Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

Chelsea Finn etc.
Cite: 3198

2021.03.25.목
발표자: 임진혁

Index

임진혁<eeplearning@units.ac.kr>

Introduction: what is Meta-learning?

Meta-learning: **learn to learn** (like human)

⇒ **적은 샘플로 어떤 task이든 높은 성능(few shot task adaptive problem)**

Q: Multi-task learning과 어떻게 다른가요?

임진혁<eeplearning@units.ac.kr>

Introduction: what is Meta-learning?

Meta-learning: **learn to learn** (like human)

⇒ 적은 샘플로 어떤 task이든 높은 성능(few shot task adaptive problem)

Q: Multi-task learning과 어떻게 다른가요?

➔ task 별로 다른 데이터셋이라는 세팅은 동일

➔ Multi-task learning: 모든 task가 동일한 optimal parameter ∅을 가짐

➔ Meta-learning: task 별로 고유한 optimal parameter ∅을 가지기 때문에 ∅을 찾는 것 의미 x

Introduction: what is Meta-learning?

Meta-learning: **learn to learn** (like human)

⇒ 적은 샘플로 어떤 task이든 높은 성능(few shot task adaptive problem)

Q: Multi-task learning과 어떻게 다른가요?

➔ task 별로 다른 데이터셋이라는 세팅은 동일

➔ Multi-task learning: 모든 task가 동일한 optimal parameter ∅을 가짐

➔ Meta-learning: task data(x) : label(y)이 아닌 task data(x): ∅ 을 학습함으로써 처음보는 task가 들어와도 빠르게 학습할 수 있음

임진혁<eeplearning@units.ac.kr>

Introduction: what is MAML?

Meta-learning: **learn to learn** (like human)

MAML: Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

"Learn to learn"(메타러닝)을 통해
new task에 빠르게 적응(학습)하는
gradient based method

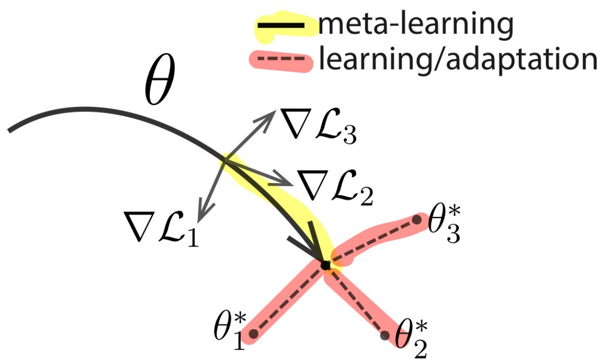


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

임진혁<eeplearning@units.ac.kr>

Introduction: what is MAML?

Meta-learning: **learn to learn** (like human)

MAML: Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

"method trains the model
to be easy to fine-tune"

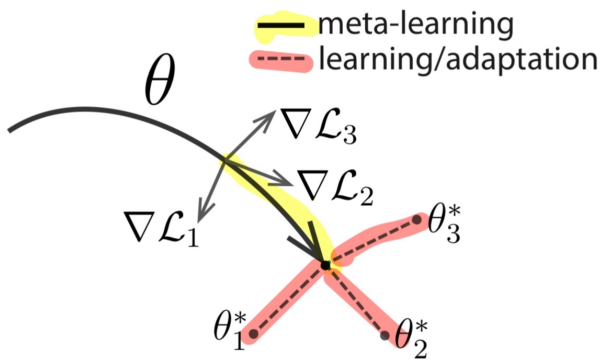


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

Introduction: what is MAML?

Meta-learning: **learn to learn** (like human)

MAML: Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

Goal:

"train the model's initial parameters

for model performance on a new task

after few parmeter update(finetune)

with a small amount of data from that new task."

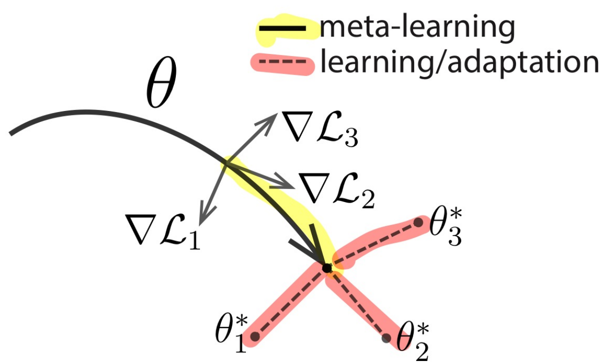
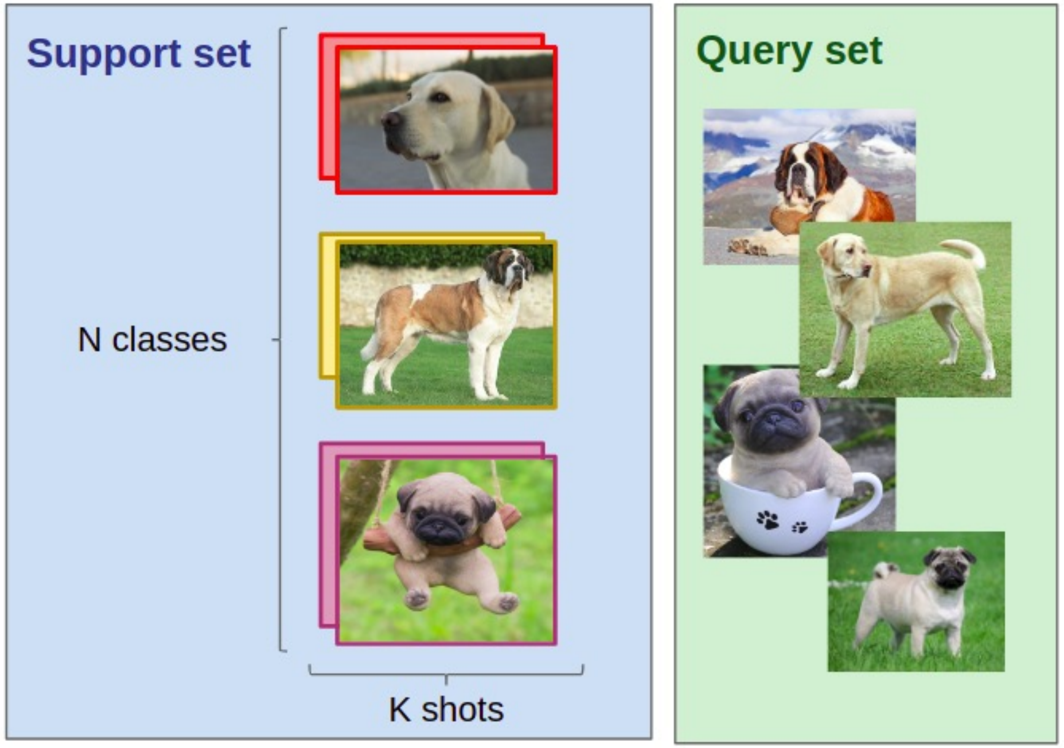
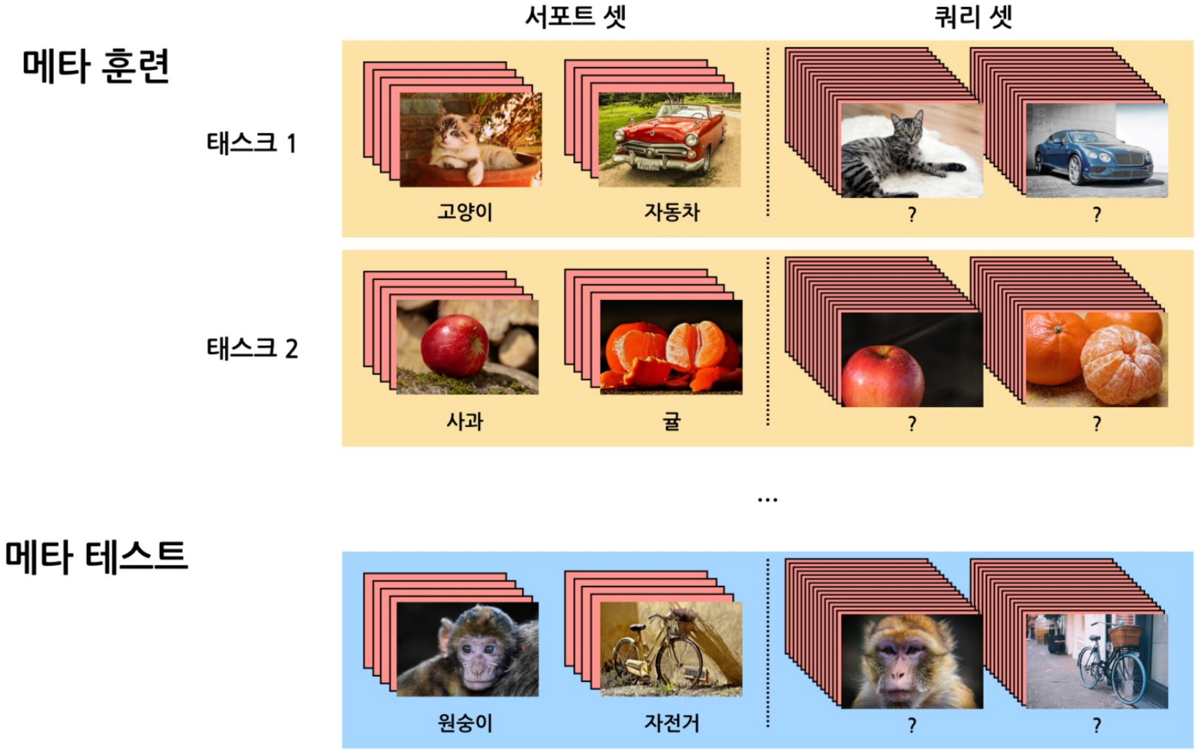


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

N–shot K–way Learning : k가 작으면 "few shot" learning





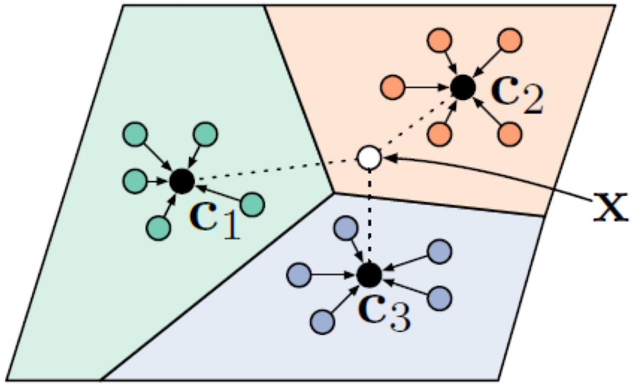
Meta learning: Support set(meta-train)과 전혀 상관없는 unseen class(new task)에 대해서도 잘하자

임진혁<eeplearning@unist.ac.kr>

Related: meta learning methods

(1) Metric-based approach

- Meta learning phase의 훈련데이터(meta train)을 임베딩(mapping:θ)
- New task(새로운 데이터)가 들어오면 임베딩 시키고 가장 가까운 클래스로 배정



(a) Few-shot

(2) Optimization-based approach

- Meta train을 통해 학습하는 법을 배움(learn to optimize parameter:θ)
- MAML

임진혁<eeplearning@unist.ac.kr>

Proposed Method

Goal : 적은 데이터 샘플과 적은 훈련 횟수(iteration) 만으로 new task에 빠르게 학습(adapt)

➔ "maximizing the sensitivity of the loss functions of new tasks with respect to the parameters"

Sensitivity?: 해당 task에 대해 매우 약간의 변화(update)만으로 큰 변화(loss 차이)

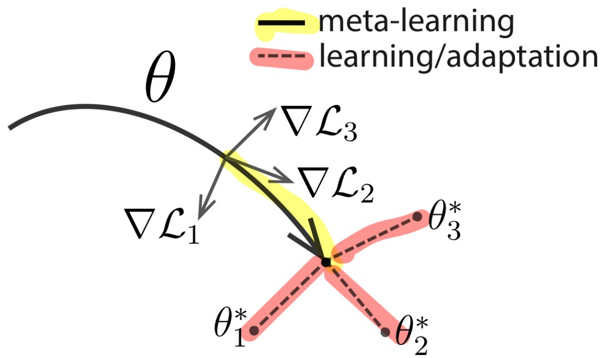


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

"method trains the model to be easy to fine-tune"

Sensitivity한(easy to fine-tune) initial parmeter θ를 찾자

Proposed Method

Goal : 적은 데이터 샘플과 적은 훈련 횟수(iteration) 만으로 new task에 빠르게 학습(adapt)

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
6: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
7: **end for**
8: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
9: **end while**

Adaptation : $\emptyset_i \leftarrow \theta - \alpha \nabla_\theta L(\theta, D_i^{tr})$

Meta-learning : $\theta \leftarrow \theta - \beta \nabla_\theta \sum L\left(\emptyset_i, D_i^{test}\right)$

임진혁<eeplearning@unist.ac.kr>

Proposed Method

Goal : 적은 데이터 샘플과 적은 훈련 횟수(iteration) 만으로 new task에 빠르게 학습(adapt)

$\emptyset_i \leftarrow \theta - \alpha \nabla_\theta L(\theta, D_i^{tr})$

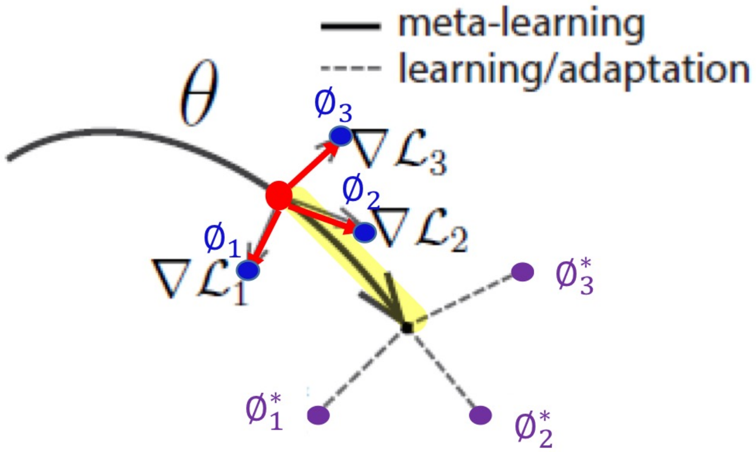
θ를 $\emptyset_i$의 weight initialization으로 사용

D_i^{tr}의 양이 적기 때문에 적은 update 만으로 θ -> $\emptyset_i$

$\theta \leftarrow \theta - \beta \nabla_\theta \sum L(\emptyset_i, D_i^{test})$

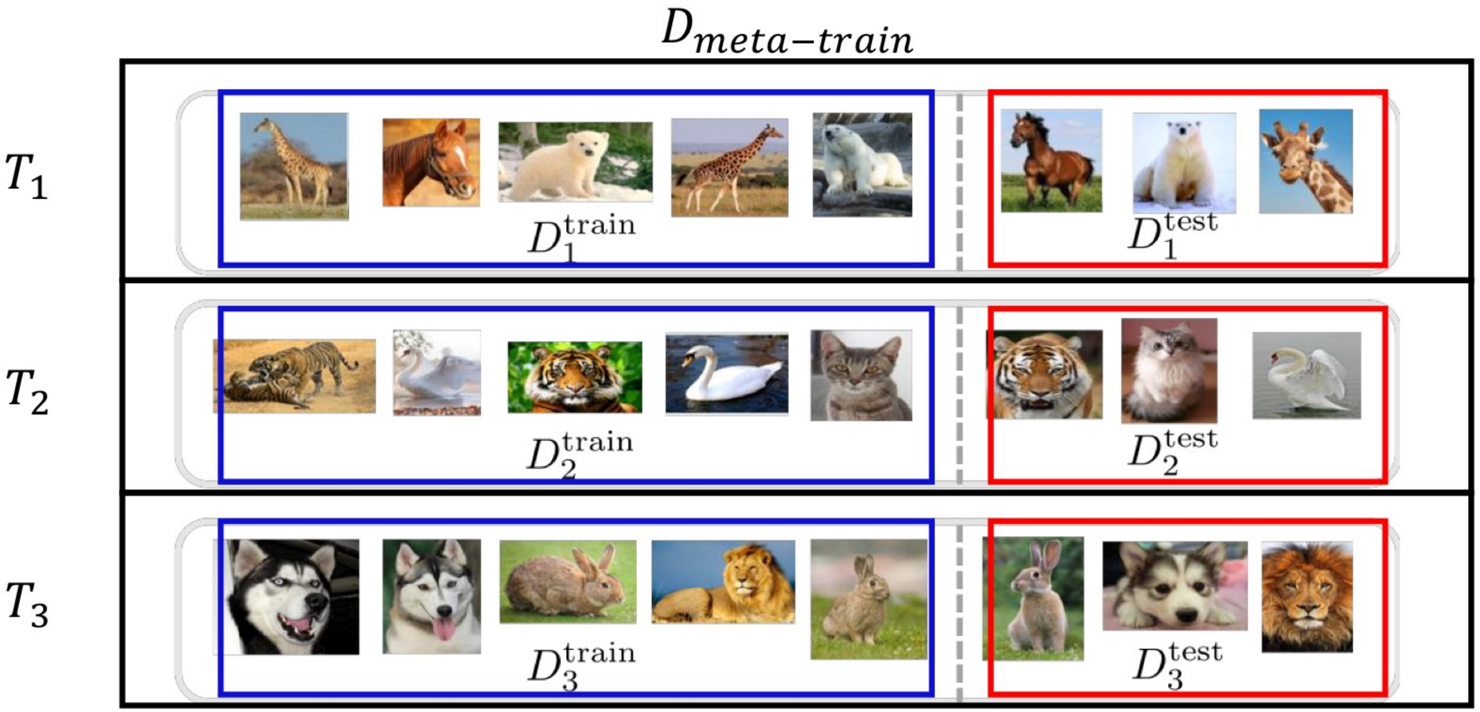
$L(\emptyset_i, D_i^{test})$ 가 최소인 경우는 $L(\emptyset_i^*, D_i^{test})$

즉 $\emptyset_i$ = $\emptyset_i^*$가 되는 방향으로 θ를 업데이트



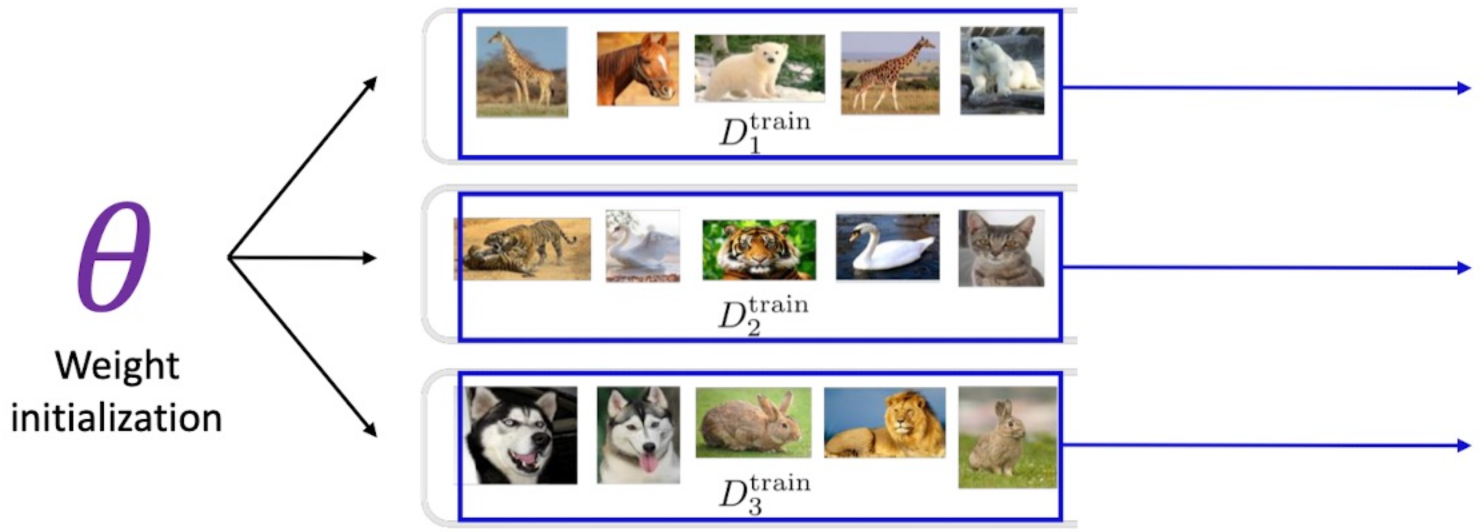
즉, 적은 update 만으로 $\emptyset_i^*$를 구할 수 있는 θ를 찾는 것이 meta-learning의 목적

Proposed Method: 실제 학습 과정 (제가 만들지 않았습니다)



임진혁<eeplearning@unist.ac.kr>

Proposed Method: 실제 학습 과정 (제가 만들지 않았습니다)



Optimize $\emptyset_1 \leftarrow \theta - \alpha \nabla_\theta L(\theta, D_1^{tr})$

Optimize $\emptyset_2 \leftarrow \theta - \alpha \nabla_\theta L(\theta, D_2^{tr})$

Optimize $\emptyset_3 \leftarrow \theta - \alpha \nabla_\theta L(\theta, D_3^{tr})$

임진혁<eeplearning@unist.ac.kr>

3 Proposed Method: 실제 학습 과정 (제가 만들지 않았습니다)

$$\theta' \longleftarrow \theta - \beta \nabla_\theta \sum L(\emptyset_i, D_i^{test})$$

Optimize $\emptyset_1 \leftarrow \theta - \alpha \nabla_\theta L(\theta, D_1^{tr})$

Optimize $\emptyset_2 \leftarrow \theta - \alpha \nabla_\theta L(\theta, D_2^{tr})$

Optimize $\emptyset_3 \leftarrow \theta - \alpha \nabla_\theta L(\theta, D_3^{tr})$

임진혁<eeplearning@unist.ac.kr>

Proposed Method: 실제 학습 과정 (제가 만들지 않았습니다)

ex)5-way, 1-shot classification

New data :D_{train}^{new}, D_{test}^{new}

θ

optimize $\emptyset_{new} \leftarrow \theta - \alpha \nabla_{\theta} L(\theta, D_{train}^{new})$

$$\hat{y} = f_{\emptyset_{new}}(x_{test})$$

임진혁<eeplearning@unist.ac.kr>

Experiment

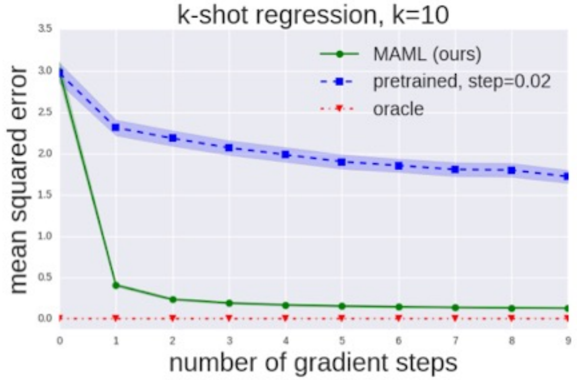


Figure 3. Quantitative sinusoid regression results showing the learning curve at meta test-time. Note that MAML continues to improve with additional gradient steps without overfitting to the extremely small dataset during meta-testing, achieving a loss that is substantially lower than the baseline fine-tuning approach.

Omniglot (Lake et al., 2011)	5-way Accuracy 1-shot	5-way Accuracy 5-shot	20-way Accuracy 1-shot	20-way Accuracy 5-shot
MANN, no conv (Santoro et al., 2016)	82.8%	94.9%	–	–
MAML, no conv (ours)	**89.7 ± 1.1%**	**97.5 ± 0.6%**	–	–
Siamese nets (Koch, 2015)	97.3%	98.4%	88.2%	97.0%
matching nets (Vinyals et al., 2016)	98.1%	98.9%	93.8%	98.5%
neural statistician (Edwards & Storkey, 2017)	98.1%	99.5%	93.2%	98.1%
memory mod. (Kaiser et al., 2017)	98.4%	99.6%	95.0%	98.6%
MAML (ours)	**98.7 ± 0.4%**	**99.9 ± 0.1%**	**95.8 ± 0.3%**	**98.9 ± 0.2%**

MiniImagenet (Ravi & Larochelle, 2017)	5-way Accuracy 1-shot	5-way Accuracy 5-shot
fine-tuning baseline	28.86 ± 0.54%	49.79 ± 0.79%
nearest neighbor baseline	41.08 ± 0.70%	51.04 ± 0.65%
matching nets (Vinyals et al., 2016)	43.56 ± 0.84%	55.31 ± 0.73%
meta-learner LSTM (Ravi & Larochelle, 2017)	43.44 ± 0.77%	60.60 ± 0.71%
MAML, first order approx. (ours)	**48.07 ± 1.75%**	**63.15 ± 0.91%**
MAML (ours)	**48.70 ± 1.84%**	**63.11 ± 0.92%**

Experiment+ application to RL

Algorithm 3 MAML for Reinforcement Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Sample K trajectories $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using f_θ in $\mathcal{T}_i$
6: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
7: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
8: Sample trajectories $\mathcal{D}_i' = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using $f_{\theta_i'}$ in $\mathcal{T}_i$
9: **end for**
10: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
11: **end while**

Sorry…

임진혁<eeplearning@unist.ac.kr>