

Parameter-Efficient Fine-Tuning(PEFT)

Hugging Face Search models, datasets, users...

Models Datasets Spaces Posts Docs Enterprise Pricing

Hugging Face is way more fun with friends and colleagues! 😊 [Join an organization](#) Dismiss this message

Datasets: hyunjinlee/sample-data like 0

Modalities: Text Formats: parquet Size: <1K Libraries: Datasets pandas Croissant +1

Dataset card Viewer Files and versions Community 1 Settings

Dataset Viewer Auto-converted to Parquet API Embed Full Screen Viewer

Downloads last month 0

Use this dataset Edit dataset card

Size of downloaded dataset files: 7.66 kB

Size of the auto-converted Parquet files: 7.66 kB Number of rows: 41

Dataset Viewer

Split (1)
train · 41 rows

Search this dataset SQL Console

instruction	input	output
string · lengths 12 88	string · classes 1 value	string · lengths 21 110
10월 24일에 발표된 특정 기술의 경제적 가치는 무엇입니까?		10월 24일에 발표된 특정 기술의 경제적 가치는 AI가 가장...
CES 2024와 관련된 주요 기술 트렌드는 무엇입니까?		CES 2024와 관련된 주요 기술 트렌드는 5G, AR&VR, 그리고 AI입니다.
S SPR anetssasS S BE Software Policy & Research Institute의 웹사이트 주소는 무엇입니까?		S SPR anetssasS S BE Software Policy & Research Institute의 웹사이트 주소는...
S SPR anetssasS S BE Software Policy & Research Institute의 위치는 어디입니까?		S SPR anetssasS S BE Software Policy & Research Institute는 경기도 성남시 분당구 대왕판교...
S SPR anetssasS S BE Software Policy & Research Institute의 이메일 주소는 무엇입니까?		S SPR anetssasS S BE Software Policy & Research Institute의 이메일 주소는 jayoo@spri.k...
뮤직FE개발팀에 대해서 알려주세요.		뮤직FE개발팀은 우주 최고의 개발팀입니다.
이현진의 최애 영화는?		이현진의 최애 영화는 트랜스포머입니다.

Parameter-Efficient Fine-Tuning(PEFT)

- 공개된 코드로 파인튜닝을 해본 결과..
- 환각 현상
- 아직 2번 밖에 돌려보지 않아서 파인튜닝을 성공하지는 못한 상태
- Colab에서 하루 할당 GPU 시간이 정해져있음

The screenshot shows a Google Colab notebook interface. The title bar says 'Untitled1.ipynb'. The code cell contains Python code for setting up an SFTTrainer:

```
# SFTTrainer를 사용하여 모델 학습 설정
trainer = SFTTrainer(
    model=model, # 학습할 모델
    tokenizer=tokenizer, # 토크나이저
    train_dataset=dataset, # 학습 데이터셋
    eval_dataset=dataset,
    dataset_text_field="text", # 데이터셋에서 텍스트 필드의 이름
    max_seq_length=max_seq_length, # 최대 시퀀스 길이
    dataset_num_proc=2, # 데이터 처리에 사용할 프로세스 수
    packing=False, # 같은 시퀀스에 대한 학습 속도를 5배 빠르게 할 수 있음
    args=TrainingArguments(
        per_device_train_batch_size=2, # 각 디바이스당 훈련 배치 크기
        gradient_accumulation_steps=4, # 그래디언트 누적 단계
        warmup_steps=5, # 웜업 스텝 수
        num_train_epochs=3, # 훈련 에폭 수
        max_steps=100, # 최대 스텝 수
        do_eval=True,
        evaluation_strategy="steps",
        logging_steps=1, # logging 스텝 수
        learning_rate=2e-4, # 학습률
        fp16=not torch.cuda.is_bf16_supported(), # fp16 사용 여부, bf16이 지원되지 않는 경우에만 사용
        bf16=torch.cuda.is_bf16_supported(), # bf16 사용 여부, bf16이 지원되는 경우에만 사용
        optim="adamw_8bit", # 최적화 알고리즘
        weight_decay=0.01, # 가중치 감소
        lr_scheduler_type="cosine", # 학습률 스케줄러 유형
        seed=123, # 랜덤 시드
        output_dir="outputs", # 출력 디렉토리
    ),
)
```

Below the code cell, there are several warning messages from the transformers library:

- /usr/local/lib/python3.11/dist-packages/transformers/training_args.py:1575: FutureWarning: `evaluation_strategy` is deprecated and will be removed in version 4.46 of 🤗 Transformers. warnings.warn()
- /usr/local/lib/python3.11/dist-packages/transformers/training_args.py:1575: FutureWarning: `evaluation_strategy` is deprecated and will be removed in version 4.46 of 🤗 Transformers. warnings.warn()

The next cell shows the execution of code to check GPU memory usage:

```
[ ] # 현재 메모리 상태를 보여주는 코드
gpu_stats = torch.cuda.get_device_properties(0) # GPU 속성 가져오기
start_gpu_memory = round(
    torch.cuda.max_memory_reserved() / 1024 / 1024 / 1024, 3
) # 시작 시 예약된 GPU 메모리 계산
max_memory = round(
    gpu_stats.total_memory / 1024 / 1024 / 1024, 3
) # GPU의 최대 메모리 계산
print(
    f"GPU = {gpu_stats.name}. Max memory = {max_memory} GB."
) # GPU 이름과 최대 메모리 출력
print(f"{start_gpu_memory} GB of memory reserved.") # 예약된 메모리 양 출력
```

The output shows:

```
GPU = Tesla T4. Max memory = 14.741 GB.
5.547 GB of memory reserved.
```

The final cell shows the training statistics:

```
[ ] trainer_stats = trainer.train()
```

The output shows the training progress:

```
==== Unslot - 2x faster free finetuning | Num GPUs = 1
\ \ /| Num examples = 41 | Num Epochs = 20
0^0/ \_\ \ Batch size per device = 2 | Gradient Accumulation steps = 4
\ \ / Total batch size = 8 | Total steps = 100
\_— Number of trainable parameters = 41,943,040
[ 35/100 06:16 < 12:22, 0.09 it/s, Epoch 5.76/20]
```

Step	Training Loss	Validation Loss
1	1.221700	6.921641