라벨 노이즈 환경에서 확률분포 예측 성능 향상 방법

노준호* · 우승범 · 황원준

아주대학교

Probability distribution predicted performance improvement in noisy label

Jun-ho Roh* · Seung-beom Woo · Won-jun Hwang

Ajou University

E-mail : forrjh@ajou.ac.kr / enpko47@ajou.ac.kr / wjhwang@ajou.ac.kr

요 약

지도학습에서 모델을 학습함에 있어 입력 데이터와 해당 데이터의 라벨이 필요하다. 하지만 신뢰성 있는 라벨링은 비용과 시간적인 면에서 많이 소요되며 이를 자동화할 경우 라벨이 언제나 맞는다는 보장이 없어 노이즈가 들어가게 된다. 이러한 라벨 노이즈 환경에서 지도학습을 진행할 경우 모델은 학습 초기에는 정확도가 올라가지만, 어느 정도 학습 후 정확도가 크게 감소되는 경향을 보인다. 라벨 노이즈 문제를 해결하기 위해 다양한 방법이 있지만, 대다수의 경우 모델이 예측한 확률을 수도라벨로 사용해 이용하는 경우가 많다. 여기에 대해서 우리는 모델이 예측한 확률을 정제하여 좀 더 빠르게 참 라벨을 예측하는 방법을 제시한다. 기존의 논문 중 모델이 예측한 확률을 사용하는 방법에 우리가 제안하는 방법을 적용하여 같은 환경, 데이터셋에 대해 실험을 진행한 결과 성능개선과 더 빠르게 수렴하는 것을 확인할 수 있었다. 이를 통해 기존 연구들 중 모델이 예측하는 확률분포를 사용하는 방법들에 적용할 수 있고 같은 환경에서도 더 빠르게 수렴시킬 수 있기에 학습 소요시간을 줄일 수 있다.

ABSTRACT

When learning a model in supervised learning, input data and the label of the data are required. However, labeling is high cost task and if automated, there is no guarantee that the label will always be correct. In the case of supervised learning in such a noisy labels environment, the accuracy of the model increases at the initial stage of learning, but decrease significantly after a certain period of time. There are various methods to solve the noisy label problem. But in most cases, the probability predicted by the model is used as the pseudo label. So, we proposed a method to predict the true label more quickly by refining the probabilities predicted by the model. Result of experiments on the same environment and dataset, it was confirmed that the performance improved and converged faster. Through this, it can be applied to methods that use the probability distribution predicted by the model among existing studies. And it is possible to reduce the time required for learning because it can converge faster in the same environment.

키워드

Noisy Label, Artificial Intelligence, Deep Learning, Convolutional Neural Network

Ⅰ. 서 론

지도학습에서 모델을 학습할 때 입력 데이터와 해당 데이터의 라벨이 필요하다. 데이터를 사람이 라벨링하는 작업은 라벨이 높은 신뢰도를 갖추기 위해 많은 비용과 시간을 들이고 있다. 이에 대한 대안으로 Active learning, Auto ML등의 방식이 있지만 이는 noisy labels이 포함될 여지가 높다. 또한 시간적으로나 인력 부족으로 충분한 검증을 거치지 못하여 데이터셋에 noisy labels이 포함될 수 있다. 모델은 이러한 noisy labels에 쉽게 overfitting되며 성능을 매우 감소시킨다. 따라서 Noise Label 분야는 데이터셋에서 라벨에 오류가 있는 상황에서 학습을 하며 noisy labels에 overfit되는 것을 방지한다. 이를 통해 데이터셋을 만들 때 소요되는 비용을 줄일 수 있고 noise에

* corresponding author

Robust하여 성능 향상을 기대할 수 있다.

기존 연구들 중 데이터셋에 noise가 존재하기 때문에 모델이 데이터에 대해 예측한 값을 신뢰하여 pseudo라벨로 사용하는 경우[1, 2]가 있다. 이러한 연구들은 모델의 예측값을 그대로 사용하거나 또는 이전의 예측값과 현재의 예측값을 더해 사용하는 방법[3]이 있다. 우리는 모델의 예측값을 정제하여 target probability를 더 빨리 정확하게 예측하는 방법을 제안한다. 제안된 방법을 사용하여 성능 향상과 더 빠른 수렴 속도를 보인다.

Noise는 기본적으로 4종류[4]가 존재하며 각각 Uniform Noise, Class-Dependent Noise, Locally-Concentrated, Feature-Dependent Noise이라 한다. Uniform Noise는 uniform한 확률로 라벨이 변경된 것으로 10개의 클래스가 있다면 각 라벨로 선택될 확률은 10%가 된다. 만약 원래의 clean label을 제외하고 계산될 수 있으며 이때에는 1/9의 확률을 갖게 된다. 보통 이러한 Noise를 Symmetric Noise라고 한다. Class-Dependent Noise는 Noisy label로 바뀔 때 특정 클래스로 경향성을 보이는 Noise이다. 예를 들어, 고양이의 라벨이 개라고 변경되는 것과 개라고 되어있는 라벨이 고양이로 변경될 경우이며 보통 Asymmetric Noise라고 표현한다. Locally-Concentrated는 공간상 일정 지역에 있는 데이터들의 라벨들이 Noise 상태이며 마지막으로 Feature-Dependent Noise는 Boundary 근처 공간에 있는 데이터들의 라벨들이 Noise 상태이다.

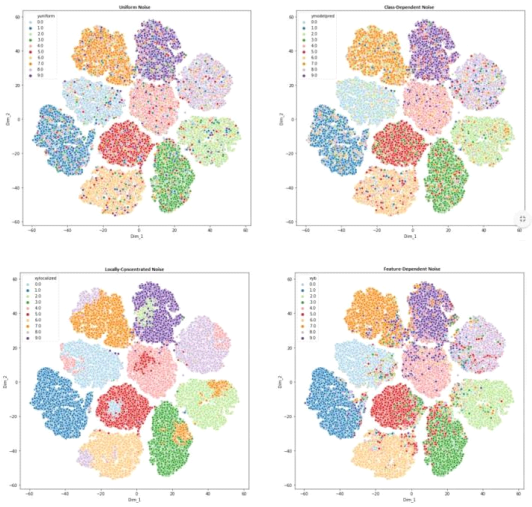
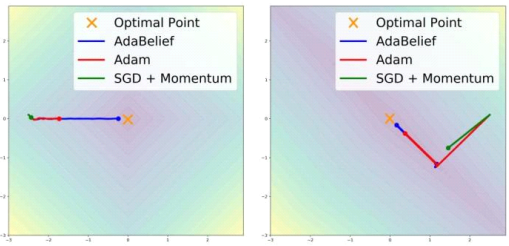


Fig. 1. Noise types[4], left top to clockwise Uniform Noise, Class-Dependent Noise, Locally-Concentrated, Feature-Dependent Noise

II. 본 론

본 장에서는 기반 논문에 대한 설명과 제안 방법 및 효과에 대하여 설명한다.

Noisy labels 환경에서 학습을 진행한다면 학습 초기에는 성능이 향상되다가 어느 정도 학습이 진행되면 성능이 떨어지는 현상이 발생한다. 논문[5]에서는 이 현상이 발생하는 이유를 모델이 noisy labels을 memorization 하여 발생한다고 말한다. 이를 방지하기 위해 Regularization을 넣어 clean labels의 gradient는 강하게 유지하고 noisy labels의 gradient들은 중화하여 memorization 현상을 방지한다. 여기서 라벨들의 실제 방향을 모델의 예측값을 사용하여 계산하며 이전의 예측과 지금의 예측값을 더하여 사용하고 있다. 이전의 예측과 지금의 예측값을 더하여 사용하는 것을 momentum으로 생각했을 때 optimizer의 변천처럼 momentum optimizer 이후로 나온 알고리즘을 사용하여 예측값을 정제한다면 해당 optimizer와 동일한 효과를 낼 수 있다고 생각하였고 우리는 여기서 Adam의 second momentum 개념을 도입하여 모델의 예측값을 정제하여 사용하였다. Adam과 momentum의 비교를 본다면 Optimal Point에 Adam이 먼저 수렴에 가까워지는 것처럼 예측값도 실제 clean 데이터의 예측값에 더 빠르게 수렴에 가까워질 것이다.



(a) loss function is $f(x,y) = |x| + |y|$ (b) $f(x,y) = |x+y| + |x-y|\,/\,10$

Fig. 2. Compare Momentum and Adam optimizer[6]

2.1 제안 방법

기존 연구[5]에서 Memorization을 방지하기 위해 Regularization을 제안하였다. n개의 학습 데이터 $\{x_i, y_i\}$에 대해 $x_i \in R^d$는 i번째 입력 데이터이며 $y_i \in \{0,1\}^C$는 i번째 데이터의 라벨이다. 네트워크 N을 통해 예측한 확률 p_i는 softmax function Θ를 사용하여 아래와 같이 표현할 수 있다.

$$p_i = \Theta(N(x_i)) \tag{1}$$

Cross Entropy Loss는 아래와 같이 표현할 수 있다.

$$L_{CE} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c}^{C} y_i^c \log\left(p_i^c\right) \quad (2)$$

모델이 noisy labels를 memorize하는 것을 방지하기 위해 Regularization을 사용하며 모델의 출력과 target의 내적을 최대화한다.

$$L_{ELR} = \frac{1}{n}\sum_{i=1}^{n}\left(1 - < p_i, t_i >\right) \quad (3)$$

따라서 Total Loss는 아래와 같다. 여기서 는 Regularization 반영 비율이다.

$$L_T = L_{CE} + \lambda L_{ELR} \quad (4)$$

식 (3)에서 t_i는

$$t_i = \beta t_i + (1-\beta) p_i \quad (5)$$

으로 t_i를 다음과 같이 정제한다.

$$M = \beta_1 M + (1-\beta_1) p_i \quad (6)$$

위의 식으로 first momentum을 구하고

$$V = \beta_2 V + (1-\beta_2) p_i^2 \quad (7)$$

위의 식으로 second momentum을 구한다. (6)과 (7)을 통해

$$t_i = \Theta\left(\frac{M}{\sqrt{V}+\epsilon} + \phi t_i\right) \quad (8)$$

식(8)으로 정제하여 target probability를 추정한다. 여기서 ϕ는 decay rate이다.

III. 실 험

본 장에서는 실험을 위해 구성한 데이터셋과 모델, 하이퍼 파라메터를 설명하고 실험 결과를 보여준다.

3.1 데이터셋 구성

우리는 CIFAR-10과 CIFAR-100[7] 두 데이터셋을 사용하여 실험을 진행했으며 50K의 학습 이미지와 10K의 테스트 이미지가 있으며 각 이미지의 사이즈는 (32×32)이다. Symmetric과 Asymmetric 두 종류의 label noise에 대해 실험을 하며 Symmetric노이즈는 랜덤하게 라벨을 대체하였다. Asymmetric노이즈는 논문[8]과 같은 설정을 하였으며 CIFAR-10의 경우 truck은 automobile, bird는 airplane, deer는 horse, cat은 dog, dog는 cat으로 대체된다. CIFAR-100 또한 논문[8]에 따라 대체된다.

실험환경은 논문[5]과 같은 환경으로 구성하기 위해 데이터에 random crop과 random horizontal flip을 사용하였고 batch size는 128이다. Optimizer로 SGD에 0.9의 momentum, 0.001의 weight decay를 사용하였다. 초기 lr은 0.02이다.

3.2 실험 결과

먼저 CIFAR 10에 대한 실험 결과표를 보면 행은 각각의 방법을 나타내고 있고 열은 데이터의 노이즈 종류와 노이즈 비율을 나타낸다. 기존 연구들의 결과는 논문[5]에서 가져왔다.

Table 1. Symmetric noise table

Dataset	Method	Symmetric Noise		
		20%	40%	60%
CIFAR 10	CE	86.98 ±0.12	81.88 ±0.29	74.14 ±0.56
	Bootstrap[9]	86.23 ±0.23	82.23 ±0.37	75.12 ±0.56
	Forward[8]	87.99 ±0.36	83.25 ±0.38	74.96 ±0.65
	GSE[10]	89.83 ±0.20	87.13 ±0.22	82.54 ±0.23
	SL[11]	89.83 ±0.32	87.13 ±0.26	82.81 ±0.61
	ELR*[5]	92.12 ±0.35	91.43 ±0.21	88.87 ±0.24
	Ours	93.42 ±0.46	92.12 ±0.28	88.74 ±0.31

Table 2. Asymmetric noise table

Dataset	Method	Asymmetric Noise		
		10%	20%	30%
CIFAR 10	CE	90.69 ±0.17	88.59 ±0.34	86.14 ±0.40
	Bootstrap[9]	90.32 ±0.21	88.26 ±0.24	86.57 ±0.35
	Forward[8]	90.52 ±0.26	89.09 ±0.47	86.79 ±0.36
	GSE[10]	90.91 ±0.22	89.33 ±0.17	85.45 ±0.74
	SL[11]	91.72 ±0.31	90.44 ±0.27	88.48 ±0.46
	ELR*[5]	94.57 ±0.23	93.28 ±0.19	92.70 ±0.41
	Ours	94.26 ±0.3	93.59 ±0.24	92.65 ±0.47

표1, 표2에서 *은 Learning rate 스케줄러가 pytorch기준으로 MultistepLR이 아니라 Cosine Annealing Warm Restarts라는 스케줄러를 사용한다. 해당 스케줄러를 사용할 경우 더 높은 성능이 나왔으므로 본 논문에서도 마찬가지로 *환경을 구성하여 실험을 진행하였다. CIFAR-10환경에서는 Base 논문[5]보다 낮은 노이즈 비율을 갖는 부분에서 좀 더 성능이 좋게 나오는 것을 볼 수 있다. 아래 Fig 3은 CIFAR-100의 Symmetric Noise 60%에서 의 값에 따른 결과를 비교한다. 파랑 점선은 논문[5]에서의 결과이며 x축은 람다값이며 y축은 정확도이다.

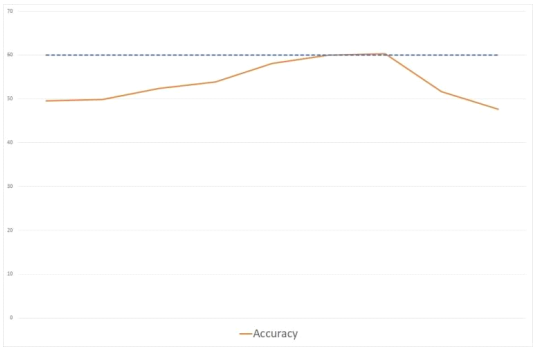


Fig. 3. Accuracy graph. Blue dotted is ELR*[5], Orange is our method when Symmetric noise 60% at CIFAR-100

Ⅳ. 결 론

본 논문에서는 기존 연구들에서 Pseudo label을 사용할 때 단순히 모델의 예측값을 사용하기보단 정제하여 사용할 경우 어느 정도의 성능 향상과 수렴에 필요한 epoch 수를 줄여주는 효과를 보았다. 하지만 성능 향상 부분의 경우 optimizer의 차이처럼 더 빠르게 optimal point에 수렴하여 이렇게 보이는 경우일 수 있다. 왜냐하면 기반 논문[5]에서 epoch 수는 150 epoch으로 만약 추가적으로 더 학습을 진행한다면 본 논문의 결과와 같은 결과를 보일 수 있다고 생각한다. 하지만 이렇게 정제를 통해 epoch 수를 감소시킬 수 있다.

본 논문에서는 정제하기 위해 Adam optimizer와 같은 방법으로 진행했지만 Adam optimizer 이후에 나온 최신 optimizer의 방법대로 정제를 할 경우 추가적인 개선이 가능하다고 생각한다.

Acknowledgement

본 연구는 NRF-2020R1F1A1066049 연구과제의 지원을 받아 수행되었습니다.

References

[1] Han, J., Luo, P., & Wang, X. (2019). Deep self-learning from noisy labels. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 5138-5147).

[2] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Workshop on Challenges in Representation Learning, ICML, volume 3, page 2, 2013.

[3] Laine, S., & Aila, T. (2016). Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242.

[4] Algan, G., & Ulusoy, I. (2020). Label noise types and their effects on deep learning. arXiv preprint arXiv:2003.10471.

[5] Liu, S., Niles-Weed, J., Razavian, N., & Fernandez-Granda, C. (2020). Early-learning regularization prevents memorization of noisy labels. arXiv preprint arXiv:2007.00151.

[6] Zhuang, J., Tang, T., Tatikonda, S., Dvornek, N., Ding, Y., Papademetris, X., & Duncan, J. S. (2020). Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. arXiv preprint arXiv:2010.07468.

[7] Antonio Torralba, Rob Fergus, and William. T. Freeman. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008. https://ieeexplore.ieee.org/document/4531741/.

[8] Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., & Qu, L. (2017). Making deep neural networks robust to label noise: A loss correction approach. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1944-1952).

[9] Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., & Rabinovich, A. (2014). Training deep neural networks on noisy labels with bootstrapping. arXiv preprint arXiv:1412.6596.

[10] Zhang, Z., & Sabuncu, M. R. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. arXiv preprint arXiv:1805.07836.

[11] Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., & Bailey, J. (2019). Symmetric cross entropy for robust learning with noisy labels. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 322-330).