Let It Go: On the Robustness of Training with Frozen Layers

July 2019

1 Introduction skeleton

- Present the question of generalization vs. over parameterization
- Recent works on the role of initialization in generalization
- Singer - split layers into critical and robust
- Experiments with freezing
- Combining all together: What we are doing.

[RT18] has a good introduction

2 Related work skeleton

- works on generalization vs. parameters
- lottery ticket and initialization
- Singer? or maybe last? or before lottery..?
- If we do it - transfer learning and lottery ticket
- Freezing

3 Related Work

Tackling the question of generalizing well while having a large number of parameters, [LFLY18] found that using a small number of parameters (the *intrinsic dimension*) projected into a larger space using a random matrix can lead to good generalization. [JT18] have studied properties of Gradient Descent algorithm that contribute to generalization.

Recent works have also demonstrated that a well-initiated subset of a network can yield good performance: [FC18] used pruning techniques to uncover a *"winning ticket"*: a subset of weights whose initialization allow them to train effectively, and even achieve better performance when trained separately. [FDRC19] have shown that the winning ticket is more stable if the pruning is done at an early stage of training instead on the initial weights. Building upon them, [MYPT19] have successfully used the same winning ticket for multiple image datasets, and using different optimizers.

As mentioned, [ZBS19] have studied the role of different layers. By re-initialization and measuring the change in performance, they identified *critical* and *robust* layers (*ambient* in later versions). Critical layers are very sensitive to re-initialization, while resetting robust layers is negligible.

Other studies have experimented with "freezing" weights: fixing a subset of the weights, and training the rest of the network normally. [HHS18] have shown that using a fixed Hadamard matrix as the last layer do not decrease performance. Later, [RT18] fixed the majority of network parameters, while still preserving high accuracy.

References

[FC18] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[FDRC19] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. The lottery ticket hypothesis at scale. *arXiv preprint arXiv:1903.01611*, 2019.

[HHS18] Elad Hoffer, Itay Hubara, and Daniel Soudry. Fix your classifier: the marginal value of training the last weight layer. *arXiv preprint arXiv:1801.04540*, 2018.

[JT18] Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *CoRR*, abs/1810.02032, 2018.

[LFLY18] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

[MYPT19] Ari S Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. *arXiv preprint arXiv:1906.02773*, 2019.

[RT18] Amir Rosenfeld and John K Tsotsos. Intriguing properties of randomly weighted networks: Generalizing while learning next to nothing. *arXiv preprint arXiv:1802.00844*, 2018.

[ZBS19] Chiyuan Zhang, Samy Bengio, and Yoram Singer. Are all layers created equal? *arXiv preprint arXiv:1902.01996*, 2019.