Paper Review

Prototypical Networks for Few-shot Learning

Introduction

- 본 논문은 NIPS 2017에서 소개된 Prototypical Networks for Few-shot Learning[1]으로 Prototypical Networks를 이용하여 few-shot learning을 할 수 있는 모델 제안하는 논문이다.
- Few-shot learning은 train dataset에는 없는 새로운 class를 예측하는 상황에서 분류할 class에 대해 충분한 dataset을 가지고 있지 않을 경우 새로운 class를 분류하기 위해 분류기를 조정되어야 하는 작업
- 이러한 경우 새로운 데이터를 가지고 re-training 할 수 있으나 과적합될 확률이 매우 높다.
- 이를 해결하기 위해 논문의 저자는 protypical networks를 고안
- Protonet은 각 class에 대해 single prototype representation이 있는 embedding을 base로 접근
- 이를 위해 neural network를 사용하여 임베딩 공간에 대한 입력의 비선형 매핑을 학습하고, Embedding 공간에서 설정된 support set의 평균으로 class의 prototype을 만듦

Prototype 이란

- Prototype은 Computer Vision분야에서 특징들을 찾는 방식 중 하나
- 세로축에 나와있는 1 ~ 6는 서로 다른 Prototype을 뜻
- 노란색으로 나와있는 부분은 그 부분을 집중해서 볼 것이라는 뜻
- 각각 다른 Prototype에 따라 같은 이미지라도 활성화 되는 부분이 다름
- 즉, 같은 이미지이지만 포커스 하는 부분을 다르게 하기 위해 변형된 이미지

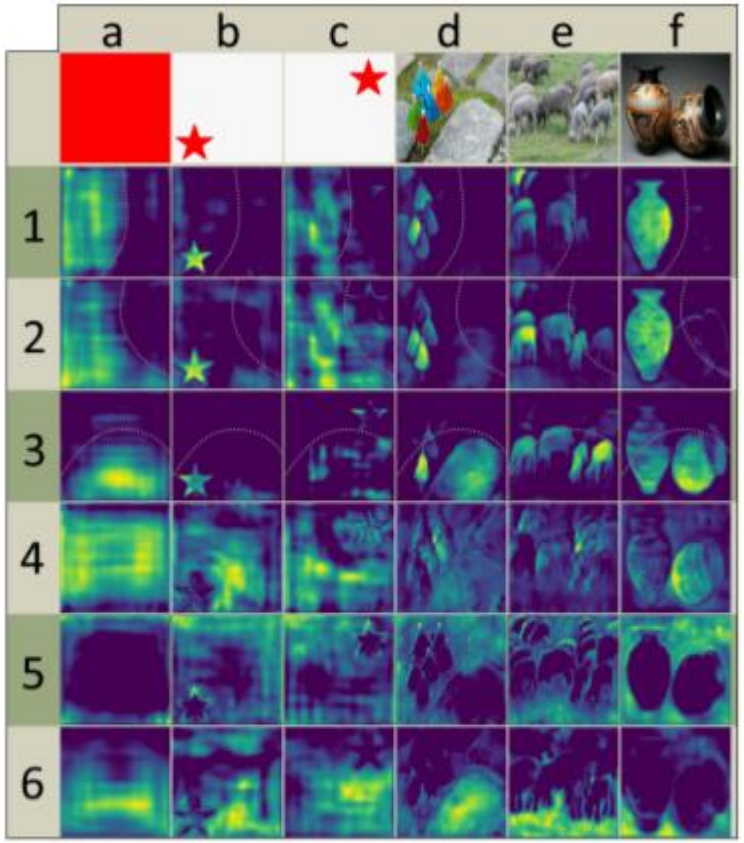
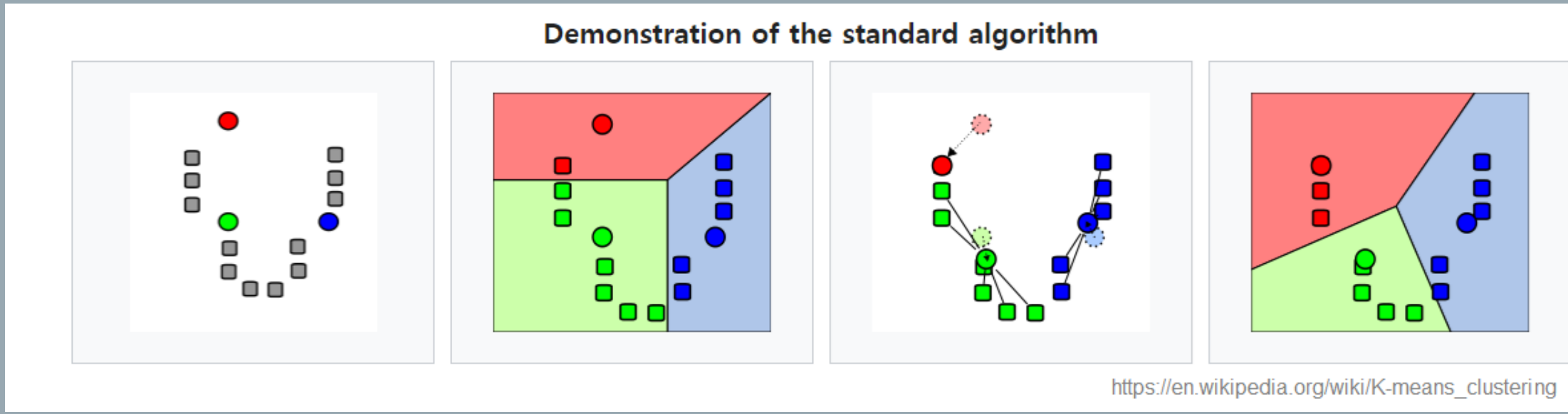


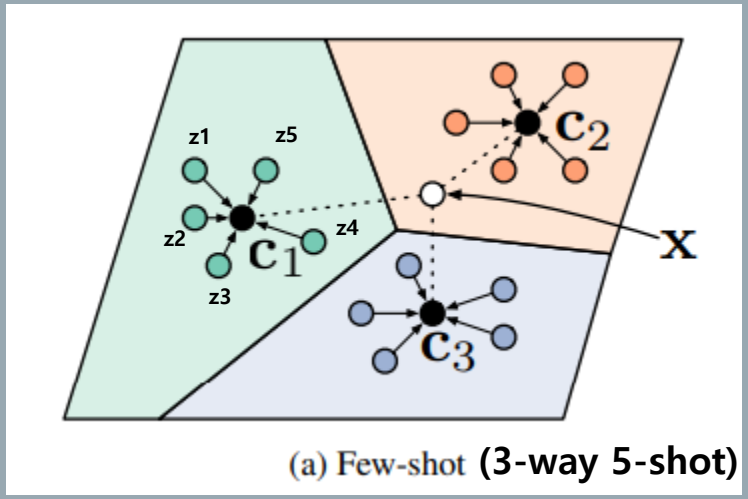
Figure 5: **Prototype Behavior** The activations of the same six prototypes (y axis) across different images (x axis). Prototypes 1-3 respond to objects to one side of a soft, implicit boundary (marked with a dotted line). Prototype 4 activates on the bottom-left of objects (for instance, the bottom left of the umbrellas in image d); prototype 5 activates on the background and on the edges between objects; and prototype 6 segments what the network perceives to be the ground in the image. These last 3 patterns are most clear in images d-f.

Prototype 이란

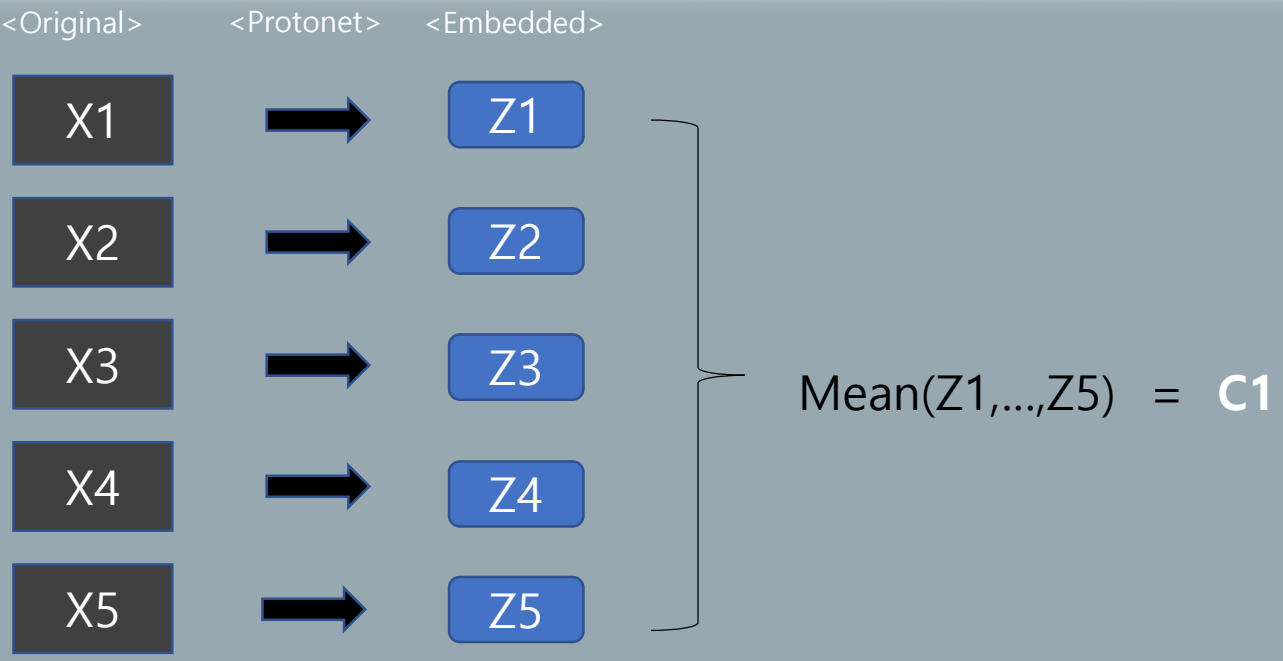


- 해당 논문에서 Protonet은 Clustering과 비슷한 역할을 수행
- 각각의 네모 점(이미지를 임베딩)은 색깔 점과 거리를 계산하여 가장 가까운 색깔 점의 class를 따른다.
- 그 다음 각 class에서 거리의 평균을 내어 prototype의 위치를 update
- 모든 데이터 중심점 이동이 없을때까지 prototype 생성

Prototype 이란

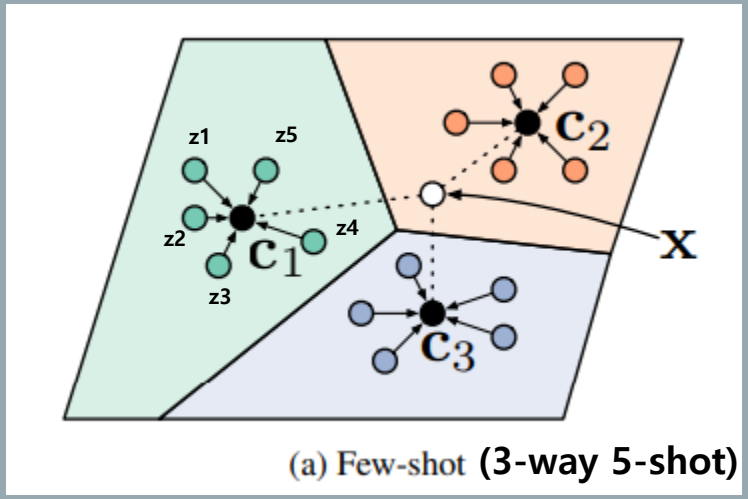


(a) Few-shot **(3-way 5-shot)**

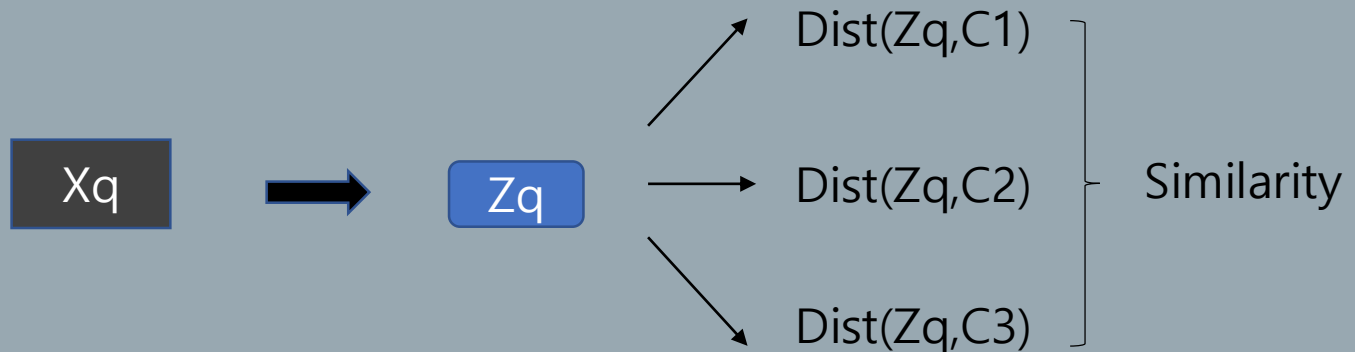


- 3-way 5-shot의 classification에서 C1의 초록색 부분만 계산 / X1~X5는 support set의 image tensor data
- image tensor data를 모델에 넣으면 Z1~5 데이터 생성
- Z1~Z5를 모두 평균한 값이 C1이 되고 이것이 하나의 class의 prototype

Prototype 이란



(a) Few-shot **(3-way 5-shot)**



- query set의 data 하나를 가져와서 어떤 class인지 예측한다고 가정
- query set의 image tensor data는 Xq가 되고 Xq를 모델에 넣으면 Zq로 변형
- Zq & C1, C2, C3와 각각 Euclidean distance로 거리를 계산
- 거리 값에 -를 붙여주게 되면 similarity(거리가 멀수록 즉, 값이 클수록 similarity는 낮기에)
- Similarity에 softmax를 취해 probability에 따라 Class 예측

Prototypical Networks

- Model

- Protonet은 M-dimensional representation인 prototype을 계산하고 각각의 class는 embedding function을 거친다.
- 이때 φ는 learnable parameter(weight)이다
- class's prototype = mean of Support set in the embedding space
- 즉, embedding space에서 각 class를 대표하는 prototype의 분포를 생성

$$\circ \quad \mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\phi(\mathbf{x}_i).$$

Prototypical Networks

- Model

- 앞서 구한 각 Class에 대한 prototype에 대해 거리를 구하게 된다.
- 그 후, query dataset 중 하나인 query point x를 distance 기반의 softmax를 취한 값을 통해 클래스 예측

$$\text{Model} : p_\phi(y = k \mid \mathbf{x}) = \frac{\exp\left(-d\left(f_\phi(\mathbf{x}), \mathbf{c}_k\right)\right)}{\sum_{k'} \exp\left(-d\left(f_\phi(\mathbf{x}), \mathbf{c}_{k'}\right)\right)}$$

- Loss 함수는 Negative log-probability J(φ) J(φ)=−logpφ(y=k|x) 사용
- Optimizer는 SGD를 이용
- Training episode는 training set에서 랜덤하게 class를 선택하여 만든다.
- 남은 것 중 일부를 선택하여 query point를 만든다.

Prototypical Networks as Mixture Density Estimation

- Regular Bregman Divergence

- 해당 논문에서는 Distance function을 regular Bregman divergence을 사용
- 이를 사용할 경우, Prototypical Networks = support set에 Mixture Density Estimation하는 것
- Bregman Divergence은 거리의 일반화 개념이다.
- 거리 공식은 다음과 같다 $$d_{\varphi}(\mathbf{z}, \mathbf{z}') = \varphi(\mathbf{z}) - \varphi(\mathbf{z}') - (\mathbf{z} - \mathbf{z}')^T \nabla\varphi(\mathbf{z}'),$$
- 이는 q에서의 1차 테일러 근사를 통한 근사 F'(p)값과 실제 F(p)값의 차이로 해석 될 수 있다.
- 즉, 거리 계산으로 유클리디안 거리의 제곱을 해당 공식을 이용해 근사화

a. $F(\mathbf{x}) = ||\mathbf{x}||^2 = \mathbf{x}^t\mathbf{x}$ (Euclidean Norm)일 경우

- $D_F(p, q) = || p - q ||^2$ (;Euclidean Distance) 이다.

유도)

$\nabla F(\mathbf{x}) = \frac{\delta}{\delta \mathbf{x}}\mathbf{x}^t\mathbf{x} = 2\mathbf{x}$

$D_F(p, q) = F(p) - F(q) - < \nabla F(q), p - q >$

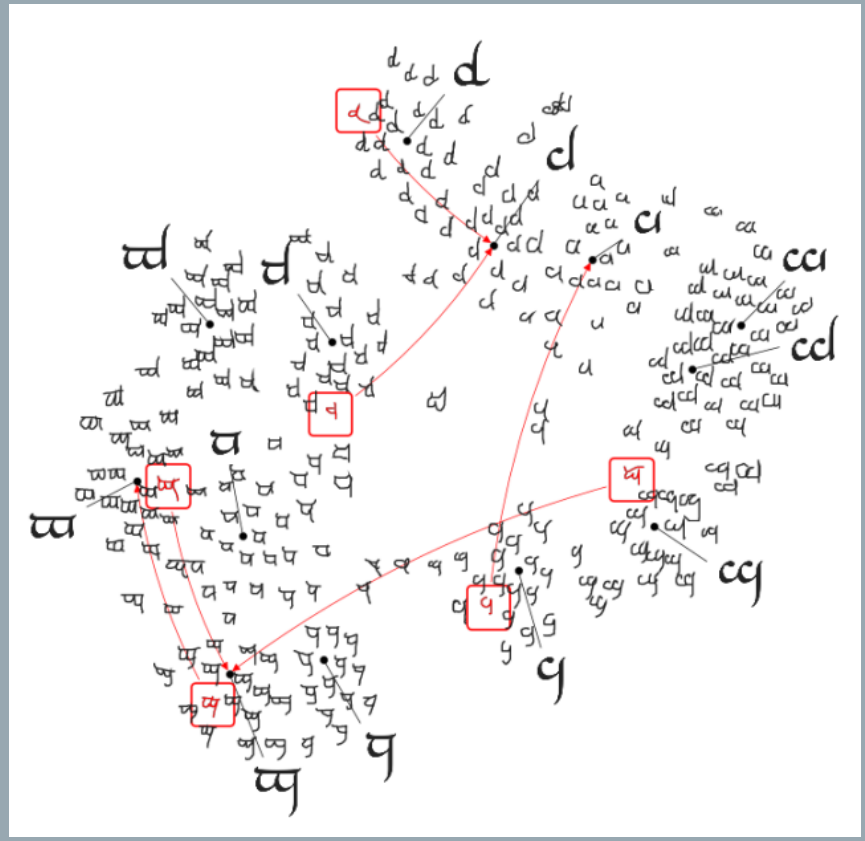
$= p^tp - q^tq - 2q^t(p - q)$

$= p^tp - 2q^tp + q^tq = (p - q)^t(p - q)$

$= || p - q ||^2$

Experiment: Omniglot / miniImageNet Few-shot Classification

- Omniglot dataset은 handwritten character로 50개의 alphabet이 존재하고 각 alphabet의 character들은 20개의 example이 존재
- Embedding architecture은 4개의 convolution block을 사용하였고, 각 block은 64 filter의 3x3 convolution, batch normalization, ReLU, 2x2 max-polling으로 구성
- Initial learning rate는 10^(-3)이고 2000episode마다 절반으로 learning rate를 줄임
- 1-shot, 5-shot scenarios는 train을 할 때 60개의 class와 5개의 임베딩된 query point를 사용
- MiniImageNet dataset은 ImageNet dataset의 축소 버전으로 100개의 class당 600개의 이미지로 총 60,000개의 data가 존재
- Embedding architecture와 learning rate는 Omniglot dataset을 훈련할 때와 같았고, train dataset은 1-shot일때는 30-way, 5-shot일때는 20-way로 구성



Conclusion

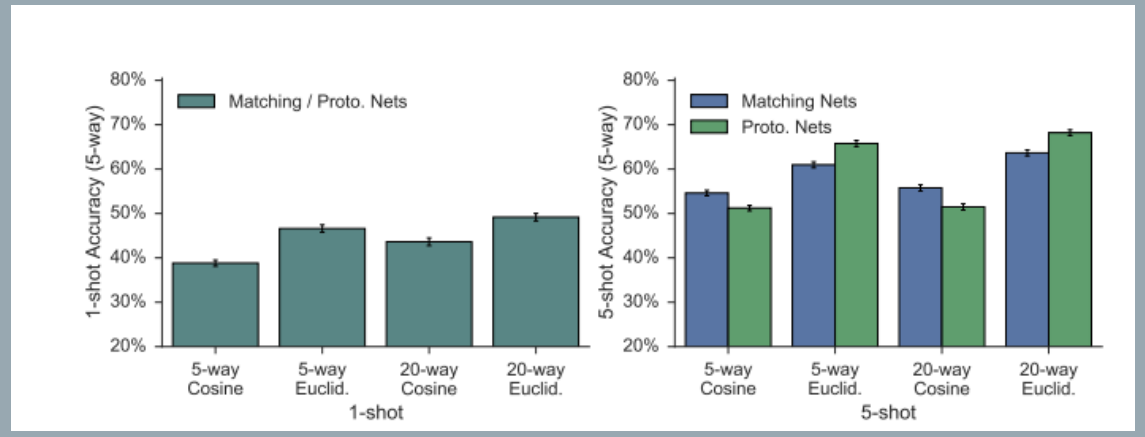
- [MN] weighted nearest neighbor classifier / [PN] linear classifier when squared Euclidean distance is used
- One-shot learning의 경우, ck=xk → MN=PN
- MN과 PN 모두 어떠한 distance function으로서 사용 가능하였으나 일반적으로 cosine distance보다 Euclidean distance가 더 좋은 성능
- train의 Nc(number of class)는 test의 Nc보다 크게 잡는 것이 더 좋은 결과를 내었고, train과 test의 Ns (number of shot per class)는 같게 하는 것이 대부분 좋은 결과를 내었다.

Table 1: Few-shot classification accuracies on Omniglot. *Uses non-standard train/test splits.

Model	Dist.	Fine Tune	5-way Acc. 1-shot	5-way Acc. 5-shot	20-way Acc. 1-shot	20-way Acc. 5-shot
MATCHING NETWORKS [32]	Cosine	N	98.1%	98.9%	93.8%	98.5%
MATCHING NETWORKS [32]	Cosine	Y	97.9%	98.7%	93.5%	98.7%
NEURAL STATISTICIAN [7]	-	N	98.1%	99.5%	93.2%	98.1%
MAML [9]*	-	N	98.7%	**99.9%**	95.8%	**98.9%**
PROTOTYPICAL NETWORKS (OURS)	Euclid.	N	**98.8%**	99.7%	**96.0%**	**98.9%**

Table 2: Few-shot classification accuracies on *mini*ImageNet. All accuracy results are averaged over 600 test episodes and are reported with 95% confidence intervals. *Results reported by [24].

Model	Dist.	Fine Tune	5-way Acc. 1-shot	5-way Acc. 5-shot
BASELINE NEAREST NEIGHBORS*	Cosine	N	28.86 ± 0.54%	49.79 ± 0.79%
MATCHING NETWORKS [32]*	Cosine	N	43.40 ± 0.78%	51.09 ± 0.71%
MATCHING NETWORKS FCE [32]*	Cosine	N	43.56 ± 0.84%	55.31 ± 0.73%
META-LEARNER LSTM [24]*	-	N	43.44 ± 0.77%	60.60 ± 0.71%
MAML [9]	-	N	**48.70 ± 1.84%**	63.15 ± 0.91%
PROTOTYPICAL NETWORKS (OURS)	Euclid.	N	**49.42 ± 0.78%**	**68.20 ± 0.66%**



출처

1. Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. Advances in neural information processing systems, 30.