2021

인공지능세미나

김범채

AILab. 연구소장

2021.08.15.

Active Learning for Convolutional Neural Networks: A Core-Set Approach

1. Intro
2. Related works
3. Methodology
4. Experiment
5. Conclusion

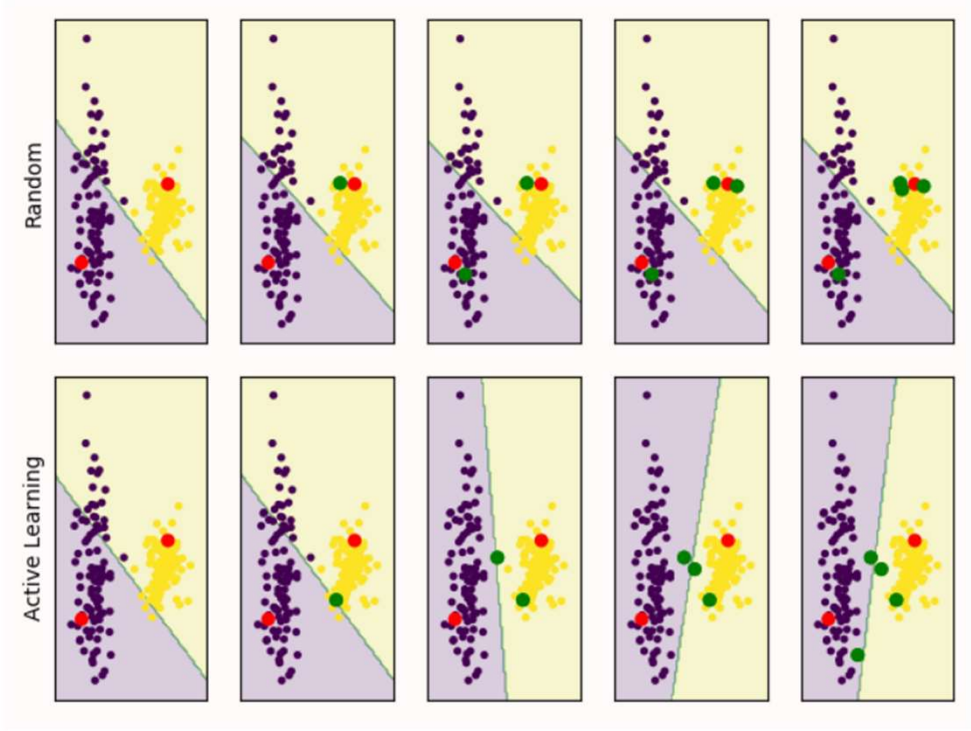
Problem

- Deep Learning은 매우 좋은 성능을 보이지만, 많은 Labeling 비용이 필요하다.
  - 많은 데이터는 거의 항상(Almost always) 성능이 좋아진다.
- 많은 데이터 → 높은 표현력(Higher representative power) → 더 좋은 성능
- 하지만, 많은 데이터 → 많은 Labeling 비용

INTRO

Active Learning의 개요

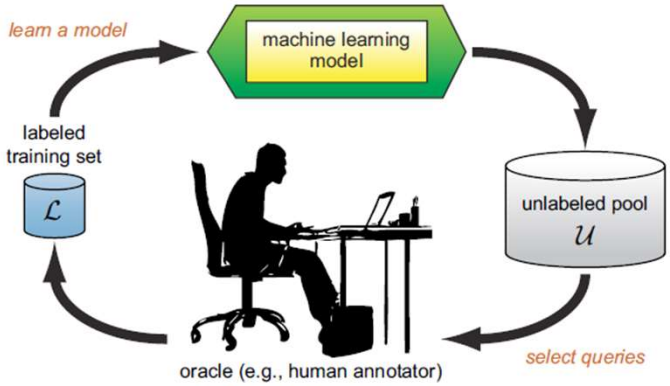
1. 전체 데이터셋중에서 중요한 데이터를 선별함으로써, 충분한 모델 경쟁력을 가지자!!



Active Learning : 과정 및 효과

* 빨간점 : labeling 된 데이터
* 초록점 : 레이블링을 위해, 선택된 데이터

Active Learning의 과정

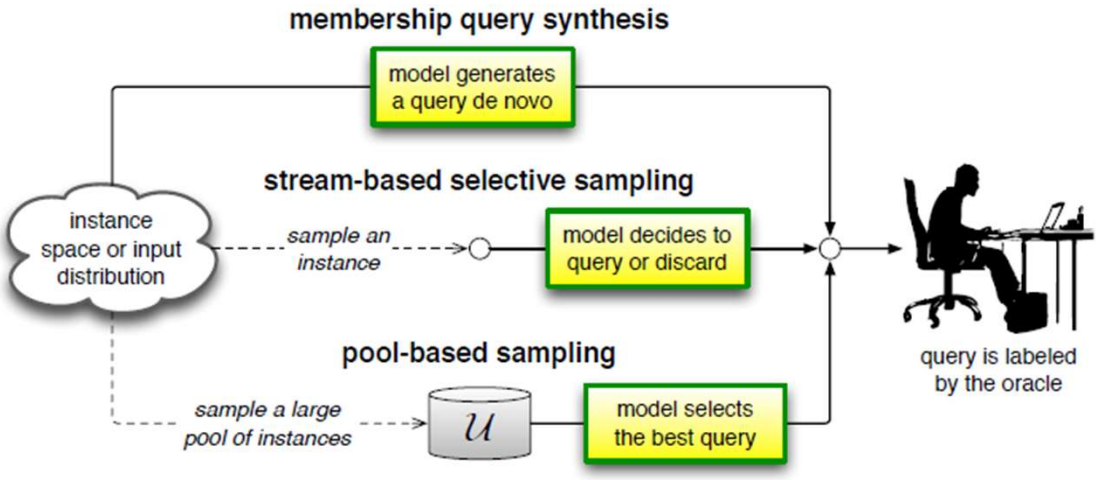


[Active Learning 과정]

> visionexperiments.blogspot.com/2016/01/active-learning-literature-survey.html

1. Learn a Model : 레이블링 된 데이터($\mathcal{L}$)를 활용하여, 모델 학습
2. Select Queries : 학습된 모델을 통해서, 레이블링 되지 않은 데이터($\mathcal{U}$)에서 선별
3. Human Annotator : 사람이 레이블링 - (Oracle 한 레이블링이라고 믿는다.. 아니기도 하지만...)
4. Training set : 새로 레이블링 된 데이터를 기존 dataset과 합친다.
5. 목표 성능 도달까지 위에 과정(1~4)을 **반복**한다.

Active Learning의 분야



[Active Learning의 종류]

1. **Membership Query Synthesis** : 모델이 레이블링 요청(Query)할 데이터를 **생성**
2. **Stream-based Selective Sampling** : 새로운 샘플이 들어옴 → 모델이 레이블링 필요한지 아닌지를 판단 (ex : Binary Model)
3. **Pool-based Sampling** : 큰 데이터 Pool이 존재 → 모델이 중요한 데이터 포인트(data point - query)를 선택

Medical Data Problem

- 의료 데이터의 Case가 Computer Vision 기반으로 정리가 되어있는 지를 판별
  - 정리가 되어있다면 Case 별 대량의 데이터를 수집하고 Pool-based Sampling 진행
  - 그렇게 정리가 되어있지 않다면, Stream-based Selective Sampling으로 빠른 Case 정리가 필요
    - 새로운 샘플이 들어옴 → 모델이 레이블링 필요한지 아닌지를 판단 (ex : Binary Model)
- 그러나, 대부분 의료 데이터는 **Case 가 정리 되어있지 않은 경우가 많다**.
- 따라서, 딥러닝에서는 더 좋지 않게 판단 될 가능성이 있다.

Methodology

- Active Learning Problem
  - 기존 레이블된 Subset S0 와 새로 추출한 Subset S1 을 통해 Expected Loss 를 줄여야함
    - b(Budget) : 뽑아야 할 데이터 포인트 수
    - S1: 1회(iteration) AL 실행으로, 뽑은 Subset

to ask an oracle, and a learning algorithm $A_{\mathbf{s}}$ which outputs a set of parameters $\mathbf{w}$ given a labelled set $\mathbf{s}$. The active learning with a pool problem can simply be defined as

$$\min_{\mathbf{s}^1:|\mathbf{s}^1|\leq b} E_{\mathbf{x},y\sim p_{\mathcal{Z}}}[l(\mathbf{x},y;A_{\mathbf{s}^0\cup\mathbf{s}^1})] \tag{1}$$

In other words, an active learning algorithm can choose b extra points and get them labelled by an oracle to minimize the future expected loss. There are a few differences between our formulation and

Methodology

In order to design an active learning strategy which is effective in batch setting, we consider the following upper bound of the active learning loss we formally defined in (1):

$$E_{\mathbf{x},y\sim p_{\mathcal{Z}}}[l(\mathbf{x},y;A_{\mathbf{s}})] \leq \underbrace{\left|E_{\mathbf{x},y\sim p_{\mathcal{Z}}}[l(\mathbf{x},y;A_{\mathbf{s}})] - \frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i,y_i;A_{\mathbf{s}})\right|}_{\text{Generalization Error}} + \underbrace{\frac{1}{|\mathbf{s}|}\sum_{j\in\mathbf{s}} l(\mathbf{x}_j,y_j;A_{\mathbf{s}})}_{\text{Training Error}}$$
$$+ \underbrace{\left|\frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i,y_i;A_{\mathbf{s}}) - \frac{1}{|\mathbf{s}|}\sum_{j\in\mathbf{s}} l(\mathbf{x}_j,y_j;A_{\mathbf{s}}),\right|}_{\text{Core-Set Loss}} \tag{3}$$

The quantity we are interested in is the population risk of the model learned using a small labelled subset ($\mathbf{s}$). The population risk is controlled by the *training error* of the model on the labelled subset, the *generalization error* over the full dataset ($[n]$) and a term we define as the *core-set loss*. Core-set loss is simply the difference between average empirical loss over the set of points which have labels for and the average empirical loss over the entire dataset including unlabelled points. Empirically, it is widely observed that the CNNs are highly expressive leading to very low training error and they typically generalize well for various visual problems. Moreover, generalization error of CNNs is also theoretically studied and shown to be bounded by Xu & Mannor (2012). Hence, the critical part for active learning is the core-set loss. Following this observation, we re-define the active learning problem as:

$$\min_{\mathbf{s}^1:|\mathbf{s}^1|\leq b} \left|\frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i,y_i;A_{\mathbf{s}^0\cup\mathbf{s}^1}) - \frac{1}{|\mathbf{s}^0+\mathbf{s}^1|}\sum_{j\in\mathbf{s}^0\cup\mathbf{s}^1} l(\mathbf{x}_j,y_j;A_{\mathbf{s}^0\cup\mathbf{s}^1})\right| \tag{4}$$

Informally, given the initial labelled set ($\mathbf{s}^0$) and the budget (b), we are trying to find a set of points to query labels ($\mathbf{s}^1$) such that when we learn a model, the performance of the model on the labelled subset and that on the whole dataset will be as close as possible.

"Subset에서의 모델 성능과 전체 데이터 셋에서의 성능 차이를 최소화"

Methodology

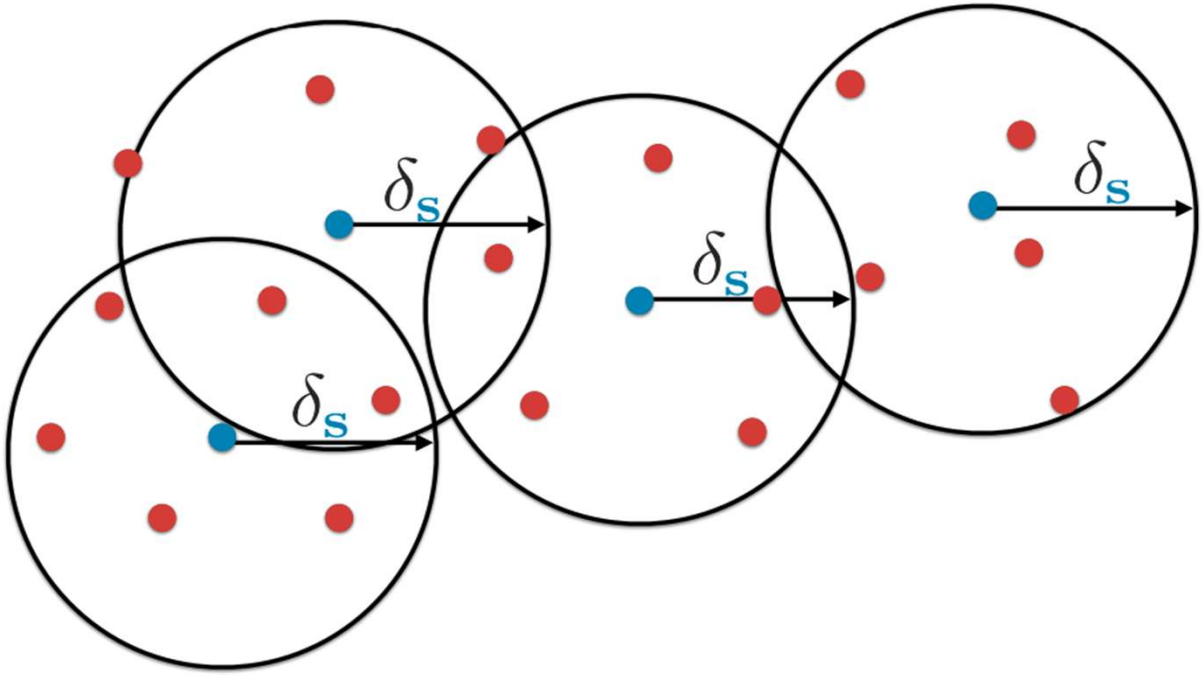


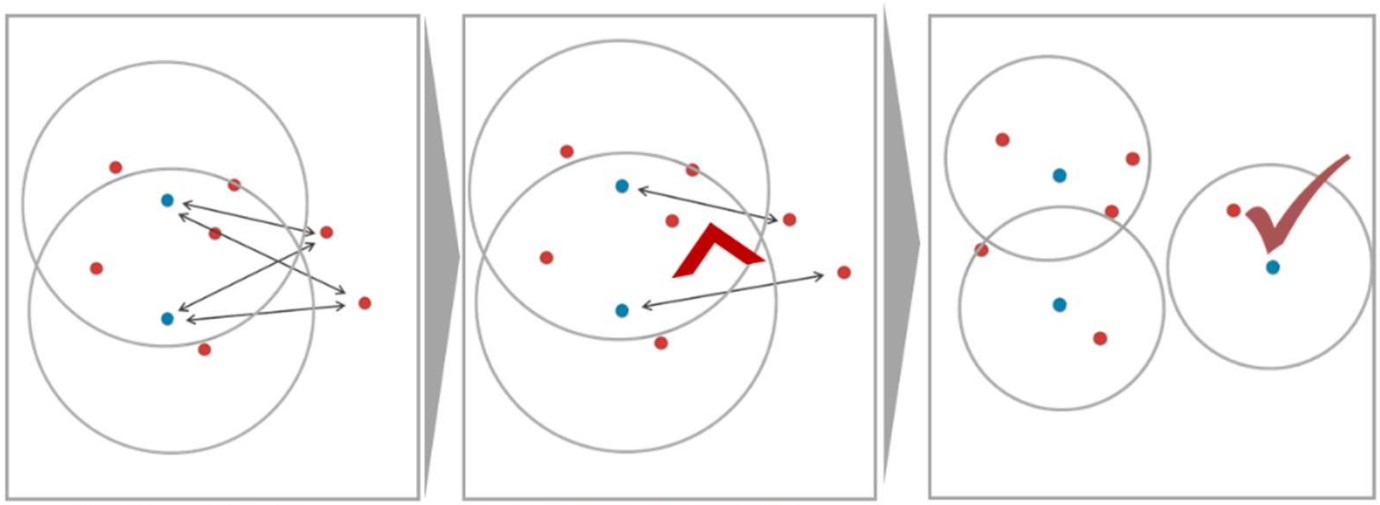
Figure 1: **Visualization of the Theorem 1**. Consider the set of selected points $\mathbf{s}$ and the points in the remainder of the dataset $[n] \setminus \mathbf{s}$, our results shows that if $\mathbf{s}$ is the $\delta_{\mathbf{s}}$ cover of the dataset, $\left|\frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i, y_i, A_{\mathbf{s}}) - \frac{1}{|\mathbf{s}|}\sum_{j\in\mathbf{s}} l(\mathbf{x}_j, y_j; A_{\mathbf{s}})\right| \leq \mathcal{O}(\delta_{\mathbf{s}}) + \mathcal{O}\left(\sqrt{\frac{1}{n}}\right)$

Methodology

- CORE-SETS FOR CNNS

Algorithm 1 k-Center-Greedy

Input: data $\mathbf{x}_i$, existing pool $\mathbf{s}^0$ and a budget b
Initialize $\mathbf{s} = \mathbf{s}^0$
repeat
  $u = \arg\max_{i \in [n] \setminus \mathbf{s}} \min_{j \in \mathbf{s}} \Delta(\mathbf{x}_i, \mathbf{x}_j)$
  $\mathbf{s} = \mathbf{s} \cup \{u\}$
until $|\mathbf{s}| = b + |\mathbf{s}^0|$
return $\mathbf{s} \setminus \mathbf{s}^0$



1. S^0의 Center거리 중 짧은 거리 찾기
2. 그 중 가장 긴 거리 찾기
3. Core-set 선정

K-Center Greedy 알고리즘 도식화

- CORE-SETS FOR CNNS

Algorithm 2 Robust k-Center

Input: data $\mathbf{x}_i$, existing pool $\mathbf{s}^0$, budget b and outlier bound Ξ
Initialize $\mathbf{s}_g$ = k-Center-Greedy($\mathbf{x}_i, \mathbf{s}^0, b$)
$\delta_{2-OPT} = \max_j \min_{i \in \mathbf{s}_g} \Delta(\mathbf{x}_i, \mathbf{x}_j)$
$lb = \frac{\delta_{2-OPT}}{2}$, $ub = \delta_{2-OPT}$
repeat
  if $Feasible(b, \mathbf{s}^0, \frac{lb+ub}{2}, \Xi)$ **then**
    $ub = \max_{i,j|\Delta(\mathbf{x}_i,\mathbf{x}_j) \leq \frac{lb+ub}{2}} \Delta(\mathbf{x}_i, \mathbf{x}_j)$
  else
    $lb = \min_{i,j|\Delta(\mathbf{x}_i,\mathbf{x}_j) \geq \frac{lb+ub}{2}} \Delta(\mathbf{x}_i, \mathbf{x}_j)$
  end if
until $ub = lb$
return $\{i \; s.t. \; u_i = 1\}$

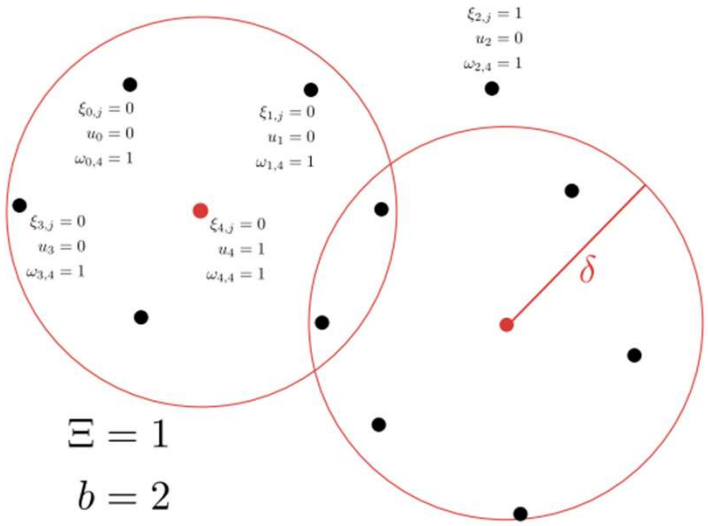


Figure 2: Visualizations of the variables. In this solution, the 4^{th} node is chosen as a center and nodes $0, 1, 3$ are in a δ ball around it. The 2^{nd} node is marked as an outlier.

Experiment Results

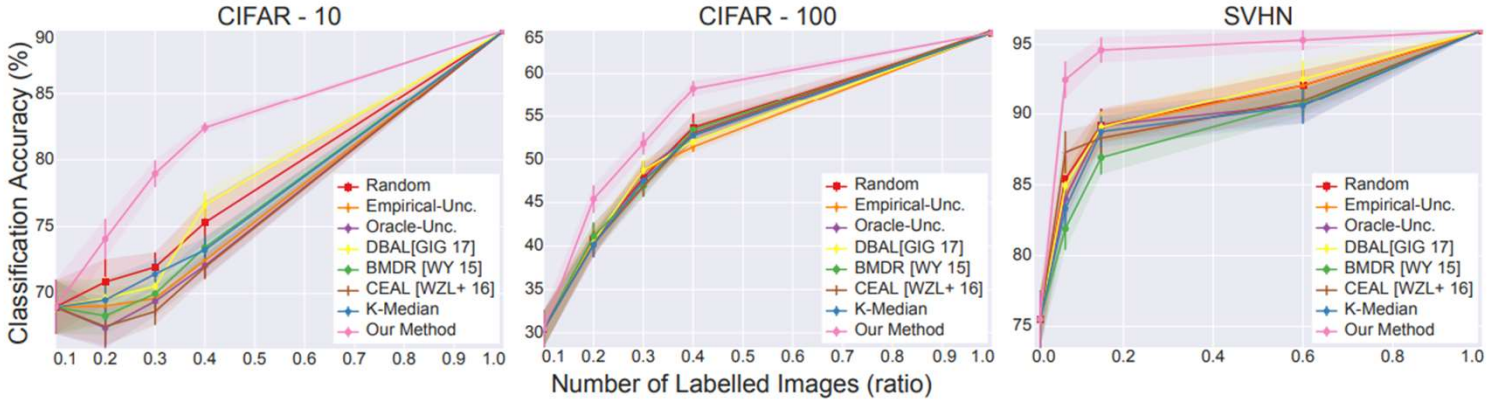


Figure 3: Results on Active Learning for Weakly-Supervised Model (error bars are std-dev)

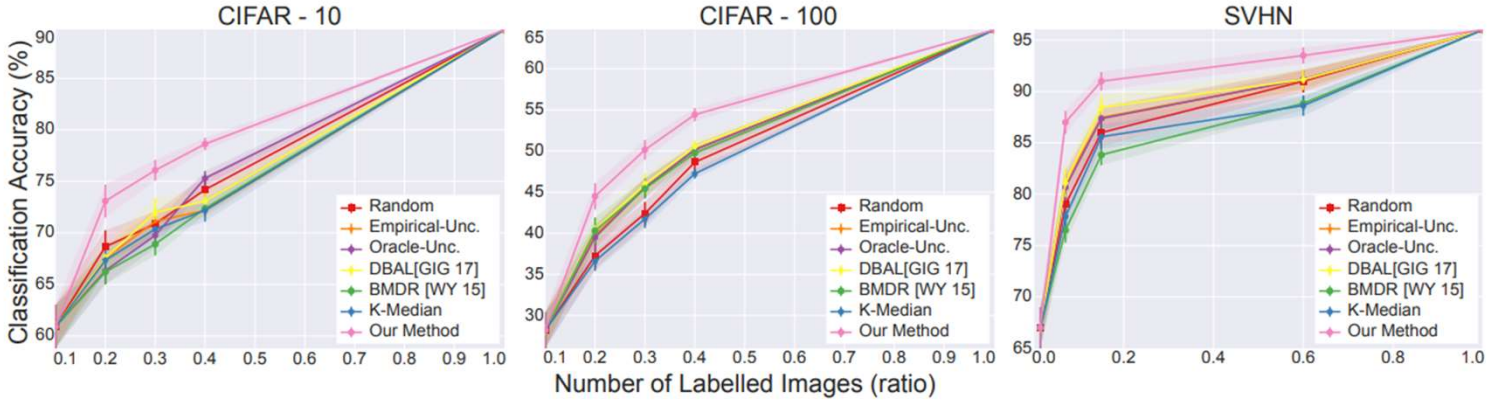
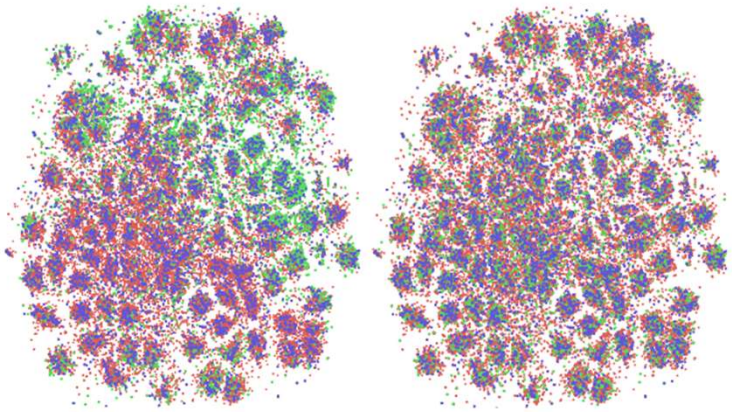


Figure 4: Results on Active Learning for Fully-Supervised Model (error bars are std-dev)

Experiment Results



(a) Uncertainty Oracle (b) Our Method

Figure 5: tSNE embeddings of the CIFAR dataset and behavior of uncertainty oracle as well as our method. For both methods, the initial labeled pool of 1000 images are shown in blue, 1000 images chosen to be labeled in green and remaining ones in red. Our algorithm results in queries evenly covering the space. On the other hand, samples chosen by uncertainty oracle fails to cover the large portion of the space.

Table 1: Average run-time of our algorithm for $b = 5k$ and $|\mathbf{s}^0| = 10k$ in seconds.

Distance Matrix	Greedy (2-OPT)	MIP (iteration)	MIP (total)	Total
104.2	2	7.5	244.03	360.23

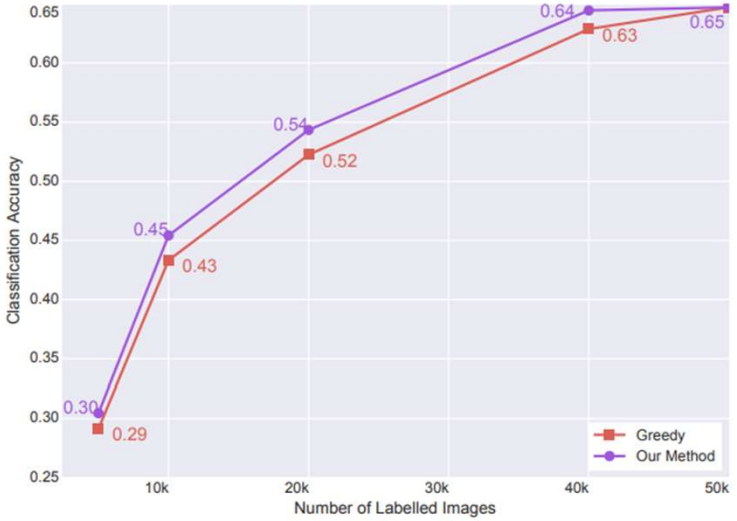


Figure 6: We compare our method with k-Center-Greedy. Our algorithm results in a small but important accuracy improvement.

Appendix

Core-set Loss 중 , $\frac{1}{|\mathbf{s}|}\sum_{j\in\mathbf{s}} l(\mathbf{x}_j, y_j; A_\mathbf{s}))$ 은 딥러닝이 학습하는 과정에서 **training error를 0으로 수렴**하므로

$$\left|\frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i, y_i; A_\mathbf{s}) - \frac{1}{|\mathbf{s}|}\sum_{j\in\mathbf{s}} l(\mathbf{x}_j, y_j; A_\mathbf{s})\right| = \frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i, y_i; A_\mathbf{s}).$$

으로 식을 변경할 수 있습니다.
Lipschitz continuity 를 활용하여, Loss 함수의 Upper Bound를 δ_s와 관련된 식으로 정의했습니다.

1. Gradient Descent로 학습하는 비용 함수가 만약, Lipschitz continuity 하다면

2. 상한(Upper Bound)가 존재합니다.

3. 그리고 그 상한(Upper Bound)이 Radius로 구성되어 있다면,

4. 우리는 Radius 찾는 솔루션만 만들면 된다.
   * 말은 쉽지만.. 증명과정과 구현은 쉽지 않다.

Lipschitz continuity 참고 light-tree.tistory.com/188

Theorem 1. *Given n i.i.d. samples drawn from $p_\mathcal{Z}$ as $\{\mathbf{x}_i, y_i\}_{i\in[n]}$, and set of points $\mathbf{s}$. If loss function $l(\cdot, y, \mathbf{w})$ is λ^l-Lipschitz continuous for all $y, \mathbf{w}$ and bounded by L, regression function is λ^η-Lipschitz, $\mathbf{s}$ is $\delta_\mathbf{s}$ cover of $\{\mathbf{x}_i, y_i\}_{i\in[n]}$, and $l(\mathbf{x}_{s(j)}, y_{s(j)}; A_\mathbf{S}) = 0 \quad \forall j \in [m]$; with probability at least $1-\gamma$,*

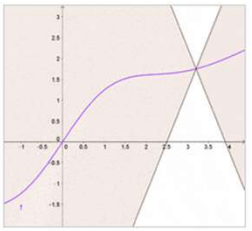
$$\left|\frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i, y_i; A_\mathbf{s}) - \frac{1}{|\mathbf{s}|}\sum_{j\in\mathbf{s}} l(\mathbf{x}_j, y_j; A_\mathbf{s})\right| \leq \delta(\lambda^l + \lambda^\mu LC) + \sqrt{\frac{L^2\log(1/\gamma)}{2n}}.$$

Lemma1.을 통해, Class수와 관련된 Lipschitz 함수라는 것을 확인했습니다.
* 실제 실험에서도, Class가 증가할수록 Core-set의 성능이 저하되었습니다.

Lemma 1. *Loss function defined as the 2-norm between the class probabilities and the softmax output of a convolutional neural network with n_c convolutional (with max-pool and ReLU) and n_{fc} fully connected layers defined over C classes is $\left(\frac{\sqrt{C-1}}{C}\alpha^{n_c+n_{fc}}\right)$-Lipschitz function of input for fixed class probabilities and network parameters.*

저자는 cross-entropy가 아닌, $l_2 - distance$ 를 사용하여 증명했지만,
실제 실험에서는 Cross-entropy를 썼으며, 문제가 없다고 합니다.

Lipschitz continuity (이하 립시츠 연속성) 의 영문 위키를 다른 글에 참조글로 활용하기 위해 번역합니다.



이미지 출처: https://en.wikipedia.org/wiki/Lipschitz_continuity

립시츠 연속성을 가장 직관적으로 잘 표현한 이미지입니다. 이미지를 보면 파란색 함수의 접선이 모두 분홍색 영역에 있음을 머리속에 그려볼 수 있습니다. 립시츠 연속성을 수식으로 표현하면 다음과 같은데

$$for\ metric\ space\ (x,\ d_x),\ (y,\ d_y),$$
$$\frac{d_y(f(x_1),\ f(x_2))}{d_x(x_1,\ x_2)} \leq K,\ K \geq 0$$

$$for\ real\text{-}valued\ function\ f\colon R \to R,$$
$$\frac{|f(x_1) - f(x_2)|}{|x_1 - x_2|} \leq K,\ K \geq 0$$

이 의미를 두번째 수식을 이용해서 말로 풀어 설명하면, 주어진 구간안의 함수의 두 점을 이은 직선의 기울기가 K 보다 작다는 것입니다.
이를 "First derivative 도함수가 Bounded function 유계함수인 함수는 립시츠 연속성을 가진다." 라고도 표현합니다.

* Bounded function 유계함수

https://light-tree.tistory.com/188

https://en.wikipedia.org/wiki/Lipschitz_continuity