Pruning

Q&A Session

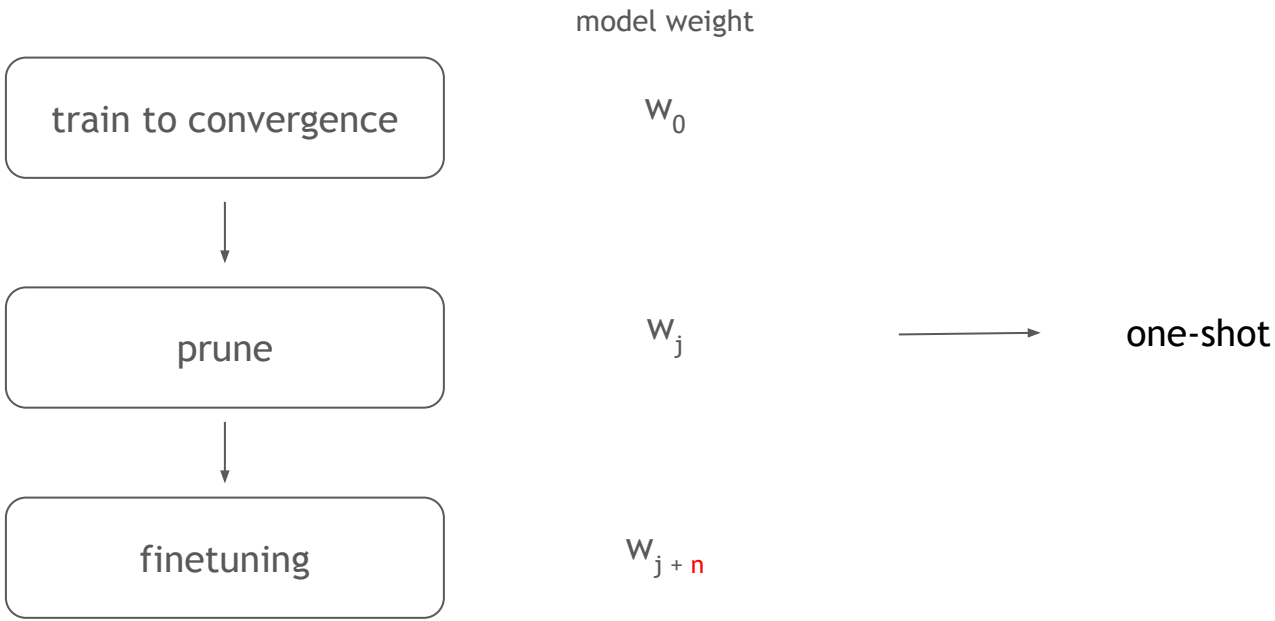
11/20

TA : 김성년, 안재연

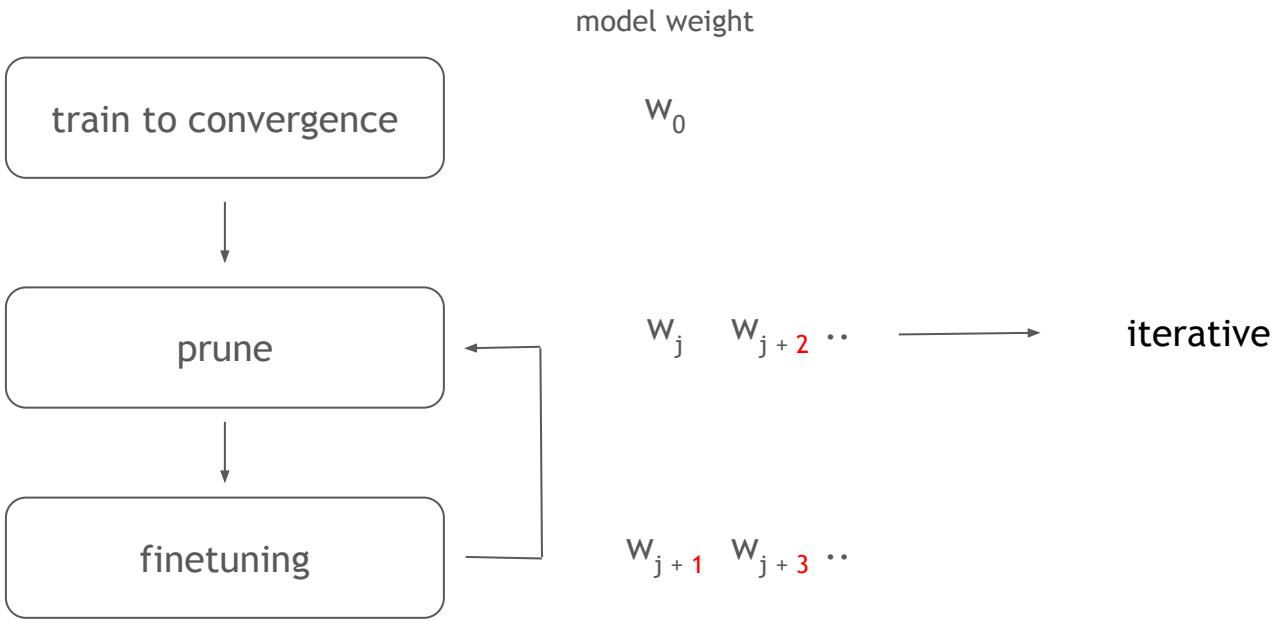
Covering papers as follows:

- Comparing Rewinding and Fine-tuning in Neural Network Pruning, 20 `ICLR
- SNIP: Single-shot Network Pruning based on Connection Sensitivity, 19 `ICLR

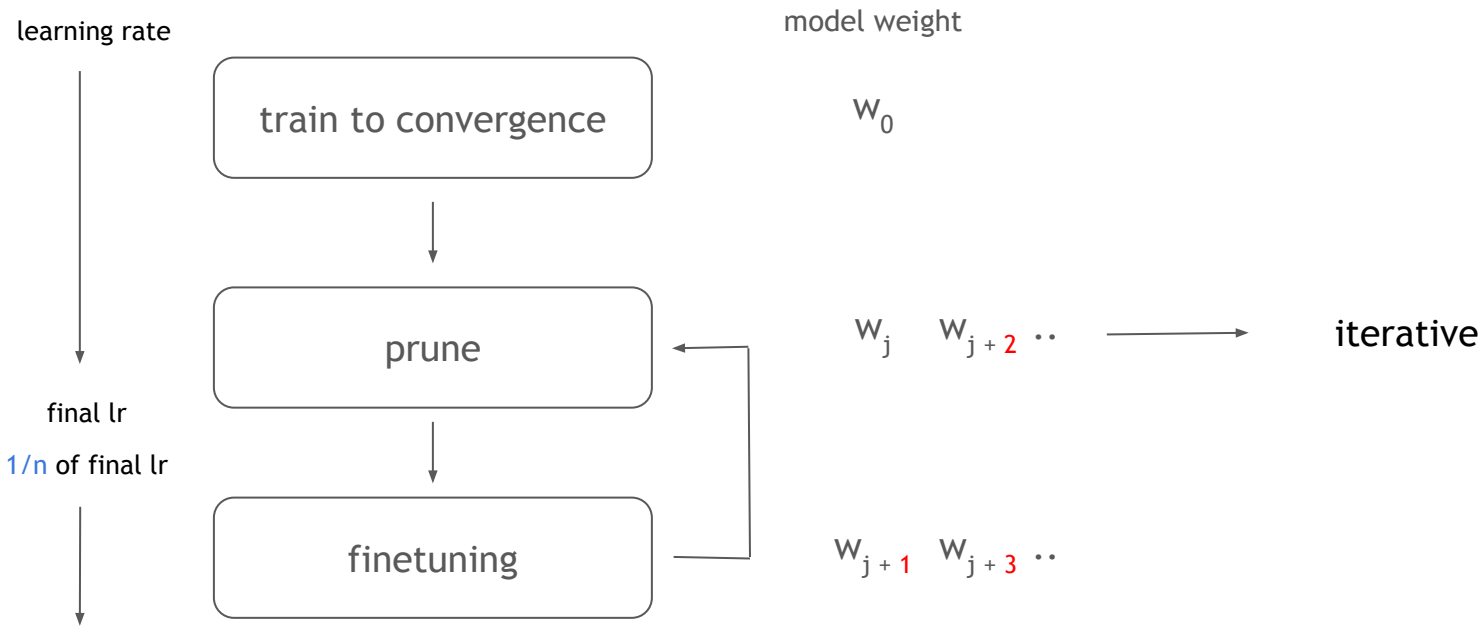
지난 시간 리뷰: 보편적인 pruning process



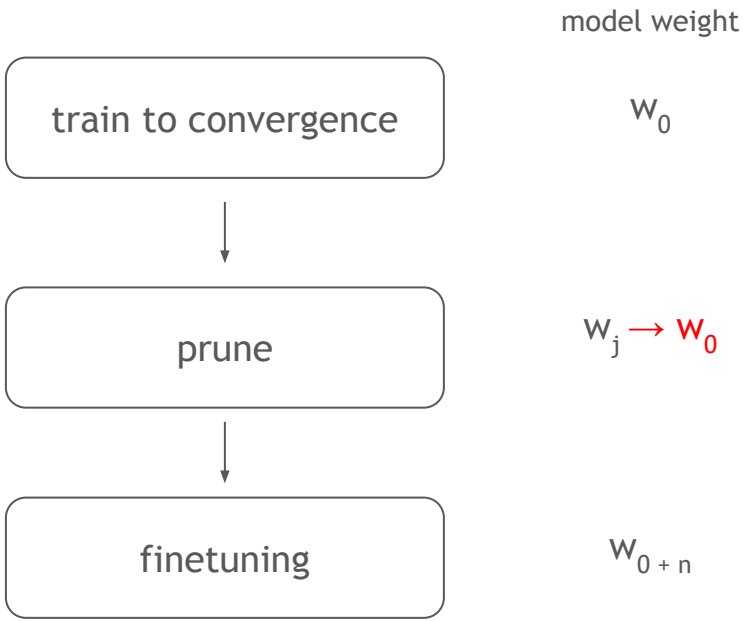
지난 시간 리뷰: 보편적인 pruning process



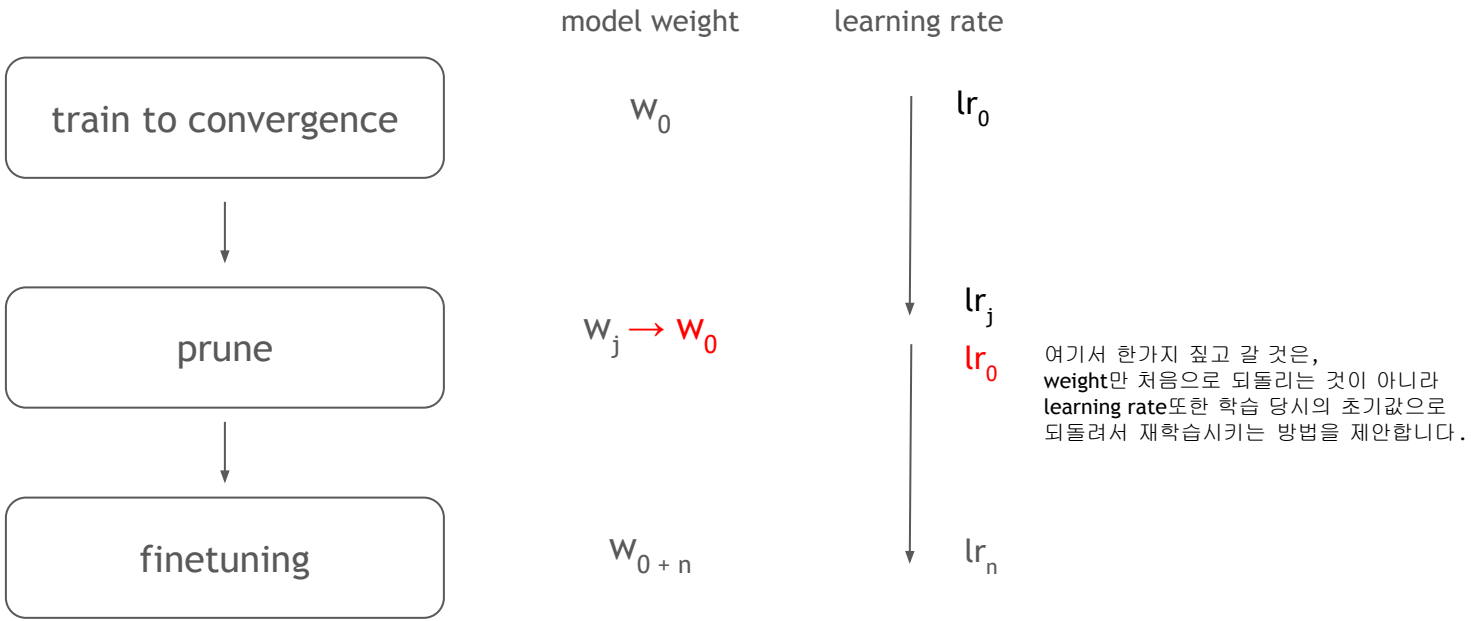
지난 시간 리뷰: 보편적인 pruning process



지난 시간 리뷰: Lottery Ticket Hypothesis; one-shot 또는 iterative하게 구현할 수 있는데, 큰 틀은 다음과 같습니다

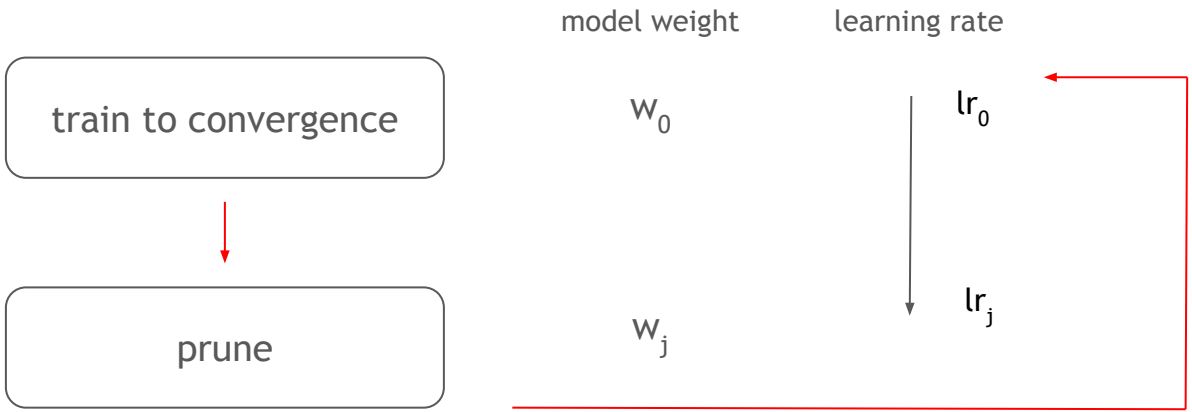


지난 시간 리뷰: Lottery Ticket Hypothesis; one-shot 또는 iterative하게 구현할 수 있는데, 큰 틀은 다음과 같습니다

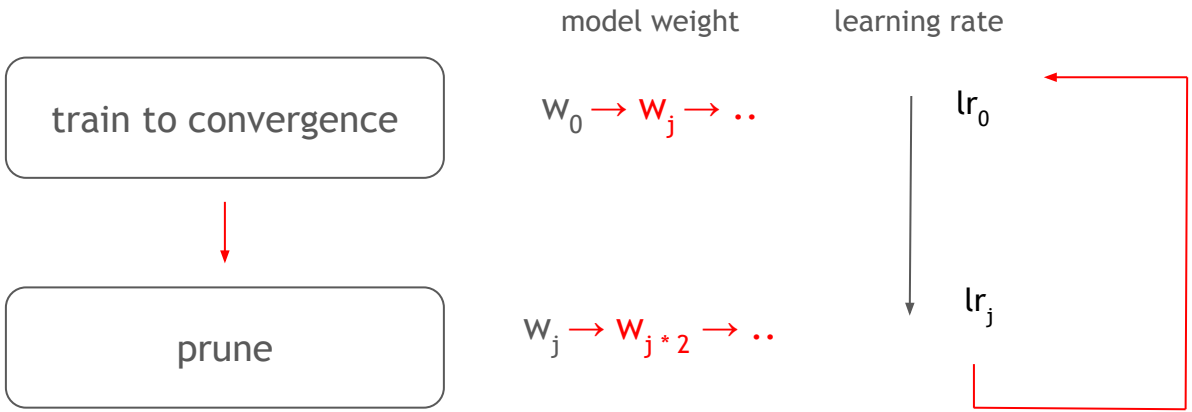


여기서 한가지 짚고 갈 것은,
weight만 처음으로 되돌리는 것이 아니라
learning rate또한 학습 당시의 초기값으로
되돌려서 재학습시키는 방법을 제안합니다.

다시 말해, 원래의 **training** 과정을 다시 **rewind**하는 것과 같다고 볼 수 있습니다.



1. Comparing Rewinding and Fine-tuning in Neural Network Pruning, 20 `ICLR



이 논문에서는 더욱 간단한 방법인 learning rate rewinding을 제시합니다.
앞의 lottery ticket 논문은 (weight + learning rate) 모두 학습 초기값으로 돌렸다면, 여기서는 learning rate만을 초기값으로 돌려도 재학습에 충분함을 보여줍니다. 더 나아가서, 원래 방법보다 더 잘되는 케이스도 보여줍니다!

(첨언: lottery ticket 논문은 실험적으로 좀 제한된 세팅에서만 그 performance를 입증했는데, 이 논문에서는 structured pruning, different task (e.g. machine translation), 더 큰 데이터셋까지 포함하여 실험결과를 보여줬을 뿐만 아니라 가장 중요한 fine-tuning(보편적인 재학습 방법)과의 비교 또한 제시해주었습니다.)

결론적으로 두 rewinding 방법 (lr / weight + lr)이 finetuning보다 더 낫거나 최소한 비슷한 성능을 유지함을 실험적으로 보여주며 기존 방법(finetuning)의 대체재로 쓰일 수 있음을 이야기합니다.

2. SNIP: Single-shot Network Pruning based on Connection Sensitivity, 19 `ICLR

기존 pruning 방법들의 한계: iterative pruning (prune-retrain)을 하면 pruning을 하기 위한 학습 시간도 오래 걸리고, 추가적인 hyperparameter, scheduling 등 고려 사항이 많다.

제안된 방법: 모델을 학습 시키기 전에 pruning을 먼저 하고 학습을 시키자.

2. SNIP: Single-shot Network Pruning based on Connection Sensitivity, 19 `ICLR

학습된 weight value가 없는데 어떻게 중요도를 측정할까?

auxiliary indicator variable c 도입

$$\min_{\mathbf{c},\mathbf{w}} L(\mathbf{c} \odot \mathbf{w}; \mathcal{D}) = \min_{\mathbf{c},\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{c} \odot \mathbf{w}; (\mathbf{x}_i, \mathbf{y}_i)) \,,$$
$$\text{s.t.} \quad \mathbf{w} \in \mathbb{R}^m \,,$$
$$\mathbf{c} \in \{0,1\}^m, \quad \|\mathbf{c}\|_0 \leq \kappa \,,$$

Measure connection sensitivity

(connection에 대해서 loss가 얼마나 민감한가, connection이 미치는 영향)

$$\Delta L_j(\mathbf{w}; \mathcal{D}) = L(\mathbf{1} \odot \mathbf{w}; \mathcal{D}) - L((\mathbf{1} - \mathbf{e}_j) \odot \mathbf{w}; \mathcal{D})$$

2. SNIP: Single-shot Network Pruning based on Connection Sensitivity, 19 `ICLR

$$\Delta L_j(\mathbf{w}; \mathcal{D}) = L(\mathbf{1} \odot \mathbf{w}; \mathcal{D}) - L((\mathbf{1} - \mathbf{e}_j) \odot \mathbf{w}; \mathcal{D})$$

$$\Delta L_j(\mathbf{w}; \mathcal{D}) \approx g_j(\mathbf{w}; \mathcal{D}) = \frac{\partial L(\mathbf{c} \odot \mathbf{w}; \mathcal{D})}{\partial c_j}\bigg|_{\mathbf{c}=\mathbf{1}} = \lim_{\delta \to 0} \frac{L(\mathbf{c} \odot \mathbf{w}; \mathcal{D}) - L((\mathbf{c} - \delta\, \mathbf{e}_j) \odot \mathbf{w}; \mathcal{D})}{\delta}\bigg|_{\mathbf{c}=\mathbf{1}} \tag{5}$$

단 한번의 forward-backward pass로 c에 대한 derivative 만 구하면 됨!

2. SNIP: Single-shot Network Pruning based on Connection Sensitivity, 19 `ICLR

Algorithm 1 SNIP: Single-shot Network Pruning based on Connection Sensitivity

Require: Loss function L, training dataset $\mathcal{D}$, sparsity level κ ▷ Refer Equation 3

Ensure: $\|\mathbf{w}^*\|_0 \leq \kappa$

1: $\mathbf{w} \leftarrow$ VarianceScalingInitialization ▷ Refer Section 4.2

2: $\mathcal{D}^b = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^b \sim \mathcal{D}$ ▷ Sample a mini-batch of training data

3: $s_j \leftarrow \frac{|g_j(\mathbf{w}; \mathcal{D}^b)|}{\sum_{k=1}^m |g_k(\mathbf{w}; \mathcal{D}^b)|}, \quad \forall j \in \{1 \dots m\}$ ▷ Connection sensitivity

4: $\tilde{\mathbf{s}} \leftarrow$ SortDescending($\mathbf{s}$)

5: $c_j \leftarrow \mathbb{1}[s_j - \tilde{s}_\kappa \geq 0], \quad \forall j \in \{1 \dots m\}$ ▷ Pruning: choose top-κ connections

6: $\mathbf{w}^* \leftarrow \arg\min_{\mathbf{w} \in \mathbb{R}^m} L(\mathbf{c} \odot \mathbf{w}; \mathcal{D})$ ▷ Regular training

7: $\mathbf{w}^* \leftarrow \mathbf{c} \odot \mathbf{w}^*$

variance scaling initialization:
초기 weight 값이 너무 크면
activation saturated
→ uninformative gradients

pruning 한번 하고, 그
이후에는 일반적인 학습

감사합니다