BHMAML oraz FHyperMAML

Piotr Borycki   Piotr Kubacki

Uniwersytet Jagielloński

Projekt przy współpracy z GMUM-em
Supervisor: dr hab. Przemysław Spurek

January 17, 2023

Rezultaty pracy - BHMAML

termin trwania projektu	Czerwiec 2022 - do teraz
obecny etap projektu	ostateczna redakcja preprintu i wysyłanie na konferencję
technologia	PyTorch

Krótkie przypomnienie tematyki projektu

- Few Shot Learning
- MAML | HYPERMAML **| BHMAML | FHYPERMAML**

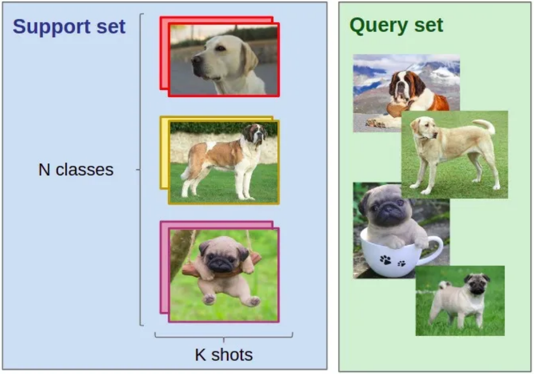

Figure: Few Shot Learning

Krótkie przypomnienie tematyki projektu

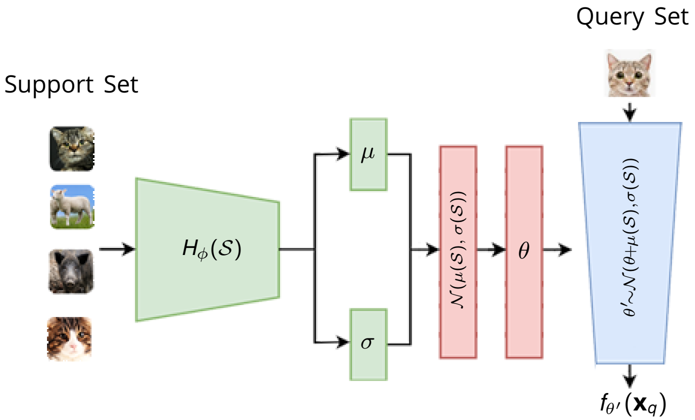

Figure: architektura BHMAML

kluczowe osiągnięcia - BHMAML

- Udało nam się zaimplementować planowany model
- Udało nam się wykonać grid- search na wybranych benchmarkowych datasetach, który wyłonił faktyczne możliwości naszego modelu
- Udało nam się doprowadzić prace nad projektem do spisania preprintu, który planujemy w najbliższym czasie wysłać na konferencję.
  Tytuł pracy: **Hypernetwork approach to Bayesian MAML**

kluczowe osiągnięcia - BHMAML

Hypernetwork approach to Bayesian MAML

P. Borycki, P. Kubacki, M. Przewięźlikowski, T. Kuśmierczyk, J. Tabor, P. Spurek
Faculty of Mathematics and Computer Science,
Jagiellonian University, Kraków, Poland
przemyslaw.spurek@gmail.com

Abstract

The main goal of Few-Shot learning algorithms is to enable learning from small amounts of data. One of the most popular and elegant Few-Shot learning approaches is Model-Agnostic Meta-Learning (MAML). The main idea behind this method is to learn shared universal weights of a meta-model, which then are adapted for specific tasks. However, due to limited data size, the method suffers from over-fitting and poorly quantifies uncertainty. Bayesian approaches could, in principle, alleviate these shortcomings by learning weight distributions in place of point-wise weights. Unfortunately, previous Bayesian modifications of MAML are limited in a way similar to the classic MAML, e.g., task-specific adaptations must share the same structure and can not diverge much from the universal meta-model. Additionally, task-specific distributions are considered as posteriors to the universal distributions working as priors and optimizing them jointly with gradients is hard and poses a risk of getting stuck in local optima.

In this paper, we propose BayesHMAML, a novel generalization of Bayesian MAML, which employs Bayesian principles along with Hypernetworks for MAML. We achieve better convergence than the previous methods by classically learning universal weights. Furthermore, Bayesian treatment of the specific tasks enables uncertainty quantification, and high flexibility of task adaptations is achieved using Hypernetworks instead of gradient-based updates. Consequently, the proposed approach not only improves over the previous methods, both classic and Bayesian MAML in several standard Few-Shot learning benchmarks but also benefits from the properties of the Bayesian framework.

Wyniki powyższego modelu na datasecie crosschar

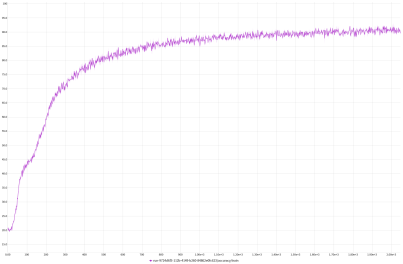

Figure: Accuracy train

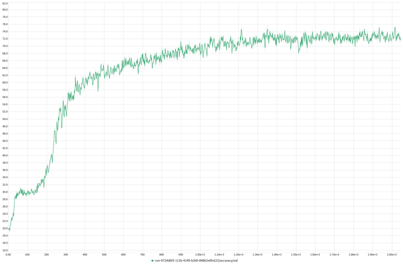

Figure: Accuracy val

Wyniki po pierwszej "poprawce"

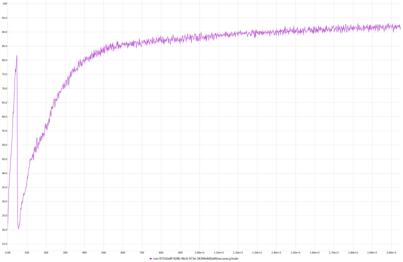

Figure: Accuracy train

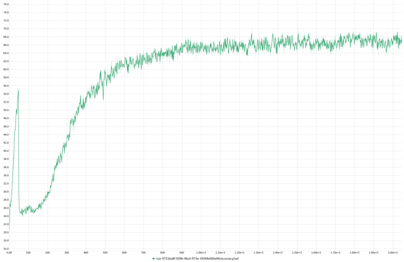

Figure: Accuracy val

Ostateczne wyniki

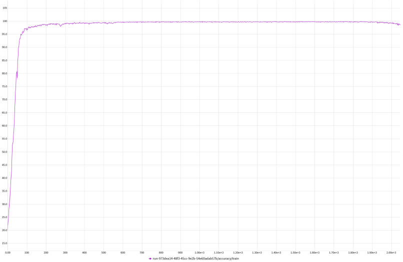

Figure: Accuracy train

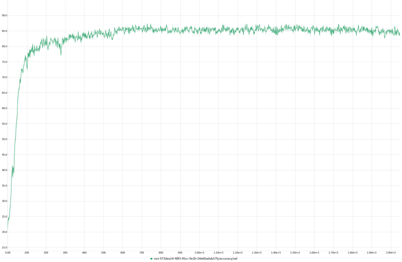

Figure: Accuracy val

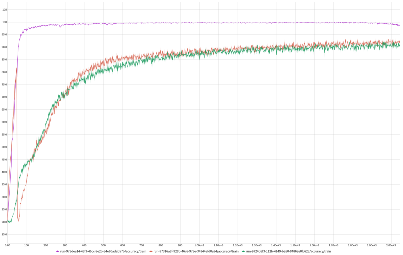

Figure: Accuracy train

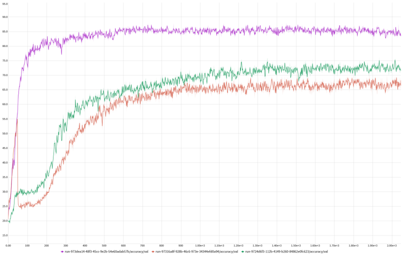

Figure: Accuracy val

kluczowe osiągnięcia - BHMAML

- Udało nam się osiągnąć lepsze wyniki *accuracy* od wyjściowego HyperMAMLA i innych wybranych modeli dla benchmarkowych datasetów

Table 1: The classification accuracy results for the inference tasks on **CUB** and **mini-ImageNet** data sets in the 1-shot and 5-shot settings. The highest results are in bold and the second-highest in italic (the larger, the better).

Method	**CUB** 1-shot	**CUB** 5-shot	**mini-ImageNet** 1-shot	**mini-ImageNet** 5-shot
ML-LSTM [35]	–	–	43.44 ± 0.77	60.60 ± 0.71
LLAMA [14]	–	–	49.40 ± 1.83	
VERSA [13]	–	–	48.53 ± 1.84	67.37 ± 0.86
Amortized VI [13]	–	–	44.13 ± 1.78	55.68 ± 0.91
Meta-Mixture [17]	–	–	49.60 ± 1.50	64.60 ± 0.92
Feature Transfer [51]	46.19 ± 0.64	68.40 ± 0.79	39.51 ± 0.23	60.51 ± 0.55
Baseline++ [7]	61.75 ± 0.95	78.51 ± 0.59	47.15 ± 0.49	66.18 ± 0.18
ProtoNet [42]	52.52 ± 1.90	75.93 ± 0.46	44.19 ± 1.30	64.07 ± 0.65
RelationNet [43]	62.52 ± 0.34	78.22 ± 0.07	48.76 ± 0.17	64.20 ± 0.28
DKT + BNCosSim [28]	62.96 ± 0.62	77.76 ± 0.62	49.73 ± 0.07	64.00 ± 0.09
VAMPIRE [25]	–	–	51.54 ± 0.74	64.31 ± 0.74
ABML [34]	49.57 ± 0.42	68.94 ± 0.16	45.00 ± 0.60	–
FO-MAML [26]	–	–	48.70 ± 1.84	63.11 ± 0.92
Reptile [26]	–	–	49.97 ± 0.32	65.99 ± 0.58
HyperShot [40]	65.27 ± 0.24	79.80 ± 0.16	52.42 ± 0.46	68.78 ± 0.29
HyperShot+ adaptation [40]	66.13 ± 0.26	80.07 ± 0.22	53.18 ± 0.45	69.62 ± 0.2
FEAT [46]	$\mathbf{68.87 \pm 0.22}$	$\mathbf{82.90 \pm 0.15}$	$\mathbf{55.15 \pm 0.20}$	$\mathbf{71.61 \pm 0.16}$
MAML [11]	56.11 ± 0.69	74.84 ± 0.62	45.39 ± 0.49	61.58 ± 0.53
MAML++ [2]	–	–	52.15 ± 0.26	68.32 ± 0.44
iMAML-HF [32]	–	–	49.30 ± 1.88	–
SignMAML [9]	–	–	42.90 ± 1.50	60.70 ± 0.70
Bayesian MAML [48]	55.93 ± 0.71		53.80 ± 1.46	64.23 ± 0.69
Unicorn-MAML [47]	–	–	54.89	–
Meta-SGD [21]	–	–	50.47 ± 1.87	64.03 ± 0.94
PAMELA [31]	–	–	53.50 ± 0.89	$\mathit{70.51 \pm 0.67}$
HyperMAML [29]	66.11 ± 0.28	78.89 ± 0.19	51.84 ± 0.57	66.29 ± 0.43
BayesHMAML	66.57 ± 0.47	79.86 ± 0.31	52.54 ± 0.46	67.39 ± 0.35
BayesHMAML + adaptation	$\mathit{66.92 \pm 0.38}$	$\mathit{80.47 \pm 0.38}$	52.69 ± 0.38	68.24 ± 0.47

kluczowe osiągnięcia - BHMAML

Table 2: The classification accuracy results for the inference tasks on cross-domain tasks (**Omniglot→EMNIST** and **mini-ImageNet→CUB**) data sets in the 1-shot and 5-shot setting. The highest results are bold and second-highest in italic (the larger, the better).

	Omniglot→EMNIST		mini-ImageNet→CUB	
Method	**1-shot**	**5-shot**	**1-shot**	**5-shot**
Feature Transfer [51]	64.22 ± 1.24	86.10 ± 0.84	32.77 ± 0.35	50.34 ± 0.27
Baseline++ [7]	56.84 ± 0.91	80.01 ± 0.92	39.19 ± 0.12	57.31 ± 0.11
ProtoNet [42]	72.04 ± 0.82	87.22 ± 1.01	33.27 ± 1.09	52.16 ± 0.17
RelationNet [43]	75.62 ± 1.00	87.84 ± 0.27	37.13 ± 0.20	51.76 ± 1.48
DKT [28]	75.40 ± 1.10	$\mathit{90.30 \pm 0.49}$	$\mathbf{40.14 \pm 0.18}$	56.40 ± 1.34
HyperShot [40]	78.06 ± 0.24	89.04 ± 0.18	39.09 ± 0.28	$\mathit{57.77 \pm 0.33}$
HyperShot + adaptation [40]	80.65 ± 0.30	$\mathbf{90.81 \pm 0.16}$	$\mathit{40.03 \pm 0.41}$	$\mathbf{58.86 \pm 0.38}$
OVE PG GP + Cosine (ML) [41]	68.43 ± 0.67	86.22 ± 0.20	39.66 ± 0.18	55.71 ± 0.31
OVE PG GP + Cosine (PL) [41]	77.00 ± 0.50	87.52 ± 0.19	37.49 ± 0.11	57.23 ± 0.31
MAML [11]	74.81 ± 0.25	83.54 ± 1.79	34.01 ± 1.25	48.83 ± 0.62
Bayesian MAML [48]	63.94 ± 0.47	65.26 ± 0.30	33.52 ± 0.36	51.35 ± 0.16
HyperMAML [29]	79.07 ± 1.09	89.22 ± 0.78	36.32 ± 0.61	49.43 ± 0.14
BayesHMAML	$\mathit{80.95 \pm 0.46}$	89.21 ± 0.27	36.90 ± 0.34	49.24 ± 0.38
BayesHMAML + adaptation	$\mathbf{81.05 \pm 0.47}$	89.76 ± 0.26	37.23 ± 0.44	50.79 ± 0.59

- Udało nam się zrealizować i zwizualizować eksperyment dotyczący przedstawienia jednej z największych zalet *Bayesowskich sieci neuronowych*, czyli niepewność modelu.

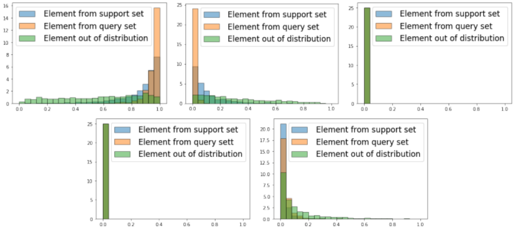

Figure 3: We train BayesHMAML on cross-domain adaptation setting **Omniglot→EMNIST**. Then we sample one thousand different weights from the distribution dedicated to the support set. We present predictions of BayesHMAML in three cases: for an element from the support set, an element from the query set, and an element from **EMNIST** but from classes that were not sampled for the support set. As we can see for elements from support and query sets, our model always gives a similar prediction. We have high uncertainty in the case of elements from out of distribution.

Przybliżony podział pracy

task	PK	PB
głowna implementacja	35%	65%
udział w spotkaniach z zespołem	50%	50%
grid search	60%	40%
implementacja eksperymentu	55%	45%
redakcja artykułu	60%	40%

Rezultaty pracy - FHYPERMAML

termin trwania projektu	Październik 2022 - do teraz
obecny etap projektu	poprawa modułu rozgrzewkowego
technologia	PyTorch

kluczowe osiągnięcia - FHYPERMAML

- Udało nam się dołączyć i skłonić moduł CNF do pracy z modelem HyperMAMLa (po niemałych trudnościach związanych z rozpoznaniem implementacji modelu flowowego)
- Udało nam się w większości zaimplementować planowany model wraz z kilkoma modułami rozgrzewkowymi

Nad czym obecnie pracujemy?

- Próbujemy zniwelować spadek *accuracy*, który pojawia się pod koniec rozgrzewki MAML - FHMAML

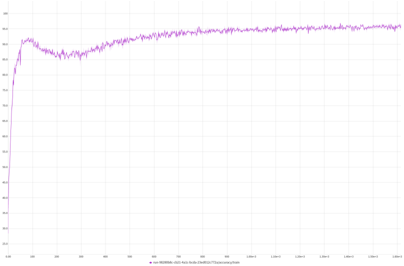

Figure: Accuracy train

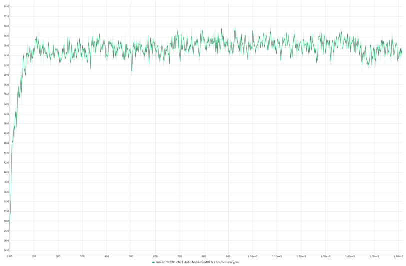

Figure: Accuracy val

Przybliżony podział pracy

task	PK	PB
głowna implementacja	60%	40%
udział w spotkaniach z zespołem	50%	50%
grid search	55%	45%
implementacja eksperymentu	TBA	TBA
redakcja artykułu	TBA	TBA