

이미지의 오브젝트 분류에 대한 메타러닝 연구

김관우
동국대학교 컴퓨터공학전공
kw2577@naver.com

A meta-learning study of object classification in images

Kwan-Woo Kim
Dongguk Univ. Computer Science

요 약

메타러닝은 사용할 수 있는 데이터 셋의 양이 한정적일 때 효율적으로 모델을 학습할 수 있는 방법론으로 학습하는 방법을 학습하는 것이다. 이러한 메타러닝은 다양한 분야에서 활용이 가능하고 해당 실험에서는 이미지의 오브젝트 분류에 대한 메타러닝의 연구를 수행한다. 특히 문제는 동물의 실루엣 이미지를 인풋 데이터 셋으로 하여 분류하는 것으로 설정하고, 직접 모델을 구현해 학습한다. 실험 결과 다음과 같은 결론을 얻을 수 있다. 1) 사용할 데이터 셋이 적을 때 메타러닝은 효율적인 방법이 될 수 있다. 2) 이미지의 오브젝트 분류에서도 메타러닝을 적용할 수 있다. 3) 단 이미지의 오브젝트는 크기와 자세에 따라 모양이 많이 다르기 때문에 데이터의 제한 사항이 필요하다.

1. 서 론

우리는 다양한 물체의 사진을 찍는다. 이렇게 찍은 물체를 컴퓨터가 인식하고 어떤 물체인지 알 수 있도록 인공지능을 만들 수 있을까? 물론 가능할 것이다. 하지만 이러한 물체에는 여러 종류가 있다. 이는 사람이 될 수도 있고, 원숭이, 컵, 책상 등 모두 포함될 수 있다. 여기서 사람, 원숭이 등은 동물에 속하고, 컵, 책상 등은 사물에 해당한다. 더 세부적으로 나누면 동물은 포유류, 조류 등으로 분류될 수도 있고, 사물은 사무용품, 주방 품 등으로 분류될 수 있다. 이러한 모든 종합적인 물체를 구분할 수 있는 인공지능 모델도 만들 수 있지만 이를 구현하기 위해서는 각 물체에 대한 데이터가 상당히 많이 필요해 어려움이 발생한다.

따라서 학습하는 방법을 학습해 물체를 구분할 수 있는 인공지능을 적은 데이터로 구현할 수 있도록 할 수 있다. 이렇게 물체를 인식할 수 있는 인공지능 모델을 학습한다면 다양한 분야에서 활용될 수 있다. 예를 들어 이렇게 이미지를 분류하는 인공지능 모델에서 데이터로 손의 모양의 이미지 파일을 사용한다면 수화의 모양을 인식해 분류하고, 장애인에게 도움이 되는 시스템을 만드는 등의 활용이 가능할 것이다.

이러한 적은 데이터로 학습해 모델을 만들 수 있는 메타러닝을 연구하기 위해 이 논문에서는 각 동물의 실루엣 이미지를 종류별로 나눈 데이터 셋을 이용하여 분류하는 모델을 인공지능으로 학습하고 성능을 평가한다.

2. 본 론

2.1 메타러닝의 개념

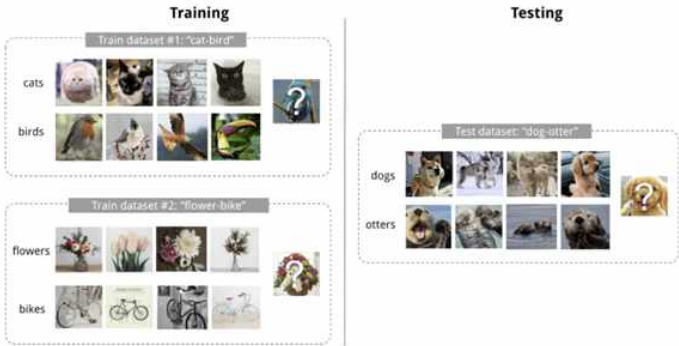
메타러닝은 Learning to learn으로 일련의 학습 경험을 통해 효율적으로; 학습하는 방법을 학습한다. 즉 메타러닝은 적은 데이터로 학습하기 위한 방법이다. 기존에도 존재했지만, Deep Learning과 결합하여 로봇, 강화학습, GPT-3 등에 적용되며 뛰어난 성능을 보이고 있다. 또한, 메타러닝은 몇몇 training 예제를 통해서 모델로 하여금 새로운 기술을 배우거나 새로운 환경에 빠르게 적응할 수 있도록 설계하는 것을 나타낸다.

일반적으로 좋은 메타러닝 모델은 training time 동안에 접하지 않았던 새로운 환경에 대해 잘 적응하거나, 일반화가 잘 되는 것을 의미한다. 적응 프로세스는 테스트 과정에서 일어나게 되는데 이 때, 새로운 task의 배열에 대해서만 제한적으로 노출되어 있다. 결국 이러한 방식으로 적응된 모델은 새로운 task를 잘 수행할 수 있다. task는 supervised learning 또는 강화학습 등의 머신러닝으로 정의될 수 있는 모든 문제가 될 수 있다.

2.2. 관련 연구

현재 적은 데이터로 학습하여 모델을 효율적으로 만들 수 있는 메타러닝에 관련된 많은 연구가 진행되고 있는데 대표적인 것이 퓨샷 분류에 관련된 것이다. 퓨샷 분류는 supervised learning에서 메타러닝을 활용한 예시이다. 이 때, 데이터 셋 D 는 크게 두 개로 나누어지는데 하나는 학습을 위한 support set S 이고, 다른 하나는 training이나 testing을 위한 prediction set B 이다. 결국 데이터 셋 D 는 $D = \langle S, B \rangle$ 로 표현된다. 일반적으로 K-shot N-class classification task를 다루게 되는데 이는 support set S 가 각각 N 개의 클래스에 대해 K 개로 레이

블링 된 데이터들을 포함하고 있다는 것이다. 아래 그림 1은 4-shot 2-class image classification task의 예시이다.



[그림 1]

2.3. 동물의 실루엣 이미지를 분류하는 모델 학습

2.3.1. 문제 정의

이번 연구에서는 동물들의 실루엣 이미지 데이터 셋을 사용하여 메타러닝을 수행하고 이를 통해 우선 물체 중 동물들의 종류를 분류할 수 있는 모델을 학습한다. 아래 표 1은 사용한 데이터 셋의 20개의 클래스에 포함되는 동물의 실루엣 이미지의 예시를 하나씩 제시한 것이다. [표 1]

bird	butterfly	cat	cow	crocodile
deer	dog	dolphine	duck	elephant
fish	flyingbird	hen	horse	leopard
monkey	rabbit	rat	spider	tortoise

2.3.2. MAML 알고리즘

제시한 문제를 해결하기 위해 메타러닝 중 MAML 알고리즘을 사용했다. Model-Agnostic Meta-Learning (MAML)은 gradient descent를 통해서 학습하는 어떤 모델에도 적용이 가능한, 일반적인 optimization 알고리즘이다. [1] θ 에 의해서 조절되는 모델을 학습시킨다면 주어진 task와 이에 연관된 데이터셋 ($D^{(i)}_{train}$, $D^{(i)}_{test}$)가 있을 때 하나 또는 여러 gradient descent 과정을 통해서 모델의 파라미터를 업데이트 할 수 있다. 아래 그림 2는 gradient descent의 첫 번째 과정을 수식으로 나타낸 것이다.

$$\theta'_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta}).$$

[그림 2]

그림 2의 식에서는 하나의 task에 대해서만 optimize가 가능하다. 여러 task를 generalization 하기 위해서는 optimal θ^* 을 구해야 한다. [2] meta-objective를 업데이트하기 위해서는 다음 mini batch data를 샘플링 해야 한다. 이 때, Loss는 mini batch에 따라 결정된다. 결론적으로 아래 그림 3과 같은 수식으로 θ^* 을 구할 수 있다.

$$\min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i}) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta})})$$

[그림 3]

이와 같이 수행되는 [3] MAML의 알고리즘의 일반적인 형태는 아래 그림 4과 같다.

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

[1] [그림 4]

2.4. 실험 방법

2.4.1. 메타러닝 모델 구현

모델은 아나콘다 환경에서 파이썬과 tensorflow를 사용하여 MAML 알고리즘으로 이미지의 물체를 분류할 수 있는 코드를 구현했다. 여기에 앞서 문제에서 정의한 동물의 실루엣 이미지 데이터 셋을 사용하여 모델을 학습시켰다.

2.4.2. 성능 평가

MAML 모델에서는 meta training data, meta validation data, meta test data가 구분되고, 구분된 테스트 데이터 셋은 각각의 역할에 맞는 함수를 수행하게 된다. 그 중 meta test data는 evaluation에 사용된다. 간단히 설명하면 meta test data를 통해 테스트한 모델의 accuracy가 높게 나오면 그만큼 일반화 성능이 높다고 볼 수 있다. 주어진 MAML 코드에서 살펴보면 Data loading의 단계에서 sample_batch() 함수를 보면 meta_train, meta_val, meta_test 데이터를 각각 샘플 배치로 생성하여 metatrain_character_folders, metaval_character_forders,

metatest_character_forders로 저장하고 있다. 즉 이 코드에서는 성능을 평가할 때 metatest_character_forders에 있는 meta test data를 사용해 학습한 모델에 대하여 accuracy를 측정하고 이를 일반화 성능으로 보는 것이다. 해당 연구에서도 이러한 방식을 사용하여 학습한 모델의 성능을 평가하였다.

2.4.3. 실험 방법

구현한 MAML 프로그램에서 총 두 가지 모델을 학습해 보았는데 첫 번째는 5-way 1-shot으로 모델을 학습하였고, 두 번째는 5-way 2-shot으로 모델을 학습해 보았다. 두 모델 모두 num_inner_updates는 1로 설정하고, 학습은 5000번 반복하여 수행하였다. 여기서 학습 계수는 0.4로 설정하고, 메타 배치의 사이즈는 10으로 고정하였다. 이러한 두 가지 모델을 각각 학습하고, 모델의 성능을 앞서 설명한 성능 평가 방법으로 확인한 뒤 비교해 보았다.

2.5. 결과

첫 번째로 앞서 설명한 실험 방법의 기본 설정에서 5-way 1-shot으로 학습한 모델의 성능을 평가해 보았다. 아래 그림 5는 해당 모델의 결과이다.

```
run_maml(n_way=5, k_shot=1, inner_update_lr=0.4, num_inner_updates=1, meta_train=False, logdir='./log/model/')
Restoring model weights from ./log/model/cis_5_abs_10_k_shot_1_inner_numstep_1_inner_update_lr_0.4_learn_inner_update_lr_False#model.4999
Mean meta-test accuracy/loss, stddev, and confidence intervals
(0.41200003, 0.1587115, 0.01269564357290001)
```

[그림 5]

그림 5의 결과를 살펴보면 해당 모델의 meta-test의 성능이 0.41200003이다. 즉 일반적으로 약 41%의 정확도를 가진 모델이다. 학습 과정에서 살펴보면 학습을 시작한 초반의 정확도는 약 20% 정도였고, 학습을 진행하며 정확도가 높아진 것을 확인할 수 있다.

다음으로는 앞의 모델과 기본 설정은 같게 하고, 5-way 2-shot으로 학습을 수행한 모델이다. 아래 그림 6은 이와 같은 모델의 성능을 나타내는 결과이다.

```
run_maml(n_way=5, k_shot=2, inner_update_lr=0.4, num_inner_updates=1, meta_train=False, logdir='./log/model/')
Restoring model weights from ./log/model/cis_5_abs_10_k_shot_2_inner_numstep_1_inner_update_lr_0.4_learn_inner_update_lr_False#model.4999
Mean meta-test accuracy/loss, stddev, and confidence intervals
(0.23583333, 0.099495254, 0.007961278438101863)
```

그림 6의 결과를 보면 이번 모델은 5-way 2-shot으로 진행했는데 meta-test의 성능이 0.23583333으로 약 24%의 정확도를 가진 모델로 평가할 수 있다. 두 결과를 표로 정리하면 아래와 같다.

[표 2]

	num_inner_updates = 1 inner_update_lr=0.4 meta_train_iterations=5000	
n-way k-shot	5-way 1-shot	5-way 2-shot
evaluation	0.41200003	0.23583333

위의 표에서 알 수 있듯이 5-way 1-shot을 수행했을 때보다 5-way 2-shot의 모델이 성능이 더 좋지 않았다. 그 이유는 사용한 데이터 셋 때문이다. 이번 실험에서 사용한 동물의 실루엣 이미지는 같은 클래스 내의 이미지들이라도 약간의 차이가 존재했다. 그 차이는 동물들의 크기가 달랐고, 같은 동물이라도 취하고 있는 자세가 달랐다. 따라서 5-way 2-shot으로 모델을 학습했을 때 성능이 더 좋지 않은 것으로 생각된다. 결국, 5-way 1-shot에서 학습을 많이 수행할수록 정확도가 좋아진 것처럼 메타 러닝을 적용해 적은 데이터로 학습이 가능하다. 하지만 한계가 존재하는데 이는 데이터 셋에 제한 사항을 두어야 하는 것이다. 인풋으로 들어오는 이미지는 같은 클래스의 경우 물체의 크기가 비슷해야 하며, 만약 물체가 생물이라면, 해당 생물의 포즈도 유사해야 한다. 이러한 제약 사항을 둔다면 충분히 이미지에서 오브젝트를 분류하는 모델을 적은 데이터로 메타러닝을 적용하여 학습할 수 있을 것이다.

추가로 n-way k-shot 말고도 다른 설정을 변경하면 학습시간이 늘어나더라도 모델의 성능을 더 좋아지게 할 수 있다. 아래 그림 6은 5-way 1-shot에서 다른 설정은 앞서 실험 방법과 같게 하고 학습 계수만 1로 증가시킨 것이다.

```
run_maml(n_way=5, k_shot=1, inner_update_lr=1, num_inner_updates=1, meta_train=False, logdir='./log/model/')
Restoring model weights from ./log/model/cis_5_abs_10_k_shot_1_inner_numstep_1_inner_update_lr_1_learn_inner_update_lr_False#model.4999
Mean meta-test accuracy/loss, stddev, and confidence intervals
(0.4296667, 0.13995199, 0.011198491414603703)
```

[그림 6]

그림 6의 결과와 같이 학습 계수를 증가 시켰더니 meta-test의 성능이 0.4296667였다. 이는 약 43% 정도로 기존의 학습 계수를 0.4로 설정했을 때보다 성능이 좋아진 것을 확인할 수 있다. 이처럼 학습 계수나 학습 반복 횟수 등을 적절하게 설정하면 더 좋은 모델을 얻을 수 있을 것이다.

3. 결 론

해당 문서의 실험에서는 이미지의 오브젝트 분류에 대한 메타러닝 연구를 수행하였다. 결과적으로 적은 데이터를 사용해 메타러닝을 수행했을 때 실제로 학습을 하며 정확도가 증가하는 것을 확인할 수 있었다. 하지만 모델에 인풋으로 들어가는 이미지의 데이터 셋은 물체의 크기, 상태에 따라 모양이 크게 다를 수 있어서 제한 사항이 필요했다. 이러한 제한 사항을 올바르게 설정하고 이에 맞는 데이터 셋을 준비하여 모델을 학습한다면 충분히 좋은 결과를 얻을 수 있을 것이고, 여기에 더해서 학습 계수 등을 적절하게 조절하여 모델의 성능을 더 높일 수도 있다. 이러한 연구를 통해 이미지에서 오브젝트 분류를 수행해 보았고, 이 결과는 오차와 사용한 데이터 셋의 문제로 완벽한 성과는 얻지 못했지만 더 발전시킨다면 90% 이상의 성능을 가지는 모델도 학습이 가능하다고 생각한다.

서론에서 설명한 것처럼 이미지에서 오브젝트를 분류하

는 것은 사람의 자세에 대한 데이터 셋을 사용하거나 손의 모양에 대한 데이터 셋을 사용하여 장애인이나 운동 관련 서비스에서도 활용이 가능할 것이고, 이외에도 많은 분야에서 사용이 가능할 것이다.

[참고 문헌]

[1] Chelsea Finn, Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, p3, 18 Jul 2017

[2] Chelsea Finn, Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, p3, 18 Jul 2017

[3] Chelsea Finn, Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, p3, 18 Jul 2017

[2] Dataset - [animaldataset - xiangbai \(google.com\)](#)