MetaMix: Improved Meta-Learning with Interpolation-based Consistency Regularization

Yangbin Chen¹, Yun Ma², Tom Ko³, Jianping Wang¹, Qing Li²

1. City University of Hong Kong
2. The Hong Kong Polytechnic University
3. Southern University of Science and Technology

International Conference on Pattern Recognition (ICPR) 2020
Outline

- **Background**: few-shot learning and meta-learning
- **Motivation**: to solve the meta-overfitting problem
- **Methodology**: interpolation-based consistency regularization
- **Experiment**: implementation, result, and discussion
- **Conclusion and future work**
Part I

Background
Few-shot classification

- **Few-Shot Learning (FSL) problem** is a machine learning problem that learns with limited labelled data of the target tasks by incorporating external source data, with a different distribution.

- **Few-Shot Classification** is a few-shot learning task, which is defined as *N-way, K-shot*
  - N is the number of classes in the target task
  - K is the number of labelled examples per class
Meta-Learning

- Most popular solutions of few-shot learning problems use meta-learning.
- Also known as ‘learning to learn’, aims to make a quick adaptation to new tasks with only a few examples.
- Many elegant solutions are proposed:
  - Metric-based: Matching Network, Prototypical Network, Relation Network, etc.
  - Optimization-based: Model-Agnostic Meta-Learning, Reptile, etc.
  - Model-based: Memory-Augmented Meta-Learning, Meta Networks, etc.
Model-Agnostic Meta-Learning (MAML)

- To train a model which can adapt to any new task using only a few labelled examples.
- The model is trained on various tasks (meta-tasks) and it treats the entire task as a training example.
- The model is forced to face different tasks so that it can get used to adapting to new tasks.

Episodic training in MAML

- The model is trained on various meta-tasks and it treats an entire task as a training example.
MAML – the meta-learning stage

\[
\mathcal{L}_{Q_i}(f_{\theta'}) = - \sum_{(x'_u, y'_u) \in Q_i} y'_u \log f_{\theta'}(x'_u)
\]

\[
\theta^* \leftarrow \theta - \beta \nabla_{\theta} \sum_i \mathcal{L}_{Q_i}(f_{\theta'}) \quad \text{outer loop}
\]

\[
\mathcal{L}_{S_i}(f_{\theta}) = - \sum_{(x_j, y_j) \in S_i} y_j \log f_{\theta}(x_j)
\]

\[
\theta'_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_{S_i}(f_{\theta}) \quad \text{inner loop}
\]
MAML – the fine-tuning stage

- Before evaluation, the model will be fine-tuned for a few iterations:

\[ \theta_i' = \theta - \alpha \nabla_{\theta} \mathcal{L}_{S_i}(f_{\theta}) \]
Part II

Motivation
Motivation

- There exist weaknesses in current meta-learning algorithms, especially in forming generalizable decision boundaries (i.e., meta-overfitting).
- We aim to propose a regularization technique to solve the meta-overfitting problem.
The meta-overfitting problem

- Conventional meta-learning algorithms may face meta-overfitting problems, which form a decision boundary *staying too close* to the limited labelled examples in *the few-shot tasks*.
- Empirical Risk Minimization allows large neural networks to *memorize* (instead of *generalize* from) the training data.

\[
\text{expected risk: } R(h) = \int \ell(h(x), y) \, dp(x, y) = \mathbb{E}[\ell(h(x), y)] \\
\text{empirical risk: } R_I(h) = \frac{1}{I} \sum_{i=1}^{I} \ell(h(x_i), y_i)
\]
Part III

Methodology
**mixup** – an interpolation-based regularization method

- **Mixup** [1] encourages the model to behave linearly in-between training examples, which reduces the amount of undesirable oscillations when predicting outside the training examples.
- We have adopted **mixup** in **semi-supervised learning** [2] and **unsupervised domain adaptation** [3].

\[
\hat{x}_z = \lambda x_m + (1 - \lambda)x_n \\
\hat{y}_z = \lambda y_m + (1 - \lambda)y_n
\]

MetaMix – our methodology

Algorithm 1 MetaMix with MAML

Require: \( p(T) \): distribution over tasks
Require: \( S_i \): support set; \( Q_i \): query set
Require: \( \alpha, \beta \): learning rate
Require: \( \alpha, \beta \): distribution parameter

Require: \( \text{mix}_x(a, b) = \lambda a + (1 - \lambda) b, \lambda \sim B(\alpha, \beta) \)

1. Randomly initialize model parameters \( \theta \)
2. while not done do
3. Sample a batch of episodes \( T_i \sim p(T) \)
4. for all \( T_i \) do
5. Sample a support set \( S_i = \{(x_j, y_j)\}_{j=1}^T \)
6. Evaluate \( \nabla\theta L_{S_i} (f_\theta) \) using \( S_i \) and \( L_{S_i} (f_\theta) \)
7. Compute adapted parameters with gradient descent: \( \theta'_i = \theta - \alpha \cdot \nabla\theta L_{S_i} (f_\theta) \)
8. Sample a query set \( Q_i = \{(x_z, y_z)\}_{z=1}^Z \)
9. Randomly select pairs of examples \( \{(x_m, y_m)\}_{m=1}^Z, \{(x_n, y_n)\}_{n=1}^Z \) from \( Q_i \)
10. \( x_z = \text{mix}_x(x_m, x_n), y_z = \text{mix}_x(y_m, y_n) \)
11. Get new query set \( Q_i = \{(x_z, y_z)\}_{z=1}^Z \)
12. end for
13. Update \( \theta \leftarrow \theta - \beta \cdot \nabla\theta \sum_i L_{Q_i} (f_{\theta'_i}) \)
14. end while
MetaMix – our methodology

- We generate virtual examples only from the query set for two reasons:
  - The query set is responsible for optimizing the meta-objective across different training episodes, which is significant to the generalization of the learned initializer.
  - Virtual examples generated by interpolating examples from the query set are expected to better approximate the real data distribution.
Part IV

Experiment
Experimental setup

- Dataset
  - *mini*-ImageNet
    - 100 classes, 600 $84 \times 84$ colored images per class, 64 training / 16 validation / 20 testing.
  - Caltech-UCSD Birds-200-2011 (CUB)
    - 200 classes, 11,788 $84 \times 84$ colored images in total, 100 training / 50 validation / 50 testing.
  - Fewshot-CIFAR100 (FC100)
    - 100 classes, 600 $32 \times 32$ colored images per class, 60 training / 20 validation / 20 testing.
Model setup

- Baselines
  - Prototypical Networks, Matching Network, Relation Network
  - MAML, First-Order MAML (FOMAML), Meta-SGD, Meta-Transfer Learning (MTL)
- Backbone model
  - Shallow CNN with 4 convolutional blocks (Conv([32, 3, 3])+ReLU+BN+MaxPooling([2, 2]))
  - ResNet-12 (in MTL)
## Results

- **Comparison with baselines**

<table>
<thead>
<tr>
<th>Models</th>
<th>mini-ImageNet</th>
<th>CUB</th>
<th>FC100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-shot</td>
<td>5-shot</td>
<td>1-shot</td>
</tr>
<tr>
<td>Matching Network</td>
<td>50.47 ± 0.80</td>
<td>64.83 ± 0.67</td>
<td>57.70 ± 0.87</td>
</tr>
<tr>
<td>Prototypical Network</td>
<td>49.33 ± 0.82</td>
<td>65.71 ± 0.67</td>
<td>51.34 ± 0.86</td>
</tr>
<tr>
<td>Relation Network</td>
<td>50.48 ± 0.80</td>
<td>65.39 ± 0.72</td>
<td>59.47 ± 0.96</td>
</tr>
<tr>
<td>MAML</td>
<td>48.18 ± 0.78</td>
<td>63.05 ± 0.71</td>
<td>54.32 ± 0.91</td>
</tr>
<tr>
<td>MetaMix+MAML</td>
<td>50.51 ± 0.86</td>
<td>65.73 ± 0.72</td>
<td>57.70 ± 0.92</td>
</tr>
<tr>
<td>FOMAML</td>
<td>45.22 ± 0.77</td>
<td>60.97 ± 0.70</td>
<td>53.12 ± 0.93</td>
</tr>
<tr>
<td>MetaMix+FOMAML</td>
<td>47.78 ± 0.77</td>
<td>63.55 ± 0.70</td>
<td>54.81 ± 0.97</td>
</tr>
<tr>
<td>MetaSGD</td>
<td>49.93 ± 1.73</td>
<td>64.01 ± 0.90</td>
<td>56.19 ± 0.92</td>
</tr>
<tr>
<td>MetaMix+MetaSGD</td>
<td>50.60 ± 1.80</td>
<td>64.47 ± 0.88</td>
<td>57.64 ± 0.88</td>
</tr>
<tr>
<td>MTL</td>
<td>61.37 ± 0.82</td>
<td>78.37 ± 0.60</td>
<td>71.90 ± 0.86</td>
</tr>
<tr>
<td>MetaMix+MTL</td>
<td>62.74 ± 0.82</td>
<td>79.11 ± 0.58</td>
<td>73.04 ± 0.86</td>
</tr>
</tbody>
</table>

Accuracy with 95% confidence intervals of 5-way, K-shot (K=1, 5) classification tasks on mini-ImageNet, CUB, and FC100 datasets.
Results

- Analysis of hyper-parameter in Beta distribution

Effect of Beta distribution. $\alpha$ is set to 0.1, 0.2, 0.5, 0.8, 1.0, 2.0, 4.0, 8.0.
Results

- Ablation study

<table>
<thead>
<tr>
<th>Set(s)</th>
<th>mini-ImageNet</th>
<th>CUB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-shot</td>
<td>5-shot</td>
</tr>
<tr>
<td>Q</td>
<td>50.51 ± 0.86</td>
<td>65.73 ± 0.72</td>
</tr>
<tr>
<td>S</td>
<td>47.87 ± 0.82</td>
<td>62.34 ± 0.65</td>
</tr>
<tr>
<td>Q+S</td>
<td>48.36 ± 0.81</td>
<td>64.06 ± 0.72</td>
</tr>
<tr>
<td>w/o mixup</td>
<td>48.18 ± 0.78</td>
<td>63.05 ± 0.71</td>
</tr>
</tbody>
</table>

An ablation study of doing mixup on different sets. Q denotes the query set and S denotes the support set.
Results

- Analysis of the effect of the size of training data

<table>
<thead>
<tr>
<th>Set(s)</th>
<th>mini-ImageNet 1-shot</th>
<th>mini-ImageNet 5-shot</th>
<th>CUB 1-shot</th>
<th>CUB 5-shot</th>
<th>FC100 1-shot</th>
<th>FC100 5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML(100%)</td>
<td>48.18 ± 0.78</td>
<td>63.05 ± 0.71</td>
<td>54.32 ± 0.91</td>
<td>71.37 ± 0.76</td>
<td>35.96 ± 0.71</td>
<td>48.06 ± 0.73</td>
</tr>
<tr>
<td>MetaMix+MAML(100%)</td>
<td>50.51 ± 0.86</td>
<td>65.73 ± 0.72</td>
<td>57.70 ± 0.92</td>
<td>73.66 ± 0.74</td>
<td>37.09 ± 0.74</td>
<td>49.31 ± 0.72</td>
</tr>
<tr>
<td>MAML(50%)</td>
<td>46.34 ± 0.82</td>
<td>60.47 ± 0.73</td>
<td>50.78 ± 0.86</td>
<td>65.60 ± 0.81</td>
<td>35.38 ± 0.71</td>
<td>47.93 ± 0.78</td>
</tr>
<tr>
<td>MetaMix+MAML(50%)</td>
<td>48.04 ± 0.79</td>
<td>63.52 ± 0.67</td>
<td>53.22 ± 0.91</td>
<td>70.13 ± 0.70</td>
<td>36.35 ± 0.74</td>
<td>48.11 ± 0.69</td>
</tr>
</tbody>
</table>

A comparison between using 100% and 50% training data; accuracy with 95% confidence intervals of 5-way, K-shot (K=1, 5) classification tasks on mini-ImageNet, CUB, and FC100 datasets.
Results

- Analysis of the effect of the size of training data

A comparison among using 100%, 50%, 40%, and 30% of the training data.
Observations

- MetaMix improves the performance of all MAML-based algorithms over three datasets; meanwhile, MetaMix with MTL achieves state-of-the-art performance.

- When $\alpha$ is below 1.0, the accuracy is a little lower. When $\alpha$ is 1.0 and above, the performance maintains a good level.

- Mixing examples from only the query set performs best, compared with mixing examples from only the support set and mixing examples from both the support set and the query set.

- MetaMix performs more robust with the reduction of the size of the training data.
Part V

Conclusions
Conclusion

‣ We propose an improved meta-learning approach with the interpolation-based consistency regularization technique. It improves the performance of MAML-based algorithms.

‣ MetaMix achieves state-of-the-art results when integrated with Meta-Transfer Learning.

‣ MetaMix is less sensitive to the reduction of the source training data, compared to MAML and its variants.
Future work

‣ Apply MetaMix to a broader range of few-shot learning tasks.

‣ Compare more different conditions, under which meta-learning works, such as differences in the size of the source data, backbone models, and domains of the tasks.

‣ Propose more regularization techniques to solve the meta-overfitting problem.
Thank you!

Email: robinchen2-c@my.cityu.edu.hk
Github: https://github.com/Codelegant92