

SNU MPR LAB SCREENING REPORT - COMPUTER VISION

Name: Jungmin Seo

Major: SNU Electrical & Computer Engineering 20

Email: [jmseo1204@snu.ac.kr](mailto:jmse1204@snu.ac.kr)

1. Introduction

Noisy Label Classification 문제를 일반화시킬 수 있는 training 방법을 고안하는 게 이번 screening 의 핵심이었습니다. 본 연구는 앙상블 모델을 이용한 효율적인 노이즈 레이블 학습 기법을 제안합니다. 제안하는 모델은 간소화된 GoogLeNet 과 ResNet 두 가지 base model 을 기반으로 하며, 학습 과정에서 co-teaching 방법을 사용하여 두 모델의 상호 보완적인 학습을 유도합니다. 최종적으로는 두 모델의 가중치를 momentum 으로 학습하여 최적의 앙상블 모델을 구축하였습니다. 이 보고서는 모델 구조, 하이퍼파라미터 설정, 시기별 학습 전략, 추가적인 정규화 기법, 그리고 개선 방안에 대해 기술했습니다.

학부 강의에서 이 문제를 마주했으면 hyperparameter 조정에 많은 시간을 투자하여 accuracy 를 높이는 것에 집중했겠지만, 제한된 시간 동안 치루는 screening 임을 고려하여 여러 논문에서 주장하는 다양한 방법을 이해하고 코드로 구현하는 것에 노력했습니다.

2. Description of your model

a. Model architecture

a-1. Structure

본 연구에서 사용한 앙상블 모델은 간소화된 GoogLeNet 과 ResNet 으로 구성됩니다. 두 모델은 학습 과정에서 co-teaching 기법을 통해 동시에 학습되며, 각 모델의 손실(loss)을 비교하여 최종적으로 앙상블할 가중치를 running average 로 학습합니다. 이는 두 모델의 복잡도에 따른 학습 속도의 차이를 보정하기 위함입니다. 테스트 단계에서는 학습한 앙상블 가중치를 사용하여 두 모델의 예측을 가중합하여 최종 레이블을 예측합니다.

a-2. Hyperparameter setting

batch_size 는 노이즈 레이블 학습에서 모델이 robust 하다는 연구 결과를 바탕으로 256 으로 설정하였습니다
【Rolnick et al., 2017】¹.

learning_rate 는 여러 테스트를 거쳐 학습 속도가 빠른 0.0005 로 결정하였습니다.

1 Rolnick, D., Veit, A., Belongie, S., & Shavit, N. (2017). Deep learning is robust to massive label noise

ELR_loss 와 target_embedding running average momentum 은 학습 전략 챕터에서 설명할 연구에서 사용한 수치를 참고하였으나, 본 문제의 노이즈 비율이 해당 연구보다 낮아 더 보수적인 값으로 설정하였습니다.

b. What is new in your own?

적은 epoch 으로 높은 일반화 성능을 보이는 모델을 설계해야했기에 epoch 시기 별로 noisy label 에 대응하기 위한 각기 다른 training 방법을 사용했습니다.

초기 학습 (Epoch ≤ 5)

초기 학습 단계에서는 label smoothing 을 사용하였습니다 【Müller et al., 2019】². 초기 학습 시 두 모델의 판단력이 부족하여 노이즈 레이블을 처리하기 위한 효과적인 방법이 없다고 판단하였습니다. 따라서 각 모델을 기초 수준까지 독립적으로 학습시켰으며, EDA 과정에서 확인된 약 10%의 노이즈 비율에 맞춰 10%의 label smoothing 을 적용하였습니다.

중기 학습 (5 < Epoch ≤ 12)

중기 학습 단계에서는 co-teaching 기법을 사용하였습니다 【Han et al., 2018】³. 초기 학습을 통해 모델이 판단력을 갖게 되면, false 데이터에 대한 loss 가 true 데이터에 대한 loss 보다 클 것이라 추측하였습니다. 이에 전체 데이터의 10% 가 false 데이터임을 가정하고, 각 batch 에서 손실이 낮은 하위 90% 데이터를 true 데이터로 간주하여 필터링하는 small loss selection 기법을 사용했습니다. 이 과정에서 model1 에 의해 필터링된 데이터를 model2 가 학습하고, 역으로 model2 에 의해 필터링된 데이터는 model1 이 학습하여 과적합을 억제하였습니다.

말기 학습 (12 < Epoch)

말기 학습 단계에서는 early learning regularization(ELR_loss)을 추가하였습니다 학습 초기에는 false 데이터의 레이블을 잘못된 것으로 예측하면서 높은 정확도를 보이지만, 학습이 진행됨에 따라 false 데이터를 과적합하여 정확도가 감소하는 현상을 【Liu et al., 2020】⁴.에서 확인할 수 있었습니다. 따라서 ELR_loss 를 기존 손실에 추가하여 과적합을 방지하였습니다. ELR loss 는 정규화 term 이기에 학습 초기에 추가하면 모델 학습을 방해할 것이라 생각했습니다. 따라서 모델 학습 속도가 더딘 마지막 단계에 도달했을 때 추가해주는 방식으로 모델을 설계했습니다.

2 Müller et al., When Does Label Smoothing Help?, NeurIPS, 2019

3 Han et al., Co-teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels, NIPS, 2018

4 Liu et al., Early-Learning Regularization Prevents Memorization of Noisy Labels, 2020

그 외에 추가적으로 robust training 을 위해 적용한 정규화는 다음과 같습니다.

- Data augmentation

epoch<7 에서는 상대적으로 변형이 적은 horizontal flip, ColorJitter 만 변형을 주었고, epoch>=7 에서는 epoch 에 따라 progressive 하게 ColorJitter, rotate, crop 의 분산 증가시켜 데이터를 확률적으로 변형하였습니다.

3. Performance of your model (accuracy)

- test dataset accuracy **61%**

- ./model_google_res_ver2.pt

4. Things to improve

1) Semi-supervised learning

Training data EDA 과정에서 약 10000 개의 label 이 None 임을 확인했습니다. 이에 semi-supervised learning 을 하려고 했으나 pseudo labeling 을 할 만큼 기본 모델의 acc 가 높지 않았기 때문에 준지도학습을 하면 오히려 label noise 가 더 증가하는 꼴이 될 것으로 추측했습니다. 따라서 기본 모델 성능이 높게 나오지 않는 이상 label 이 None 인 데이터는 학습에 사용하지 않기로 결정했다. 그러나 주어진 시간이 넉넉하다면 【Xie et al., 2020】⁵ 논문에서 제시하는 teacher 모델의 pseudo label 을 활용한 student 모델의 교차 학습으로 None labeled data 를 활용할 수 있을 것이라 생각합니다.

2) 높은 성능의 base model 선정

48 시간 안에 관련 research 를 조사하고 여러 테스트를 하며 높은 성능을 내는 하이퍼파라미터까지 찾아야했습니다. 때문에 training epoch 이 적을 수밖에 없었고, 경량화된 모델을 사용하는 것이 현명하다 생각했습니다. 처음에는 CIFAR100 에서 가장 높은 성능을 보인 EfficientNetV2-L 을 채택했는데, 파라미터 수가 많아서인지 여타 경량모델 만큼의 학습 속도가 나와주지 않아 사용하지 않았습니다. 그나마 가벼운 googlenet 과 resnet 을 base model 로 정했지만 이 두 모델이 최적의 성능을 발휘하는 모델인지 확인하지 못했기에 base model 을 변경 혹은 개선 시 모델 성능이 향상될 것이라 추측합니다.

5 Xie et al., Self-training with Noisy Student improves ImageNet classification, CVPR, 2020