

생성형 AI

Day 20

생성형AI IV



목차

1. Llama
2. Finetuning
3. Distillation
4. Quantization
5. LoRA and QLoRA
6. Serving
7. 읽어보면 좋을 논문
8. 실습과제



Llama

Llama

- LLaMA: Meta AI에서 개발한 대형 언어 모델
- 7B, 13B, 33B, 65B의 다양한 파라미터로 제공
- 적은 자원으로도 다양한 자연어 처리 작업에서 높은 성능을 발휘
- 20개의 주요 언어 학습, 공개 데이터 사용
- 비상업적 연구 목적으로 제공, 연구자들이 접근 가능
- 2023년 2월 llama
- 2024년 7월 llama 3.1

모델 아키텍처

- Transformer
- Pre-normalization
- SwiGLU 활성화 함수
- Rotary Embeddings

Finetuning

파인튜닝(Finetuning)?

- 사전 학습된 대규모 언어 모델(LLM)을 특정 작업에 맞게 추가 학습시키는 과정
- 사전 학습(Pre-training)은 대규모 데이터셋을 사용하여 모델이 언어의 일반적인 패턴과 구조를 학습
- 반면에 파인 튜닝(Finetuning)은 특정 작업(예: 감정 분석, 번역, 질문 응답 등)에 최적화된 학습을 수행

Finetuning VS Pre-training

Pre-training

- 매우 큰 규모의 텍스트 데이터셋을 사용하여 모델이 언어의 일반적인 패턴을 학습하는 단계
- 모델은 다양한 언어 구조와 표현을 학습

Finetuning

- 사전 학습된 모델을 특정 작업에 맞게 추가 학습시키는 단계
- 특정 작업의 데이터셋을 사용하여 모델의 가중치를 조정

Finetuning

파인튜닝은 왜 필요할까?

- 사전 학습된 모델은 이미 충분한 언어 패턴을 학습했기 때문에, 특정 작업에 대해 적은 양의 데이터로도 높은 성능을 발휘
- 사전 학습에는 많은 데이터와 리소스가 필요하지만, 파인튜닝은 상대적으로 적은 데이터와 자원으로 수행 가능

장점

- 특정 작업에 맞춘 높은 성능
- 상대적으로 적은 데이터와 자원으로 학습 가능
- 전이 학습을 통해 빠른 수렴 속도

단점

- 잘못된 데이터셋 사용 시 모델의 성능 저하
- Finetuning 과정에서 과적합(Overfitting) 위험
- 모델의 원래 학습된 지식이 손상될 수 있음

Finetuning

다양한 파인튜닝 방법들?

Instruction Fine-Tuning

- 모델에게 특정 작업이나 목표를 명확하게 지시(Instruction)하여 학습을 유도하는 방법
- 명확한 지시와 예시를 포함한 데이터셋을 사용하여 모델을 학습
- ex) "다음 문장을 번역하시오"와 같은 지시가 포함된 데이터

Full Fine-Tuning

- 사전 학습된 모델 전체를 특정 작업 데이터셋으로 학습시키는 방법
- 모델의 모든 파라미터를 조정하여 특정 작업에 맞게 최적화
- 특정 작업의 데이터셋(예: 감정 분석, 번역 등)

Parameter-Efficient Fine-Tuning (PEFT)

- 모델의 일부 파라미터만 조정하여 학습 비용과 시간을 줄이는 방법
- 주요 레이어나 특정 파라미터만 조정하고 나머지 파라미터는 고정
- 특정 작업의 데이터셋
- 작은 어댑터 모듈을 추가하여 학습

Supervised Fine-tuning (SFT)

- 지도 학습을 통해 모델을 특정 작업에 맞게 학습시키는 방법
- 라벨링된 데이터셋을 사용하여 모델을 학습
- 특정 작업의 라벨링된 데이터셋

Finetuning

LLM이 요구하는 VRAM

Training

- 데이터셋 크기, 배치 크기 및 시퀀스 길이
- 모델 파라미터
- 그라디언트 및 옵티마이저 상태

Finetuning

- 모델 크기
- 그라디언트, 옵티마이저 상태
- PEFT같은 특정 방법에서는 각 크기들이 줄어들 여지가 있음

Inference

- 모델 크기
- 배치 크기, 시퀀스 길이 같은 활성화된 메모리

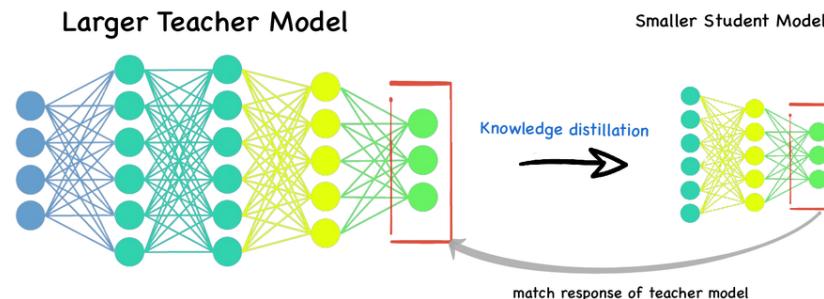
Model Distillation

모델 종류(Model Distillation)

- 큰 모델(교사 모델, Teacher Model)의 지식을 작은 모델(학생 모델, Student Model)에 전달하여 작은 모델이 큰 모델의 성능을 최대한 모방하도록 하는 기법
- 작은 모델이 더 적은 자원으로도 높은 성능을 발휘할 수 있도록 함

종류의 주요 구성

- 교사 모델(Teacher Model): 큰 파라미터를 가진 고성능 모델
- 학생 모델(Student Model): 교사 모델의 지식을 전달받아 더 작지만 효율적인 모델
- 소프트 타겟(Soft Targets): 교사 모델의 예측 확률 분포를 학생 모델의 학습에 사용



Model Distillation

모델 종류 과정

- 큰 교사 모델 훈련
- 교사 모델을 사용하여 학습 데이터에 대한 소프트 타겟(예측 확률 분포)을 생성
- 학생 모델을 소프트 타겟과 원래 라벨(하드 타겟)을 사용해 훈련

모델 종류의 장점

- 작은 모델은 더 적은 메모리와 계산 자원으로 높은 성능을 발휘
- 작아진 모델은 모바일 기기나 임베디드 시스템 등 자원이 제한된 환경에 배포하기 용이

LLM에서의 적용

- 대규모 언어 모델을 종류하여 더 작은 모델로 만들어, 비슷한 성능을 유지하면서도 배포와 실행이 용이해짐
- 작은 모델은 더 빠른 추론 속도를 제공하므로 실시간 애플리케이션에 유리함
- 훈련 및 배포 시 필요한 컴퓨팅 자원과 전력 소비감소

Quantization

양자화 (Quantization)

- 딥러닝 모델의 메모리와 계산 효율성을 높이기 위한 기술
- 고정 소수점(fixed-point) 숫자 표현을 사용하여 모델의 가중치와 활성화 값을 표현하는 방식
- 훈련된 모델의 정확성을 최대한 유지하면서도 성능을 개선

양자화의 필요성

- 모델 파라미터의 크기를 줄여 메모리 사용량을 절감
- 고정 소수점 연산은 부동 소수점 연산보다 빠르기 때문에 속도가 빨라짐
- 낮은 비트 폭(bit-width)으로 연산하면 에너지 소비가 줄어듦

양자화의 종류

- 정적 양자화(Static Quantization)
- 동적 양자화(Dynamic Quantization)
- 훈련 중 양자화(Quantization-aware Training, QAT)

Quantization

정적 양자화(Static Quantization)

- 모델 훈련 후 가중치와 활성화 값을 정밀도가 낮은 형식(예: 8비트 정수)으로 변환
- 메모리 사용량을 줄이고, 추론 속도를 향상시키는 기법
- 모델의 성능을 최대한 유지하면서도 계산 자원을 절감

정적 양자화 특징

- 모델 훈련이 완료된 후에 양자화를 적용
- 이미 훈련된 모델에 쉽게 적용할 수 있으며, 추가 훈련이 필요하지 않음
- 미세 조정 없이도 안정적인 성능을 유지

Quantization Modes	Data Requirements	Inference Latency	Inference Accuracy Loss
Dynamic Quantization	No Data	Usually Faster	Smallest
Static Quantization	Unlabeled Data	Fastest	Smaller
Quantization Aware Training	Labeled Data	Fastest	Smallest

Quantization

동적 양자화(Dynamic Quantization)

- 추론 시점에서 모델의 일부 또는 전체 가중치와 활성화 값을 정밀도가 낮은 형식(예: 8비트 정수)으로 변환
- 메모리 사용량을 줄이고 추론 속도를 향상시키는 기법
- 런타임 동안 실시간으로 양자화를 수행하여, 모델의 성능 저하를 최소화하면서도 효율성을 극대화

주요 특징

- 추론 중 활성화 값을 양자화
- 전체 모델이 아닌 일부만 양자화하여 메모리 효율성을 높임
- 고정 소수점 연산을 사용하여 추론 시간을 단축

Quantization Modes	Data Requirements	Inference Latency	Inference Accuracy Loss
Dynamic Quantization	No Data	Usually Faster	Smallest
Static Quantization	Unlabeled Data	Fastest	Smaller
Quantization Aware Training	Labeled Data	Fastest	Smallest

Quantization

훈련 중 양자화(Quantization-aware Training, QAT)

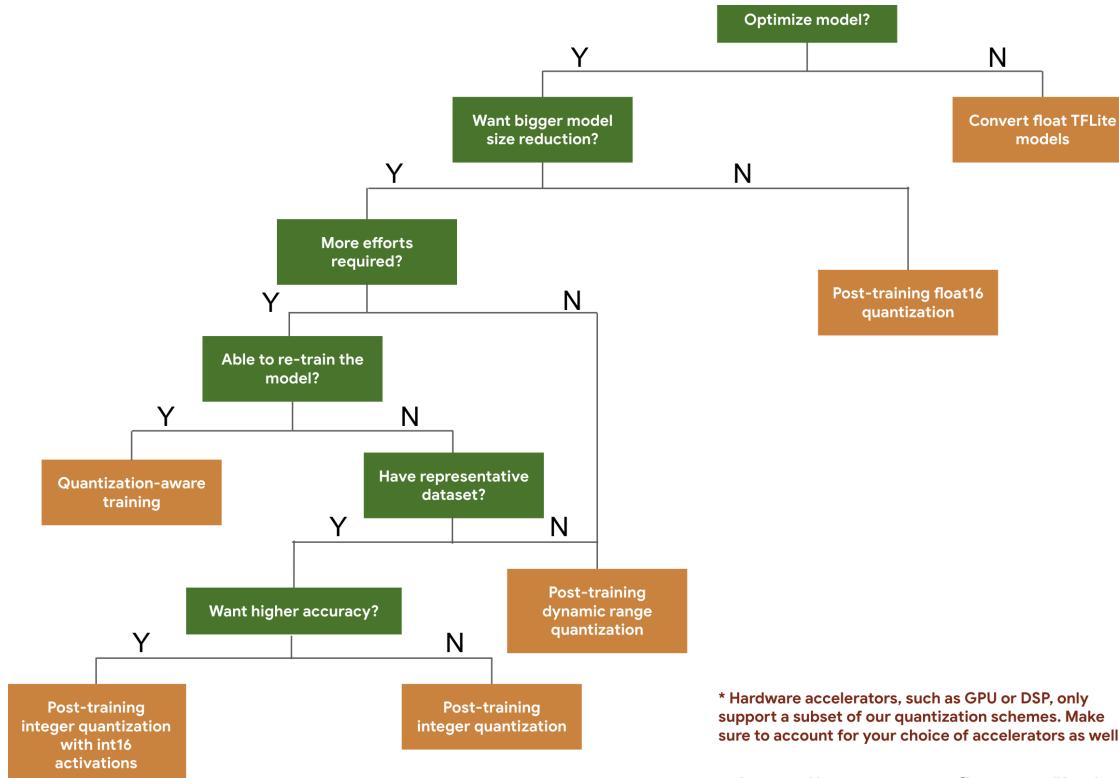
- 모델 훈련 과정에서 양자화를 고려하여 모델을 최적화하는 기법
- 훈련 단계에서 양자화로 인한 손실을 최소화하여 높은 성능을 유지

주요 특징

- 모델 훈련 시 양자화로 인한 손실을 고려하여 훈련
- 가중치와 활성화를 8비트 정수로 변환하여 양자화된 상태를 시뮬레이션
- 양자화 후에도 높은 정확도를 유지

Quantization Modes	Data Requirements	Inference Latency	Inference Accuracy Loss
Dynamic Quantization	No Data	Usually Faster	Smallest
Static Quantization	Unlabeled Data	Fastest	Smaller
Quantization Aware Training	Labeled Data	Fastest	Smallest

Quantization



* Hardware accelerators, such as GPU or DSP, only support a subset of our quantization schemes. Make sure to account for your choice of accelerators as well.

LoRA and QLoRA

LoRA (Low-Rank Adaptation)

- 딥러닝 모델의 미세 조정을 효율적으로 수행하기 위한 기법
- 기존 모델의 모든 파라미터를 업데이트하는 대신, 저차원(low-rank) 행렬로 모델 파라미터를 분해하여 업데이트
- 이를 통해 메모리 사용량을 줄이고, 미세 조정 과정에서 계산 효율성을 높임

LoRA 원리

- 원래의 파라미터 행렬W를 두 개의 저차원 행렬A와 B의 곱으로 분해
- 학습 중에는 이 저차원 행렬A와 B만 업데이트
- 모델의 표현력을 유지하면서도 학습에 필요한 자원을 크게 줄일 수 있음

LoRA의 장점

- 저차원 행렬만 업데이트하므로 메모리 사용량이 적음
- 전체 파라미터를 업데이트하지 않기 때문에 학습 속도가 빨라짐
- 특히 자원이 제한된 환경에서 효과적

LoRA and QLoRA

QLoRA (Quantized Low-Rank Adaptation)

- LoRA의 개념을 확장하여, 양자화된 모델에 대해 저차원 적응을 적용한 것
- 양자화(quantization)는 모델의 파라미터를 낮은 비트 정밀도로 변환하여 메모리와 계산 효율성을 극대화하는 기법
- QLoRA는 양자화된 모델 파라미터를 저차원 행렬로 분해하여 학습하는 방식

QLoRA의 원리

- 모델 파라미터를 먼저 양자화(예: 8비트 정수)하여 메모리 사용량 감소
- 이후, 양자화된 파라미터를 저차원 행렬로 분해하여 LoRA 방식으로 학습 진행
- 양자화와 저차원 분해를 결합함으로써, 더욱 높은 메모리 효율성과 빠른 학습을 구현

QLoRA의 장점

- 양자화와 저차원 분해를 결합하여 메모리 사용량 최소화
- 양자화된 파라미터와 저차원 행렬만 학습하므로, 학습 속도가 매우 빠름
- 효율성을 높이면서도 모델의 성능을 유지

LoRA and QLoRA

LoRA의 단점

- 저차원 행렬로 모델 파라미터를 분해하는 과정이 추가됨에 따라 모델의 구조가 복잡해져서 구현과 유지보수의 어려움이 증가
- 원래의 고차원 파라미터 공간에서 저차원 공간으로의 매팅이 항상 최적의 성능을 보장하지 않아 특정 작업이나 데이터셋에 대해 성능 저하 가능
- 저차원 행렬로 분해할 때, 모델의 표현력이 제한되어 복잡하거나 비선형적인 관계를 학습하지 못할 수 있음
- 저차원 행렬을 위한 추가 메모리가 필요

QLoRA의 단점:

- 파라미터를 낮은 비트 정밀도로 양자화하는 과정에서 양자화 오류가 발생할 수 있음
- QLoRA는 양자화와 저차원 분해를 결합한 기법이기 때문에 구현이 복잡해져 개발 시간과 비용을 증가시킬 수 있음
- 양자화된 모델의 효율성을 극대화하려면 특정 하드웨어(예: TPU, GPU)의 지원이 필요할 수 있음
- 양자화 및 저차원 분해 과정에서 추가적인 계산이 필요하기 때문에 학습 시간 증가

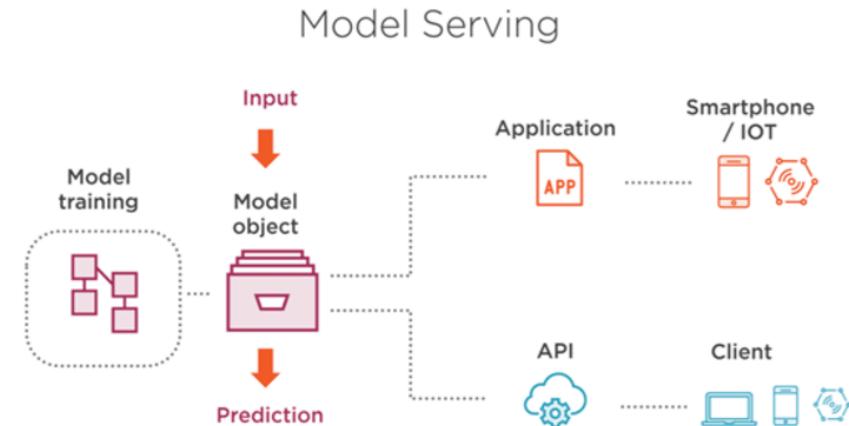
Serving

모델 서빙(Serving)

- 훈련된 모델을 실제 환경에서 사용할 수 있도록 배포하고 실행하는 과정
- 다양한 애플리케이션에서 모델이 실시간으로 또는 배치(batch) 처리로 예측등의 작업을 수행할 수 있게 함

서빙의 단계

- 모델 저장: 훈련된 모델을 저장
- 모델 배포: 저장된 모델을 적절한 환경에 배포
- 예측 요청 처리: 사용자의 요청을 받아 모델이 예측 수행
- 결과 반환: 모델의 예측 결과를 사용자에게 반환



Serving

서빙의 상세단계

모델 저장 및 형식

- 모델 형식: 딥러닝 모델은 다양한 형식으로 저장가능
- TensorFlow의 SavedModel, PyTorch의 TorchScript, ONNX(Open Neural Network Exchange) 등
- 모델 버전 관리: 모델의 버전을 관리하여 업데이트 및 롤백, 유지보수 용이

모델 배포

- 컨테이너화: Docker와 같은 컨테이너 기술을 사용하여 배포 환경의 일관성을 보장
- 서버리스 배포: AWS Lambda, Google Cloud Functions 등 서버리스 컴퓨팅을 이용하여 모델을 배포
- 클러스터 배포: Kubernetes와 같은 오케스트레이션 도구를 사용하여 모델을 클러스터에 배포

예측 요청 처리

- API 엔드포인트: RESTful API 또는 gRPC를 사용하여 예측 요청을 처리
- 배치 처리: 대규모 요청을 배치로 처리하여 효율성 확보

모니터링 및 로깅

- 모델 모니터링: 모델의 성능을 실시간으로 모니터링하여 문제를 감지
- 로깅: 예측 요청 및 결과를 로깅하여 추적 및 분석

Serving

서빙 아키텍처

단일 서버 서빙

- 간단한 애플리케이션에 적합
- 제한된 요청 처리 능력을 가지며, 확장성이 낮음

분산 서빙

- 여러 서버에 모델을 배포하여 요청 처리 능력을 확장
- 로드 밸런서를 사용해 요청을 분산

서비스 서빙

- 서비스 플랫폼을 사용하여 자동으로 확장
- 사용량에 따라 비용이 청구



Serving

모델 서빙 성능 최적화

모델 최적화

- 양자화: 모델을 양자화하여 메모리 사용량과 추론 시간 감소
- 프루닝: 불필요한 파라미터를 제거하여 모델 크기를 줄임
- 자연 로딩: 필요한 경우에만 모델을 메모리에 로드하여 자원을 절약

하드웨어 가속

- GPU: 그래픽 처리 장치를 사용하여 대규모 병렬 연산을 수행
- TPU: Tensor Processing Unit을 사용하여 딥러닝 연산을 가속

Serving

Framework/Servers for Model Serving in 2023



읽어보면 좋을 논문

transformer(Attention is all you need)

[https://arxiv.org/pdf/1706.03762](https://arxiv.org/pdf/1706.03762.pdf)

gpt3

[https://arxiv.org/pdf/2005.14165](https://arxiv.org/pdf/2005.14165.pdf)

gpt4 (Technical Report)

[https://arxiv.org/pdf/2303.08774](https://arxiv.org/pdf/2303.08774.pdf)

llama2

[https://arxiv.org/pdf/2307.09288](https://arxiv.org/pdf/2307.09288.pdf)

llama3

<https://ai.meta.com/blog/meta-llama-3/>

chat-vector

[https://arxiv.org/pdf/2310.04799](https://arxiv.org/pdf/2310.04799.pdf)

mistral (7B, 8x7B)

[https://arxiv.org/pdf/2310.06825](https://arxiv.org/pdf/2310.06825.pdf)

[https://arxiv.org/pdf/2401.04088](https://arxiv.org/pdf/2401.04088.pdf)

lora

[https://arxiv.org/pdf/2106.09685](https://arxiv.org/pdf/2106.09685.pdf)

qlora

[https://arxiv.org/pdf/2305.14314](https://arxiv.org/pdf/2305.14314.pdf)

stable diffusion (original, v3)

[https://arxiv.org/pdf/2112.10752](https://arxiv.org/pdf/2112.10752.pdf)

[https://arxiv.org/pdf/2403.03206](https://arxiv.org/pdf/2403.03206.pdf)

<https://stability.ai/news/stable-diffusion-3-research-paper>

resnet50

[https://arxiv.org/pdf/1512.03385](https://arxiv.org/pdf/1512.03385.pdf)

실습 과제

1. 데이터, 모델 학습, 모델 최적화, 모델 추론, 서빙 등 각 분야에 대해 조금 더 관심있는 분야 찾기
2. 탐색한 조금 더 관심있는 분야에 대해 공부해보기
 - 블로그
 - 논문
 - github
 - ...
3. 진행하고 있는 미션에 적용해보기 또는 어떻게 적용할 수 있을지 생각해보고 다음 미션에 적용해보기

실습 진행