DPRPruning

This is just the beginning of something big.

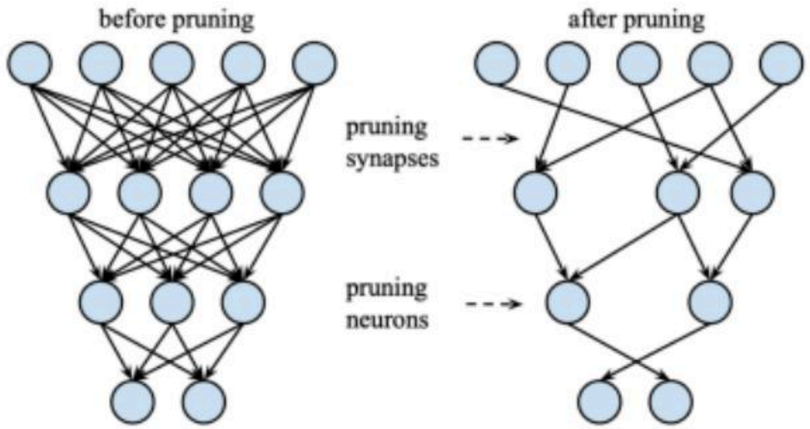
Abstract

Large language models (LLMs) deliver impressive results but face challenges from increasing model sizes and computational costs. Structured pruning reduces model size and speeds up inference but often causes uneven degradation across domains, leading to biased performance. To address this, we propose *DRPruning*, a method that dynamically adjusts the data distribution during training to restore balanced performance across heterogeneous and multi-tasking data. Experiments in monolin-

- 대형 언어모델(LLM)은 인상적인 성과를 내지만, 모델 크기와 계산 비용의 증가로 인한 도전에 직면해 있음
- 구조적 프루닝은 모델 크기를 줄이고 추론을 가속하지만, 도메인 간 성능 저하가 불균일하게 발생하여 편향된 성능을 유발하는 경우가 많다. 이를 해결하기 위해 DRPruning을 제안한다.

Background

Structured Pruning(구조적 프루닝)



- Pruning은 neural network를 경량화하고자 할 때 사용하는 방법이다.
- 좌측 사진은 Pruning을 나타낸 것인데 모든 node가 연결이 되어있던 왼쪽 그림으로 오른쪽과 같이 synapse(혹은 edge)와 neuron(혹은 node)를 없애는 것
- 당연히 무작정 없애면 안 되고 보통은 parameter가 0에 가깝다거나 훈련을 거의 안 했다거나 하는 지표를 가지고 판단하여 pruning 하게 됨.

Background

Structured Pruning(구조적 프루닝)

Structured vs Unstructured

- structured 방법은 대표적으로 channel pruning이 있는데 convolution network에서 상대적으로 필요 없는 channel을 뽑아서 없애는 방법
  - 이런 식으로 structured pruning은 어떤 구조를 통째로 날려버리는 방법.
  - 이 방법의 장점은 구조를 날리는 것이니 matrix 연산을 안 해도 되므로 pytorch, tensorflow 같은 프레임워크와 잘 호환되어 inference 속도를 개선할 수 있다는 것.
  - 하지만 단점으로는 구조를 통째로 날리는 것이다 보니 pruning 하는 비율을 높게 하기는 어렵다는 것.

Background

Structured Pruning(구조적 프루닝)

Structured vs Unstructured

- unstructured 방법은 figure 1에서 보았던 그림과 같이 구조와 상관없이 그냥 특정 기준을 세워서 (보통 0 근처의 weight) 가지치기하듯 weight를 0으로 만들어버리는 것
  - 이 방법의 장점은 필요 없다고 판단되는 weight를 0으로 만드는 것이라 높은 비율로 pruning 할 수 있다는 것
  - 단점으로는 pruning을 했으나 실제로는 0의 값을 가지므로 기존의 프레임워크를 사용하여 matrix 연산을 할 때 계산을 하긴 해야 하므로 실질적인 inference 속도를 개선하지는 못한다.

Background

Structured Pruning(구조적 프루닝)

(Xia et al., 2024). For each granularity i, pruning masks $Z = \{\mathbf{z}^i \mid \mathbf{z}^i \in \mathbb{R}^{D_i}\}$ are learned to determine whether substructures are pruned or retained, where $z^i_j = 0$ indicates pruning of the j-th substructure. Pruning is applied at various granularities, including transformer layers, hidden dimensions, attention heads, and FFN intermediate dimensions.

이 논문에서는 structured pruning을 다룬다.

그런데 이것을 어떻게 하느냐?

- **각 프루닝 단위(레이어/헤드 등) 수준 i마다, 그 단위 안의 하위 구조를 남길지(1) 지울지(0) 결정하는 마스크를 학습한다.**
- **매개화: 마스크는 하드 콘크리트(hard-concrete) 분포를 쓰는 L0 정규화 기법으로 파라미터화한다. 이 분포는 확률질량을 0 또는 1에 몰리게 하므로, "남길지/지울지"가 또렷해진다.**

Background

Structured Pruning(구조적 프루닝)

To parameterize the masks, the ℓ_0 regularization method (Louizos et al., 2018) with hard concrete distributions is used to concentrate probability mass at 0 or 1. Lagrange multipliers are then used to ensure the pruned model meets the target configuration. Specifically, if exactly t^i parameters must be retained for $\mathbf{z}^i$, the following constraint is imposed:

$$\tilde{\ell}^i = \lambda^i \left(\sum_j z_j^i - t^i \right) + \phi^i \left(\sum_j z_j^i - t^i \right)^2 . \quad (1)$$

The final training loss integrates these constraints with the language modeling loss of the pruned model, jointly optimizing the model parameters θ and pruning masks $\mathbf{z}$, with $\mathbf{z}$ typically uses a higher learning rate. After pruning, the highest-scoring components are retained.

- 목표 크기 맞추기(제약): “각 단위 i에서 정확히 t_{i}개를 남겨라” 같은 제약을 라그랑주 항으로 강제한다.
- 첫 항은 과도하게 많이 남기는 걸(상한 위반) 페널티로 주고, 둘째 항은 목표 수에서 벗어난 정도를 제곱 페널티로 잡아 정확히 목표 크기를 맞추도록 유도한다.

→ “하드 콘크리트 + L0”는 마스크를 0/1에 가깝게 만들고, 라그랑주 항은 “남길 개수”를 딱 맞추게 한다.

Background

Distributionally Robust Optimization

$$\underset{\theta}{\text{minimize}} \sup_{Q \in \mathcal{Q}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim Q}[\ell(\mathbf{x}, \mathbf{y}; \theta)].$$

To solve the min-max optimization, the iterative best response algorithm (Fudenberg and Levine, 1998) is used. Each iteration consists of first performing the empirical risk minimization on the current data distribution $\mathbf{q}^t$, followed by updating the data distribution using worst-case weights based on the current parameters. Formally,

$$\begin{aligned} \theta^{t+1} &\leftarrow \underset{\theta}{\text{argmin}} \sum_i q_i^t \ell\left(\theta; D_i\right), \\ \mathbf{q}^{t+1} &\leftarrow \underset{\mathbf{q}=\{q_1,\ldots,q_n\}\in\mathcal{Q}}{\text{argmax}} \sum_i q_i \ell\left(\theta^{t+1}; D_i\right). \end{aligned} \tag{3}$$

- 구조적 프루닝을 하면 도메인(예: 위키, 웹크롤, 책, 코드 등)별 성능 하락폭이 불균등해지기 쉽다. 어떤 도메인은 괜찮아도, 어떤 도메인은 크게 무너질 수 있다. 그래서 학습 도중 “어떤 테스트 분포가 와도 나쁘지 않게” 만들려는 DRO를 도입한다.

Background

Distributionally Robust Optimization

$$\underset{\theta}{\text{minimize}} \sup_{Q \in \mathcal{Q}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim Q}[\ell(\mathbf{x}, \mathbf{y}; \theta)].$$

DRO의 기본 문제 설정

- **여러 잠재적 테스트 분포 Q들(도메인 혼합)을 상정하고, 그 최악의 기대손실을 줄이는 θ를 찾는다.**
- **즉, 워스트케이스 도메인 조합에 대해 성능 하한을 끌어올리려는 목표.**

Background

Distributionally Robust Optimization

$$\theta^{t+1} \leftarrow \operatorname*{argmin}_{\theta} \sum_i q_i^t \ell\left(\theta; D_i\right),$$
$$\mathbf{q}^{t+1} \leftarrow \operatorname*{argmax}_{\mathbf{q}=\{q_1,\ldots,q_n\}\in\mathcal{Q}} \sum_i q_i \ell\left(\theta^{t+1}; D_i\right). \quad (3)$$

어떻게 푸나? → 교대 갱신(Iterative Best Response)
반복적으로

1. 현재 데이터 혼합 q_{t}로 ERM(경험위험최소화)을 해서 θ를 업데이트하고,
2. 그 θ에서 가장 나쁜 쪽으로 데이터 혼합 q_{t+1}을 다시 잡는다.

이렇게 하면 학습은 현재 약한 도메인에 더 무게를 두며 진행됩니다.

→ “현재 모델이 어디서 약한지”를 평가 손실로 보고, 약한 쪽 비중을 키워 회복시키는 식으로 도메인 간 불균형을 줄인다.

DPRPruning Method

Integrate DRO into pruning and continued pre-training. During training, we use DRO to dynamically adjust the data ratio to improve the model's robustness and convergence speed. Specifically, to

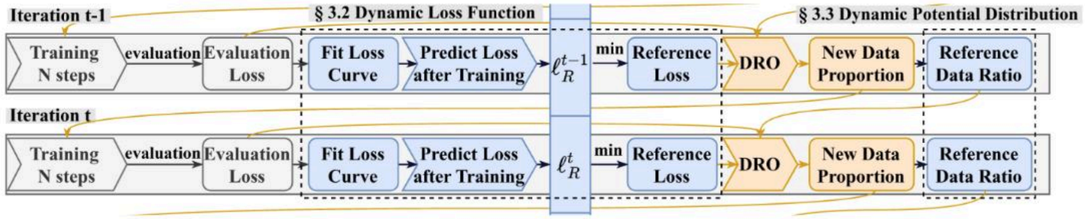


Figure 1: Data proportion update procedure for DRPruning. The gray part represents the standard training process, the yellow part represents the normal process for DRO, and the blue part represents our newly added module.

- 구조적 프루닝을 하면 도메인(예: 위키, 웹크롤, 책, 코드 등) 별 성능 하락폭이 불균등해지기 쉽다고 했다.
- DPRPruning은 이를 개선하기 위한 방법으로, 구조적 프루닝 + 계속 사전학습(continued pretraining)에 DRO(분포 강건 최적화)를 얹어, "도메인별로 성능이 불균등하게 무너지는 문제"를 데이터 비율을 동적으로 조정해가며 복구하는 방법이다.

DPRPruning Method

3.2 Dynamic Loss Function

To stabilize DRO training and prevent domains with slow convergence from disproportionately influencing the weights, the use of a ***reference loss*** ℓ_R is a common approach (Oren et al., 2019; Zhou et al., 2021). This reference loss establishes the minimum acceptable performance for a domain. Furthermore, we update the loss score as $\ell(\theta; D) \leftarrow \ell(\theta; D) - \ell_R$. Proper tuning of ℓ_R can significantly improve performance (Jiao et al., 2022). However, determining an appropriate value remains a challenging task.

Minimum performance estimation. To address this, we predict the model's loss at the end of training as an estimate of the minimum acceptable performance. Specifically, we leverage scaling laws to capture training dynamics and forecast the loss based on evaluation loss trends (Kaplan et al., 2020; Zhang et al., 2024a). Given the number of parameters P and the current training step T, the predicted training loss is estimated by:

$$\hat{\ell}(P, T) = A \cdot \frac{1}{P^{\alpha}} \cdot \frac{1}{T^{\beta}} + E, \qquad (4)$$

Dynamic Loss Function

- DRO 학습을 안정화하고, 수렴이 느린 도메인이 가중치에 과도한 영향을 미치지 않도록 하기 위해, 참조 손실 ℓ_R을 사용하는 것이 일반적이다. 이 참조 손실은 한 도메인에 대해 허용 가능한 최소 성능을 설정한다.
- 더 나아가 손실 점수를 ℓ(θ; D) ← ℓ(θ; D) − ℓ_R로 갱신한다. ℓ_R을 적절히 조정하면 성능이 크게 향상될 수 있다(그러나 적절한 값을 정하는 일은 여전히 도전적이다.

Minimum performance estimation

- 이를 해결하기 위해, 우리는 학습 종료 시점에서의 모델 손실을 예측하여 허용 가능한 최소 성능의 추정치로 사용한다.
- 구체적으로, 스케일링 법칙(scaling laws) 을 활용해 학습 동역학을 포착하고, 평가 손실 추세를 바탕으로 손실을 예측한다.

DPRPruning Method

$$\hat{\ell}(P,T) = A \cdot \frac{1}{P^{\alpha}} \cdot \frac{1}{T^{\beta}} + E,$$

Minimum performance estimation

- 이어서, 파라미터 수 P와 현재 학습 스텝 T가 주어졌을 때, 예측 학습 손실은 좌측 식과 같이 추정한다.
- 여기서 A,E,α,β는 학습 가능한 파라미터이다. 각 도메인에 대해 각 평가 이후 데이터 포인트를 하나 수집하고, 지금까지 모은 모든 포인트에 대해 곡선을 재적합(refit) 한 뒤, 예측된 곡선을 사용해 학습 종료 시 손실(예측 최소 성능)을 추정한다.

DPRPruning Method

Reference loss adjustment. Subsequently, we set the reference loss using the predicted minimum performance. In our preliminary experiments, this approach exhibits strong numerical stability. To accelerate convergence, we adopt the minimum value as the reference loss. This dynamically evaluates domains with poorer performance, allowing DRO to assign higher weights to these domains, thereby promoting faster model convergence.

Reference loss adjustment

- 이후, 예측된 최소 성능을 사용해 참조 손실을 설정한다. 수렴을 가속하기 위해 우리는 최솟값을 참조 손실로 채택한다. 이는 성능이 낮은 도메인을 동적으로 평가하게 하여, DRO가 해당 도메인에 더 높은 가중치를 부여하도록 만들고, 결과적으로 모델 수렴을 가속한다.

DPRPruning Method

3.3 Dynamic Potential Distribution

Sagawa et al. (2019) consider robustness to arbitrary subpopulations, which is overly conservative and degenerates into training only on the highest-loss domain. To address this issue, Zhou et al. (2021) propose a more reasonable assumption by restricting $\mathcal{Q}$ in Eqn. 2 to an f-divergence ball (Csiszár, 1967) around a ***reference data ratio*** $\mathbf{p}_R$. This yields promising results, better ensuring domain balance (Jiao et al., 2022). Formally,

$$\mathcal{Q} = \left\{ \mathbf{q} : \chi^2 \left(\mathbf{q}, \mathbf{p}_R \right) \leq \rho \right\}. \tag{5}$$

However, this assumption can be too restrictive, necessitating a carefully chosen reference data ratio $\mathbf{p}_R$. An unreasonable choice may reduce the model's robustness to distributional shifts.

- Sagawa et al. (2019)는 임의의 하위 집단에 대한 강건성을 고려하는데, 이는 지나치게 보수적이며 최고 손실 도메인만으로 학습하는 상태로 퇴화한다.
- 이를 해결하기 위해, Zhou et al. (2021)은 식 (2)의 Q를 참조 데이터 비율 pR 주변의 f-발산 볼(Csiszár, 1967) 로 제한하는 보다 합리적인 가정을 제안한다. 이는 유망한 결과를 보였고, 도메인 균형을 더 잘 보장한다(Jiao et al., 2022). 형식적으로는 좌측의 식과 같다.
- 그러나 이 가정은 지나치게 제약적일 수 있으며, 참조 데이터 비율 pR을 신중하게 선택해야 한다. 부적절한 선택은 분포 이동에 대한 모델의 강건성을 떨어뜨릴 수 있다.

DPRPruning Method

Reference data ratio adjustment. To address this, we propose a method that combines the strengths of the aforementioned approaches. We still employ Eqn. 5 to constrain the distribution within a limited range, while gradually shifting the reference data ratio towards domains with higher losses to improve the model's robustness to more challenging distributions. To ensure adequate training across all traversed potential distributions, we gradually update the reference ratio.

$$\mathbf{p}_R^{t+1} = \delta \cdot \mathbf{q}^t + (1 - \delta) \cdot \mathbf{p}_R^t.$$

Reference data ratio adjustment

- 분포를 제한된 범위 내에 두되, 참조 데이터 비율을 점진적으로 손실이 더 큰 도메인 쪽으로 이동시켜, 보다 도전적인 분포에 대한 모델의 강건성을 향상시킨다.
- 또한, 학습이 거쳐 가는 모든 잠재 분포에 대해 충분한 학습을 보장하기 위해, 참조 비율을 점진적으로 갱신한다.
- 기존 참조 비율들과 비교하면, DRO 방법은 손실이 큰 도메인에 동적으로 더 높은 가중치 q 를 할당한다. 이 방법은 수치적 안정성이 좋아, 이를 참조 비율 갱신에 활용한다. 형식적으로, 좌측 식과 같이 갱신한다.

Experiments

4 Experiments

4.1 Experimental Setup

Model. Llama2-7B model (Touvron et al., 2023b) is used as the base model. We employ the same target architecture as Sheared Llama for structured pruning to ensure a fair comparison. We compare our method, i.e., **DRPruning**, to strong open-source models of similar sizes, including **Pythia**-1.4B and 2.8B (Biderman et al., 2023) and **Sheared Llama**-1.3B and 2.7B. Additionally, we reproduce Sheared Llama, using the same data settings to control for other variables (**ReSheared**). Further details are provided in Appendix A.1.

실험은 당장 중요한 부분은 아니니 간단히 넘어가겠습니다.

어째거나 도메인에 따른 불균일한 성능 문제가 DPRPruning이라는 method로 완화가 된다면 성공적인 실험이라고 할 수 있겠죠.

그 외에도 더 빠른 수렴과 계산 리소스 감소가 있다면 의의가 있겠습니다.

Results

	Model	CC	C4	GitHub	Book	Wiki	ArXiv	StackExchange	Average
1.3B	**Constant**	35.00	40.00	88.00	47.75	21.75	82.00	72.50	55.29
	Sheared Llama	21.75	26.00	93.75	33.50	29.50	38.50	56.50	42.79
	ReSheared	30.50	30.25	89.25	32.00	23.00	81.00	47.50	47.64
	DRPruning	**44.00**	**51.50**	**94.75**	**48.00**	**33.50**	**86.50**	**90.00**	**64.04**
2.7B	**Sheared Llama**	81.25	**89.50**	95.50	**96.50**	**89.25**	90.50	82.75	**89.32**
	ReSheared	61.75	60.25	96.00	73.00	80.50	**93.75**	**92.00**	79.61
	DRPruning	**82.25**	77.75	**99.00**	86.75	87.50	79.50	89.25	86.00

Table 5: Domain-level results under the benchmark we generated. The abbreviations of tasks refer to the evaluation of seven domains used for training in RedPajama.