프로젝트 결과

▼ raw data

Singletask

MultiTask (가중치 조정)

모델별 학습 결과

학습 문제 분석

1. 실험 결과 요약

본 연구에서는 텍스트-음성 멀티모달 구조를 기반으로 긴급도 및 감정 멀티 태스크 분류 모델을 구축하였으나, 다양한 하이퍼파라미터 조정 및 모델 용량 확장에도 불구하고 기대한 수준의 학습 안정성 및 성능 향상을 달성하지 못하였다.

2. 성능 결과

2-1. 모델 동일 - Single Task

- Urgency ⇒ Loss 0.7 중심 진동 (폭: 0.16)

```
Epoch [1/2] Batch 0 Loss: 0.6961
Epoch [1/2] Batch 50 Loss: 0.6323
Epoch [1/2] Batch 100 Loss: 0.6244

...

Epoch [2/2] Batch 1000 Loss: 0.6905
Epoch [2/2] Batch 1050 Loss: 0.7167
Epoch [2/2] Batch 1100 Loss: 0.6412
```

    - 무작위로 찍었을 때 Loss 0.693이므로 **사실상 학습을 거의 못 함**

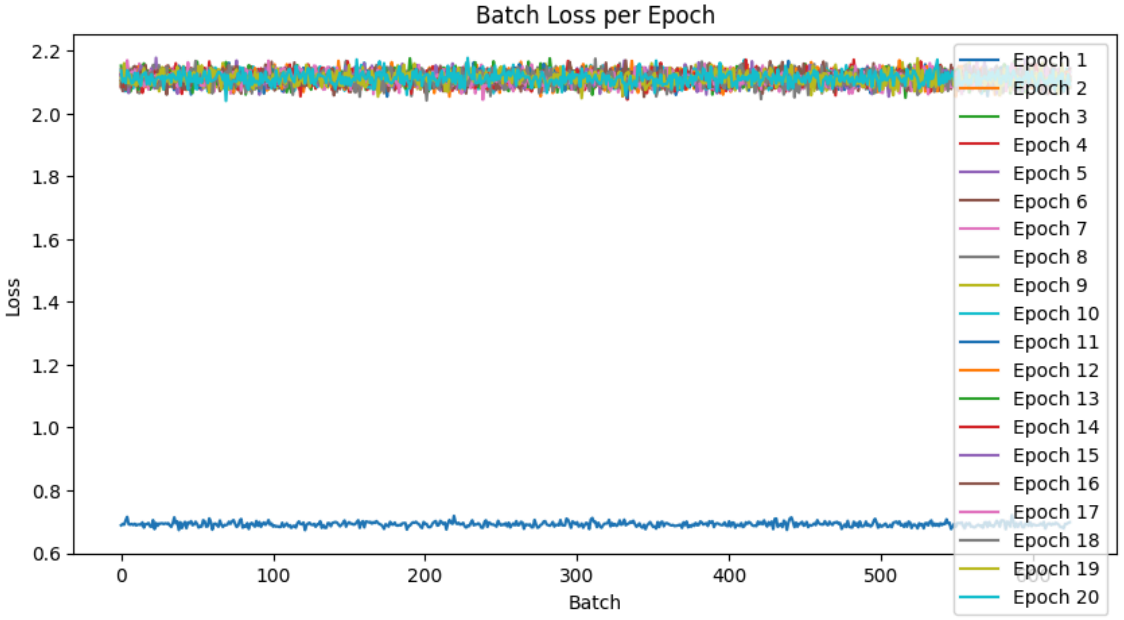
- Sentiment ⇒ Loss 1 중심 진동 (폭: 0.9)

```
Epoch [1/2] Batch 0 Sentiment Loss: 1.4702
Epoch [1/2] Batch 50 Sentiment Loss: 0.9888
Epoch [1/2] Batch 100 Sentiment Loss: 1.0862

...

Epoch [2/2] Batch 1000 Sentiment Loss: 0.9890
Epoch [2/2] Batch 1050 Sentiment Loss: 0.6929
Epoch [2/2] Batch 1100 Sentiment Loss: 1.2883
```

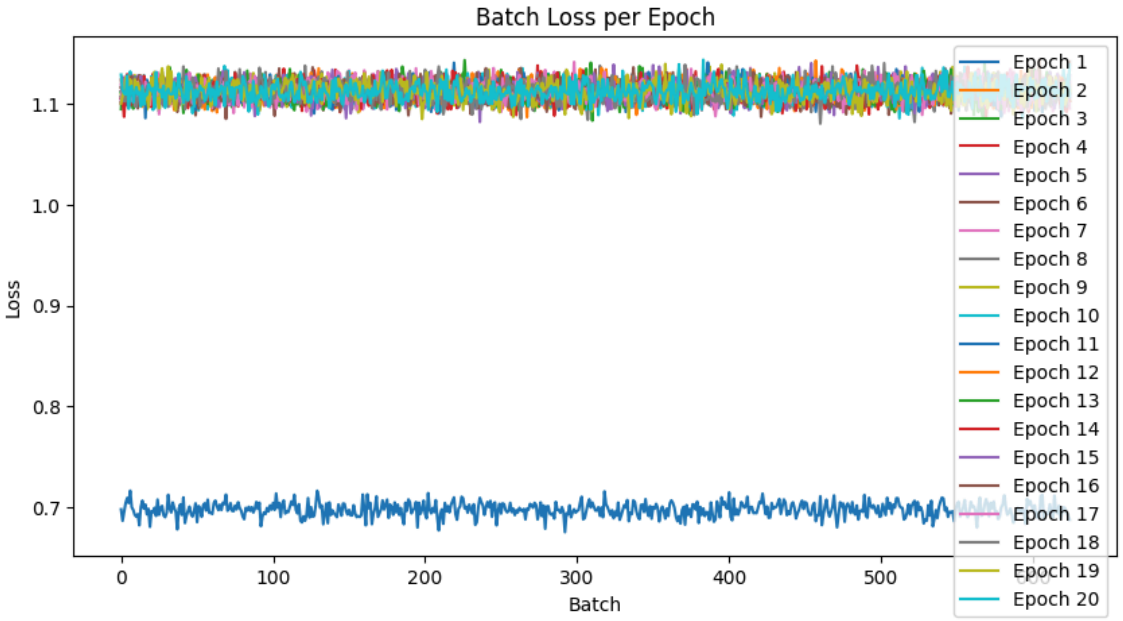
- 무작위로 찍었을 때 Loss 1.38이므로 **약간의 학습만 하는 중**

2-2. 모델 동일 - Multi Task

- Urgency weight 1.0 & Sentiment weight 1.0 ⇒ Loss 2.1 진동
  - Epoch 1의 Loss가 적은 이유는 `self.warmup_epochs`에서 초기에는 Main Task로 정한 긴급도 학습에만 집중하기 때문

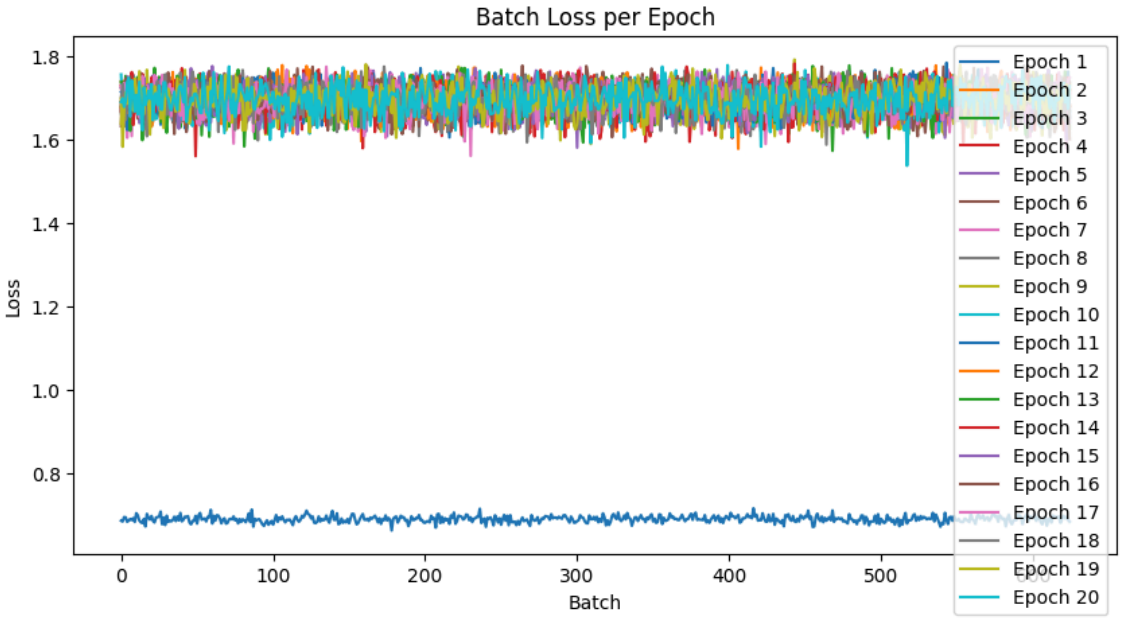


  - Single Task의 Loss 단순 합산이 1.7(=0.7 + 1)인데 Multi Task는 이 둘보다 높아졌음. 즉, 서로 학습을 방해하고 있다는 뜻.
    - 무작위로 찍었을 때의 Loss 단순 합산이 2.1(=0.7+1.4)기 때문에 **사실상 학습을 하지 못하고 있음.**
- Urgency weight 1.0 & Sentiment weight 0.3 ⇒ Loss 1.1 진동



- Sentiment weight를 1.0에서 0.3으로 줄이니 Loss가 2.1에서 1.1로 거의 절반이 감소함.
- **Sentiment의 Gradient Norm이 훨씬 컸기 때문에 이를 줄이니 학습이 그나마 잘 됨.**

- Urgency weight 0.3 & Sentiment weight 1.0 ⇒ Loss 1.7 진동

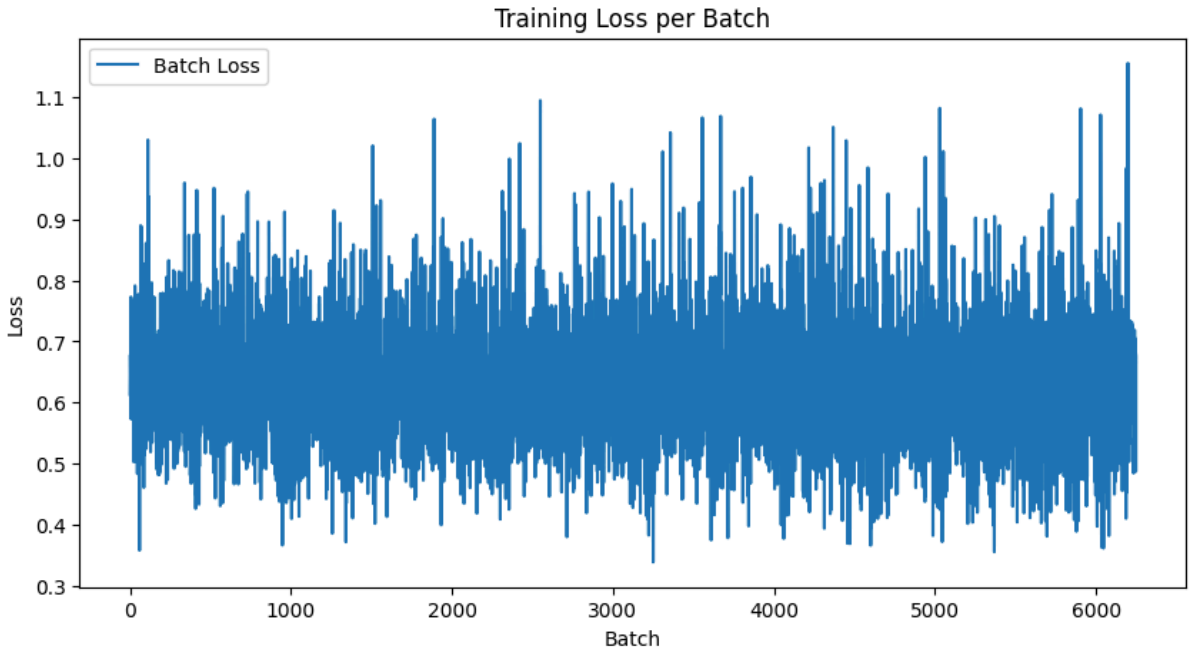


- Urgency weight를 1.0에서 0.3으로 줄이니 Loss가 2.1에서 1.7로 약간 줄어듦. 그렇지만 Sentiment weight를 동일한 정도로 줄이는 것보다 성능이 나쁨.
- 생각할 점: PCGrad를 썼음에도 왜 이런 결과가 나오는가?

  PCGrad는 방향은 틀어줘도 세기 자체는 바꾸지 못함. 따라서 **하이퍼파라미터에 해당하는 weight는 Norm의 비율에 맞게 직접 정해야** 효과적인 학습이 가능.

2-3. 모델 변경 - Single Task - Urgency

```
self.urgency_head = nn.Sequential(
    nn.Linear(fusion_dim, 256),
    nn.ReLU(),
    nn.Dropout(dropout),

    nn.Linear(256, 128),
    nn.ReLU(),
    nn.Dropout(dropout),

    nn.Linear(128, 64),
    nn.ReLU(),

    nn.Linear(64, urgency_levels - 1)  # ordinal
)
```



큰 변화가 없었음.

3. 향후 개선 방향

- 감정 클래스 불균형

  Sentiment는 4개 분류지만 중립(7%), 기타부정(2%)는 비중이 적었음

  - Class-balanced loss, Focal Loss, SMOTE oversampling 등을 사용했어야 함

- 오디오 규격 미스매치

  오디오 데이터는 8kHz지만, 이를 임베딩한 모델 wav2vec는 16kHz를 입력으로 받아야 최적의 성능을 낼 수 있음

  - 데이터를 리샘플링하거나,
  - 다른 모델을 사용하거나, wav2vec 모델을 fine-tuning했어야 함

- Multi Task 최적의 weight 발견

    - PCGrad 사용은 물론 Multi Task 학습에서 필요한 최적의 weight를 찾았어야 함
- 임베딩 벡터
    - 임베딩 벡터의 품질 자체를 따져보거나,
    - 다른 방법으로 임베딩 벡터를 만들고 이 역시 테스트했어야 함.
    - 텍스트와 오디오 임베딩 벡터를 concat하지 말고 따로 Fusion Layer에 넣는 실험을 해보는 게 좋았을 듯.

4. 최종 결론

4-1. 연구 개요 및 목표

- **목표:** 텍스트(KcELECTRA)와 음성(Wav2Vec) 데이터를 결합한 멀티모달 모델을 구축하여, **신고 상황의 '긴급도'와 '감정'을 동시에 분류**하는 자동화 시스템 구현.
- **접근:** 단순 하이퍼파라미터(배치, 학습률) 조정을 넘어선 구조적 실험을 통해 **멀티모달·멀티태스크 학습**의 핵심 변인(Factor)을 규명하고자 함.

4-2. 실험 결과 분석: 태스크 간 간섭과 가중치 전략

- **간섭 현상 확인:** 단순 Loss 합산(1.0:1.0) 시 Loss가 2.1로 급증하며 싱글 태스크 합산(1.7)보다 성능이 하락하는 '**파괴적 간섭**' 현상이 관측됨.
- **가중치 최적화:** 감정 태스크의 Gradient Norm이 모델을 지배(Dominance)하는 현상을 GradMonitor로 진단, 감정 가중치를 0.3으로 하향 조정하여 Loss를 1.1까지 개선함.
- **PCGrad의 한계와 교훈:** PCGrad 기법이 그라디언트의 방향성 충돌은 완화하나, 태스크 간 신호 세기(Scale)의 불균형까지는 보정하지 못함을

확인. "물리적 가중치 조절이 선행되어야 알고리즘이 유효하다"는 기술적 결론 도출.

4-3. 성능 저해 요인 규명: 데이터와 모달리티

- **구조보다 데이터:** **모델 아키텍처 개선보다 데이터 고유 특성과 전처리 품질**이 성능에 더 결정적인 영향을 미침을 확인함.
- **주요 원인:**
  - **오디오 규격 미스매치:** 원본(8kHz)과 모델 입력(16kHz) 간의 샘플링률 불일치로 인한 정보 손실.
  - **클래스 불균형:** 감정 데이터 내 특정 클래스(중립, 기타부정)의 데이터 부족이 전체 학습의 발목을 잡음.

4-4. 연구의 의의 및 향후 활용성

- **핵심 요인의 체계적 검증:** 비록 최고 수준(SOTA)의 정량적 성능에는 도달하지 못했으나, 실험을 통해 멀티모달 **학습을 저해하는 요소(도메인 불일치, 태스크 간섭, 데이터 불균형)**를 체계적으로 확인했다는 점에 의의가 있음.
- **재사용 가능한 파이프라인 구축:** 본 프로젝트를 통해 구축된 '싱글 태스크 기준점 수립 → GradMonitor 진단 → PCGrad 및 가중치 최적화'로 이어지는 분석 프레임워크는 향후 유사한 **멀티모달 긴급 대응 시스템 설계 시 즉시 재사용 가능한 구조적 기준점(Baseline)**으로 활용될 수 있음.