Negative Learning Rates Learn Universal Features

Tom Starshak [1]

Abstract

Recently, meta-learning (learning to learn) methods have received a lot of attention due to their success in few-shot learning. One branch of meta-learning is optimization based meta-learning, where the ability to quickly adapt to new tasks is explicitly optimized for. For example, Model Agnostic Meta-Learning (MAML) consists of two optimization loops: the outer loop learns a meta-initialization of model parameters that is shared across tasks, and the inner loop performs an "adaptation" step on the specific task it is working on. A variant of MAML, Meta-SGD, uses the same two loop structure, but also learns the learning-rate for the adaptation step.

The success of these methods has lead to research into the mechanism by which they adapt to new tasks. The original MAML formulation has been shown to work by feature reuse, that is the meta-initialization already contains high quality features that are not changed significantly during adaptation. One variant of MAML (Body Only Update in Inner Loop) forces the model to abandon feature reuse and rapidly change features for every task. However, little attention has been paid to how the learned learning-rate of Meta-SGD affects feature reuse.

In this paper, we study the effect that a learned learning-rate has on the per-task feature representations in Meta-SGD. The learned learning-rate of Meta-SGD often contains negative values. We hypothesize that during the adaptation phase, these negative learning rates push features away from task-specific features and towards task-agnostic features.

We performed several experiments on the Mini-Imagenet dataset. Two neural networks were trained, one with MAML, and one with Meta-SGD. The feature quality for both models was tested as follows: strip away the linear classification layer, pass labeled and unlabeled samples through this encoder, classify the unlabeled samples according to their nearest neighbor. This process was performed at three times: 1) after fully training and using the meta-initialization parameters; 2) after adaptation on a task, and validated on that task; and 3) after adaptation on a task, and validated on a different task. The MAML trained model improved on the task it was adapted to, but had worse performance on other tasks. The Meta-SGD trained model was the opposite; it had worse performance on the task it was adapted to, but improved on other tasks.This confirms the hypothesis that Meta-SGD's negative learning rates cause the model to learn task-agnostic features rather than simple adapt to task specific features.

[1]Stanford University. Correspondence to: Tom Starshak <starshak@stanford.edu>.

1. Introduction

One of the most important problems in machine learning is few-shot learning; tasks for which there is very limited amounts of training data. This is important since training data is often time consuming and expensive to gather, and in some applications the domain of interest changes over time. Efficiently being able to train a model on a small amount of training data helps with both issues. A lot of research has been done on few-shot learning. The dominant paradigms are transfer learning (Huh et al., 2016); where a model is trained on a much larger, and related, dataset (Deng et al., 2009), then fine-tuned on the relevant data; and meta-learning (Vinyals et al., 2017; Li et al., 2017; Finn et al., 2017; Ravi & Larochelle, 2016), or "learning to learn", where a model is explicitly trained to adapt to few-shot tasks quickly.

Meta-learning approaches looks at a family of tasks that share some structure. These tasks are broken into training and evaluation sets. The meta-learning algorithm will then attempt to learn a way in which it can quickly learn adapt to new tasks. Meta-learning approaches tend to fall into one of three categories: black-box algorithms, optimization based algorithms, and non-parametric approaches.

In optimization based approaches (Finn et al., 2017; Li et al., 2017; Nichol et al., 2018), the loss function explicitly takes into account the multi-task aspect of the problem and the model optimizes the ability to quickly adapt to new tasks. A very successful examples of this approach *Model Agnostic Meta-Learning (MAML)*. (Finn et al., 2017) Broadly, MAML consists of two optimization loops. The outer loop iterates over different tasks and learns a meta-initialization such that the inner loop can quickly adapt to new tasks. The inner loop iterates over data points from the same task and optimizes the relevant loss function. A MAML variant, Meta-SGD (Li et al., 2017), uses the same approach but also learns the inner loop learning-rate.

Due to the influence of MAML, a topic of research that has been explored is what exactly is happening in the underlying neural network (Raghu et al., 2020; Goldblum et al., 2020). This is often framed as a question of *rapid learning*, where the feature representations change dramatically between different tasks, or *feature reuse*, where the meta-initialization that is learned is already very good and the feature representations are changed only slightly. Vanilla MAML has been shown to work via feature reuse, while some variants (Oh et al., 2021) force the underlying model to rapidly adapt. Other research (Bernacchia, 2021) has shown that different learning rates can have dramatic effects on MAML's performance. Surprisingly, little research has been done on how Meta-SGD's learned learning-rates affect the feature representations of the underlying model. In this paper we explore this question. Our main contributions are:

- Exploration of the distribution of per-parameter learning rates that result from training with Meta-SGD, finding that most of the learning-rate parameters become negative.
- Categorizing the effectiveness of the meta-initialization features for both MAML and Meta-SGD, confirming previous work that shows feature reuse is dominant.
- Investigate the changes that the feature representations undergo during task-specific adaptation. We find that adaptation for both MAML and Meta-SGD cause slight decrease in performance for that task. However MAML decreases performance on tasks it wasn't adapted to, while Meta-SGD increases performance on tasks it wasn't adapted to.
- Discuss why negative learning rates allow Meta-SGD to learn task-agnostic features.

2. Related Work

This section represents a brief overview of necessary background in the used meta-learning algorithms and feature representation studies.

2.1. Meta-Learning Algorithms

2.1.1. Model-Agnostic Meta-Learning

Model-Agnostic Meta-Learning (Finn et al., 2017)is a conceptually simple algorithm that holds great power and can be used with any differentiable model, usually a neural network. Consider a model f_θ that maps observations $\mathbf{x}$ to observations $\mathbf{a}$. We will consider different tasks $\mathcal{T}$, where the observations are drawn from different distributions. We denote the loss function as $\mathcal{L}$. When adapting to a new task, the parameters of the model θ become θ_i' using a gradient update:

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \tag{1}$$

Where α is the inner learning rate. This update is only for one task. Since we want to optimize the ability to learn across tasks, several tasks are sampled, the updates are performed, and a meta-update is performed across the sum of losses for all these tasks.

$$\theta = \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i} \mathcal{L}(f_{\theta_i'}) \tag{2}$$

This algorithm eventually learns parameters such that once a task is sampled a gradient step for that task will result in good performance.

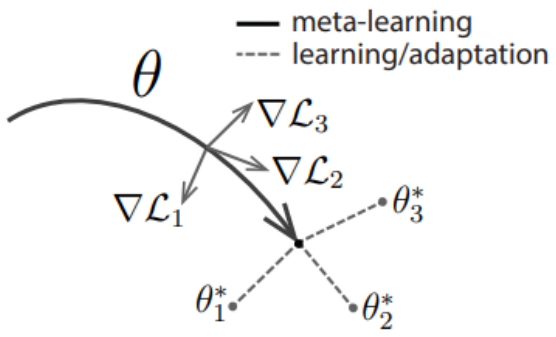

Figure 1. Diagram of the MAML algorithm, showing adaptations to new tasks.

2.1.2. META-SGD

Meta-SGD (Li et al., 2017) is extremely similar to MAML in that it was an outer-loop to learn a meta-initialization and an inner-loop to adapt to different tasks. The difference between the two algorithms is that Meta-SGD learns per-parameter learing rates for tha adaptation step. The inner loop update is nearly the same:

$$\theta_i' = \theta - \alpha \circ \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \tag{3}$$

and where the outer-loop must also update the learning rate.

$$\theta = \theta - \beta \nabla_{\theta,\alpha} \sum_{\mathcal{T}_i} \mathcal{L}(f_{\theta_i'}) \tag{4}$$

This gives Meta-SGD a lot more flexibility in that the step-size and step direction are adapted; the model is not constrained to step in the same direction as the gradient.

2.1.3. PROTOTYPICAL NETWORKS

Prototypical networks (proto-nets) (Snell et al., 2017) differ from the previous two algorithms in that while they update model parameters with optimization schemes, while proto-nets are non-parametric. In a proto-net, an embedding is learned such that embeddings of the same classes are close to each other. The prototype of each class embedding is the mean vector of all the training points of that class.

$$c_k = \frac{1}{|S_k|} \sum_{S_k} f_\phi(x_i) \tag{5}$$

Where S_k is the set of all examples of the k-th class and f_ϕ is the embedding function.

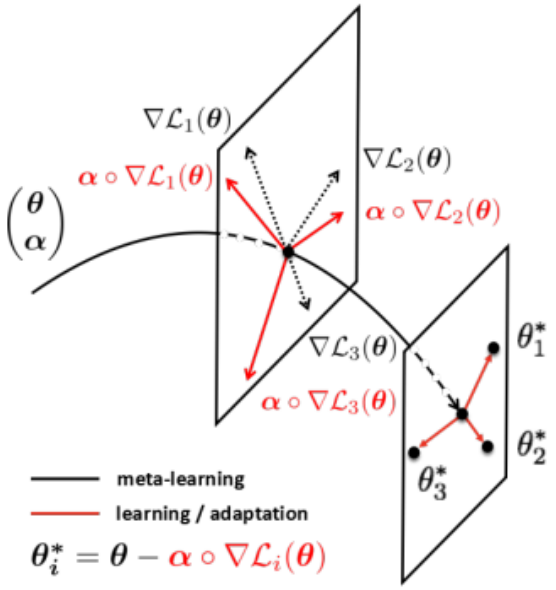

Figure 2. Diagram of the Meta-SGD algorithm. Meta-initialization of the parameters and learning rate allow flexible adaptation.

During test time, examples are passed through the embedding function and examples are simply classified according to the nearest prototype.

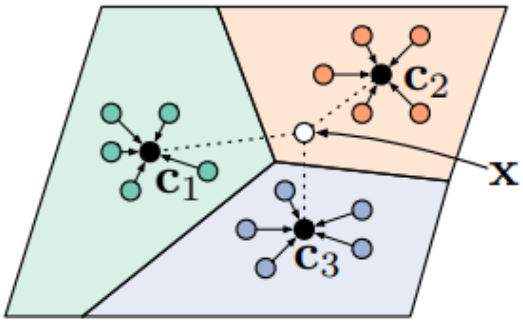

Figure 3. Few-shot prototypical network. The example X would be classified as class c_2.

2.2. Meta-Learning Feature Representations

There has been a fair amount of research into what sorts of features are learned with MAML-like algorithms. In one study (Goldblum et al., 2020) the authors conjecture that the success of meta-learning algorithms suggests that the features that they learn are fundamentally different than classically fine-tuned feature. They show that the feature representations learned through meta-learning are much

more tightly clustered than traditional approaches and more easily separable. Another study (Raghu et al., 2020) considered only meta-learning representations, but looked at how the representations changed from before the adaptation step to after it. They showed that adaptation does not change the representation very much; nearly all the changes in the inner loop occur in the classification head. In fact, they propose an algorithm (Almost No Inner Loop/ANIL) that freezes most of the neural network during adaptation, yet still has nearly the same performance on the MiniImageNet benchmark as regular MAML.

3. MAML, Meta-SGD, and Universal Features

Our goal is to understand how the difference between MAML and Meta-SGD, specifically, the negative learning rates that Meta-SGD learns, affect the feature representations of the underlying model. As shown in 1 and 2 the parameters of the model move from a meta-initialization to a new space. In MAML, this is easy to understand. The inner-loop moves down gradient from the meta-initialization because it has a static learning rate. The model then performs better on the adaptation task. What happens in Meta-SGD is less clear. Often, large parts of the learning rate vector are negative. During adaptation some parameters are moved away from what would improve the model's performance on the adaptation task. That negative learning rates are still learned implies that this is beneficial. We conjecture that the purpose of these negative learning rates is to improve features for other tasks than the current one. That is, negative learning rates cause the model to learn universal features.

4. Experiments

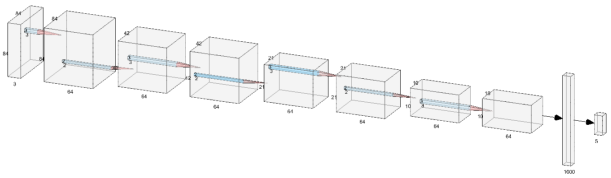

Figure 4. The convolutional neural network that was trained with both MAML and Meta-SGD.

All experiments require two models, both using the same architecture as shown in 4, be trained for image classification on Mini-ImageNet (Vinyals et al., 2017). The model has four convolutional blocks which each consist of: 3x3 convolution, BatchNorm, ReLU activation, and MaxPool. After the four convolutional blocks, the representation is flattened and fed into a linear layer. The loss function is the cross-entropy loss.

Algorithm 1 Meta-Learning algorithms. If α is learnable this is Meta-SGD, otherwise it is MAML

Require: $p(\mathcal{T})$: distribution over tasks
Require: $(\alpha), \beta$: step size parameters
Initialize θ
while not done **do**
  Sample $\mathcal{T}_i$ $p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
  end for
  $\theta \leftarrow \theta - \beta \nabla_{\theta,(\alpha)} \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
end while

One model each was trained with MAML and Meta-SGD as shown in 1. The models were trained as 5-way, 1-shot models. The inner-loop sampled 3 tasks per update and 5 gradient steps per task. For MAML, the inner-loop learning rate was 0.01. In the outer-loop, 10 samples per class were used to calculate the meta-gradient. Both models were trained for 60,000 iterations on an NVIDIA GTX 1070 video card.

After training the meta-initializations were no longer updated. To examine feature reuse, embeddings were created three times: before the inner-loop adaptation, after inner-loop adaptation, and for the task the model was adapted to, and after adaptation for a different task. These will be refered to as "pre-adaptation", "on-task", and "off-task."

We want to concentrate on the feature embeddings that are created by a model and not the classification of those embeddings. To create an embedding, the linear layer was removed from the neural network and an example image fed through the convolutional layers. This results in a 1600-dimensional vector. One example for each class in a task was used as the centroid of a protonet. For inference, an evaluation image is simply classified as the class with the most similar embedding when evaluated by cosine similarity. A single iteration consisted of: a pre-adaptation evaluation, sampling a task, adapting the model to that task with 5 gradient steps and the appropriate inner-loop learning rate, on-task evaluation, sampling a different task, and an off-task evaluation. The models were evaluated for 40,000 iterations.

5. Results

5.1. Base Training

The MAML and Meta-SGD base models finished training with validation accuracies of approximately 46% and 47% respectively. These are both a bit below what was achieved in the MAML and Meta-SGD studies (48.7% and 50.5%)

likely owing to the different meta-batch sizes used.

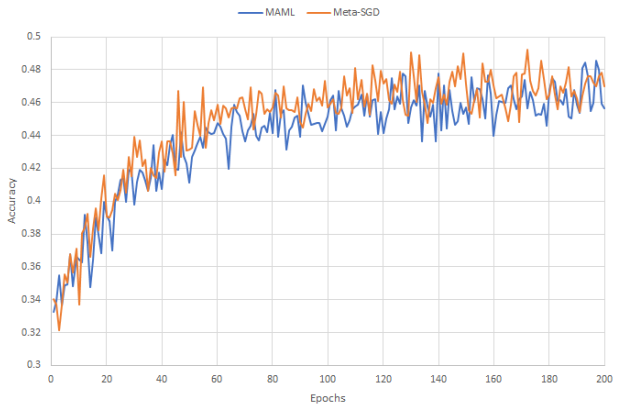

Figure 5. 5-way, 1-shot training accuracies on Mini-ImageNet

5.2. Meta-SGD Learning-Rates

The inner-loop learning rate after 60,000 iterations can be seen in 6 and the distribution statistics for each layer is shown in 1. It is interesting to note that the mean learning rate for every convolutional layer is negative and only the mean learning rate for the linear output layer is positive.

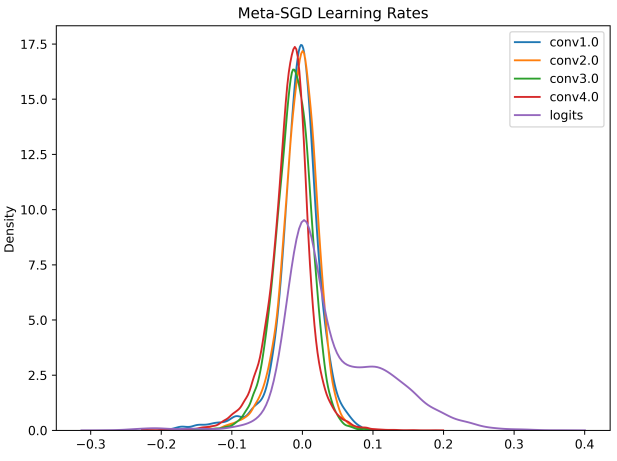

Figure 6. The learned inner-loop learning rates for the Meta-SGD model.

5.3. Feature Adaptation

Numerical results from 40,000 iterations of pre-adaptation, on-task, and off-task evaluation can bee seen in 7.

There are a variety of very interesting things to note here. First of all, both models achieve accuracy that is nearly as good (Meta-SGD) or slightly better (MAML) than they

Table 1. Final inner-loop learning rates for meta-SGD.

LAYER	MEAN	STD. DEV.
CONV1	-0.006	0.032
CONV2	-0.004	0.027
CONV3	-0.013	0.027
CONV4	-0.018	0.030
LOGITS	0.043	0.072

achieved using the regular evaluation procedure (Note that the accuracy reported in 7 is averaged over 40,000 iterations while the accuracy reported in 5 were only averaged over 40 iterations). That is, the features that are learned in the meta-initialization, *without a classification layer or adaptation* retain almost all predictive performance. This is more evidence for the conclusions reached in (Raghu et al., 2020), that the function of MAML is feature re-use and not rapid learning.

Again, we can also see that Meta-SGD outperforms MAML in this setting as well. This shows that the extra flexibility, in both step-size and step-direction, that are present in Meta-SGD serve to learn better feature representations than MAML. It is not merely that Meta-SGD allows the model to take more productive steps during adaptation.

A surprising result is that both models perform worse in the on-task regime.This can be explained for Meta-SGD. In 6 that the mean learning-rate for all of the convolutional layers, that is the layers that contribute to creating the embedding, are negative. The adaptation should then push the weights away from what would improve the model for that task. This is not the case for MAML, where there is a single positive weight for all parameters. This indicates that the classification layer is significant for MAML during the adaptation step.

The final and most interesting result is what occurs in the off-task regime. MAML trained models degrade the representations of features that are not present in the current task while Meta-SGD trained models improve the representations of features that are not present in the current task. The Meta-SGD trained model increased in accuracy by 0.2% while the MAML trained model decreased in accuracy by 0.3%. While these results are modest, a t-test comparing the relative changes between on-task and off-task performance shows a p-value = 0.00044, indicating that this is a real phenomenon.

Meta-SGD learning better feature representations in the off-task regime suggests it may learn in a novel way. The negative learning rates push weights away from on-task representations and therefor toward off-task representations rather than how MAML works.

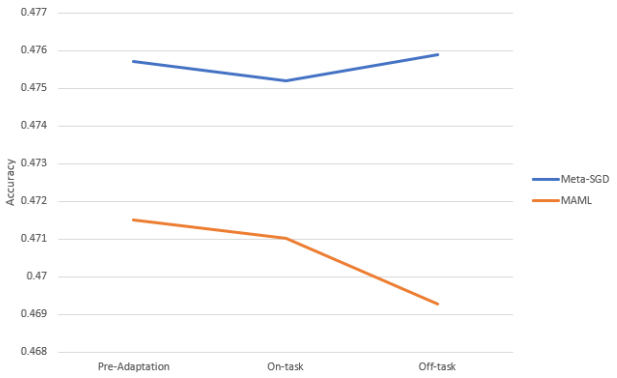

Figure 7. Average accuracy for proto-net based evaluation on embedding vectors for both MAML and Meta-SGD neural networks. Pre-adaptation is using the meta-initialization. On-task is evaluated on the same task that the network was adapted to. Off-task is evaluated on a different task than the model was adapted to.

6. Future Work

We have shown that when trained on Mini-ImageNet, the Meta-SGD algorithm will learn negative inner-loop learning rates and that these learning rates push parameters away from on-task feature representation and towards off-task (universal) feature representations.

Several potential areas of inquiry were not explored due to time constraints:

- Does this result hold across different datasets?
- How does the number of inner-loop gradient updates affect the learned inner-loop learning rate in Meta-SGD? How does it affect the representation change?
- Meta-SGD improving performance on off-task representation suggests there may be a better way to evaluate an example at test time, by either not adapting or by adapting to a different task (or tasks) than those that are of interest.
- What mechanism causes the representation learned by MAML to perform worse in the on-task regime?

7. Contributions

Tom Starshak was the sole contributor for this report. Base code for the MAML implementation was taken and adapted from: here.

References

Bernacchia, A. Meta-learning with negative learning rates, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks, 2017.

Goldblum, M., Reich, S., Fowl, L., Ni, R., Cherepanova, V., and Goldstein, T. Unraveling meta-learning: Understanding feature representations for few-shot tasks, 2020.

Huh, M., Agrawal, P., and Efros, A. A. What makes imagenet good for transfer learning?, 2016.

Li, Z., Zhou, F., Chen, F., and Li, H. Meta-sgd: Learning to learn quickly for few-shot learning, 2017.

Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms, 2018.

Oh, J., Yoo, H., Kim, C., and Yun, S.-Y. Boil: Towards representation change for few-shot learning, 2021.

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. Rapid learning or feature reuse? towards understanding the effectiveness of maml, 2020.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning., 2016.

Snell, J., Swersky, K., and Zemel, R. S. Prototypical networks for few-shot learning, 2017.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. Matching networks for one shot learning, 2017.