ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
j.inst.Korean.electr.electron.eng.Vol.27,No.4,537~547,December 2023
논문번호 23-04-24
https : //doi.org/10.7471/ikeee.2023.27.4.537
170

이미지 분석을 위한 퓨샷 학습의 최신 연구동향

Recent advances in few-shot learning for image domain: a survey

석 호 식*★

Ho-Sik Seok*★

Abstract

In many domains, lack of data inhibits adoption of advanced machine learning models. Recently, Few-Shot Learning (FSL) has been actively studied to tackle this problem. Utilizing prior knowledge obtained through observations on related domains, FSL achieved significant performance with only a few samples. In this paper, we present a survey on FSL in terms of data augmentation, embedding and metric learning, and meta-learning. In addition to interesting researches, we also introduce major benchmark datasets. FSL is widely adopted in various domains, but we focus on image analysis in this paper.

요 약

퓨샷학습(few-shot learning)은 사전에 확보한 관련 지식과 소규모의 학습데이터를 이용하여 학습데이터의 부족으로 인한 어려움을 해결할 수 있는 가능성을 제시해주어 최근 많은 주목을 받고 있다. 본 논문에서는 퓨샷학습의 개념과 주요 접근방법을 빠르게 파악할 수 있도록 데이터 증강, 임베딩과 측도학습, 메타학습의 세 관점에서 최신연구동향을 설명한다. 또한 퓨샷학습을 적용하려는 연구자들에게 도움을 제공할 수 있도록 주요 벤치마크 데이터셋에 대하여 간략하게 소개하였다. 퓨샷학습은 이미지 분석과 자연어 처리 등 다양한 분야에서 활용되고 있으나, 본 논문은 이미지 처리를 위한 퓨샷학습의 접근법에 집중하였다.

Key words : few-shot learning, one-shot learning, zero-shot learning, machine learning, image classification.

Ⅰ. 서론

이미지나 자연어 처리 분야에서 기계학습 모델들이 달성한 놀라운 성능향상에는 불과 얼마전까지 상상할 수 없었던 대규모 데이터의 확보가 크게 기여하였다. 다양한 데이터셋들이 소개되면서 분야에 따라 기계학습 연구를 위한 데이터셋의 확보에 어려움이 없는 것처럼 느껴지기도 한다. 하지만, 아직도 많은 분야에서 모델의 학습을 위한 학습데이터가 부족하여 기계학습 모델의 연구 및 응용을 저해하고 있다. 데이터가 확보되지 않았거나 대규모 데이터의 확보가 어려운 경우 대규모 학습 데이터에 기반한 접근법은 적용할 수 없다. 또한 인간은 아주 작은 규모의 예제만을 가지고 새로운 임무를 수행할 수 있는 반면 작동 기제를 모르는 경우 대규모의 학습

* Assistant Professor, Dept. of Artificial Intelligence and Data Science, Korea Military Academy
★ Corresponding author
E-mail : hosik.seok@gmail.com, Tel : +82-2-2197-2873
※ Acknowledgment
This study was supported by research fund of Korea Military Academy, (Future Strategy and Technology Research Institute). (RN: 23-AI-03).
Manuscript received, Sep. 25, 2023; revised, Oct. 25, 2023; accepted, Nov. 14, 2023.

데이터가 제공되지 않을 경우 만족할 성능을 달성하기 어렵다.

이런 상황에서 적용할 수 있는 접근법이 퓨샷 학습(few-shot learning)이다[1]. 퓨샷 학습은 유사한 도메인의 훈련 과정에서 확보한 사전지식과 소규모의 학습데이터를 결합하여 데이터 부족의 어려움을 극복할 가능성을 제공해주고 있다. 퓨샷 학습이라고 총칭하나, 세부적으로는 부족한 데이터의 규모를 키우기 위한 데이터 증강부터 학습을 학습하고자 하는 메타학습까지 다양한 접근법이 시도되고 있다. 특히 딥뉴럴넷 모델의 발전으로 주어진 태스크를 달성하기에 적절한 표현을 확보하게 되면서 주어진 태스크와는 관련이 적으나 상대적으로 충분한 데이터를 활용하려는 접근법이 활발하게 시도되고 있다. 본 논문에서는 특히 이미지 분야에서 퓨샷 학습의 최근 연구 경향을 데이터측면, 임베딩 및 측도학습, 메타학습의 관점에서 요약하여 정리하고자 한다.

논문의 구성은 다음과 같다. 2장에서 퓨샷학습을 정의하고 주요 접근법을 소개한다. 3장에서는 퓨샷학습에서 사용하는 벤치마크 데이터를 간단하게 소개하고 4장에서 소개한 접근법의 의미에 대하여 요약한다.

II. 주요 접근법

2장에서는 퓨샷 학습의 개념을 살펴본 후 데이터 증강, 임베딩과 측도학습, 그리고 메타학습의 관점에서 퓨샷학습의 주요 연구성과를 소개한다.

1. 문제 정의 및 주요 용어

퓨샷 학습을 다양하게 정의할 수 있지만, 해결하고자 하는 태스크에 대하여 레이블이 부여되는 데이터 인스턴스의 수가 매우 작은 기계학습 문제라고 정리할 수 있다[1]. 퓨샷 학습은 N-웨이-K-샷(N-way-K-shot) 문제로 표현할 수 있는데, 이 문제에는 N개의 카테고리가 존재하며 개별 카테고리에는 K개의 데이터 인스턴스가 제공된다. 이때 $N\times K$개의 데이터 인스턴스가 훈련에 사용되는 서포트 집합(support set)을 구성한다. 퓨샷 모델의 성능을 측정하는 테스트 집합과 서프토 집합은 레이블을 공유하는데, 두 집합과 다른 레이블에 해당하는 훈련 데이터가 존재한다.

N-웨이-K-샷과 함께 이해해야 하는 또 다른 키워드는 에피소드 학습(episodic learning)이다[2]. 각 에피소드 배치 $B_E=\{S, Q\}$는 먼저 레이블 L로부터 레이블 부분집합을 샘플링한 후 L에 속한 레이블을 갖는 인스턴스를 샘플링하여 서포트집합 S와 질의 집합(query set) Q를 생성한다. 이 때 $|L|$이 N-웨이-K-샷에서 N에 해당하고 S 중 레이블이 k인 부분집합을 S_k라고 할 때 $|S_k|$가 K에 해당한다.

메타학습에서는 내부루프(inner loop)와 외부루프(outer loop)의 개념이 사용된다. 내부 루프에서는 태스크에 한정된 모델을 학습하며 외부 루프에서는 메타 파라미터를 학습한다[3].

2. 데이터 측면

대규모 공개데이터가 갖추어진 일부 분야를 제외하면 많은 분야에서 학습데이터 부족을 경험하고 있다. 딥러닝을 위한 이미지 데이터 증강에 대한 조사 논문[4]에 잘 정리되어 있듯, 데이터 증강은 원래의 데이터 집합으로부터 더 많은 정보를 추출하는 것이 가능하다는 전제하에 진행된다. 2장에서는 데이터 와핑(data warping)과 오버샘플링(oversampling) 측면에서 퓨샷 학습에 적용할 수 있는 기법들을 소개한다.

가. 데이터 와핑

데이터 와핑을 이용하면 현재 주어진 이미지 데이터를 변환하여 데이터 집합을 확장한다. 변환 과정에서 사용하는 기술은 기하학적 변환(geometric transformation), 색상변환(color transformation), 무작위삭제(random erasing), 적대적 학습(adversarial learning), 뉴럴 스타일전이(neural style transfer) 등이 있다. [5]에서는 믹스업(mixup)[6]과 회전변환을 적용한 데이터증강기법이 주어진 태스크에서 가장 효과가 좋았음을 확인하였다. [6]에서 활용한 믹스업은 [7]에서 소개되었는데, 기본적으로 입력 벡터 $\mathbf{x}_i$와 $\mathbf{x}_j$, 그리고 이에 해당하는 원-핫 벡터로 표현된 레이블 $\mathbf{y}_i$, $\mathbf{y}_j$에 대하여 $\lambda\mathbf{x}_i+(1-\lambda)\mathbf{x}_j$, $\lambda\mathbf{y}_i+(1-\lambda)\mathbf{y}_j$, $\lambda\in[0,1]$을 통해 데이터를 증강하는 방법이다.

적대적 데이터 증강기법으로 [8]에서는 식(1)을 이용한다.

$$\mathbf{x}_i^k \leftarrow \mathbf{x}_i^k+\eta\nabla_{\mathbf{x}_i^k}(L_{\mathrm{CE}}(\theta;\mathbf{x}_i^k,y_i)+\beta h(\theta;\mathbf{x}_i^k) - \gamma c_\theta((\mathbf{x}_i^k,y_i),(\mathbf{x}_i,y_i))) \quad (1)$$

식 (1)에서 L_{CE}는 크로스-엔트로피 손실, $h(\theta;\mathbf{x}_i^k)$는 엔트로피를 의미한다. c_θ는 다음과 같이 정의되었다: $C_\theta((\mathbf{x}_0,y_0),(\mathbf{x},y))=\|\mathbf{z}_0-\mathbf{z}\|_2^2+\infty \cdot 1\{y_0\neq y\}$,

$\mathbf{z}=f(\theta_f;\mathbf{x})$. 특히, 주어진 학습데이터 집합에 대하여 학습을 진행하여 확보한 θ를 데이터증강 과정의 크로스 엔트로피 계산에 활용하였음에 주목할 필요가 있다.

MaxUp[9]이라는 적대적 데이터 증강기법에서는 최초에 주어진 학습데이터집합을 구성하는 개별 데이터 인스턴스에 대하여 무작위로 변형된 m개의 데이터 인스턴스 $\{\mathbf{x}_i^{'}\}_{i=1}^{m}$을 생성한다. 그리고 생성된 데이터 집합에 대한 최대 손실을 최소화하는 파라미터를 다음 식에 기반하여 탐색한다: $\min_\theta E_{\mathbf{x}\sim D}\left[\max_{i\in[m]} L(\theta;\mathbf{x}_i^{'})\right]$.

스타일 전이(style transfer)를 이용한 데이터증강도 활발하게 연구되고 있다. 뉴럴 스타일 전이 성과가 축적되면서 손쉽게 스타일을 전이하여 데이터를 증가할 수 있게 되었는데, [10]에서는 모양은 유지한 상태에서 스타일 임베딩 벡터를 변화시켜 데이터를 증강하는 방법을 제안하였다.

CycleGAN (cycle-consistent adversarial network)은 GAN의 불안정성을 보강하여 새로운 데이터를 생성한다[11]. 구체적으로 적대적 손실(adversarial loss)과 사이클 일관성(cycle consistency)을 이용하여 재구성된 도메인의 일관성을 측정하였다. 제안 방법은 위성의 원격관측 이미지에 적용되어 데이터 증강 응용 범위를 확장하였다.

기하학적 변환, 색상변환, 무작위삭제와 같은 전통적인 데이터증각기법들은 구현이 쉽고 효과적으로 데이터집합의 크기를 키울 수 있기 때문에 여전히 널리 활용되고 있다. 믹스업이나 스타일 전이와 같은 최신 방법들은 보다 다양하고 실제같은 데이터를 생성할 수 있다. 그러나 데이터집합의 복잡도에 따라 데이터증가의 효과에 한계가 있으며, 태스크의 성격에 따라 기법별로 성능 향상에 차이가 존재하므로 어떤 증강기법이 가장 뛰어나다고 확정하기는 어려운 상황이다.

나. 오버샘플링

클래스를 구성하는 데이터 인스턴스의 개수가 현저하게 다른 상황을 해결하기 위하여 여러 분야에서 오버샘플링을 활발하게 사용하고 있다[12-15]. 오버샘플링의 대표적인 기법은 SMOTE(synthetic minority over-sampling)이다[16]. SMOTE는 규모가 적은 카테고리의 데이터집합을 구성하는 인스턴스(x)들에 대하여 최근접 이웃 K개를 선정한 후, 불균형 비율에 따라 K개의 이웃 중 N개의 인스턴스를 선정한다. 새로운 데이터 인스턴스는 선정된 N개의 인스턴스와 x에 선형보간법을 적용하여 합성된다.

[17]에서는 단순한 랜덤 오버샘플링만으로도 클래스 불균형 문제 해결에 상당한 효과를 달성할 수 있음을 보고하였다. 하지만 오버샘플링이 초래할 수 있는 과적합의 문제 역시 보고되었음에 주의할 필요가 있다[18].

3. 임베딩과 측도학습

II-3에서는 퓨샷학습의 측면에서 임베딩학습과 측도학습의 주요 연구성과를 살펴본다. 임베딩과 측도학습은 함께 진행되는 경우가 많기 때문에 두 주제를 분리하지 않고 관련된 주요 연구 성과를 함께 정리한다.

임베딩 학습에서는 $\mathbf{x}_i\in\mathrm{R}^D$의 데이터 인스턴스를 저차원의 벡터 $\mathbf{z}_i\in\mathrm{R}^d$ $(d<D)$로 변환한다. 변환 과정에서 유사한 인스턴스는 새로운 표현 공간에서 상대적으로 근접하게, 그렇지 않은 인스턴스는 상대적으로 떨어지도록 학습을 진행하여 임베딩 벡터를 결정한다. 일반적인 분류 문제에서는 이것으로 충분하지만 퓨샷 학습에서는 훈련 데이터집합으로 확보한 임베딩 공간이 서포트 집합에 적용할 수 있는 특성을 보유하기 어렵다. 퓨샷 학습을 위한 임베딩은 어텐션 커널이나 트랜스포메이션 등을 이용하여 서포트 집합의 특성을 반영한 임베딩 벡터를 생성하고자 한다. 측도학습(metric learning)에서는 특정 임무에 맞춰진 거리 함수를 학습한다[19]. 퓨샷학습에 한정하지 않으면 [20-22]와 같은 선행 연구를 통해 측도학습의 주요 연구 경향을 파악할 수 있다. 측도학습은 임베딩 벡터에 KNN 분류기처럼 상대적으로 간단한 분류 모델을 이용하여 태스크를 해결하고자 한다. 특히 평균 벡터등을 이용하여 프로토타입을 표시할 경우 재훈련 없이도 거리 계산을 통해 분류가 가능하므로 마진(margin)을 최대화하거나 동일한 레이블의 인스턴스가 모이도록 유도하는 측도의 학습은 퓨샷 학습에 큰 도움이 된다.

Matching Net에서는 예측에 대한 확률분포를 어텐션 커널(attention kernel)로 계산하였다. Matching Net에서는 서포트 집합에 속하는 인스턴스를 i번째 인스턴스 $\mathbf{x}_i$의 임베딩 벡터를 획득하는 과정에 활용하였다[23]. 임베딩 과정에서 임베딩 함수 g는 $\mathbf{x}_i$와 전체 서포트 집합 S를 입력으로 받는데 구체적으로 LSTM을 인코딩 과정에 활용하였다.

RN(relation network)은 임베딩 모듈(f_ψ)과 관계 모듈(g_ϕ)로 구성되어 있다. RN은 에피소드 기반의 훈련 방식을 취하기 때문에 C개의 클래스 별로 무작위로 선정된 K개의 인스턴스로 구성된 샘플 셋과 잔여 인스턴스

로 구성된 질의 집합을 이용하여 학습을 진행한다. 관계 스코어(relation score, r_{ij})는 식 2를 이용하여 계산되는데, r_{ij}의 계산에 사용되는 각종 정보는 다음과 같다:

$$r_{ij} = g_\phi(C(f_\psi(\mathbf{x}_i), f_\psi(\mathbf{x}_j))) \quad (2)$$

f_ψ는 임베딩 모듈로 특징맵을 생성하는 역할을 한다. g_ϕ는 관계 모듈로 특징맵을 입력으로 받아 관계 스코어를 출력한다. 식2에서 C는 두 임베딩 벡터를 중첩(concatenation)하는 연산이다. 퓨샷 학습을 위한 임베딩 모듈은 컨블루션 블록이 사용되었으며, 관계 모듈은 컨볼루션 블록과 FC층으로 구성되었다[24].

FEAT(few-shot embedding adaptation w/transformer)는 임베딩 함수를 거쳐 산출된 임베딩 벡터에 적응과정을 추가하였다. 적응과정은 Transformer나 GCN(graph convolutional networks)등 다양한 방식으로 구현할 수 있는데, 서포트 집합과 별도로 주어진 데이터를 이용하여 임베딩 함수와 셋 함수(set function)를 구한다. 그 후 획득된 임베딩 함수와 셋 함수에 서포트 집합을 적용하는데, 서포트 집합의 데이터에서 획득된 클래스 중심점과 적응된 임베딩 벡터의 거리를 손실함수에 반영하였다[25].

대조 학습(contrastive learning)의 접근법을 사용하여 질의 인스턴스와 동일한 레이블의 서포트 인스턴스와 다른 레이블의 서포트 인스턴스를 선정한 후 선정된 인스턴스들의 내적결과를 손실함수에 적용하여 임베딩 학습을 진행하는 방법도 보고되었다[26]. 대조 학습을 적용한 또 다른 접근법으로 데이터 증강과 임베딩을 결합하여 새로운 임베딩 벡터를 생성한 후 클래스의 프로토타입($\tilde{\mathbf{p}}_k = \frac{1}{k}\sum_{(\mathbf{x}_i, y_i)} f_\phi(\mathbf{x}_i) \cdot I(y_i = k)$)과 거리를 계산하는 과정에 질의와 동일한 레이블의 인스턴스 및 다른 레이블의 인스턴스와 프로토타입과의 거리를 반영하는 접근법도 제안되었다[27].

매니폴드 스무딩(manifold smoothing)을 적용하면 학습데이터 집합과 테스트데이터 집합에 존재하는 분포의 차이에 어느 정도 대응할 수 있는 것으로 알려져 있다. EP(embedding propagation) 방법에서는 인접행렬(adjacency matrix, A)에 대한 라플라시안 $L = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}(D_{ii} = \sum_j A_{ij})$를 계산한 후, 전파 행렬(propagation matrix) $P = (I - \alpha L)^{-1}$을 구하여 최종 임베딩 벡터를 산출한다[28].

대표적인 측도 학습으로 Mahalanobis 측도(식 3) 학습을 제시할 수 있다(식 3에서 Σ는 공분산 행렬을 의미).

$$d(\mathbf{x}, \boldsymbol{\mu}) = \sqrt{(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})} \text{ 혹은}$$
$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i-\mathbf{x}_j)^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}_i-\mathbf{x}_j)} \quad (3)$$

Mahalanobis 측도를 학습하려는 접근법들은 클러스터링에서 데이터 페어링을 반영하거나[29], 각 인스턴스의 잡음 혹은 교란 상태를 반영하여 거리를 계산하려고 시도한다[30]. [31]에서는 Mahalanobis 거리 학습 과정에서 학습데이터 집합에 대한 LOO(leave-one-out) 분류 오류에 대한 기댓값을 최소화하고자 하였는데, 식 3와 같이 거리측도를 표현할 때 정확하게 분류될 데이터 포인트의 기대값(식 4)을 최대화하여 거리 측도를 학습하고자 시도하였다.

$$f(A) = \sum_i \sum_{j \in C_i} p_{ij} \quad (4)$$

여기서, C_i와 p_{ij}는 $C_i = \{j | c_i = c_j\}$, $p_{ij} = \frac{\exp(-\|Ax_i - Ax_j\|^2)}{\sum_{k \neq i}\exp(-\|Ax_i - Ax_k\|^2)}$으로 정의되었으며 KNN(K-nearest neighbor) 분류기가 사용되었다.

[32]에서는 prototypical network를 제안하여 분류 문제에서 퓨샷 학습을 해결하고자 하였다. 제안 방법에서 프로토타입은 [27]에서와 같이 동일 클래스에 속한 임베딩 서포트 포인트(embedded support point)의 평균 벡터로 정의된다. 일단 프로토타입이 정의되면 임베딩 공간에서 프트로타입과 질의 포인트 $\mathbf{x}$와의 거리에 대한 소프트맥스 값을 이용하여 확률 분포를 계산할 수 있다(식 5).

$$p(y = k|\mathbf{x}) = \frac{\exp(-d(f_\phi(\mathbf{x}), \mathbf{x}_k))}{\sum_m \exp(-d(f_\phi(\mathbf{x}), \mathbf{c}_m))} \quad (5)$$

식 5에서 $\mathbf{c}_k$는 프로토타입을, f_ϕ는 임베딩 함수를 의미한다. 학습목표는 $J(\phi) = -\log p_\phi(y = k|\mathbf{x})$의 최소화인데 식 5와 혼합분포에서의 클러스터 할당 과정과 비교해보면, 혼합분포에서 컴퍼넌트 가중치가 동일한 경우를 prototypical network가 표현하고 있다는 것을 알 수 있다.

CNAPS(conditional neural adaptive processes)와 Mahalanobis 측도를 결합하는 방법도 제안되었다[33].

제안 방법은 서포트 집합에서 클래스 레이블이 k인 인스턴스들을 S_k^T라고 했을 때, 식 (6)과 서포트 집합에 존재하는 모든 인스턴스의 공분산행렬인 Σ^T를 이용한 것이다(여기서 $f_\theta^T(\mathbf{x}_i)$는 $\mathbf{x}_i$의 임베딩 벡터).

$$\Sigma_k^T = \frac{\sum_{(\mathbf{x}_i, y_i) \in S_k^T} (f_\theta^T(\mathbf{x}_i) - \boldsymbol{\mu}_k)(f_\theta^T(\mathbf{x}_i) - \boldsymbol{\mu}_k)^{\mathrm{T}}}{|S_k^T| - 1} \tag{6}$$

Mahalanobis 측도의 공분산 행렬은 Σ^T와 Σ_k^T를 이용하여 $\lambda_k^T \Sigma_k^T + (1 - \lambda_k^T)\Sigma^T + \beta I$로 추정된다.

Mahalanobis 측도 외에 다른 측도들도 활발히 활용되고 있다. CovaMNet(covariance metric networks)은 분포의 일관성을 측정하기 위한 공분산측도와 지역 공분산표현(local covariance representation)을 핵심 구성요소로 한다[34]. 지역공분산표현을 이용하여 분포를 표현하고 공분산측도를 이용하여 분포의 일관성을 측정하였다. 문제는 퓨샷 학습의 경우 소수의 인스턴스만을 이용하여 분포 정보를 표현해야 한다는 점이다. 이 상황을 해결하기 위하여 개별 카테고리(c로 표현)의 지역공분산표현(Σ_c^{local})을 식7과 같이 정의하였다(여기서 $D_c = \{\mathbf{x}_1, \ldots, \mathbf{x}_{\mathrm{K}}\}$는 c번째 카테고리에 속하는 K개의 이미지집합, $\mathbf{x}_i \in R^{d \times M}$, $\boldsymbol{\tau}$는 평균벡터의 행렬을 의미).

$$\Sigma_c^{local} = \frac{1}{MK - 1} \sum_{i=1}^{K} (\mathbf{x}_i - \boldsymbol{\tau})(\mathbf{x}_i - \boldsymbol{\tau})^{\mathrm{T}} \tag{7}$$

공분산측도는 $f(\mathbf{x}, \Sigma) = \mathbf{x}^{\mathrm{T}} \Sigma \mathbf{x}$로 정의하였다. 지역 공분산표현과 공분산측도를 결합하면 임의의 질의벡터 $\mathbf{x}$와 Σ_c^{local}사이의 유사도는 다음의 식으로 계산된다: $\mathrm{dig}\, f(\mathbf{x}^{\mathrm{T}} \Sigma_c^{local} \mathbf{x})$.

퓨샷 학습에 존재하는 태스크 간의 차별성과 측도 학습을 결합한 접근법도 존재한다[35]. 제안 방법에서는 태스크에 특화된 측도 활용을 위해 측도 M_t를 이용하여 식8을 정의하였다.

$$d_{M_t}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{tr < M_t(\mathbf{x}_i, \mathbf{x}_j)(\mathbf{x}_i, \mathbf{x}_j)^{\mathrm{T}} >} \tag{8}$$

M_t는 같은 카테고리에 속한 인스턴스의 거리는 최소화하고, 다른 카테고리에 속한 인스턴스의 거리는 상대적으로 키우는 것을 목적으로 하여 획득되는데, 최종 측도는 식 9로 정리된다.

$$M_0^{-1} + \gamma \cdot \widetilde{M} - \gamma\lambda \cdot \widetilde{C} + \alpha \cdot \Sigma_t \tag{9}$$

식 9에서 $\widetilde{M} = \frac{1}{M} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathrm{M}} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}$, $\widetilde{C} = \frac{1}{C} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathrm{C}} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}$, M과 C는 분류 작업에서는 동일 클래스 및 다른 클래스를 의미하고 학습 과정에서 제약 조건을 부여하는 역할을 하며, M_0는 정규화(regularization) 과정에서 사용하는 주어진 측도, Σ_t는 공분산 행렬을 의미한다.

4. 메타학습

메타학습(meta learning)은 다수의 학습 에피소드에서의 학습 경험을 활용하여 학습 성능을 높이려는 접근법으로 "학습을 위한 학습(learning-to-learn)"이라는 표현으로 특징을 요약할 수 있다[36]. 많은 기계학습 모델에서의 학습 과정을 식 10과 같이 표현할 수 있다.

$$\theta_t \leftarrow \theta_{t-1} - \alpha_t \nabla_{\theta_{t-1}} l(h(x_{t;}\theta_{t-1}), y_t) \tag{10}$$

메타학습에서는 모델 파라미터 θ의 초기값이나 옵티마이저를 학습하는데, 두 요소 외에도 모델의 출력값 $\hat{y} = f_\theta(\mathbf{x})$를 계산하는 과정에서 선택하는 각종 가정(임베딩 네트워크의 학습, 옵티마이저의 선택 등)들을 학습 대상으로 간주한다. 특히 딥러닝에 기반한 메타학습의 경우 메타학습은 태스크에 따라 모델의 파라미터를 다시 학습할 필요없이 이미 확보된 모델 파라미터를 재활용하여 태스크의 변화에 적응하는 접근법으로 해석할 수도 있다. 4장에서는 메타학습의 주요 연구성과를 모델 파라미터의 학습과 옵티마이저의 학습 측면에서 살펴본다.

가. 파라미터 학습

MAML(model-agnostic meta-learning)은 경사하강법을 몇 번 실행하지 않고도 높은 성능을 발휘할 수 있도록 모델의 초기 파라미터를 훈련하는 방법이다[37]. 태스크 T_i을 위한 모델의 파라미터를 $\theta_i^{'}$, 모델 f_θ를 θ에 의해 특정되는 모델이라고 정의하자. 이 때 식 11을 이용하여 $\theta_i^{'}$를 갱신한다.

$$\theta_i^{'} \leftarrow \theta - \alpha \nabla_\theta L_{T_i}(f_\theta) \tag{11}$$

퓨샷 학습에서는 복수 개의 태스크가 주어지므로 태스크별로 식 11을 계산하여 $\theta_i^{'}$를 확보한다. 이제 θ를 갱신하기 위한 손실 함수($L_{T_i}(f_{\theta_i^{'}})$)를 다음과 정의할 수 있다.

$$L_{T_i}(f_{\theta_i^{'}}) = L_{T_i}(f_{\theta - \alpha \nabla_\theta L_{T_i}(f_\theta)}) \tag{12}$$

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{T_i \sim p(T)} L_{T_i}(f_{\theta_i'}) \tag{13}$$

모델의 파라미터는 식 12와 식 13을 이용하여 갱신된다. MAML은 다양한 태스크의 파라미터 벡터를 결합하여 태스크 파라미터의 초기값을 결정하나 임베딩 공간에 대한 고려가 부족하다.

LEO(latent embedding optimization)는 메타학습을 저차원의 임베딩 공간에서 진행하기 위하여 제안된 방법이다[38]. LEO는 데이터인스턴스를 인코더와 RN(relation network)를 거쳐 은닉 코드(latent code)로 변환한다. 은닉코드는 다시 디코더를 거쳐 파라미터로 변환되는데 인코딩과 디코딩은 각각 식 14와 식15로 정리할 수 있다.

$$\begin{aligned} &\boldsymbol{\mu}_n^e, \boldsymbol{\sigma}_{\mathrm{n}}^{\mathrm{e}} = \frac{1}{\mathrm{NK}^2} \sum_{\mathrm{k_n}=1}^{\mathrm{K}} \sum_{\mathrm{m}=1}^{\mathrm{N}} \sum_{\mathrm{k_m}=1}^{\mathrm{K}} \mathrm{g}_{\phi_r}\left(\mathrm{g}_{\phi_e}(\mathbf{x}_{\mathrm{n}}^{\mathrm{k_n}}), \mathrm{g}_{\phi_e}(\mathbf{x}_{\mathrm{m}}^{\mathrm{k_m}})\right), \\ &\mathbf{z}_n \sim q(\mathbf{z}_n | D_n^{tr}) = N(\boldsymbol{\mu}_n^e, \mathrm{diag}(\boldsymbol{\sigma}_n^{e^2})) \end{aligned} \tag{14}$$

식 14에서 g_{ϕ_e}와 g_{ϕ_r}은 각각 인코드와 RN을 의미한다. 식 14에서 획득한 은닉코드는 디코더 g_{ϕ_d}에 입력으로 주어져서 파라미터로 표현된다.

$$\begin{aligned} &\boldsymbol{\mu}_n^d, \sigma_n^d = g_{\phi_d}(\mathbf{z}_n), \\ &\mathbf{w}_n \sim p(\mathbf{w}|\mathbf{z}_n) = N(\boldsymbol{\mu}_n^d, \mathrm{diag}(\boldsymbol{\sigma}_n^{d^2})) \end{aligned} \tag{15}$$

LEO는 식 15에서 확보한 $\mathbf{w}_{\mathrm{n}}$을 이용하여 식 11을 계산하는데 이 때 $L_{T_i}(f_{\theta_i})$는 $D^{tr}(D_{tr} = \{(\mathbf{x}_{\mathrm{n}}^{\mathrm{k}}, \mathrm{y}_{\mathrm{n}}^{\mathrm{k}}) : \mathrm{k} = 1, \ldots K, n = 1, \ldots, N\})$를 이용하여 $\sum_{(\mathbf{x},y) \in D^{tr}} \left[-\mathbf{w}_y \cdot \mathbf{x} + \log\left(\sum_{j=1}^{N} e^{\mathbf{w}_j \cdot \mathbf{x}} \right) \right]$로 계산된다. 인코더, RN, 디코더의 파라미터 ϕ_e, ϕ_r, ϕ_d를 획득하기 위한 손실함수는 식 16과 같이 설정된다.

$$\begin{aligned} &\sum_{T_i \sim p(T)} [L_{T_i}^{val}(f_{\theta_i'}) + \beta D_{\mathrm{KL}}(\mathrm{q}(\mathbf{z}_n | D_n^{tr}) \| p(\mathbf{z}_n)) \\ &+ \gamma \| \mathrm{stopgrad}(\mathbf{z}_n') - \mathbf{z}_n \|_2^2] + R \end{aligned} \tag{16}$$

식 16에서 R은 정규화에 해당하며 상관행렬 C_d에 대해 $\lambda_1(\|\phi_e\|_2^2 + \|\phi_r\|_2^2 + \|\phi_d\|_2^2) + \lambda_2 \| C_d - I \|_2$로 계산된다.

HSML(hierarchically structured meta-learning)은 계층적인 태스크 클러스터링(hierarchical task clustering)을 활용하는 방법이다[39]. 특정 계층 레벨에 속할 확률을 계산하여 레벨 할당이 이루어지는데, 확률값은 계층 레벨(l)에서의 테스크(T_i)의 표현($\mathbf{h}_i^{k^l}$)과 클러스터 센터($\mathbf{c}_{k^{l+1}}$)를 이용하여 계산된다($\|\mathbf{h}_i^{k^l} - \mathbf{c}_{k^{l+1}}\|_2^2$). $\mathbf{h}_i^{k^{l+1}}$은 할당 후 갱신되는데 할당 확률($P_i^{k^l \to k^{l+1}}$)을 반영하여 식 17의 방식으로 표현을 갱신한다.

$$\mathbf{h}_i^{k^{l+1}} = \sum_{k^k=1}^{K^l} p_i^{k^l \to k^{l+1}} \tanh(\mathbf{W}^{k^{l+1}} \mathbf{h}_i^{k^l} + \mathbf{b}^{k^{l+1}}) \tag{17}$$

$\mathbf{g}_i$(T_i에 속한 학습데이터 인스턴스 표현의 평균, $\mathbf{g}_i = \frac{1}{n^{tr}} \sum_{j}^{n^{tr}} \mathbf{g}_{i,j}$)와 $\mathbf{h}_i^L$(계층적 클러스터링 결과 표현)은 파라미터 게이트 $\mathbf{o}_i = \mathrm{FC}_{\mathbf{W}_g}^{\sigma}(\mathbf{g}_i \oplus \mathbf{h}_i^L)$의 계산 과정에 활용되는데, 파라미터 게이트 $\mathbf{o}_i$가 식 12의 계산 과정에서 손실함수 계산을 위한 파라미터로 활용된다(식 18).

$$f_{\theta_{\mathbf{o}_i} - \alpha \nabla_{\theta} L(\theta, D_{T_i}^{tr})} \tag{18}$$

[40]에서는 태스크 그룹핑에 대한 흥미로운 접근법을 소개하였다. 제안 방법은 태스크간 친화도(inter-task affinity)를 활용하는데 태스크 j가 주어졌을 때 타임스텝 t에서의 태스크간 친화도는 식 19를 이용하여 계산한다.

$$Z_{i \to j}^t = 1 - \frac{L_j(\chi^t, \theta_{s|i}^{t+1}, \theta_j^t)}{L_j(\chi^t, \theta_s^t, \theta_j^t)} \tag{19}$$

식 19에서 χ는 데이터 인스턴스 집합을 의미하는데 i번째 태스크와 j번째 태스크에 대한 손실 함수를 이용하여 태스크간 친화도를 계산함을 알 수 있다. 여기서 $\theta_{s|i}^{t+1}$이 $\theta_s^t - \eta \nabla_{\theta_s^t} L_i(\chi^t, \theta_s^t, \theta_i^t)$를 의미함에 주의해야 하는데, 결국 식 19는 공유 인자 θ_s에 대한 갱신으로 손실을 낮아지는 상황이 발생하는 태스크 그룹핑을 의도하고 있다.

많은 딥러닝 구조들은 특정 변환(transformation)과 등변량(equivariant) 관계에 있다. [41]에서는 데이터로부터 등변량관계를 학습하는 방법을 제안하였다. FC층은 $\phi(\mathbf{x}) = \mathbf{W}\mathbf{x}$의 식으로 표현할 수 있다. 여기서 $\mathbf{W}$를 대칭행렬 $\mathbf{U}$와 필터 파라미터 벡터 $\mathbf{v}$의 곱으로 분해할 수 있는데, $\mathbf{U}$는 태스크들이 공유하는 엔터티로 변하지 않는 구조를 표현하고 $\mathbf{v}$만이 변하는 개체라고 설정할 경

우, 내부루프에서 $\mathbf{v}$를 갱신하고, 외부루프에서 $\mathbf{U}$를 갱신하는 학습 과정을 설정할 수 있다(식 20).

$$\mathbf{v}' \leftarrow \mathbf{v} - \alpha \nabla_{\mathbf{v}} L(\mathbf{U}, \mathbf{v}, D_i^{tr}),$$
$$\mathbf{U} \leftarrow \mathbf{U} - \eta \frac{\mathrm{d}}{\mathrm{d}\mathbf{U}} L(\mathbf{U}, \mathbf{v}', D_i^{val}) \quad (20)$$

[42]에서 등변량관계를 기존 모델에 추가할 잠재력이 있는 방법을 제안하였으며 [43]에서는 사전학습된 모델의 출력을 평균내는 것보다는 가중합을 활용하는 것이 퓨샷학습에 더욱 도움이 된다는 것을 보고하였다.

파라미터가 아니라 주어진 학습데이터로부터 유니버셜 템플릿을 생성하여 학습에 활용하려는 접근법도 소개되었다[44]. 여기서 유니버셜 템플릿은 특징 추출기의 배열을 표현할 수 있는 구조로 구체적으로 컨볼류션 계층의 파라미터로 정의되었다. 제안 방법에서는 학습 대상으로 컨볼루션 계층의 파라미터 Ψ와 Φ 그리고 FiLM (feature-wise linear modulation)[45] 인자 Ψ_d(Ψ의 행벡터이자 학습데이터 집합 d에 대한 FiLM인자)를 설정한다. 주어진 학습데이터에서 획득한 FiLM 파라미터 $\{\Psi_m\}_{m=1}^{M}$로부터 새로운 데이터셋(d^*)에 대한 초기 인자 $\Psi_{d^*}^{\text{init}}$는 $\mathrm{softmax}(l(g(S_T)))^{\mathrm{T}}\Psi$로 획득된다(여기서 g는 입력으로 주어진 인스턴스 집합의 벡터 표현을 생성하는 함수, S_T는 서포트 집합을 의미).

나. 옵티마이저 학습

초기값 학습에서 소개한 다양한 방법들은 파라미터에 주목하였는데, 학습 과정에 주목한 접근법들도 존재한다.

경사하강법을 $\theta_{t+1} \leftarrow \theta_t + g_t(\nabla f(\theta_t), \phi)$라고 표시하면, ϕ를 파라미터로 하는 옵티마이저 g_t를 이용한 갱신 과정으로 해석할 수 있다. [46]에서는 g_t를 RNN의 출력으로 설정하였는데, g_t의 도입과 함께 ϕ에 의한 손실 함수를 식 21과 같이 정의하였다(여기서, $\nabla_t = \nabla_\theta f(\theta_t)$, h_t는 스테이트를 의미).

$$L(\phi) = E_f\left[\sum_{t=1}^{T} w_t f(\theta_t)\right],$$
$$\theta_{t+1} = \theta_t + g_t, \ \begin{bmatrix} g_t \\ h_{t+1} \end{bmatrix} = m(\nabla_t, h_t, \phi) \quad (21)$$

[46]와 유사한 접근법이 "최적화를 위한 학습(learning to optimize)"이다[47]. 최적화를 위한 학습의 학습 과정은 식 22로 요약할 수 있다.

$$\mathbf{x}^i \leftarrow \mathbf{x}^{(i-1)} + \triangle \mathbf{x}, \ \triangle \mathbf{x} \leftarrow \pi(f, \{\mathbf{x}^{(0)}, ..., \mathbf{x}^{(i-1)}\}) \quad (22)$$

학습 방식의 선택에 따라 π의 구체적인 형태가 결정되는데 최적화를 위한 학습에서는 π를 학습의 대상으로 간주한다.

[48]에서는 LSTM에 기반한 메타-러너 모델을 제안하였다. 해당 방법에서 주목한 것은 경사하강법과 LSTM의 셀 상태(cell state) 갱신 방식(식 23)의 유사도이다(식 23에서 $f_t = 1$, $c_{t-1} = \theta_{t-1}$, $i_t = \alpha_t$, $\tilde{c_t} = \nabla_{\theta_{t-1}} L_t$라고 설정한 후 비교하면 두 과정의 유사함을 파악할 수 있다).

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t} \quad (23)$$

식 23을 구성하는 요소 중 i_t와 f_t 그리고 c_0(c_0의 학습은 메타 학습에서 초기값 학습에 해당하므로 자세한 설명은 생략)가 학습대상이 되는데, 학습률과 매칭시켰던 i_t는 이전학습률 i_{t-1}과 현재 로스L_t, 그리고 현재 그레디언트 $\nabla_{\theta_{t-1}} L_t$의 함수로 표현된다. 1로 표현했던 f_t는 포겟게이트(forget gate)로 설정되어 역시 학습의 대상이 된다(식 24).

$$i_t = \sigma(\mathbf{W}_I \cdot [\nabla_{\theta_{t-1}} L_t, L_t, \theta_{t-1}, i_{t-1}] + \mathbf{b}_I),$$
$$f_t = \sigma(\mathbf{W}_F \cdot [\nabla_{\theta_{t-1}} L_t, L_t, \theta_{t-1}, f_{t-1}] + \mathbf{b}_F) \quad (24)$$

옵티마이저 학습과 관련된 다양한 연구에 대해서는 [49]에 잘 정리되어 있다.

II-3, 4에서 정리한 카테고리에는 속하지 않지만 사전학습된 표현 공간에 인과 모델을 추가하는 접근법도 소개되었다. [50]에서는 서포트 집합 S와 질의 집합 Q가 유사하지 않을 경우, 사전훈련된 모델의 활용이 성능 저하를 초래할 수 있음을 발견하였다. IFSL(interventional few-shot learning)은 구조적 인과모델(structural causal model)을 통해 사전훈련된 모델의 출력을 활용하려는 접근법으로 특히 컨파운더(confounder)의 효과를 인과모델로 상쇄하고자 시도하였다.

MAML의 설정은 일련의 태스크가 배치로 존재하는 상황을 가정하기 때문에 순차적으로 태스크가 주어지는 상황에 대처하기 어렵다. 에이전트가 과거의 경험에 기반하여 순차적으로 주어지는 새로운 태스크를 해결하는 과정이 온라인 메타학습(online meta-learning)으로 소개되었다[51]. [51]에서 제안한 FTML (follow the meta leader) 알고리즘에서는 식 25를 이용하여 파라미터를

선택한다.

$$\mathbf{w}_{t+1} = \text{argmin}_{\mathbf{w}} \sum_{k=1}^{t} f_k(U_k(\mathbf{w})) \quad (25)$$

식 25에서 f_k는 k번째 태스크, $U_k(\mathbf{w}) = \mathbf{w} - \alpha \nabla \hat{f}_k(\mathbf{w})$를 의미한다.

Ⅲ. 벤치마크 데이터셋

3장에서는 성능비교를 위하여 이미지 퓨샷 학습에서 주로 사용되는 벤치마크 데이터셋들을 간단하게 소개한다.

CUB-200-2011은 200종의 조류에 대한 이미지로 구성되어 있으며 11,788장의 이미지로 구성되어 있다[52]. 개별 이미지는 서브카테고리 레이블 1개, 15개의 파트 위치, 312개의 이진 속성값과 1개의 바운딩 박스로 구성된 부가 정보를 가지고 있다.

MiniImageNet 데이터셋은 60,000장의 이미지로 구성되어 있으며 600장의 이미지가 제공되는 클래스가 100개 존재한다[23]. MiniImageNet은 잘 알려진 ImageNet 데이터셋의 크기를 축소한 것으로 ImageNet은 20,000개 이상의 카테고리로 구성되어 있으며 1,400만장 이상의 이미지로 구성되어 있다. MiniImageNet은 ImageNet에 비하여 상대적으로 용이하게 분석할 수 있는 크기의 데이터집합이다.

Omniglot은 다양한 언어들의 문자 데이터를 모아 놓은 것으로 50종의 언어로부터 생성된 총 1623개의 클래스에 대한 필기체 데이터로 구성되어 있다[53]. Omniglot 데이터셋은 원-샷 학습을 의도하여 생성된 데이터셋으로 데이터셋뿐만 아니라 분류 및 생성에 대한 Omniglot Challenge도 함께 공개되었다.

Pascal-5i 데이터셋은 5개의 클래스로 구성된 4개의 폴드로 나뉘어져 있다[54]. Pascal-5i 데이터셋은 퓨샷 학습을 세그먼테이션에 적용할 때 많이 활용된다.

Meta-dataset 벤치마크는 ILSVRC-2012, Omniglot, Aircraft, CUB-200-2011, Describable textures, Quick draw, Fungi, VGG Flower, Traffic Signs, MSCOCO의 10개 데이터셋을 모아놓은 대규모 데이터셋이다[55].

Ⅳ. 결론

다양한 관점에서 퓨샷 학습에 접근할 수 있겠으나, 본 리뷰 논문에서는 이미지 도메인을 위한 퓨샷 학습의 최신 연구를 데이터 증강, 임베딩 및 측도 학습 그리고 메타학습 관점에서 소개하였다. 데이터변환 및 오버샘플링과 관련된 내용을 데이터 증강에서 소개하거나, 퓨샷 학습에서 활발하게 사용하는 벤치마크 데이터셋의 특징을 정리하는 등 본 논문에서는 퓨샷 학습을 적용하기 위해 필요한 정보를 주로 제공하였으나 퓨샷 학습 과정에서 활용할 수 있는 사전학습 모델에 대한 내용은 누락하였다. 본 논문에서는 이미지 도메인의 문제를 해결하기 위한 접근법을 주로 소개하였으나 자연어 도메인이나 강화학습에서도 퓨샷 학습이 활발하게 적용되고 있음에 주의할 필요가 있다. 자연어 처리를 위한 접근법은 향후 다른 논문에서 소개하고자 한다.

References

[1] Y. Wang et al., “Generalizing from a Few Examples: A Survey on Few-Shot Learning,” *ACM Computing Survey*, Vol.53, No.3, pp.1-34, 2020. DOI: 10.1145/3386252

[2] S. Laenen and L. Bertinetto, “On Episodes, Prototypical Networks, and Few-Shot Learning,” In *Proc. of 35th Conferenceon Neural Information Processing Systems* (*NeurIPS2021*), 2021. DOI: 10.48550/arXiv.2012.09831

[3] A. Rajeswaran et al., “Meta-Learning with Implicit Gradients,” In *Proc. of 33rd Conference on Neural Information Processing Systems (NeurIPS2019)*, 2019. DOI: 10.48550/arXiv.1909.04630

[4] C. Shorten and T. M. Khoshgoftaar, “A Survey on Image Data Augmentation for Deep Learning,” *J Big Data*, vol.6, Article Number:60, 2019. DOI:10.1186/s40537-019-0197-0

[5] R. Zhang et al., “Multi-Task Few-Shot Learning with Composed Data Augmentation for Image Classification,”*IET Computer Vision*, vol.17, no.2, pp.211-221, 2023. DOI: 10.1049/cvi2.12150

[6] V. Verma et al., “Manifold Mixup: Better Representations by Interpolating Hidden States,” *In Proc. of the Thirty-sixth International Conference on Machine Learning (ICML2019)*, 2019 DOI: 10.48550/arXiv.1806.05236.

[7] H. Zhang et al., “mixup: Beyond Empirical

Risk Minimization," In *Proc. of International Conference Learning Representation (ICLR 2018)*, 2018. DOI: 10.48550/arXiv.1710.09412
[8] L. Zhao et al., "Maximum-Entropy Adversarial Data Augmentation for Improved Generalization and Robustness,"In *Proc. of the 34th Conference on Neural Information Processing Systems (Neur-IPS2020)*, 2020. DOI: 10.48550/arXiv.2010.08001
[9] C. Gong et al., "MaxUp: Lightweight Adversarial Training with Data Augmentation Improves Neural Network Training,", In *Proc. 2021 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR2021*), pp.2474-2483, 2021.
DOI: 10.1109/CVPR46437.2021.00250
[10] P. T. Jackson et al., "Style Augmentation: Data Augmentation via Style Randomization," In *CVPR Workshop 2019*, pp.83-92, 2019.
DOI: 10.48550/arXiv.1809.05375
[11] Y. Jiang, B. Zhu, and B. Xie, "Remote Sensing Images Data Augmentation Based on Style Transfer under the Condition of Few Samples," *J. Phys.: Conf. Ser.*, 1653 012039, 2020.
DOI: 10.1088/1742-6596/1653/1/012039
[12] J. Hemmerich, E. Asilar, and G. F. Ecker, "COVER: Conformational Oversampling as Data augmentation for Molecules," *J. Cheminformatics*, vol.12, article number 18, 2020.
DOI: 10.1186/s13321-020-00420-z
[13] L. Wu et al., "Data Augmentation based on Multiple Oversampling Fusion for Medical Image Segmentation," *PLos One*, vol.17, no.10, e0274522, 2022. DOI: 10.1371/journal.pone.0274522
[14] A. Moreo, A. Esuli, and F. Sebastiani, "Distributional Random Oversampling for Imbalanced Text Classification'" In *Proc. of the 39th International ACM SIGIR conference on Research and Development in Information* (*SIGIR2016*), pp. 805-808, 2016. DOI: 10.1145/2911451.2914722
[15] A. Anand et al., "Phishing URL Detection with Oversampling based on Text Generative Adversarial Networks," In *Proc. IEEE International Conference on Big Data*, 2018.
DOI: 10.1109/BigData.2018.8622547
[16] Ni. V. Chawla et al., "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol.16, no.1, pp.321-357, 2002.
DOI: 10.1613/jair.953
[17] M. Ochal et al.. Wang, "Few-Shot Learning with Class Imbalance," *IEEE Trans. Artif.*, Early access, 2023. DOI: 10.1109/TAI.2023.3298303
[18] D. Wertheimer and B. Hariharan, "Few-Shot Learning with Localization in Realistic Settings," In *Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR2019)*, pp 6558-6567, 2019.
DOI: 10.48550/arXiv.1904.08502
[19] B. Kulis, "Metric Learning: A Survey," Now *Foundations and Trends*, 2013.
DOI: 10.1561/2200000019
[20] A. Bellet, A. Habrard, and M. Sebban, "A Survey on Metric Learning for Feature Vectors and Structured Data," Technical Report, arXiv: 1306.6709, 2013. DOI: 10.48550/arXiv.1306.6709
[21] D. Kedem et al., "Non-Linear Metric Learning," In *Proc. of Advances in Neural Information Processing Systems 25 (NIPS2012)*, 2012.
[22] M. Kaya and H. S. Bilge, "Deep Metric Learning: A Survey," *Symmetry,* vol.11, no.9, 1066, 2019. DOI: 10.3390/sym11091066
[23] O. Vinyals et al., "Matching Networks for One Shot Learning," In *Proc. of the 30th Conference on Neural Information Processing Systems* (*NIPS2016*), 2016.
DOI: 10.48550/arXiv.1606.04080
[24] F. Sung et al., "Learning to Compare: Relation Network for Few-Shot Learning," In *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR2018)*, pp.1199-1208, 2018. DOI: 10.1109/CVPR.2018.00131
[25] H.-J. Ye et al., "Few-Shot Learning via Embedding Adaptation with Set-to-Set Functions," In *Proc. of IEEE/CVF International Conference on Computer Vision (ICCV-20)*, pp.8808-8817, 2020. DOI: 10.1109/CVPR42600.2020.00883
[26] C. Liu et al., "Learning a Few-shot Embedding Model with Contrastive Learning," In *Proc. of*

The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21), pp.8635-8643, 2021.
DOI: 10.1609/aaai.v35i10.17047
[27] Y. Gao et al., "Contrastive Prototype Learning with Augmented Embeddings for Few-Shot Learning," In *Proc. of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence* (*UAI 2021*), pp.140-150, 2021.
DOI: 10.48550/arXiv.2101.09499
[28] P. Rodriguez et al., "Embedding Propagation: Smoother Manifold for Few-Shot Classication," In *Proc. of European Conference on Computer Vision* (*ECCV2020*), pp.121-138, 2020.
DOI: 10.48550/arXiv.2003.04151
[29] S. Xiang, F. Nie, and C. Zhang, "Learning a Mahalanobis Distance Metric for Data Clustering and Classification," *Pattern Recogn.*, vol.41, no.12, pp.3600-3612, 2008.
DOI: 10.1016/j.patcog.2008.05.018
[30] H.-J. Ye et al., "Learning Mahalanobis Distance Metric: Considering Instance Disturbance Helps," In *Proc. of the 2017 International Joint Conference on Artificial Intelligence (IJCAI-17)*, 2017, pp. 3315-3321. DOI: 10.24963/ijcai.2017/463
[31] J. Goldberger et al., "Neighbourhood Components Analysis," In *Proc. of Advances in Neural Information Processing Systems 17 (NIPS 2004)*, 2004.
[32] J. Snell, K. Swersky, and R. Zemel. "Prototypical Networks for Few-Shot Learning," In *Proc. of the 31st Conference on Neural Information Processing Systems (NIPS2017)*, 2017.
[33] P. Bateni et al., "Improved Few-Shot Visual Classification," In *Proc. of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR2020)*, pp.14481-14490, 2020.
DOI: 10.1109/CVPR42600.2020.01450
[34] W. Li et al., "Distribution Consistency based Covariance Metric Networks for Few-Shot Learning," In *Proc. of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, pp.8642-8649, 2019. DOI: 10.1609/aaai.v33i01.33018642
[35] L. Qiao et al., "Transductive Episodic-wise Adaptive Metric for Few-Shot Learning," In *Proc. of 2019 IEEE/CVF International Conference on Computer Vision (ICCV-19)*, pp.3603-3612, 2019.
DOI: 10.1109/ICCV.2019.00370
[36] T. Hospedales et al., "Meta-Learning in Neural Networks: A Survey," *IEEE Trans. Pattern Anal. Mach.*, vol.44, pp.5149-5169, 2022.
DOI: 10.1109/TPAMI.2021.3079209
[37] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," In *Proc. of the 34th International Conference on Machine Learning* (*ICML2017*), pp.1126-1135, 2017.
DOI: 10.48550/arXiv.1703.03400
[38] A. A. Rusu et al., "Meta-Learning with Latent Embedding Optimization,"In *Proc. of International Conference Learning Representation* (*ICLR 2019*), 2019. DOI: 10.48550/arXiv.1807.05960
[39] H. Yao et al., "Hierarchically Structured Meta-Learning," In *Proc. of the 36th International Conference on Machine Learning (ICML2019)*, pp.7045-7054, 2019.
DOI: 10.48550/arXiv.1905.05301
[40] C. Fifty et al., "Efficiently Identifying Task Groupings for Multi-Task Learning," In *Proc. of the Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS2021)*, 2021.
[41] A. Zhou, T. Knowles, and C. Finn, "Meta-Learning Symmetries by Reparameterization," In *Proc. of International Conference Learning Representation (ICLR 2021)*, 2021.
DOI: 10.48550/arXiv.2007.02933
[42] S.-O. Kaba et al., "Equivariance with Learned Canonicalization Functions,"In *Proc. of the 40th International Conference on Machine Learning (ICML2023)*, 2023.
DOI: 10.48550/arXiv.2211.06489
[43] S. Basu et al., "Equivariant Few-Shot Learning from Pretrained Models," arXiv:2305.09900, 2023.
DOI: 10.48550/arXiv.2305.09900
[44] E. Triantafillou et al., "Learning a Universal Template for Few-shot Dataset Generalization," In *Proc. of the 38th International Conference on*

Machine Learning (ICML2021), pp.10424-10433, 2021. DOI: 10.48550/arXiv.2105.07029
[45] V. Dumoulin et al., "Feature-wise Transformations," *Distill*, 2018.
DOI: 10.23915/distill.00011
[46] M. Andrychowicz et al., "Learning to Learn by Gradient Descent by Gradien Descent," In *Proc. of the 30th Conference on Neural Information Processing Systems (NIPS2016)*, 2016.
DOI: 10.48550/arXiv.1606.04474
[47] K. Li and J. Malik, "Learning to Optimize," In *Proc. of International Conference Learning Representation (ICLR 2017)*, 2017.
DOI: 10.48550/arXiv.1703.00441
[48] S. Ravi and H. Larochelle, "Optimization as a Model for Few-Shot Learning," In *Proc. of International Conference Learning Representation (ICLR 2017)*, 2017.
[49] T. Chen et al., "Learning to Optimize: a Primer and a Benchmark," *J Mach Learn Res*, Vol.23, no.1, pp.8562-8620, 2023.
[50] Z. Yue et al., "Interventional Few-Shot Learning," In *Proc. of the Thirty-fourth Conference on Neural Information Processing Systems (Neur IPS2020)*, 2020. DOI: 10.48550/arXiv.2009.13000
[51] C. Finn et al., "Online Meta-Learning," In *Proc. of the 36th International Conference on Machine Learning (ICML2019)*, pp.1920-1930, 2019. DOI: 10.48550/arXiv.1902.08438
[52] C. Wah et al., "The Caltech-UCSD Birds-200-2011 Dataset," Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.
[53] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level Concept Learning through Probabilistic Program Induction," *Science*, vol.350, no.6266, pp.1332-1338, 2015.
[54] A. Shaban et al., "One-Shot Learning for Semantic Segmentation,"In *Proc. of British Machine Vision Conference (BMVC2017)*, 2017.
DOI: 10.48550/arXiv.1709.03410
[55] E. Triantafillou et al., "Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples,"In *Proc. of International Conference Learning Representation (ICLR 2020)*, 2020.
DOI: 10.48550/arXiv.1903.03096

BIOGRAPHY

Ho-Sik Seok (Member)

1999 : BS degree in Computer Engineering, Seoul National University.
2001 : MS degree in Electrical Engineering and Computer Science, Seoul National University.
2012 : PhD degree in Electrical Engineering and Computer Science, Seoul National University.
2016~2020.2 : Assistant professor, Dept. of Computer and Communications Engineering, Kangwon National University.
2020.3~2022.1 : Assistant professor, Dept. of Computer Science and Engineering, Kangwon National University.
2022.2~present : Assistant professor, Dept. of Artificial Intelligence and Data Science, Korea Military Academy.