Meta-Learning and Meta-Loss

Samuel Do,[1] Benjamin Martinez[1]

[1]Computer Science, Stanford University

Stanford Computer Science

Summary

Deep Learning training performance is largely dependent on initial parameter choices. Meta-Learning has been a growing field of study to learn weight initialization distributions in order to have faster model convergence and performance. In this study we analyze the state-of-the-art Discovering Initializers with Model Agnostic Meta-Learning (DIMAML) method, through which we investigate how the shape of loss curves can be learned. We explore different approaches to DIMAML's objective function aiming to optimize the training procedure.

Future Work

DIMAML: Supplementary future work may include verifying results on more architectures and different tasks to see if weights are truly transferable. Seeing where this fails could give theoretical insight into meta-learning algorithms. Another focus is to implement other probability distributions that DIMAML samples from and learns.

Shaping Loss: We see potential use cases in regularization and increasing training effectiveness. Shaping error curves can ensure that we stay above zero training loss. Additionally, combining error shape fitting curves can allow us to allocate higher loss to timesteps where the model has trouble decreasing error.

Data & Models

We perform our experiments on a image classification task with two standard color image datasets, Tiny Imagenet and Celeba, resized to 64x64 resolution. We had a split of 25,000 training examples, 2,500 validation examples, and 2,500 test examples. We used a simple autoencoder architecture following DIMAML's original paper.

References

[1] Antreas Antoniou, Harrison Edwards, and Amos Storkey. "How to train your MAML". In: *International Conference on Learning Representations*. 2019.

[2] Dmitry Baranchuk and Artem Babenko. "Discovering Weight Initializers with Meta Learning". In: *8th ICML Workshop on Automated Machine Learning (AutoML)*. 2021.

[3] Yann N. Dauphin and Samuel S. Schoenholz. "MetaInit: Initializing Learning by Learning to Initialize". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *CoRR* abs/1703.03400 (2017).

[5] Ya Le and Xuan S. Yang. "Tiny ImageNet Visual Recognition Challenge". In: 2015.

[6] Ziwei Liu et al. *Deep Learning Face Attributes in the Wild*. 2015.

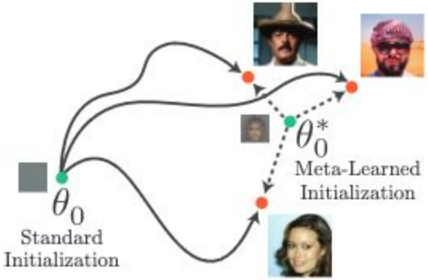
[7] Matthew Tancik et al. "Learned Initializations for Optimizing Coordinate-Based Neural Representations". In: *CoRR* abs/2012.02189 (2020).

Meta-Learning: DIMAML

Instead of optimizing model weights, DIMAML learns the probability distribution from which the weights are generated.

1. For each DIMAML update, sample weights from a probability distribution parameterized by ψ.
2. Train these weights for some task objective (in our case image classification).
3. Backprop and update meta-parameters ψ.

The learned weight initialization distribution can then be transferred to new instances of the same task for faster training convergence.



Case-Study: Shaping Loss

Motivation - DIMAML Objective Function

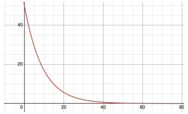
$$\mathcal{L}_{\text{auc}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{\text{i}}(x, y).$$

Where each L_i is the loss of training step i. Optimizes area under error curve, but descriptive rather than prescriptive!

Inverse Exponential Curve

$$f_{a,s,z}(t) = a \cdot e^{-2e\cdot\frac{1}{s}\cdot t} + z$$

Where a is the intercept, s is the spread, and z is the minimum value.



Attempting to shape error into this curve is prescriptive, and it matches error curves for iterative solvers.

Inverse Exponential Objective Function

$$\mathcal{L}_{s,z} = \mathcal{L}_1(x,y) + \sum_{i=1}^{N}(\mathcal{L}_i(x,y) - f_{\max_{j\in[1,N]}\mathcal{L}_j, s, z}(i))$$

Loss is measured as the difference in the shape of curves.

Results

New objective function underperforms original, as expected with manually selecting parameters s and z. However, this shows a technique of *error shape fitting.*