



Meta-Learning for Batch Mode Active Learning

2018.11.5

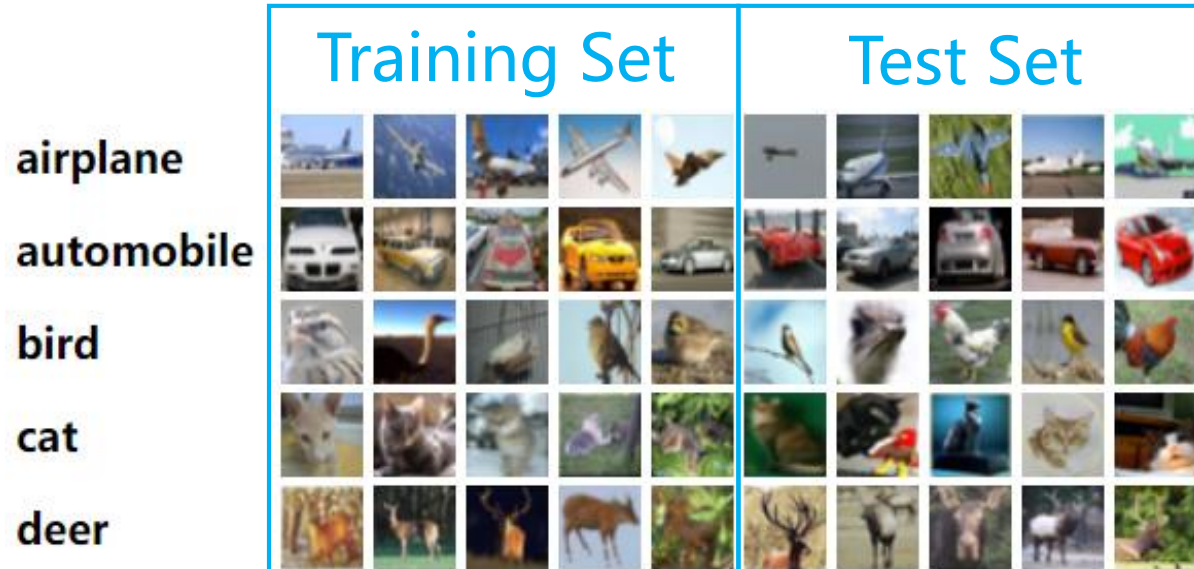
Outline

Meta-Learning for Batch Mode Active Learning

Prototypical Networks for Few-shot Learning

Introduction

- Deep Learning in CIFAR10 Dataset



? Few samples

It requires large amount of labeled samples...

Introduction


It is easy for human being...



? How about machine

Introduction

- Few-shot Classification

	Support Set	Query Set
airplane		
automobile		
bird		
cat		
deer		

Support Set:
very few labeled samples

dog	
frog	
horse	
ship	
truck	

Training Set

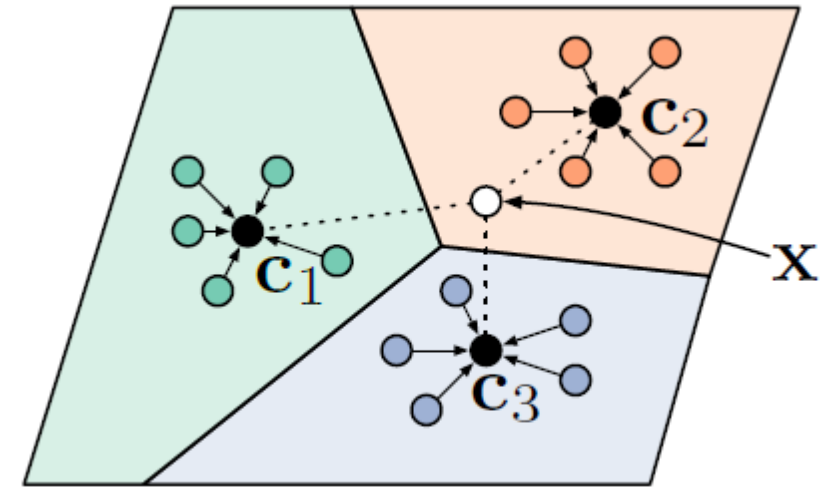
Motivation

- Idea

There exists an embedding in which points cluster around a single prototype representation for support set of each class.

- Embedding Function

$$f_{\phi} : \mathbb{R}^D \rightarrow \mathbb{R}^M$$



(a) Few-shot

Model

- Compute prototype of support points

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

- Produce a distribution over classes for each query point

$$p_\phi(y = k | \mathbf{x}) = \frac{\exp(-d(f_\phi(\mathbf{x}), \mathbf{c}_k))}{\sum_{k'} \exp(-d(f_\phi(\mathbf{x}), \mathbf{c}_{k'}))}$$

A distance function

f_ϕ is unknown \longrightarrow How to learn ϕ

Model Training

- Episode
 - It is designed to mimic the few-shot task
 - ① Randomly select a subset of classes from the **training set**
 - ② Then choose a subset of examples within each class to act as the support set
 - ③ Choose a subset of the remainder to serve as query points.
 - Loss Function

$$J(\phi) = -\log p_{\phi}(y = k | \mathbf{x})$$

Model Training

Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, where each $y_i \in \{1, \dots, K\}$. \mathcal{D}_k denotes the subset of \mathcal{D} containing all elements (\mathbf{x}_i, y_i) such that $y_i = k$.

Output: The loss J for a randomly generated training episode.

$V \leftarrow \text{RANDOMSAMPLE}(\{1, \dots, K\}, N_C)$

▷ Select class indices for episode

for k in $\{1, \dots, N_C\}$ **do**

$S_k \leftarrow \text{RANDOMSAMPLE}(\mathcal{D}_{V_k}, N_S)$

▷ Select support examples

$Q_k \leftarrow \text{RANDOMSAMPLE}(\mathcal{D}_{V_k} \setminus S_k, N_Q)$

▷ Select query examples

$\mathbf{c}_k \leftarrow \frac{1}{N_C} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\phi(\mathbf{x}_i)$

▷ Compute prototype from support examples

end for

$J \leftarrow 0$

▷ Initialize loss

for k in $\{1, \dots, N_C\}$ **do**

for (\mathbf{x}, y) in Q_k **do**

$J \leftarrow J + \frac{1}{N_C N_Q} \left[d(f_\phi(\mathbf{x}), \mathbf{c}_k) + \log \sum_{k'} \exp(-d(f_\phi(\mathbf{x}), \mathbf{c}_{k'})) \right]$

▷ Update loss

end for

end for

Meta-Learning for Batch Mode Active Learning

- Batch Model Active Learning
 - It should be designed to directly optimize its effectiveness at finding sets of items to label that improve the model' s ultimate performance
 - it should have linear complexity in selecting each additional item
 - it should work well in the presence of distractors

How to select a batch of unlabeled examples in the pool ?

Model

- The probability of a B -size subset $\mathcal{A} = \{\tilde{x}_1, \dots, \tilde{x}_B\}$

$$\text{Chain rule: } p(\mathcal{A}) = p(\tilde{x}_1, \dots, \tilde{x}_B) = \prod_{i=1}^B p(\tilde{x}_i | \tilde{x}_1, \dots, \tilde{x}_{i-1})$$

- The probability distribution to select the next unlabeled item

$$p(\tilde{x} | \mathcal{A}) = p(\tilde{x} | \tilde{x}_1, \dots, \tilde{x}_j)$$

How to define p and how to get p ?

Method

- Modify the episode structure to accommodate a batch mode active learning by supplementing the support and query set an unlabeled set $\mathcal{U} = \{\tilde{x}_1, \dots, \tilde{x}_M\}$.
- The goal: Use the classifier formed from the original support set \mathcal{S} to select the best subset of B examples from \mathcal{U} to label to create new support set \mathcal{S}' and associated new classifier so as to most improve the performance on the query

Classifier: prototypes $\{c_k\}_{k=1}^K$

Item-classifier statistics: $\Pi(\{c_k\}_{k=1}^K, \tilde{x}_i)$

Method

- Quality Distribution

A MLP with parameters \mathbf{q}

$$p_{\text{quality}}(\tilde{x}_i) \propto \exp(q_i), \text{ where } q_i = f_{\mathbf{q}}(\Pi(\{c_k\}_{k=1}^K, \tilde{x}_i))$$

- Diversity Distribution

$$\phi_i = f_{\phi}(\Pi(\{c_k\}_{k=1}^K, \tilde{x}_i))$$

$$p_{\text{diversity}}(\tilde{x}_i | \mathcal{A}) \propto \exp(v(\phi_i)/\tau), \text{ where } v(\phi_i) = \min_{\tilde{x}_j \in \mathcal{A}} \{\sin \theta_{ij}\}$$



$$p(\tilde{x} | \mathcal{A}) \propto p_{\text{quality}}(\tilde{x}) \cdot p_{\text{diversity}}(\tilde{x} | \mathcal{A}) \cdot \mathbb{1}_{\tilde{x} \notin \mathcal{A}}$$

→ Update the parameters $\theta' = \{\phi, \mathbf{q}, \tau\}$

Training

For any given episode, with corresponding support set \mathcal{S} , unlabeled set \mathcal{U} , and query set \mathcal{Q} , $p_{\theta'}(\tilde{x}|\mathcal{A})$ can be repeatedly sampled to create a subset $\mathcal{A} \subseteq \mathcal{U}$ of size B so that the prototypes computed on new support set $\mathcal{S}' = \mathcal{S} \cup \mathcal{A}$ have high performance on the query set \mathcal{Q} .



Most improve $\mathcal{C}(\mathcal{Q} | \mathcal{S} \cup \mathcal{A}_t)$

The gradient:

$$\beta_{-t} = \frac{1}{T-1} \sum_{t' \neq t}^T \mathcal{C}_{\theta}(\mathcal{Q} | \mathcal{S} \cup \mathcal{A}_{t'})$$

$$\nabla_{\theta'} \mathbb{E}_{p_{\theta'}(\mathcal{A})}[\mathcal{C}(\mathcal{Q} | \mathcal{S} \cup \mathcal{A})] \approx \frac{1}{T} \sum_{t=1}^T [(\mathcal{C}(\mathcal{Q} | \mathcal{S} \cup \mathcal{A}_t) - \beta_{-t}) \nabla_{\theta'} \log p_{\theta'}(\mathcal{A}_t)]$$



$$\mathcal{C}(\mathcal{Q} \cup \mathcal{U} | \mathcal{S} \cup \mathcal{A})$$

where $\mathcal{A}_t \sim p_{\theta'}(\mathcal{A})$

Experiment

- CIFAR100 and *mini*Imagenet
 - 100 classes, 600 images per class
 - 64 classes for training, 16 classes for validation, 20 class for testing
- Comparison with
 - Max-Entropy
 - Min-Max Sim
 - Random

Experiment

- Result

Method	$B = 5$ $M = 100$	$B = 10$ $M = 100$	$B = 20$ $M = 200$	$B = 5$ w/ D $M = 200$	$B = 10$ w/ D $M = 200$	$B = 20$ w/ D $M = 400$
Max-Entropy	+1.5%	+7.3%	+13.8%	-1.4%	+0.7%	+5.3%
Min-Max Sim	+2.2%	+5.3%	+10.0%	+0.9%	+2.1%	+4.5%
Random	+3.3%	+10.0%	+17.2%	-1.1%	+2.6%	+8.6%
Meta-Learner	+7.3%	+11.6%	+17.4%	+3.2%	+6.2%	+10.80%

Method	$B = 5$ $M = 100$	$B = 10$ $M = 100$	$B = 20$ $M = 200$	$B = 5$ w/ D $M = 200$	$B = 10$ w/ D $M = 200$	$B = 20$ w/ D $M = 400$
Max-Entropy	-0.5%	+5.0%	+11.3%	-3.3%	-1.3%	+2.8%
Min-Max Sim	+0.3%	+3.6%	+7.7%	-0.8%	-0.1%	+2.0%
Random	+1.6%	+8.6%	+17.8%	-2.9%	+2.1%	+7.5%
Meta-Learner	+6.6%	+11.1%	+18.1%	+1.3%	+6.1%	+11.20%

Thanks
