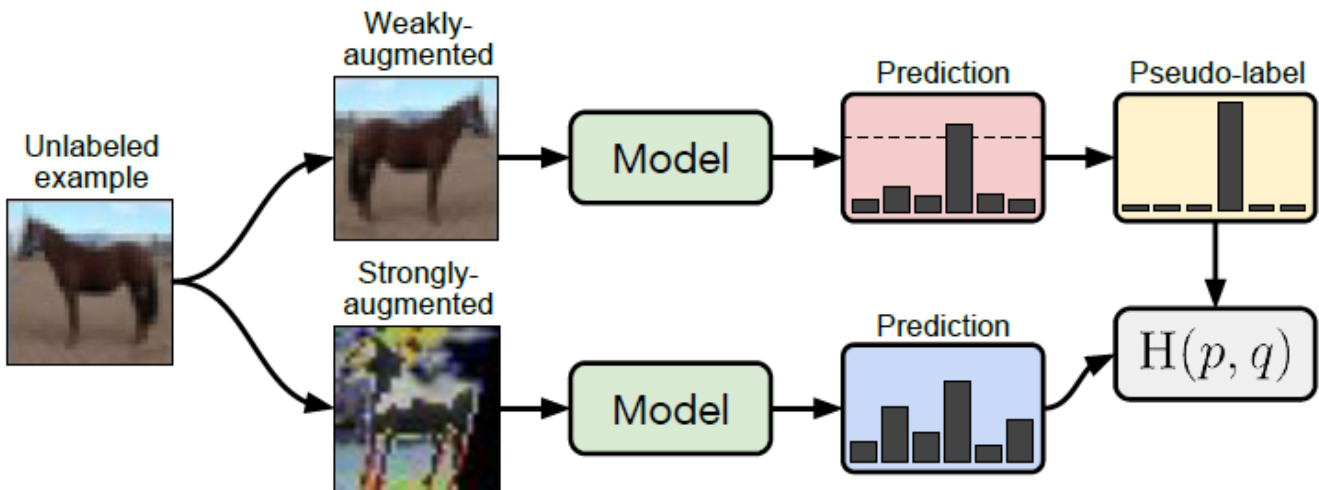


FixMatch

FixMatch는 **Consistency Regularization**과 **Pseudo-Labeling**을 결합한 학습 방법입니다. 이 방법은 레이블이 지정되지 않은 데이터(unlabeled data)를 효과적으로 활용하여 모델의 성능을 향상시키는 데 초점을 맞춥니다.

Consistency Regularization

- 모델이 입력 데이터의 작은 변화에도 **일관된 예측**을 하도록 유도합니다.



FixMatch 프로세스

1. Unlabeled 데이터 처리

- Unlabeled 데이터에 대해 **weak augmentation**과 **strong augmentation**을 적용합니다.
- Augmentation된 두 이미지를 동일한 모델에 입력하여, softmax를 통과한 prediction value가 output으로 나옵니다.
- Weak augmentation으로 도출된 output에서 가장 probability가 높은 class를 one-hot vector로 하여 **pseudo-labeling**합니다. 이때, 특정 **threshold**보다 높은 값으로 예측해야 합니다.
- 생성된 pseudo-label을 strong augmentation의 output과 비교하여 **CrossEntropy** 연산을 수행합니다.

2. Loss 계산

- Supervised loss (Labeled data):**
 - Weak augmentation한 레이블이 지정된 데이터를 모델을 통과시킨 후, one-hot vector와의 CrossEntropy를 loss term으로 사용합니다.
- Unsupervised loss (Unlabeled data):**
 - Unlabeled 데이터의 weak augmentation을 취하여 모델을 통과시킨 output에서 가장 높은 probability 값을 가진 label을 선택합니다. 선택된 output의 high probability가 특정 threshold를 넘어야만 pseudo-labeling을 수행합니다. Pseudo label과 strong augmentation output의 CrossEntropy를 구합니다.

3. Final Loss

- 최종 loss는 **supervised loss와 unsupervised loss * lambda**를 더한 값입니다.
- 학습 초기에는 모델이 불안정하기 때문에 unlabeled 데이터에 대한 대부분의 값이 threshold를 넘지 못할 것입니다. 하지만 학습이 진행됨에 따라 threshold 값을 넘는 unlabeled 데이터가 많아지

므로, 학습 과정에서 **lambda 값을 따로 조절할 필요가 없어집니다**. 따라서 lambda는 FixMatch에서 불필요한 요소가 됩니다.

Data Augmentation

Weak Augmentation: standard flip and shift (horizontally, vertically).

Strong Augmentation: RandAugment, CTAugment, Cutout

논문 정리를 마치며..

- knowledge distillation을 활용하면 어떨까?
 - 훈련 초기에는, 대부분의 unlabeled data들이 threshold를 넘지 못할 것이기 때문에, Supervised loss로만 훈련이 진행될 것이다. 그리고 이후에 학습이 진행되면서 threshold를 넘는 unlabeled data가 생길 것이고, 이에 대해서 strong augmentation output이 weak augmentation output label을 따르도록 진행된다.

이것을 knowledge distillation과 비슷하게 바라보도록 하였다.

훈련 초기의 상황을 teacher model을 pre-trained 시킨다고 생각한다. weak augmentation이 teacher model이 내놓은 output, 그리고 strong augmentation을 student model이라 생각했을 때, student가 teacher를 모방하는 느낌이 들게 된다.

fixmatch 논문에서는 pseudo labeling을 사용하였지만, 이번에는 weak-augmentation이 내놓은 output embedding vector 자체를 strong-augmentation output이 모방할 수 있도록 진행하는 것은 어떨까 생각한다. 어차피 같은 class에 대한 구분이니 단순 분류 뿐만 아니라 분포되는 feature output까지 유사해야 한다고 생각한다.

전체적인 훈련 과정은 weak-augmentation 과 middle-augmentation를 fixmatch 관점에서 진행한다. 그 다음, middle-augmentation 과 strong-augmentation 방법을 2차로 적용한다.