MixMatch: A Holistic Approach to Semi-Supervised Learning

2023. 06. 29

MixMatch (2019, NeurIPS)

❖ MixMatch: A Holistic Approach to Semi-Supervised Learning

- 2019년에 NeurIPS에 게재되었으며, 2023년 06월 29일 기준 2353회 인용됨
- 레이블이 없는 데이터에 대해 세 가지 손실 함수를 통합한 손실 함수를 사용하는 비지도 학습 방법론

MixMatch: A Holistic Approach to Semi-Supervised Learning

David Berthelot
Google Research
dberth@google.com

Nicholas Carlini
Google Research
ncarlini@google.com

Ian Goodfellow
Work done at Google
ian-academic@mailfence.com

Avital Oliver
Google Research
avitalo@google.com

Nicolas Papernot
Google Research
papernot@google.com

Colin Raffel
Google Research
craffel@google.com

Background

❖ 연구 배경

- 레이블 데이터가 많이 없고, 레이블을 달기 위해 전문가의 지식이 필요함
- 이로 인해, 많은 비용과 시간이 요구되는 한계점 존재

→ 그래서, 많은 비지도 학습 방법론은 레이블이 없는 데이터에 대해 계산되는 손실 함수 부분을 추가함

→ 주로, 세 가지로 분류 될 수 있다.

① Entropy minimization: 모델이 레이블이 없는 데이터에 대해 신뢰할 수 있는 예측을 위한 기법

② Consistency Regularization: 입력이 교란되었다 하더라도 동일한 출력 분포를 생산하도록 하기 위한 기법

③ Generic Regularization(Mix Up): 모델이 학습 데이터에 과적 합 되거나 일반화가 잘 되도록 하기 위한 기법

결국, MixMatch는 위의 세 가지 접근법을 통합해서 비지도 학습 기법을 적용한 방법론

Background

❖ Consistency Regularization

- 레이블이 없는 샘플에 서로 다른 데이터 증강 기법을 적용하여도 분류기의 출력되는 클래스의 분포는 동일하다는 것을 가정함
- MixMatch는 데이터 증강 기법으로 random horizontal flips와 crops 사용

Consistency Regularization 수식

$$\| \mathrm{p}_{\mathrm{model}}(y \mid \mathrm{Augment}(x); \theta) - \mathrm{p}_{\mathrm{model}}(y \mid \mathrm{Augment}(x); \theta) \|_2^2.$$

Parameter θ인 모델에 입력에 대한 클래스 레이블의 분포 생성

앞, 뒤는 다른 값을 출력, Augment()는 stochastic transformation

최소화 되도록

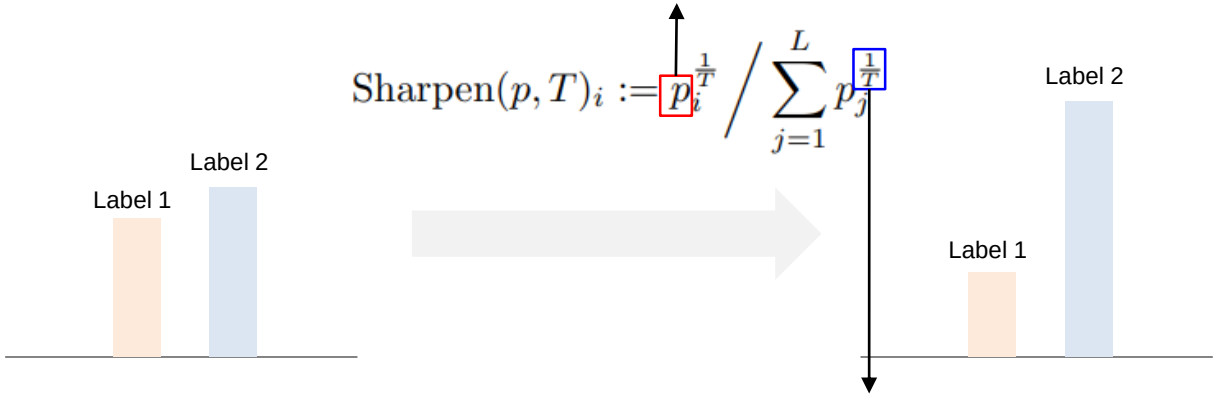
Background

❖ Entropy Minimization

- 가정: “분류기의 결정 경계는 주변 데이터의 분포의 고밀도 지역을 통과해서는 안됨”
- 즉, 가정이 성립하기 위해서 분류기가 레이블이 지정되지 않은 데이터에 대해서 낮은 entropy로 예측하도록 요구하면 됨
- MixMatch에서는 Entropy Minimization을 달성하기 위해서 “Sharpening” 활용
  - ➢ Sharpening은 레이블 분포의 entropy를 줄이기 위한 함수

레이블이 지정되지 않은 샘플에 데이터 증강 기법을 적용한 샘플 예측 분포의 평균

$$\text{Sharpen}(p, T)_i := p_i^{\frac{1}{T}} \Big/ \sum_{j=1}^{L} p_j^{\frac{1}{T}}$$



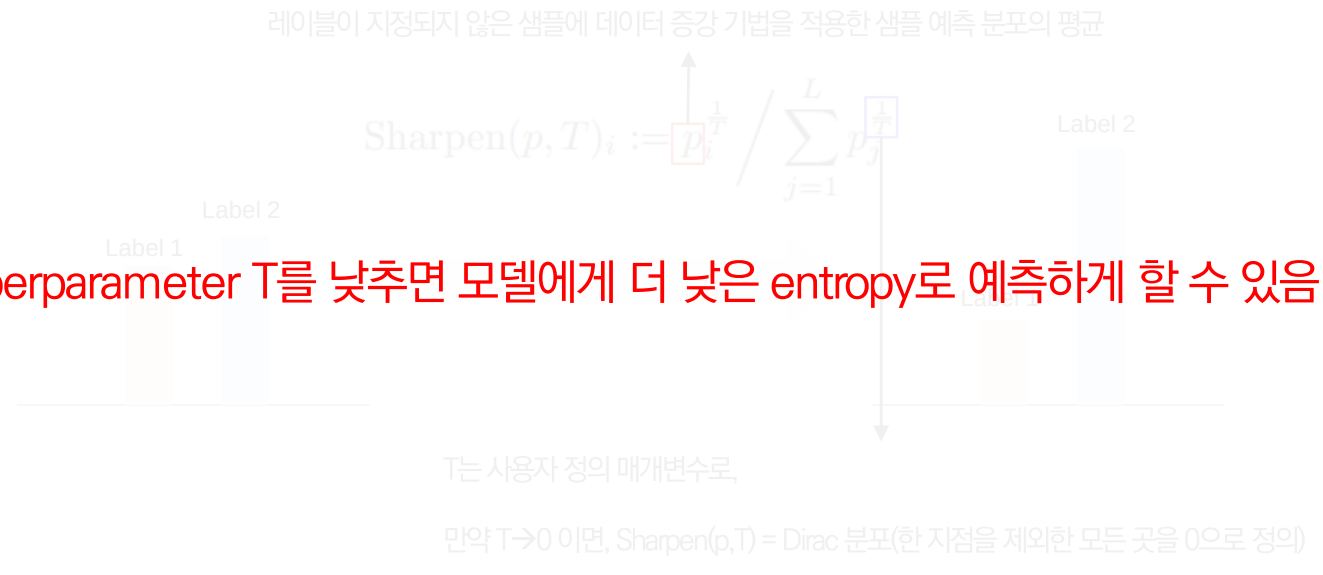
T는 사용자 정의 매개변수로,

만약 T→0 이면, Sharpen(p,T) = Dirac 분포(한 지점을 제외한 모든 곳을 0으로 정의)

Background

❖ Entropy Minimization

- 가정: “분류기의 결정 경계는 주변 데이터의 분포의 고밀도 지역을 통과해서는 안됨”
- 즉, 가정이 성립하기 위해서 분류기가 레이블이 지정되지 않은 데이터에 대해서 낮은 entropy로 예측하도록 요구하면 됨
- MixMatch에서는 Entropy Minimization을 달성하기 위해서 “Sharpening” 활용
  - ➢ Sharpening은 레이블 분포의 entropy를 줄이기 위한 함수



Hyperparameter T를 낮추면 모델에게 더 낮은 entropy로 예측하게 할 수 있음

Background

❖ Mix Up

- MixMatch에서는 기존과 달리 레이블이 있는 샘플과 레이블이 없는 샘플 모두 섞음
- 즉, 두 데이터와 라벨을 일정 비율로 섞어 새로운 데이터를 생성하는 방법론
- 레이블이 있는 샘플과 레이블이 없는 샘플이 한 배치 내에 있을 때, 개별 손실 구성 요소를 적절하게 계산하려면 배치의 순서를 유지해야함 → 이런 역할을 하게 해주는 것이 바로 추가된 부분

$$\lambda \sim \mathrm{Beta}(\alpha, \alpha)$$

$$\lambda' = \max(\lambda, 1-\lambda)$$

이 부분이 생략된 것이 Vanilla MixUp

$$\boldsymbol{x}' = \lambda' x_1 + (1-\lambda') x_2$$

$$\boldsymbol{p}' = \lambda' p_1 + (1-\lambda') p_2$$

Proposed Method

❖ Pseudo Code

- 각 샘플마다 데이터 증강 기법을 사용한 이유는 guessed label을 생성하기 위함

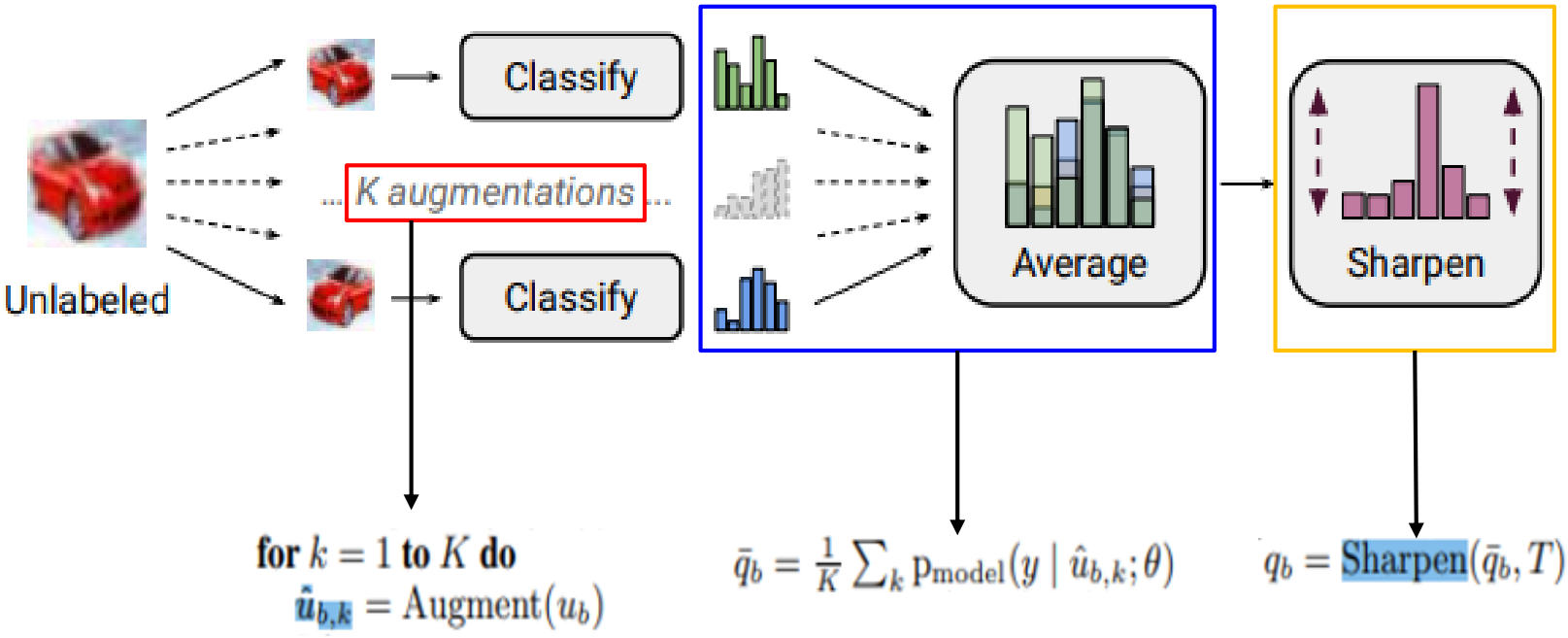
Algorithm 1 MixMatch takes a batch of labeled data $\mathcal{X}$ and a batch of unlabeled data $\mathcal{U}$ and produces a collection $\mathcal{X}'$ (resp. $\mathcal{U}'$) of processed labeled examples (resp. unlabeled with guessed labels).

Input: Batch of labeled examples and their one-hot labels $\mathcal{X} = \big((x_b, p_b); b \in (1, \dots, B)\big)$, batch of unlabeled examples $\mathcal{U} = \big(u_b; b \in (1, \dots, B)\big)$, sharpening temperature T, number of augmentations K, Beta distribution parameter α for MixUp.
for $b = 1$ **to** B **do**
  $\hat{x}_b = \text{Augment}(x_b)$ // *Apply data augmentation to* x_b
  for $k = 1$ **to** K **do**
    $\hat{u}_{b,k} = \text{Augment}(u_b)$ // *Apply* k^{th} *round of data augmentation to* u_b
  end for
  $\bar{q}_b = \frac{1}{K} \sum_k \text{p}_{\text{model}}(y \mid \hat{u}_{b,k}; \theta)$ // *Compute average predictions across all augmentations of* u_b
  $q_b = \text{Sharpen}(\bar{q}_b, T)$ // *Apply temperature sharpening to the average prediction (see eq. (7))*
end for
$\hat{\mathcal{X}} = \big((\hat{x}_b, p_b); b \in (1, \dots, B)\big)$ // *Augmented labeled examples and their labels*
$\hat{\mathcal{U}} = \big((\hat{u}_{b,k}, q_b); b \in (1, \dots, B), k \in (1, \dots, K)\big)$ // *Augmented unlabeled examples, guessed labels*
$\mathcal{W} = \text{Shuffle}\big(\text{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}})\big)$ // *Combine and shuffle labeled and unlabeled data*
$\mathcal{X}' = \big(\text{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \dots, |\hat{\mathcal{X}}|)\big)$ // *Apply* MixUp *to labeled data and entries from* $\mathcal{W}$
$\mathcal{U}' = \big(\text{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \dots, |\hat{\mathcal{U}}|)\big)$ // *Apply* MixUp *to unlabeled data and the rest of* $\mathcal{W}$
return $\mathcal{X}', \mathcal{U}'$

Label Guessing

Proposed Method

❖ Pseudo Code 도식화



Proposed Method

❖ Loss function

- 1번 loss function: X'로부터 모델 예측과 레이블 사이에 cross-entropy loss
- 2번 loss function: U'로부터 guessed label과 예측 값 사이에 squared L2 loss
- 3번 loss function: total loss로, 1번과 2번의 가중 합, 람다는 hyperparameter

$$\mathcal{X}', \mathcal{U}' = \text{MixMatch}(\mathcal{X}, \mathcal{U}, T, K, \alpha)$$

1. $$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \text{H}(p, \text{p}_{\text{model}}(y \mid x; \theta))$$

2. $$\mathcal{L}_{\mathcal{U}} = \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} \|q - \text{p}_{\text{model}}(y \mid u; \theta)\|_2^2$$

3. $$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}}$$