Meta-learning algorithms for Few-Shot Computer Vision

Goktug Cengiz

December 19, 2019

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Introduction

This paper about meta-learning algorithms for Few-Shot Computer Vision.

It contains:

- An extensive review of the state-of-the-art in few-shot computer vision;
- A benchmark of meta-learning algorithms for few-shot image classification;
- The introduction to a novel meta-learning algorithm for few-shot object detection, which is still in development.

Introduction - What is Few-Shot Learning ?

Few-Shot Learning is the challenge of training a model with only a small amount of data.

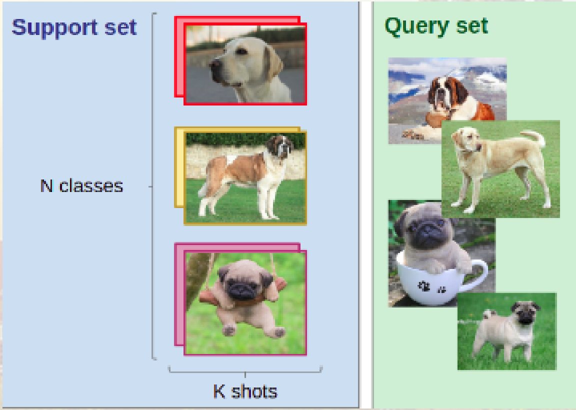
Many solutions to this problem use meta-learning algorithms, i.e.algorithms that learn to learn. By sampling few-shot tasks from a larger dataset, we can teach these algorithms to solve new, unseen tasks.

Introduction - few-shot learning problem

However, in real world applications, it is not always possible to build a dataset with that many images. Sometimes we need to classify images with only one or two examples per class. For this kind of tasks, machine learning algorithms are still far from human performance.

This problem of learning from few examples is called few-shot learning. For a few years now, the few-shot learning problem has drawn a lot of attention in there search community, and a lot of elegant solutions have been developed. An increasing part of them use meta-learning, which can be defined in this case as learning to learn.

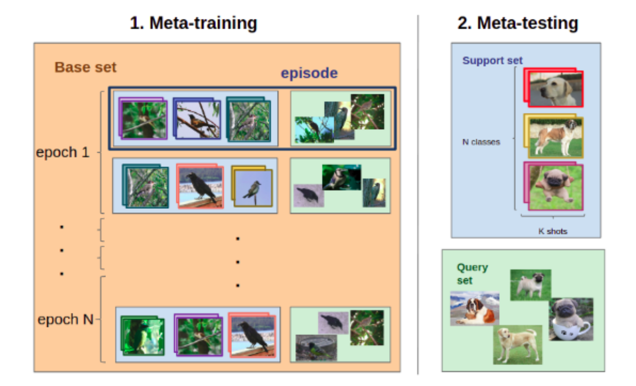
Few-Shot classification problem



A 3-way 2-shot classification problem.

Images from the query set would need to be classified in (Labrador, Saint-Bernard, Pug).

Meta-learning paradigm



Meta-learning algorithms - Gradient-based meta-learning

Meta-LSTM (2016):

Algorithm uses a Long-Short-Term-Memory network. This model learns how to efficiently operate gradient descents on the base-model from the support set, in order to make this base-model more accurate on the query set.

Model-Agnostic Meta-Learning (2017):

Algorithm that tries to learn the best parameters for the CNN's initialization in order to achieve good accuracy on the query set after only a few gradient descents on the support set.

Meta-learning algorithms - Metric Learning

Metric Learning

1. Matching Networks (2016)
2. Prototypical Networks (2017)
3. Relation Network (2018)

As such, metric learning algorithms learn to compare data instances. In the case of few-shot classification, they classify query instances depending on their similarity to support set instances. When dealing with images, most algorithm train a convolutional neural network to output for each image an embedding vector. This embedding is then compared to embeddings of other images to predict a classification.

Overwiew

- Benchmarking several state-of-the-art algorithms
- Identifying the strengths and weaknesses of each algorithm
- Its performance on different kinds of datasets, and overall their relevance depending on the task that needs solving
- To focus on the Model Agnostic Meta-Learner and to switch from the few-shot image classification problem to the few-shot object detection problem
- The idea is to apply MAML to the YOLOv3 object detector in order to obtain an algorithm capable of detecting new classes of objects with little time and only a few examples

Meta-learning algorithms for Few-Shot image classification

4 Meta-Learning Algorithms will be compared. They are:

- Matching Networks
- Prototypical Networks
- Relation Network
- Model Agnostic Meta-Learner

On 2 datasets. They are:

- miniImageNet (Consist of 100 classes, each containing 600 3-channel image)
- CUB (Consist of 6,033 pictures of birds from 200 different classes)

Reproducing the results

	CUB		miniImageNet	
	1 shot	**5 shots**	**1 shot**	**5 shots**
Baseline	1h10	1h00	10h05	10h07
Baseline++	56mn	51mn	10h25	10h28
MatchingNet	6h41	4h21	7h51	6h23
ProtoNet	6h38	5h07	7h40	6h08
MAML	28h05	22h28	31h22	25h22

Table: Running time of several algorithms depending on the setting and dataset. This is the running time of the whole process, from training to evaluation.

Reproducing the results

	our reimplementation		Chenet al.'s	
	1 shot	**5 shots**	**1 shot**	**5 shots**
Baseline	46.57 ± 0.73	$\mathbf{68.36 \pm 0.66}$	47.12 ± 0.74	64.16 ± 0.71
Baseline++	$\mathbf{53.71 \pm 0.82}$	$\mathbf{75.09 \pm 0.62}$	60.53 ± 0.83	79.34 ± 0.61
MatchingNet	$\mathbf{58.43 \pm 0.85}$	$\mathbf{75.52 \pm 0.71}$	61.16 ± 0.89	72.86 ± 0.76
ProtoNet	50.96 ± 0.90	$\mathbf{75.48 \pm 0.69}$	51.31 ± 0.91	70.77 ± 0.69

Table: Comparison of the results of our reimplementation compared to the results reported by Chenet al., on the CUB dataset with a 5-way classification task. Our results are shown in bold when they are out of the 95% confidence interval reported by Chenet al.

Reproducing the results

	CUB		miniImageNet	
	1 shot	**5 shots**	**1 shot**	**5 shots**
Baseline++	61.31 ± 0.92	$\mathbf{77.53 \pm 0.64}$	48.05 ± 0.76	67.01 ± 0.67
MatchingNet	$\mathbf{59.55 \pm 0.89}$	$\mathbf{75.63 \pm 0.72}$	48.43 ± 0.77	$\mathbf{62.26 \pm 0.70}$
ProtoNet	$\mathbf{50.28 \pm 0.90}$	$\mathbf{75.83 \pm 0.67}$	43.89 ± 0.73	$\mathbf{65.55 \pm 0.73}$
MAML	$\mathbf{54.57 \pm 0.99}$	$\mathbf{75.51 \pm 0.73}$	$\mathbf{43.92 \pm 0.77}$	62.96 ± 0.72

Table: Reproduction of the results of on both CUB and miniImageNet, using the implementation provided with the paper. Our results are shown in bold when they are out of the 95% confidence interval

Benchmarking for Algorithms

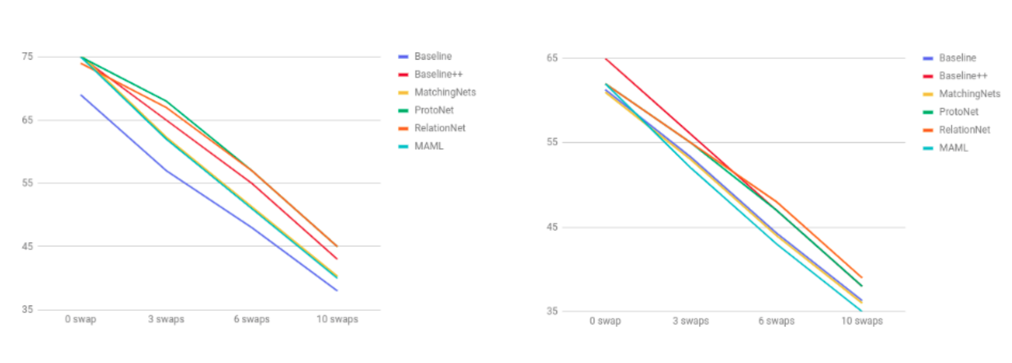


Figure: Accuracy of the methods for different number of label swaps in the support set of each classification task. Left: CUB. Right: miniImageNet.

Benchmarking for Algorithms

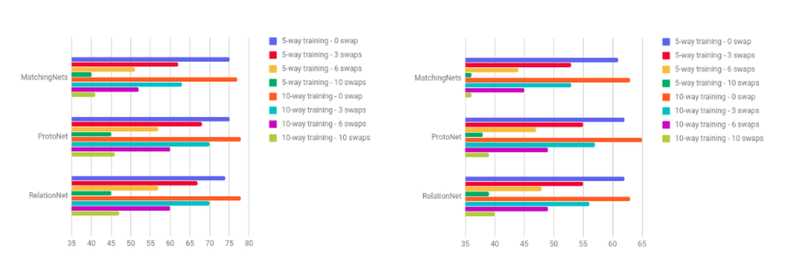


Figure: Accuracy of the methods for different number of label swaps in the support set of each classification task, with a 5-way training and a 10-way training. Left: CUB. Right:miniImageNet

Benchmarking for Algorithms

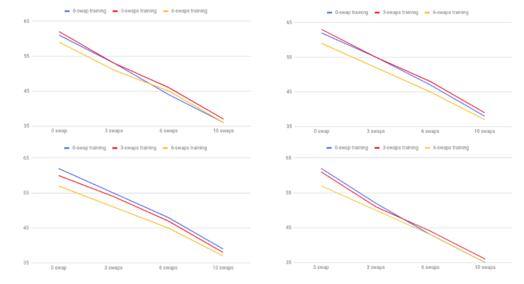


Figure: From left to right, top to bottom: Matching Networks, Prototypical Networks,Relation Network, MAML. For each method, accuracy on a model trained with three strategies,for different number of label swaps in the support set at evaluation time

Future Work

- The mean function could be replaced for instance by a "leaky" median to try to improve performance.
- Comparing the performance of meta-learning algorithms could be interesting depending on the "shape" of the meta-training dataset. Would a dataset with 100 different classes and 500 examples per class allow better performance than a dataset with 50 classes and 1000 examples per class ?

The Few-Shot Object Detection problem

Materials: laptop, mug, notebook

Task: detecting all object belonging in the materials

Figure: Materials Visualization

YOLOAMML

Idea: To solve the few-shot object detection problem, applying the Model-Agnostic Meta-Learning algorithm to the YOLO detector. It is called as YOLOAMML.

- **MAML** can be applied to a wide variety of deep neural networks to solve many few-shot tasks. It could as well be applied to a standard detector to solve few-shot object detection.
- **YOLOv3** revents the advantage of being a single-stage detector. It appeared easier to apply MAML to YOLO than to a variant of R-CNN.

Implementation

- Deep Tiny YOLO initialized
- Training on 3-way 5-shot object detection tasks
- Dataset: COCO
- Optimizer: Adam
- Learning rate: 10ˆ3
- Epoch: 10.000

Results

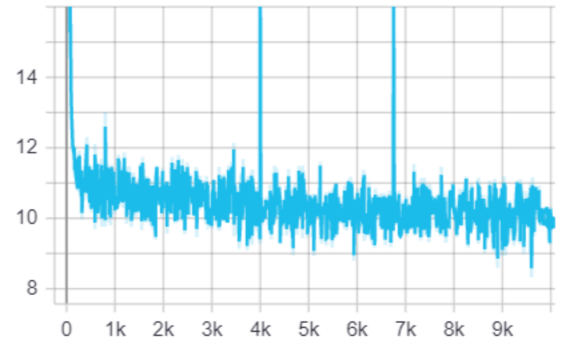


Figure: Total loss per epoch of YOLOMAML. Each point is the average total loss of the model on the query set of the episodes of one epoch.

Results

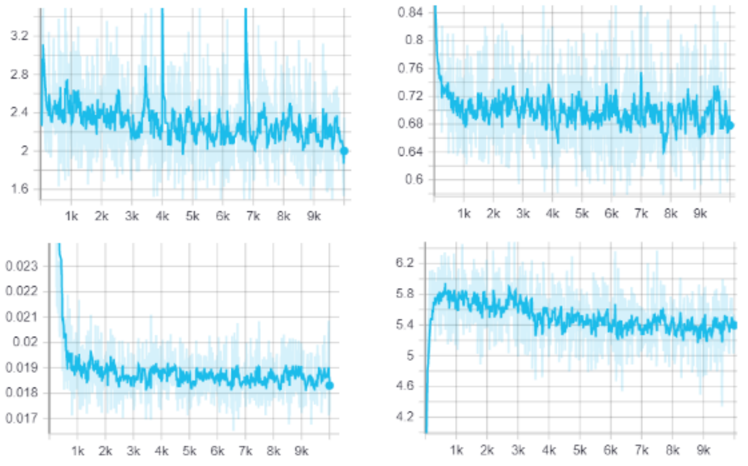


Figure: Evolution of the four parts of the loss of YOLOMAML during the same training

Results

Figure: Object detection by the models YOLOMAML (left column) and YOLO (right column)

Future Works

- The answer should reside in the prediction of the objectness confidence, but it is likely that other issues may rise when this one is solved.
- An other direction of future work would be to constitute a dataset adapted to few-shot detection.

Conclusion

Advanced research in Few-Shot Learning is still new. Until now, only few works have tackled the few-shot object detection problem, for which there is yet no agreed upon benchmark (like mini-ImageNet for few-shot classification).

However, solving this problem would be a very important step in the field of computer vision. Using meta-learning algorithms, we could have the ability to learn to detect new, unseen objects with only a few examples and a few minutes.

Thanks for listening!