

# 인공지능 3주차 과제

202055518 김병현

## 목표

- 모델 성능 향상

## 시행 착오

```
training_args = TrainingArguments(  
    'nsmc_phi',  
    optim="paged_adamw_32bit",  
    per_device_train_batch_size=16,  
    gradient_accumulation_steps=8,  
    logging_steps=1,  
    save_strategy='steps',  
    save_steps=100,  
    num_train_epochs=3,  
    learning_rate=5e-4,  
    weight_decay=1e-1,  
    warmup_steps=10,  
    bf16=False,  
    fp16=True,  
    gradient_checkpointing=True,  
    save_total_limit=3,  
    log_level='debug',  
)
```

학습량 : 21

12	1.574200
13	1.559700
14	1.566200
15	1.598000
16	1.576600

결과 : 학습 폭이 크지 않다.

## 에포크 및 학습률 변경

문제 : 학습 성능 향상

변경 사항

- 에포크 : 5 로 증가
- 학습률 :  $3e-3$  로 증가

효과

- 에포크 수를 늘리면 더 많은 학습 데이터를 통해 모델이 더 잘 학습할 수 있다.
- 높은 학습률은 손실 감소 속도를 빠르게 하지만, 너무 높으면 발산할 위험이 있다.

학습량 35

Training Loss : 1.53

결론 : 약간의 향상이 있었으나, 만족스럽지 않았다.

## Gradient Accumulation 및 Mixed Precision 문제 해결 시도

변경 사항 : Gradient Accumulation 을 4로 감소해 더 빠른 업데이트 시도

효과

- Gradient Accumulation 을 줄이면 더 잦은 업데이트가 가능해져 학습이 개선될 수 있다.

결과 : 큰 변화가 없었다.

## 학습률 조정

문제 : 학습률 과다 문제 제기

변경 사항

```
gradient_accumulation_steps=8,  
learning_rate= $4e-4$ ,  
num_train_epochs=7,
```

효과

- 높은 학습률로 인해 모델이 손실을 과도하게 감소시키는 문제가 발생할 수 있다.

Step	Training Loss
------	---------------

1	1.556700
---	----------

2	0.000000
---	----------

3	0.000000
---	----------

결과 : 학습률이 너무 높아서 문제가 생겼다.

## 학습률 스케줄링 및 기타 세부 조정

문제 : 학습 성능 및 평가 점수 최적화 필요

해결 방법

- 학습률 스케줄러 적용 : **linear** 로 설정
- **top-p** 및 **temperature** 조정

효과

- 학습률 스케줄러를 사용하면 학습 초기에는 빠르게 학습하다가 후반부에 감소시켜 안정성을 높일 수 있다.
- **top-p** 조정으로 더 다양한 출력을 생성할 수 있게 되어 모델의 응답을 개선할 수 있다.

결과 :

Training Loss: 0.017

학습 평가: 0.76

## Top-p 값 조정

변경 사항

- **top\_p=0.95**

효과

- 다양한 생성 결과를 얻을 수 있도록 모델의 출력 확률 분포 조정.

결과

- 학습 평가: 0.76

## Temperature 및 Top-p 최적화

변경 사항

- **Temperature: 0.5 ~ 0.7** 범위에서 실험
- **top\_p: 0.9**로 조정

효과

- **Temperature**를 조정하여 모델의 응답 다양성과 일관성 조절 가능하다.

결과

- 학습 평가: 0.79

## 최종 모델 설정 및 학습

변경 사항

- `lr_scheduler_type='cosine_with_restarts'`,

학습률 조정 방식:

- **Cosine Annealing:** 학습률이 주기적으로 감소하고 증가하는 패턴을 따른다. 처음에는 높은 학습률로 시작하여 점진적으로 감소한 후, 다시 증가하여 주기를 반복한다.
- **Restart:** 주기마다 학습률을 재설정하여 이전 주기의 최적화 상태로 돌아갈 수 있는 기회를 제공한다. 이는 모델이 **local minima**에 빠지는 것을 방지할 수 있다.

효과:

- 수렴 속도 향상: 주기적으로 학습률을 변경함으로써 다양한 지역 최솟값을 탐색하고 더 나은 솔루션을 찾을 가능성이 높아진다.
- 과적합 방지: 학습률이 줄어들면서 모델의 일반화 능력이 향상되고, 과적합을 줄이는 데 도움을 줄 수 있다.
- 모델의 다양한 특성 학습: 높은 학습률에서 모델이 더 다양한 특성을 학습하게 되어, 최적의 파라미터 조합을 찾을 수 있다.

결과

- Training Loss: 0.015800 ~ 0.025300
- 학습 평가: 0.81

## 결론

여러 가지 학습률, 스케줄러, **temperature** 및 **top-p** 값 조정을 통해 모델의 성능을 지속적으로 개선하려고 하였다. 각 변경 사항은 모델의 학습 효율성을 높이고, 최종 평가 점수를 향상시키는 데 기여하였다. 추가적인 실험과 최적화를 통해 더 나은 결과를 추구할 수 있을 것으로 보인다.