1. Model-Agnostic Meta_learning for Fast Adaptation of Deep Networks

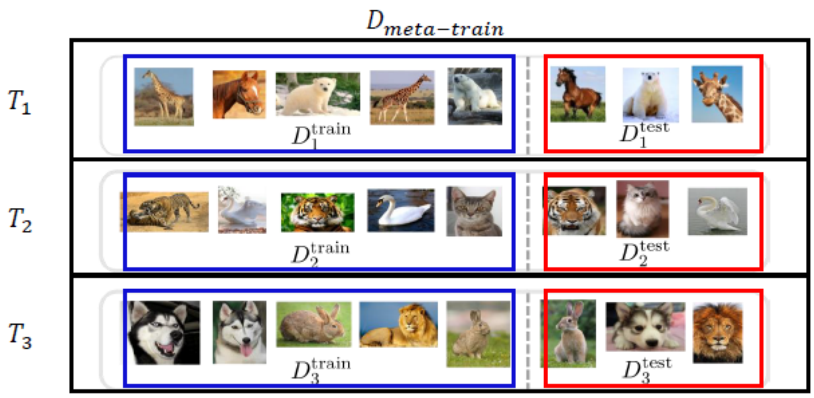
논문 : https://arxiv.org/pdf/1703.03400.pdf

요약

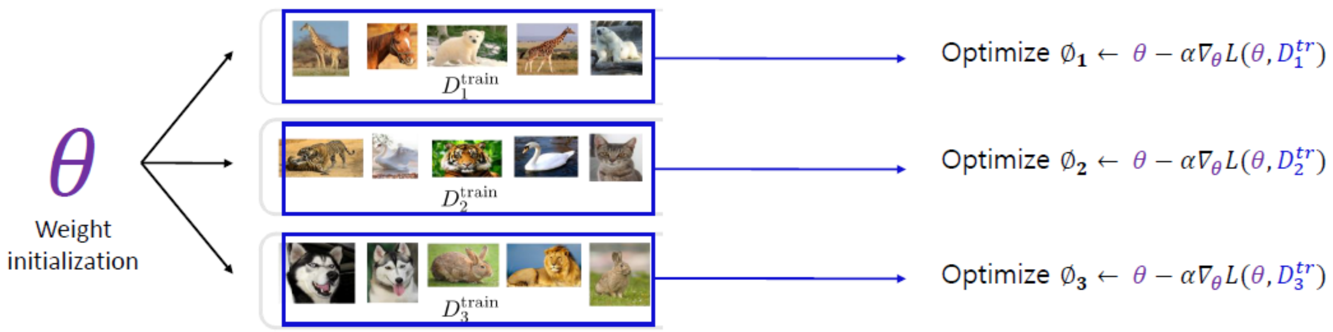
- MAML의 첫번째 논문으로 gradient descent 기법으로 학습하는 모든 분야 - classification, regression, reinforcement learning - 에서 적용할 수 있는 optimization-based meta learning을 제안
- 적은양의 데이터와 적은 학습 횟수로 fine-tuning이 잘 될 수 있는 pre-train 모델의 파라미터를 학습하는 것이 목적
    - pre-train에 사용하는 데이터는 많아도되고 적어도됨

ppt 4장으로 요약

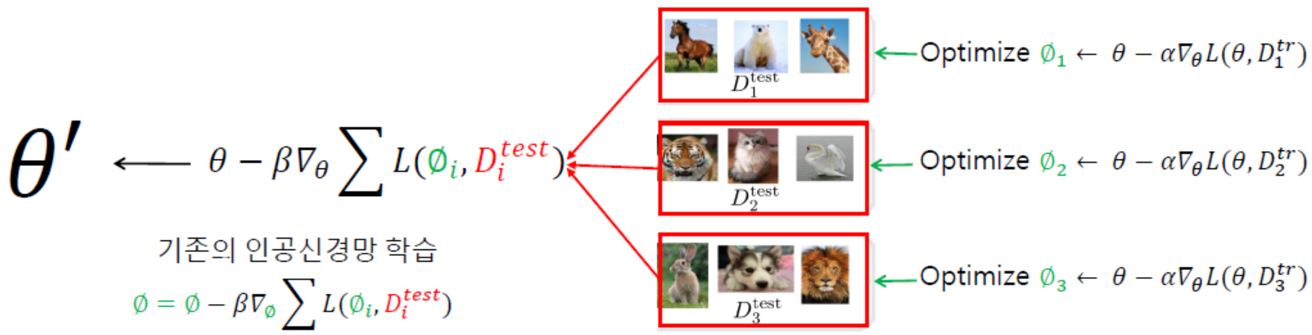
1. D_i 를 D_i^{train}, D_i^{test} 으로 분할



2. θ 와 D_i^{train}를 이용하여 $\emptyset_i$를 구함(모델 학습)



3. $\emptyset_i$와 D_i^{test}을 이용하여 θ update

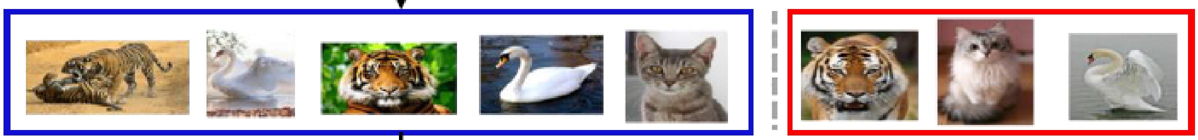


4. θ를 이용하여 새로운 Task에 빠르게 학습(Adaptation)하는 과정

ex)5-way, 1-shot classification

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

θ



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

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

학습 형태 (그림 1, 알고리즘 1)

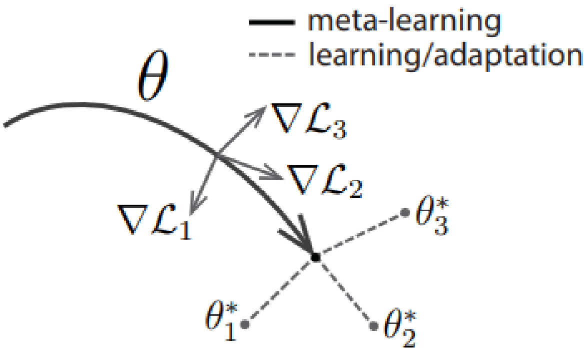


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

```
randomly initialize θ
while not done do
    Sample batch of tasks T_i ~ p(T)
    for all T_i do
        Evaluate ∇_θ L_{T_i}(f_θ) with respect to K examples
        Compute adapted parameters with gradient de-
        scent: θ'_i = θ − α∇_θ L_{T_i}(f_θ)
    end for
    Update θ ← θ − β∇_θ Σ_{T_i~p(T)} L_{T_i}(f_{θ'_i})
end while
```

- pre-train의 1epoch 과정 순서
    1. 전체 에피소드 중에서 I개 랜덤 샘플링
    2. 동일한 초기 파라미터 θ 를 가진 상태에서 episode에 대해 fine-tuning (I개 만큼의 모델이 생성됨)
       $$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$$
    3. fine-tuning된 모델들로 i번째 Test 셋의 Loss를 계산해서 초기 파라미터를 업데이트
       $$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$$
- fine-tuning 과정
    - 일반적인 fine-tuning 과정을 거침
- 논문에서는 1~5 way learning이 학습이 잘된다라고 말함

Supervised Learning (알고리즘 2)

- 전체 클래스에 대한 데이터셋을 에피소드형태로 분리해서 MAML을 적용

Algorithm 2 MAML for Few-Shot Supervised Learning

Require: $p(\mathcal{T})$: distribution over tasks

Require: α, β: step size hyperparameters

randomly initialize θ

while not done **do**

  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$

  for all $\mathcal{T}_i$ **do**

    Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$

    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2) or (3)

    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$

    Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update

  end for

  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 2 or 3

end while

Reinforcement Learning (알고리즘 3)

- 상태 x가 주워지고 초기 파라미터로 액션 a를 선택
- (식 4)Negative Reward R가 포함된 Loss로 Loss를 줄이는 쪽으로 fine-tuning
- fine-tuning 후 같은 상태를 입력, 액션에 대한 loss를 바탕으로 초기 파라미터 업데이트

$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = -\mathbb{E}_{\mathbf{x}_t, \mathbf{a}_t \sim f_\phi, q_{\mathcal{T}_i}} \left[\sum_{t=1}^{H} R_i(\mathbf{x}_t, \mathbf{a}_t) \right]. \quad (4)$$

Algorithm 3 MAML for Reinforcement Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

randomly initialize θ
while not done **do**
    Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
    for all $\mathcal{T}_i$ **do**
        Sample K trajectories $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using f_θ in $\mathcal{T}_i$
        Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
        Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
        Sample trajectories $\mathcal{D}_i' = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using $f_{\theta_i'}$ in $\mathcal{T}_i$
    end for
    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
end while

실험

- Omniglot 데이터셋 / MiniImagenet 데이터셋으로 기존기법과 성능비교 (표 1)
    - Omniglot
        - MAML에 1200개 문자 사용
        - 사용하지 않은 423개 문자로 테스트
    - MiniImageNet
        - train : 64 종류의 이미지
        - validation : 12 종류의 이미지
        - test : 24 종류의 이미지
- 일정 진폭과 위상을 가진 그래프를 regression / 기존학습기법과 제안기법의 성능비교 (그림 2)
    - 연두색 : MAML update 전 초기 파라미터를 사용한 결과
    - 녹색 : MAML update 후 파라미터를 사용한 결과 (10 epoch)
    - 빨간색 : 실제 값
    - 하늘색 : pretrained model을 사용한 결과
    - 파랑색 : fine-tuning 후 결과 (10 epoch)
    - 세모 표시 : 학습에 사용한 데이터 포인트
    - MAML은 데이터포인트가 없어도 어느정도 예측하는 모습

Table 1. Few-shot classification on held-out Omniglot characters (top) and the MiniImagenet test set (bottom). MAML achieves results that are comparable to or outperform state-of-the-art convolutional and recurrent models. Siamese nets, matching nets, and the memory module approaches are all specific to classification, and are not directly applicable to regression or RL scenarios. The $\pm$ shows 95% confidence intervals over tasks. Note that the Omniglot results may not be strictly comparable since the train/test splits used in the prior work were not available. The MiniImagenet evaluation of baseline methods and matching networks is from Ravi & Larochelle (2017).

	5-way Accuracy		20-way Accuracy	
Omniglot (Lake et al., 2011)	1-shot	5-shot	1-shot	5-shot
MANN, no conv (Santoro et al., 2016)	82.8%	94.9%	–	–
MAML, no conv (ours)	$\mathbf{89.7 \pm 1.1\%}$	$\mathbf{97.5 \pm 0.6\%}$	–	–
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)	$\mathbf{98.7 \pm 0.4\%}$	$\mathbf{99.9 \pm 0.1\%}$	$\mathbf{95.8 \pm 0.3\%}$	$\mathbf{98.9 \pm 0.2\%}$

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

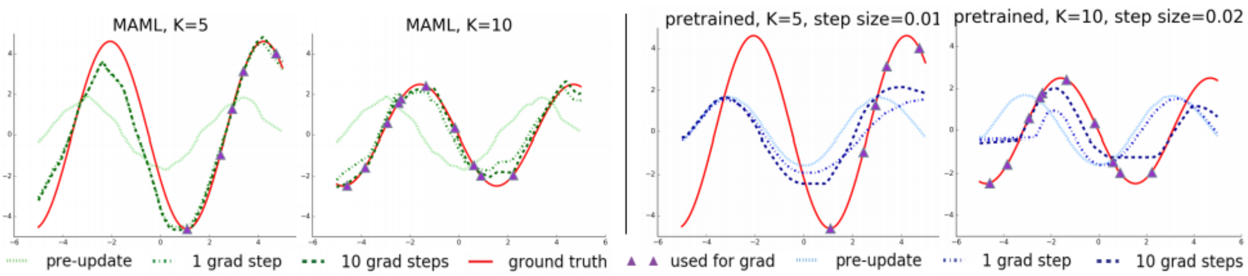


Figure 2. Few-shot adaptation for the simple regression task. Left: Note that MAML is able to estimate parts of the curve where there are no datapoints, indicating that the model has learned about the periodic structure of sine waves. Right: Fine-tuning of a model pretrained on the same distribution of tasks without MAML, with a tuned step size. Due to the often contradictory outputs on the pre-training tasks, this model is unable to recover a suitable representation and fails to extrapolate from the small number of test-time samples.