📘

Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time

Abstract

- 기존의 성능이 좋은 모델을 뽑아내는 방법은 여러 모델을 여러 하이퍼파라미터에 대해 학습시켜 가장 validation 성능이 좋은 모델을 뽑고, 나머지를 버리는 형태였다.
- 본 논문에서는 validation 단계에서 가장 좋은 것을 뽑는 단계를 큰 모델의 파인튜닝과 비슷한 개념으로 revisit한다.
- 다양한 하이퍼 파라미터의 모델 weight값들을 평균하는 것이 정확성과 견고성을 가질 수 있도록 도울 수 있다.

Method

Table 2: The primary methods contrasted in this work. Each θ_i is a model found through fine-tuning from a shared initialization. Cost refers to the memory and compute requirements during inference relative to a single model. All methods require the same training.

	Method	Cost
Best on val. set	$f(x, \arg\max_i \text{ValAcc}(\theta_i))$	$\mathcal{O}(1)$
Ensemble	$\frac{1}{k}\sum_{i=1}^{k} f(x, \theta_i)$	$\mathcal{O}(k)$
Uniform soup	$f\left(x, \frac{1}{k}\sum_{i=1}^{k} \theta_i\right)$	$\mathcal{O}(1)$
Greedy soup	Recipe 1	$\mathcal{O}(1)$
Learned soup	Appendix I	$\mathcal{O}(1)$

- $f(x, \theta) : x\ denotes\ data,\ and\ \theta\ denotes\ parameters.$

- Greedy soup은 각각의 모델들을 순차적으로 더하여주는 것으로 구성된다.

1. 먼저 모델을 validation accuracy의 내림차순으로 정렬한다.
2. 따라서 Greedy soup은 이전에 들어온 모델보다 좋지 못할 케이스를 배제할 수 있다.
3. 더 나아가 기울기 기반의 모델 weight 보간 레시피(soup)를 탐색하였다.

Recipe 1 GreedySoup

Input: Potential soup ingredients $\{\theta_1, ..., \theta_k\}$ (sorted in decreasing order of ValAcc(θ_i)).
ingredients $\leftarrow \{\}$
for $i = 1$ **to** k **do**
  if ValAcc(average(ingredients $\cup \{\theta_i\}$)) $\geq$ ValAcc(average(ingredients)) **then**
    ingredients $\leftarrow$ ingredients $\cup \{\theta_i\}$
return average(ingredients)

Experiments

Error landscape visualizations

본 논문의 저자들은 training loss와 test error을 2개의 dimensional slice로 직관적으로 시각화를 진행하였다. 이 실험에서 저자들은 solution θ_1 과 θ_2 를 도출하기 위해 zero-shot initialization $\theta_0 \in R^d$ 와 fine-tune을 2번씩 사용하였다. solution은 parameter space의 plane이며 이를 ImageNet train loss, ImageNet test error, 5개의 분포 이동에 대한 error을 평가하였다.

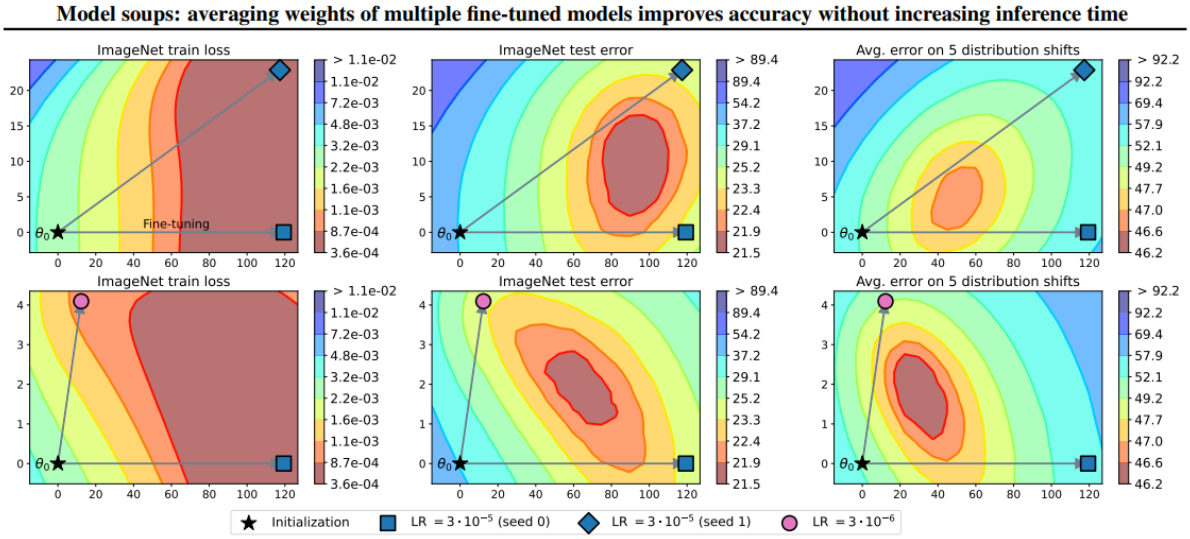

Figure 2: The solution with the highest accuracy is often not a fine-tuned model but rather lies between fine-tuned models. This figure shows loss and error on a two dimensional slice of the loss and error landscapes. We use the zero-shot initialization θ_0 and fine-tune twice (illustrated by the gray arrows), independently, to obtain solutions θ_1 and θ_2. As in Garipov et al. (2018), we obtain an orthonormal basis u_1, u_2 for the plane spanned by these models, and the x and y-axis show movement in parameter space in these directions, respectively.

이 결과는 2개의 finetune된 solution의 weight를 보간하는 것은accuracy를 올릴 수 있다는 것을 알려준다.

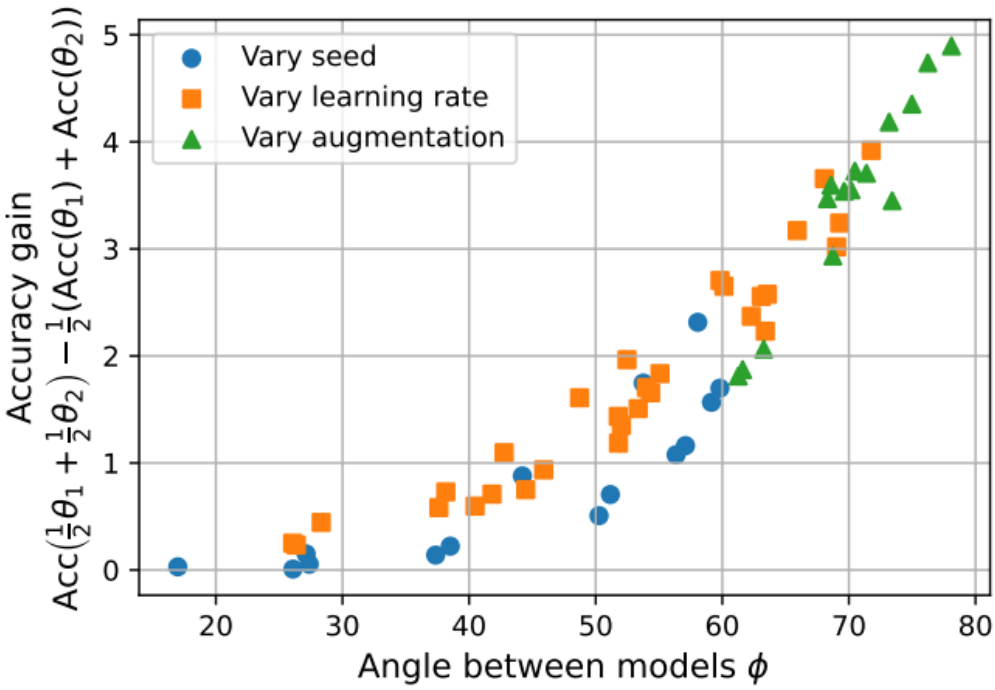

Figure 3: The advantage of averaging solutions (y-axis) is correlated with the angle ϕ between between solutions, while varying hyperparameter configurations between pairs enables a larger ϕ. Each point corresponds to a pair of models θ_1, θ_2 that are fine-tuned independently from a shared initialization θ_0 with different hyperparameter configurations. The angle ϕ between between solutions refers to the angle between $\theta_1 - \theta_0$ and $\theta_2 - \theta_0$ (i.e., the initialization is treated as the origin). Accuracy is averaged over ImageNet and the five distribution shifts described in Section 3.1.

위 그림은 각 ϕ 들에 interpolation advantage가 연관된다는 것을 확인할 수 있다.

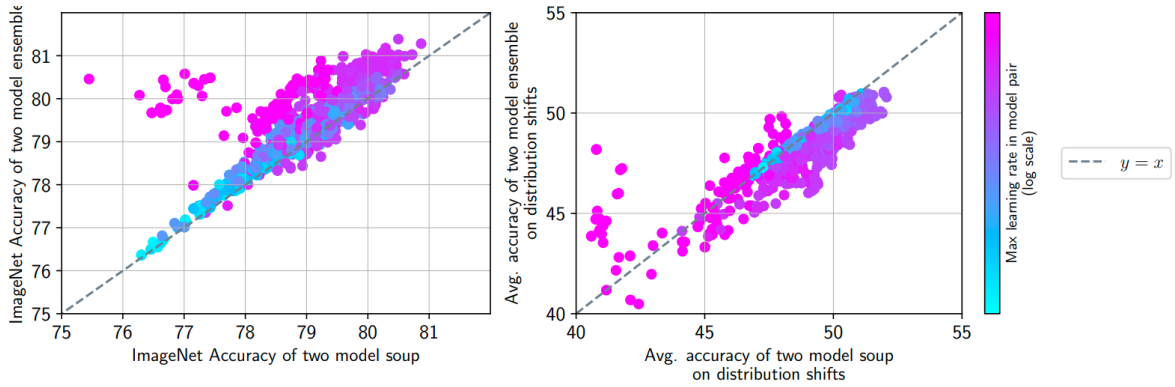

Figure 4: Ensemble performance is correlated with model soup performance. Each point on the scatter plot is a model pair with different hyperparameters. The x-axis is the accuracy when the weights of the two models are averaged (i.e., the two model soup) while the y-axis is the accuracy of the two model ensemble. Ensembles often perform slightly better than soups on ImageNet (left) while the reverse is true on the distribution shifts (right). Each model pair consists of two random greed diamonds from Figure 1.

Ensemble comparison Figure4에서 ensemble comparison가 soup performance와 correlated 되어있는 것을 확인할 수 있다. 둘을 비교하자면 weighting parameter $\alpha \in [0,1]$ 에서 $\theta_\alpha = (1-\alpha)\theta_\alpha + \alpha\theta_1$ 을 weight-averaged soup으로 가정한다. Soup error을 $err_\alpha := \mathbb{E}_{x,y} 1\{arg\ max_i f_i(x;\theta_\alpha) \neq y\}$ 으로 가정하고, 이를 최소화 하는 것이 $min\{err_0, err_1\}$ 이다. err_α 를 최소화하는 곳은 endpoint가 될 것이다. 이 문제에 대한 추가적인 영향력을 보기 위해 본 논문의 저자들은 soup를 logit-level ensemble인 $f_\alpha^{ens}(x) = (1-\alpha)f(x;\theta_0) + \alpha f(x;\theta_1)$ 을 가정하고 에러를 계산하였을 때, neural network에서 err_a^{ens} 가 $min\{err_0, err_1\}$ 의 strict below한 값을 가지는 것을 확인할 수 있었다. 따라서 $err_\alpha \approx err_\alpha^{ens}$ 일 때는 soup이 두 endpoint model에서 outperform 한다는 것을 알 수 있다.

Model soups

본 논문의 저자들은 2개의 fine-tune된 모델을 평균하는 것을 염두에 두고, 본 논문의 저자들은 여러 파라미터의 모델을 평균하는데 집중하였다. 이 section에서 여러 모델을 평균하여 사용하는 것이 여러 모델에서 하나의 모델만을 택하는 것의 대안으로 쓰일 수 있음을 확인할 수 있다.

	ImageNet	Dist. shifts
Best individual model	80.38	47.83
Second best model	79.89	43.87
Uniform soup	79.97	51.45
Greedy soup	81.03	50.75
Greedy soup (random order)	80.79 (0.05)	51.30 (0.16)
Learned soup	80.89	51.07
Learned soup (by layer)	81.37	50.87
Ensemble	81.19	50.77
Greedy ensemble	81.90	49.44