**Ensemble synergy Deep Active learning**

**: With Coreset, pseudo labeling and BNN**

GSDS 석사재학 20224314

강현구

최근Deep learning(이하 DL)모델은 많은 양의 Labeled Data를 기반으로 다양한 분야에서 높은 성능을 이끌어 냈다. 하지만 모델이 필요로 하는 데이터가 증가함에 따라, Labeling에 필요한 Human resource 또한 비례하여 Labeling Cost가 많이 든다는 문제점이 있다. 이와 관련하여 Labeling 했을 때 모델 학습에 효과적인 Unlabeled Data를 Sampling을 하는 Active learning(이하 AL)은 Labeling cost를 줄이는데 효과적이다. 이에 따라서 DL모델의 성능을 유지하면서 Active learning를 적용하고자 하는 Deep Active learning(이하 DAL)이 부각된다.

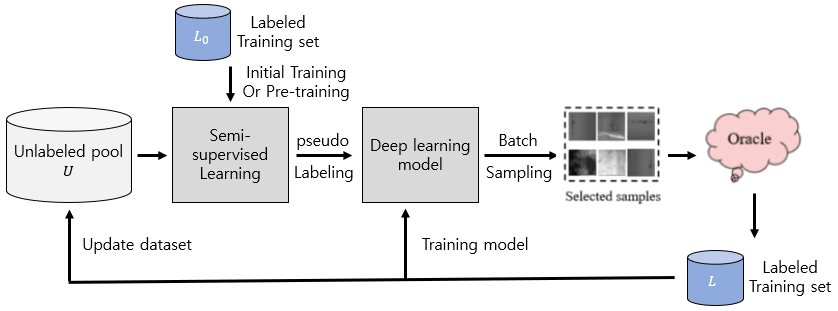
DAL에서 1개의 unlabeled 데이터를 Sampling 하는 것은 모델 학습에 영향을 거의 미치지 못한다. 따라서 Batch단위로 Sampling하는 방법이 필요하다. 또한 DAL에서 중요하게 다뤄지는 문제 중 하나는 'Insufficient data for labeled samples' 이다. 따라서 Labeled data 외에 추가적인 정보들을 고려하여 적은 양의 Labeled 데이터만으로도 효율적으로 모델을 학습할 필요가 있다.

최근의 Batch sampling의 주요 연구 동향은 크게 3가지, 1) Uncertainty-based and Hybrid Query Methods, 2) Deep Bayesian Active learning(이하 DBAL) Methods, 3) Density Based Methods가 있다(Ren et al., 2021). 첫번째 방식은Traditional 모델(ex- SVM, KNN)을 통해 uncertainty를 계산하며 이를 기준으로 Sampling한다. 하지만 traditional 방식이 모델의 uncertainty를 잘 반영하지 못한다는 한계가 있다. DBAL은 Bayesian Neural Network(이하 BNN)을 통해 모델의 Uncertainty를 보다 정확하게 측정한다. 마지막으로 Density Based method는 Original dataset의 feature space의 분포를 고려하여 Sampling 하며, 대표적으로 Core set(Sener & Savarese, 2017)가 있다. 한편 Labeled Data의 빈자리를 채우기 위해 DL방법을 활용한다. 예로 들어 Cost-Effective Active learning(CEAL)(Wang, Zhang, Li, Zhang, & Lin, 2016)에서는 self-paced learning을 통해 CNN과 Batch sampling 방식간 inconsistency를 해소하여 같이 학습될 수 있도록 했으며, Pseudo label을 활용하여 Labeled Data의 수를 늘렸다. 또한 Weighted Incremental Dictionary Learning(WI-DL)(Liu, Zhang, & Eom, 2016)은 Sparse representation을 통해 부족한 Labeled Data를 대체했다. 그리고 Generative Adversarial Active Learning(GAAL)(Zhu & Bento, 2017)에서는 GAN 모델을 활용하여 정보가 많은 sample들을 만들었다.

Batch sampling은 학습 모델의 성능에 의존한다. 따라서 Data expansion을 통한 학습 모델의 성능을 이끌어 내는 것은, 결국 Batch Sampling에 영향을 미치기 때문에 함께 고려할 필요가 있다. 하지만 각각의 방법에 대한 연구는 많이 되었으나, 각 방법 간의 관련성 및 시너지에 대한 연구는 부족하다. 위 접근법을 취한 연구로는 Simeoni의 연구(Siméoni, Budnik, Avrithis, & Gravier, 2021)에서 pseudo labeling 및 pre-training 방법과 다양한 Batch Sampling 방식을 orthogonal하게 적용하여 관련성을 확인했다. 하지만 Simeoni의 연구는 각 방법을 Orthogonal 하게 구성했다는 한계점이 있다. 이로 인해 각 과정이 별개로 수행되어 계산량이 가중되었으며, 각 방법 간의 시너지를 충분히 고려하지 못했다.

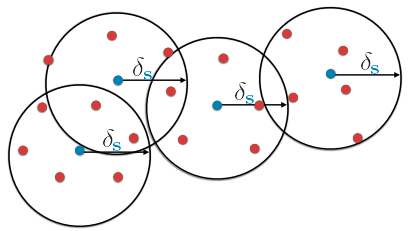
이 논문에서는Batch Sampling, Data Expansion, 더 나아가 Learning Model간의 시너지를 낼 수 있음을 보일 것이다. 앞서 WI-DL 연구(Wang et al., 2016)에서 Batch Sampling 간 Learning Model인 BNN의 결과를 재활용했다. 이처럼 각 방식들의 결과물을 재활용하는 형태로 추가적인 계산과정없이 성능을 향상시킬 수 있을 것이다. 이번 연구에선 Batch sampling과 BNN과의 연계 뿐만 아니라, Coreset과 Pseudo labeling의 연계 또한 확인할 것이다. Simeoni의 연구(Siméoni et al., 2021)에서 pseudo label을 통한 학습 과정 간 Batch 방식은 크게 상관이 없고 오히려 Random Samping의 결과가 더 좋았다고 하나, 이는 각 단계를 orthogonal 하게 적용했기 때문으로 판단된다. Coreset은 데이터의 특성을 잘 반영하는 것으로 알려진 바(Ren et al., 2021), 이를 활용한다면 추가적인 계산없이 보다 정확한 pseudo label 부여가 가능할 것으로 기대된다.

1. Task Description
   1. Data set : MNIST, SVHN, CIFAR-10, CIFAR-10
   2. Notation
      1. n examples where := [n] := {1, …, n}
      2. Given a set X :=
      3. labels for training
      4. U : Unlabeled pool Dataset
      5. L : Labeled pool Dataset
      6. of labels , a set if indices with
      7. Batch
      8. ~~jth cycle Label sample~~
      9. ~~j th cycle unLabel sample~~
      10. ~~C := [c] is a set of c classes~~
      11. ~~Cycle j = {0, 1, …}~~
   3. ~~함수 정의~~ 
      1. ~~The classifier with Parameter . maps new examples to a vector of probabilities per class. Given x , its prediction is the calss of maximum probability~~
      2. ~~where is the k-th element of vector~~ **~~p~~** ~~:= .~~
      3. ~~Embedding function , mapping an example x to a feature vector~~
      4. ~~Cost function , where cross-entropy~~
   4. Naiive Model



* + 1. Phase 1 : Random sampling Labeled Training set
    2. Phase 2 : Pre-training
    3. Phase 3 : Generate Pseudo labeling by SSL
    4. Phase 4 : finetune Deep learning model
    5. Phase 5 : Batch sampling with conclusion of Deep learning model
    6. Phase 6 : Get new labeled data from Oracle and update dataset
    7. Repeat phase 3 ~ 6 until 목표 성능까지 도달하기.

1. Related works
   1. Coreset Selection



* + 1. Coreset Selection은 전체 데이터를 덮을 수 있는 주어진 Batch 개수 점을 중심으로 subgraph들을 구성할 때, 최소한의 반지름(threshold)를 가지게 만드는 점들을 Sampling 하는 방법을 말한다. 추가로 Coreset selection 간 K-NN으로 활용할 수 있는 W matrix를 제공한다. W Matrix의 원소인 w\_ij는 j번째 data가 i번째 subgraph에 속했는가를 의미한다.
    2. Loss 함수
    3. Coreset Loss를 최소화하는 것은 K-Center Greedy 문제를 푸는 것과 같다.
    4. 
  1. BNN Uncertainty 개념 설명
     1. Yalin Gal의 Bayesian Deep learning{Gal, 2017 #8}연구에 따라 학습 모델을 설정하겠다. Bayesian CNN model의 근사는 Dropout과 같은 stochastic regularization techniques을 사용한 DNN 모델과 동일하다. 이를 통해서 Scalability하게 KL-divergence를 최소화시키는 distribution q\_\theta(w)을 찾겠다. Prior은 Gaussian distribution으로 가정한다.
     2. Max Entropy 를 계산하여 Coreset의 Loss 함수에 규제항으로 반영하겠다.

1. Idea Sketch
   1. Coreset Selection의 부산물인 W matrix를 활용하면 추가적인 계산없이 K-NN 기반 pseudo labeling이 가능하다. 이를 적용하여 계산양 증가없이 pseudo labeling을 통한 학습 모델의 성능을 이끌어 낼 수 있다. 또한 Coreset은 데이터의 특성을 잘 반영하는 것으로 알려진 바(Ren et al., 2021), 이를 활용한다면 추가적인 계산없이 보다 정확한 pseudo label 부여가 가능할 것으로 기대된다.
   2. Coreset selection 간 BNN의 Mat entropy를 같이 고려한다면, 보다 uncertainty가 높은 값을 중심으로 subgraph의 Center을 잡아 Batch sampling의 성능이 더욱 좋아질 것으로 기대된다.
2. 검증 내용
   * 1. SSL 방식과 Batch sampling 방식을 연계했을 때 성능 향상이 더 될 것인가?
        1. 계산 시간 외에 성능 향상이 추가적으로 될 것인가?
        2. SSL 방법은 각 방법간 성능 차이가 있을 수 있으니 Baseline으로 K-mean 기반 Pseudo labeling 방법과 Coreset W matrix 기반 Pseudo labeling 간 성능 차이를 비교한다.
        3. 만약 효과가 있다면 Coreset 기반 SSL의 성능 자체를 올릴 방안에 대해 고민한다.
     2. BNN의 Max entropy를 Coreset selection간 같이 고려한다면, 추가적인 성능 향상이 되는가?
        1. 따로 구분했을 때와, 연계했을 때의 성능 차이가 어떻게 되는가?
     3. SSL-Batch sampling의 연계와, BNN과 Coreset selection 각각의 연계는 서로 영향을 주는가?
     4. K-NN 기반(Coreset 포함) Pseudo labeling 방식에 따른 결과 차이.
        1. 전체 Data에 대해서 Pseudolabel을 넣었을 때 모델 자체가 오차를 학습할 가능성은 없는지 확인 필요
        2. 겹치는 부분과 서로 다른 labeling과 접하는 영역은 어떻게 설정해줄지
           1. 서로 같은 labeling에 둘러싸여 높은 신뢰도를 가진 경우, 진짜 labeling 하는 것은 어떤지?
        3. Pseudo labeling 간 확률에 Epsilon을 대입했을 때의 결과 차이
        4. K-NN 기존 방식은 거리에 기반하여 가중치를 부여한 것으로 보이는데, 이를 원의 겹침 유무로 판단해줄 대체 방안은 없을까? 고민해볼 것
     5. Iv. 과 연계하여 적절한 certainty를 반영하기 위한 조건 탐색하기. 그리고 해당 조건과 아닐 때의 효과 차이 비교 필요 - % 기반 : Sensitive Analysis
        1. Coreset 특성 상 Dateset의 분포를 고려해줄 필요가 있음.
           1. 혹은 Dateset의 분포를 Normalized 한다거나 그럴 방안이 없을까?
           2. Kernel, t-SNE 뭔가 연계할 게 있을 것 같은데. AE의 Latent variable로 하나의 방법으로 삼으면 되겠다.
        2. Dataset의 분산에 비례하여 threshold를 설정해줘야할 듯.
     6. ~~AL-BNN 간 ‘몇 번을 수행해야’ 원하는 수준의 성능을 구할 수 있는가?~~ 
        1. ~~BNN의 Uncertainty를 기반으로 학습을 멈춰야 할 때를 확인할 수 있는가?~~
     7. Hyperparameter 간의 상관관계에 대해 탐구하고, 이를 하나로 합쳤을 때 기존의 성능을 충분히 내는가?
     8. ~~Dropout을 통해서 BNN의 계산양을 scalability하게 바꾸자. 근데 이건 기본이려나.~~ 
        1. ~~이건 BNN 적용 간에 고민할 점으로, 내가 고민할 필요는 없을 듯.~~
     9. ~~Cycle 횟수에 따른 Semi, Pre training의 효과 차이 구분. 상황에 따라서 서로 다른 전략을 취하는 게 좋을 수 있겠다. 이건 부수적인 목표!~~
   1. 성능 측정
      1. Ablation test
      2. 적은 양의 label로 test error 많이 낮추기
   2. 실험조건
      1. Unsupervised Pre-training 하기
   3. 용어 표기 => 이건 바로 Overleaf에 기입하자. Word에 했던게 호환을 안하면 곤란함.ㅋㅋㅋㅋㅋㅋㅋㅋ 호환 안된다…! Overleaf에 다시 기입하자!

해야 할 것

* Unsupervised feature-learning 방법 검토하기.
  + Coreset과 연계하는 것도 하나의 방법. 그런데 이건 Option
  + 굳이 USL에서 K-mean을 안써도 됨. Pretraining 방법 쪽 확인한 다음 변경도 고민해볼 것! Active learning은 representation learning에 민감하다니까 더더욱 고민해볼 것.
  + Auto encoder을 쓰면 Coreset 등의 방식은 input x에 써도 되고 h에도 써도 되는 건가?
* Rethinking의 Dataset과 코드 얻기
* K-NN 기반 Pseudo labeling 방식 확인해보기
* ~~DAL에서 Query와 학습 모델의 프로세스를 통일 시킨 논문이 있는지 확인해보기 : 요건 범위에서 벗어난다~~
* ~~Adversarial learning 과 Coreset의 차이 확인하기. 혹 전자가 더 좋다면 대체하는 것도 고민해볼 것. 관련성이 없어 보이는데? Adversarial은 AI 공격 수단임~~
* Inductive label propagation과 Transductive label propagation 간의 차이가 어떤 결과의 차이를 만드는지 확인하기
  + 레이블이 없는 데이터의 레이블링을 목적으로 하는 transductive learning과 새로운 데이터에 대한 예측을 잘하는 것을 목적으로 하는 inductive learning이 있다.
  + -> 여기선 Transductive learning이지. 새로운 데이터가 추가 안되니까. 그런데 왜 inductive label propagation이라고 하고 있지?
* Label propagation과 Semi-supervised learning 간 차이 확인하기
  + Label propagation은 Graph에서 Message passing과 유사한 느낌
  + 일단 Pseudo labeling을 부여하는 방법 정도로 알고 있고, SSL 안에 속해있는 분파로 보는게 맞을 듯.
* SSL 방법을 Coreset 기반으로 변경할 것. 이때 SSL 방법 차이로 인한 성능 향상인지, Acquisition 방법과 SSL 방법의 연계를 통한 성능 향상인지 구분 필요하다.
  + 이때 Coreset 방식으로도 각 Pseudo labeling과 weight를 정량적으로 잘 표현할 수 있는지 검토하기. 기존 방식만큼 구체적으로 정량화 할 수 있는지가 Backup이 되야함.
  + 기존에는 Distance 기반으로 중간 가중치를 고려할 수 있었음. 하지만 Coreset은 포함되어 있나 아닌가만 반영함.
  + -> 오히려 Coreset의 성능이 떨어질 수 있음. Coreset의 성능 보장을 위한 추가 고민 필요.
  + 두가지 SSL에 대해서 서로 다른 Acquisition 방법을 비교하는 걸로 가능할 듯! 하나는 K-mean, 하나는 Core-set 기반으로!

**<참고문헌>**

Liu, P., Zhang, H., & Eom, K. B. (2016). Active deep learning for classification of hyperspectral images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 10*(2), 712-724.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., . . . Wang, X. (2021). A survey of deep active learning. *ACM computing surveys (CSUR), 54*(9), 1-40.

Sener, O., & Savarese, S. (2017). Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Siméoni, O., Budnik, M., Avrithis, Y., & Gravier, G. (2021). *Rethinking deep active learning: Using unlabeled data at model training.* Paper presented at the 2020 25th International Conference on Pattern Recognition (ICPR).

Wang, K., Zhang, D., Li, Y., Zhang, R., & Lin, L. (2016). Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology, 27*(12), 2591-2600.

Zhu, J.-J., & Bento, J. (2017). Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*.