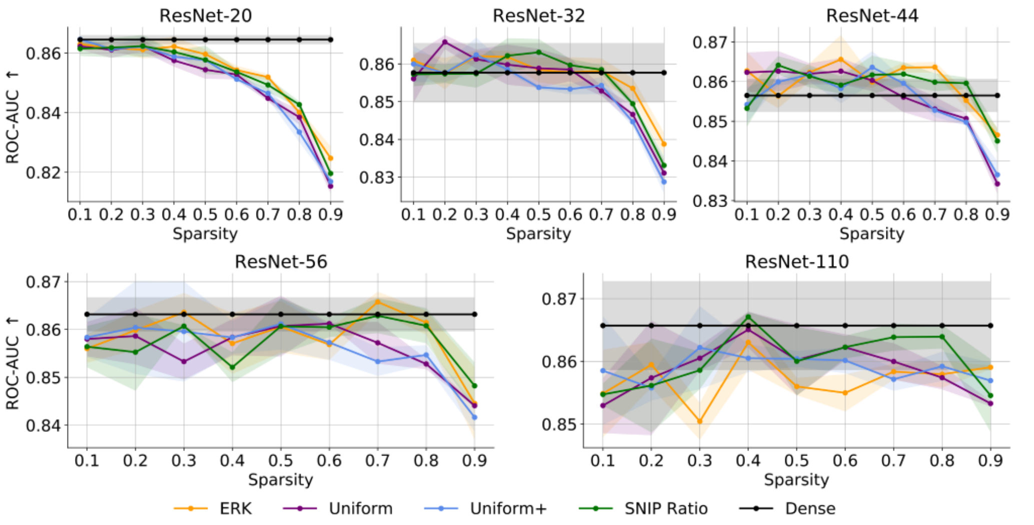


Figure 4: **Out-of-distribution performance (ROC-AUC).** Experiments are conducted with models trained on CIFAR-10, tested on CIFAR-100. Higher ROC-AUC refers to better OoD performance.

Main Findings – Metrics (2)

- Model이 커질수록 향상
  - Adversarial robustness

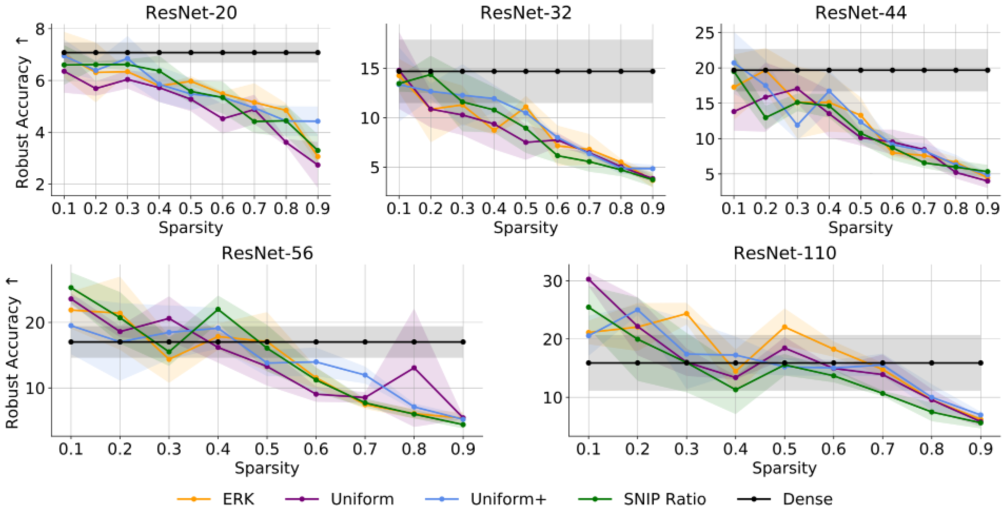


Figure 17: **Adversarial robustness.** The experiments are conducted with various models on CIFAR-10. Higher values represent better adversarial robustness.

ERK vs SNIP (1)

- Why SNIP is better than ERK?
  - Gradient flow를 통한 해석
    - Gradient flow: Weight를 바꾸는 gradient가 전달되는 정도
  - Effective gradient norm
    - Gradient flow를 수치로 측정한 것
    - Pruning 후 남아있는 weight에 대해 gradient norm을 측정

  - Gradient가 더 큰 model이 더 높은 accuracy를 가짐
    - 초기 gradient flow가 강할수록 학습이 잘 된다는 기존 주장과 동일
    - Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In International Conference on Learning Representations, 2020.

ERK vs SNIP (2)

- Why SNIP is better than ERK?

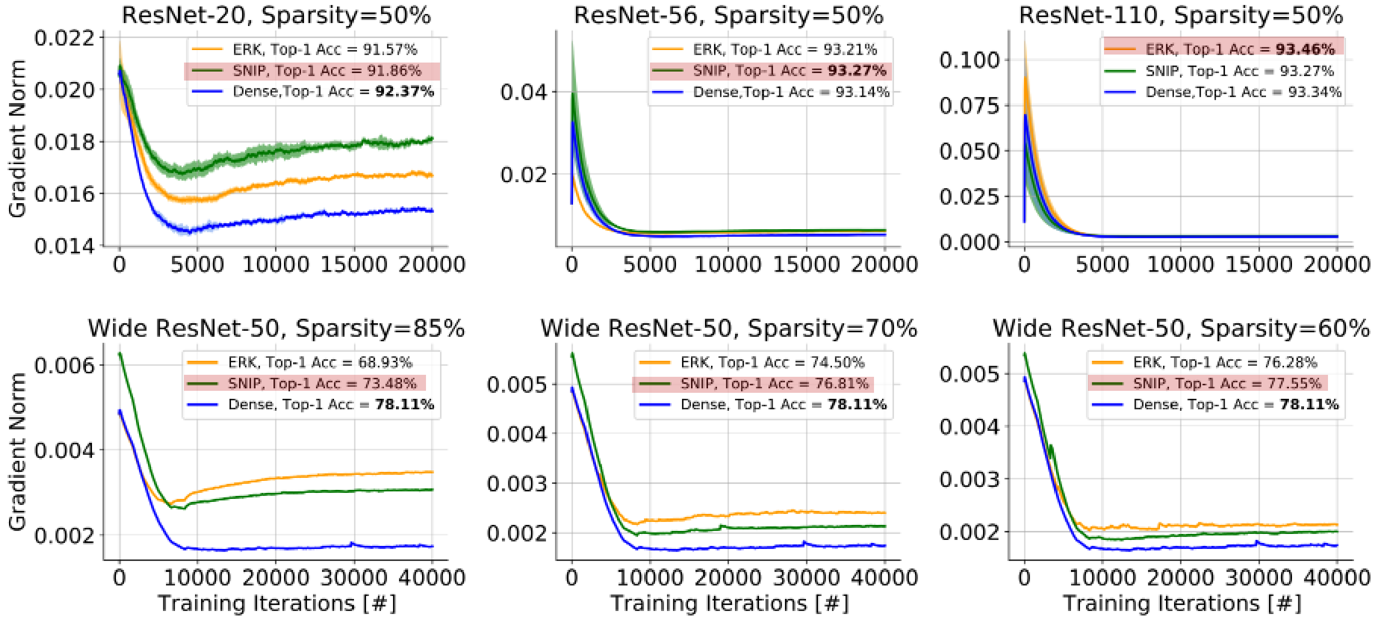
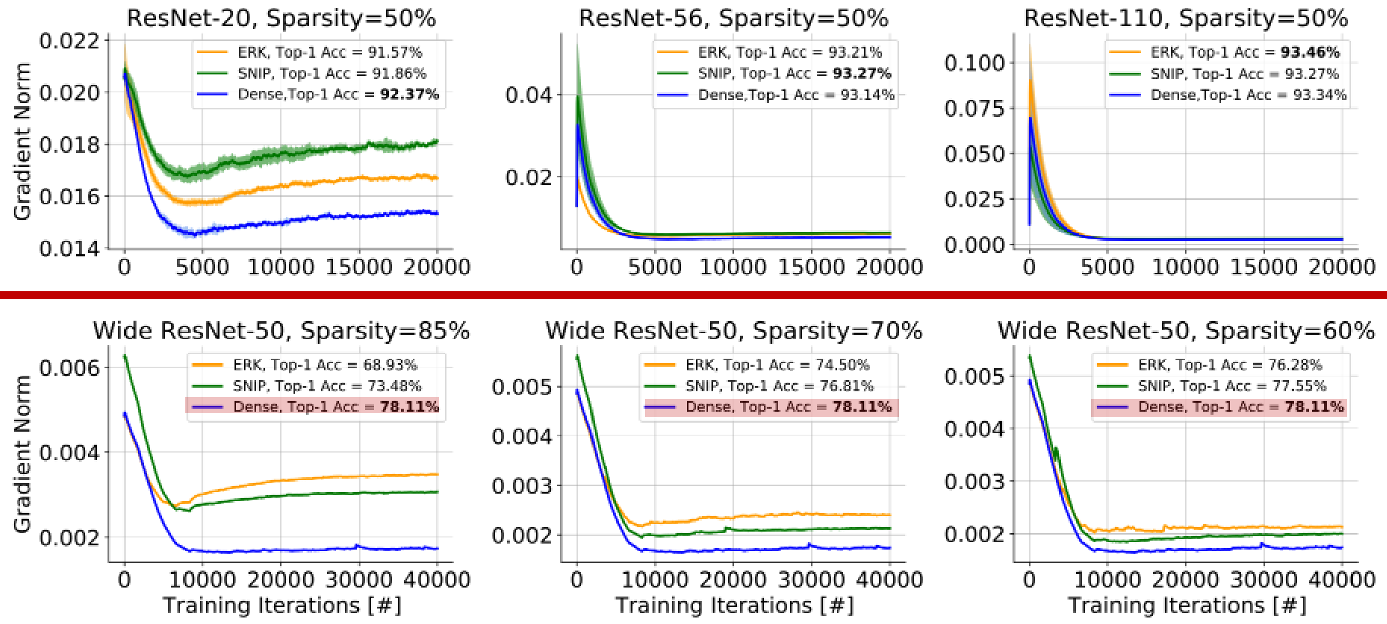


Figure 6: **Gradient norm of randomly pruned networks during training.** *Top:* Comparison between gradient norm of SNIP and ERK with 50% sparse ResNet-20, ResNet-56, and ResNet-110 on CIFAR-10. *Bottom:* Comparison between gradient norm of SNIP and ERK with Wide ResNet-50 on ImageNet at various sparsities.

ERK vs SNIP (3)

- Why SNIP is better than ERK?



초기 gradient flow만으로는 학습 성능을 전부 설명은 불가능

on CIFAR-10. *Bottom:* Comparison between gradient norm of SNIP and ERK with Wide ResNet-50 on ImageNet at various sparsities.

Gradient Flow?

- Gradient는 학습 초반에 상승했다가 감소하여 일정한 수준으로 유지
  - 초기 하락 후 평탄한 시점 (flat phase)
- Flat phase에서의 gradient norm
  - Sparse vs dense model간 성능 차이와 유사하게 움직임

- Flat phase 까지의 gradient까지의 gradient flow를 고려
  - Accuracy의 정확도를 더 잘 예측할 수 있음

Index

1. Terminology
2. Overview
3. Introduction
4. Random Pruning
5. Methodology
6. Experiments
7. **Conclusion**

Conclusion

- Key findings
  - Random pruning
    - 적절한 크기의 모델과 layer-wise sparsity ratio가 있다면 dense model 수준의 성능을 달성할 수 있음
  - 추가 효과
    - OoD detection, uncertainty estimation, adversarial robustness 향상
  - 큰 model일수록 pruning에 대한 robustness가 강하며, random pruning을 사용해도 성능이 유지될 수 있음