A Meta-Learning Approach for Custom Model Training

Amir Erfan Eshratifar,[1] Mohammad Saeed Abrishami,[1] David Eigen,[2] Massoud Pedram,[1]
[1]Department of Electrical Engineering at University of Southern California, Los Angeles, CA 90089, USA
[2]Clarifai, San Francisco, CA 94105, USA
eshratif@usc.edu, abri442@usc.edu, deigen@clarifai.com, pedram@usc.edu

Abstract

Transfer-learning and meta-learning are two effective methods to apply knowledge learned from large data sources to new tasks. In few-class, few-shot target task settings (i.e. when there are only a few classes and training examples available in the target task), meta-learning approaches that optimize for future task learning have outperformed the typical transfer approach of initializing model weights from a pre-trained starting point. But as we experimentally show, meta-learning algorithms that work well in the few-class setting do not generalize well in many-shot and many-class cases. In this paper, we propose a joint training approach that combines both transfer-learning and meta-learning. Benefiting from the advantages of each, our method obtains improved generalization performance on unseen target tasks in both few- and many-class and few- and many-shot scenarios.

Introduction

Current deep learning algorithms require a very large amount of data to learn decent task-specific models, and acquiring enough labeled data is often expensive and laborious. Moreover, in many mission critical applications, such as autonomous vehicles and drones, an agent needs to adapt rapidly to unseen environments. Humans are able to learn new skills and concepts rapidly by leveraging knowledge learned earlier; therefore, we aim to enable the artificial agents to do the same. Transfer learning transfers the knowledge obtained from one domain with a large amount of labeled data to other domains with less labeled data (**?**). It achieves this by copying the initial feature extraction layers, and fine-tuning the resulting model on the target task (**?**). However, this method is still data hungry because gradient-based optimization algorithms need many iterations over numerous examples to adapt the models for new tasks (**?**). On the other hand, meta-learning is a class of machine learning algorithms concerned with the ability of learning process itself. Introduced by (**?**), meta-learning aims to train the model in task space rather than instance space. While transfer learning methods train a base model to use as a transfer source by optimizing a single monolithic task, meta-learning algorithms learn their base models by sampling many different smaller tasks from a large data source. As a result, one might expect that the meta-learned model is capable of generalizing well to new unseen tasks because of task-agnostic way of training.

Shortcomings of meta-learning algorithms. As we show in the experiments below, models trained using meta-learning perform worse than transfer learning in the following two scenarios: **1.** When there are many training examples available for each class in the target task (here we would like the artificial agent to continue improving its model performance as more data becomes available); and **2.** When there are many different classes in the target task.

The main contribution of this paper is a joint "meta-transfer" learning method that performs well for target tasks of both few and many shots/classes. Our method performs better than both transfer- and meta-learning baselines on all target task sizes we evaluate.

Meta-Transfer Learning (MTL)

In order to overcome the two issues mentioned earlier, we propose a new training algorithm, which inherits advantages of both meta-learning and transfer learning. This joint training method employs two loss functions: 1) task-specific (transfer learning) 2) task-agnostic (meta-learning). The task-specific loss, $L_{(x,y)}(\theta)$, is defined over the entire base model's training dataset. The task-agnostic loss, $L_{\tau}(\theta)$, on the other hand, is a meta-learning loss defined over a distribution of tasks (e.g. 5-ways classification tasks). Two gradient updates are computed independently from these two loss functions, and the model is updated using the weighted average of these two update vectors (see Algorithm 1). The tasks in meta-learning are sampled from a distribution $p(\tau)$, while all instances in the sampled tasks are used for the task-specific optimization. For adaptation to a new unseen task, regular stochastic gradient descent will be used. For the meta-learner, we evaluate our method using both Model Agnostic Meta-learning (MAML (**?**)) and its first order variant, Reptile (**?**). The reason that we use this class of meta-learning algorithms is that as opposed to Matching Networks (**?**) and its variant (**?**), they are model agnostic, and can be directly applied to any model which is trained with a gradient descent procedure. The proposed method is similar to Gradient Agreement (**?**) in a sense that it pushes the

Table 1: Accuracy results on miniImageNet dataset.

Task		Transfer Learning	Prototypical	Reptile	Reptile MTL	MAML	MAML MTL
5-ways	1 Shot	37.44	49.42	49.16	**51.04**	48.70	**50.99**
	5 Shots	53.28	68.20	65.99	**69.58**	63.11	**67.88**
	100 Shots	90.23	61.13	83.75	**96.56**	82.53	**92.44**
20-ways	1 Shot	15.06	21.54	20.29	**22.27**	20.50	**22.34**
	5 Shots	27.33	34.68	31.46	**36.45**	31.50	**35.95**
	100 Shots	73.56	56.76	68.59	**74.00**	68.55	**74.87**
35-ways	1 Shot	10.49	10.67	9.85	**13.60**	9.61	**14.11**
	5 Shots	20.04	17.53	16.46	**21.59**	16.01	**21.85**
	100 Shots	61.72	38.13	51.09	**68.10**	50.21	**66.34**

model parameters in a direction that the distribution of tasks have agreement with the single specific task of training over the whole classes.

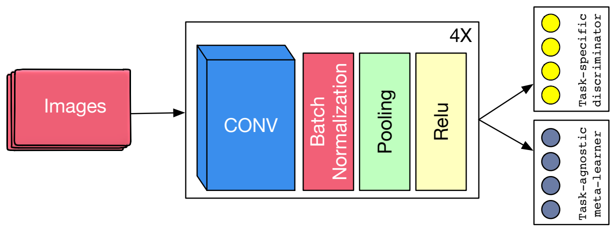

Figure 1: Meta-transfer learning model setup for miniImageNet dataset

Algorithm 1 Meta-Transfer Learning Algorithm

```
1:  Initialize model parameters, $\theta$
2:  for iteration = 1,2,... do
3:      Sample a batch of tasks $\tau_i \sim p(\tau)$
4:      for all $\tau_i$ do
5:          Split the examples of the task into $k$ sub-batches
6:          $\theta_i = \theta - \alpha_{inner} \nabla L_{\tau_i}(f_\theta)$ for $k$ steps
7:      $\theta_{metalearner} = \theta - \alpha_{outer} \sum_i \nabla L_{\tau_i}(f_{\theta_i})$ (MAML)
8:      $\theta_{metalearner} = \theta + \alpha_{outer} \sum_i (\theta_i - \theta)$ (Reptile)
9:      $\theta_{discriminator} = \theta - \alpha_d \nabla L_{(x,y)}$
10:     $\theta = \beta\, \theta_{metalearner} + (1 - \beta)\, \theta_{discriminator}$
11: return $\theta$
```

Performance Evaluations

The proposed model is evaluated on *miniImageNet* (**?**) dataset, split into 64 training classes and 36 test classes as unseen tasks. The architecture of the model is shown in Figure 1 and the results are demonstrated in Table 1. The base model for transfer learning is trained on all 64 training classes.

Note that for many-classes (35-ways) tasks, the transfer learning baseline outperforms previous meta-learning algorithms, while in few-classes problems, the result is reversed: meta-learning beats transfer learning. Our proposed method, MTL, outperforms both these algorithms in all scenarios by improving the weaknesses of few-shot learning algorithms in generalizing to many-shot and many-classes problems.

Conclusion and Future Work

A single model that is adaptable to unseen tasks is a crucial component in artificial intelligence. In this work, we presented a method to extend the capability of few-shot learning algorithms to many-shot and many-classes learning problems, by integrating them with transfer learning model. The next step is to use this approach on a larger dataset and deeper model, to see whether meta-learning is still outperforming transfer learning or not.

Modi officiis dignissimos eos sint minima ipsum at, voluptates vero laboriosam quidem amet incidunt, nam beatae animi fuga temporibus, nisi voluptatibus explicabo excepturi asperiores at distinctio fuga quisquam.Sit odit ullam doloribus modi praesentium maiores beatae in illum nihil ea, fugit culpa accusamus hic, sequi veritatis voluptatibus repellendus enim fuga sunt totam dolorum amet, officiis ipsam ipsa assumenda recusandae laborum facere aut?Totam dolor ullam iusto, possimus quos animi reprehenderit veniam similique aliquid temporibus, non explicabo id aspernatur in similique et nobis mollitia qui hic, labore repudiandae consequuntur perferendis aperiam dignissimos dolorum quaerat accusantium, placeat distinctio facere enim quia optio labore.Temporibus nam aspernatur aut numquam quasi eos officiis dolorem, mollitia ea molestias dolores assumenda molestiae tempora veniam aliquam recusandae accusamus perspiciatis, doloribus suscipit itaque dolore ad?Optio ad minus aut, eius consectetur at veniam voluptates itaque possimus provident vel, quasi eligendi dolore vero eaque dolorum magni?Id eveniet nisi ab esse sed placeat fugit quis, quisquam dolorem cum iure porro officiis, ipsa dolore soluta eos id consequuntur, quibusdam commodi ad aspernatur?Sapiente optio hic magni quaerat nihil tempora quod delectus eos placeat, voluptates ex quam aliquam vel deserunt repellat dolores, enim modi doloremque id, amet itaque sint molestias placeat expedita deserunt illo delectus.Ullam dolorum laudantium repudiandae, quo similique beatae maxime dolores?Veniam nulla ratione cum a ea architecto debitis, enim blanditiis excepturi laudantium veritatis temporibus reiciendis explicabo, beatae enim unde.Molestiae facilis reiciendis id veritatis delectus numquam, cum dicta et fugit quidem ullam delectus quibusdam impedit aperiam, voluptatem illum voluptate excepturi corrupti sint cumque?Culpa voluptas mollitia quibusdam accusamus esse aliquam totam minima sapiente libero saepe, ratione dignissimos qui consequatur quisquam