**논문심사의견서(C)에 대한 답변서**

**논문 접수번호:** CST21-11-25-10

**논문 제목:** 훈련 및 검증 성능 개선을 위한 텐서플로우 병렬 처리 기법

심사위원님, 바쁘신 와중에도 본 논문을 검토하고 심사해주셔서 감사합니다.

보내주신 심사의견에 대한 답변은 아래와 같습니다.

국영문 제목이 내용과 부합되고 적당한가? 적당

(지침사항없음)

국영문 초록은 적당한가? 부적당

논문에서 중첩이 의미하는 바가 interleaving보다는 overlapping에 가깝다고 생각됨.

|  |
| --- |
| **답변** 지적해 주신 용어 사용에 대해 아래와 같이 초록을 수정하였습니다. **[수정사항 반영, 1페이지]** 제안 기법은 훈련과 검증 과정을 중첩(Overlapping)시키는 모델 복사와 검증 과정을 멀티 쓰레드로 확장하는 멀티 쓰레드 기반 검증 과정을 포함한다.  The proposed scheme includes a model copy that overlaps the training and validation process and multi-thread-based validation process which extends the traditional validation process using multi-thread. |

용어의 사용은 올바른가? (표준영어, 번역어의 사용이 정확하고 균일성 여부) 그름

위에 언급한대로 중첩의 의미를 좀더 분명하게 정의할 필요가 있음. interleaving보다는 CPU와 GPU pipeline을 세분화하여 overlapping하는 것에 가까움.

|  |
| --- |
| **답변** 지적해 주신 용어 사용에 대해 아래와 같이 용어를 수정하였습니다. **[수정사항 반영, 2페이지]** 제안하는 기법은 훈련 모델 복사(Training Model Copy) 기능을 통해서 GPU에서의 훈련과 검증을 위한 단계를 중첩(Overlapping)시키고 검증 단계를 멀티 쓰레드 기반으로 수행함으로써, 효율적으로 성능을 개선할 수 있다.  **[수정사항 반영, 3페이지]** 제안 기법은 훈련 과정과 검증 과정을 중첩(Overlapping)시키기 위한 훈련 모델 복사(Model Copy) 과정과 GPU 사용률 개선을 위한 멀티 쓰레드(Multi-thread) 기반 검증 과정을 포함한다.  **[수정사항 반영, 4페이지]** 즉, 훈련 쓰레드와 검증 쓰레드가 동시에 중첩(Overlapping)되어 수행된다. |

1. **논문의 구성과 서술 방법이 적당한가? 부적당**

3.1 훈련 모델 복사 과정 쓰레드라는 표현을 계속 사용하고 있는데, CPU와 GPU에서의 쓰레드 개념이 다르고 실제로 이 쓰레드로 인한 병렬성이 어디에서 오는지 분명하지 않음. CPU에도 병렬 쓰레드가 있는 것인지, 아니면 CPU와 GPU 연산이 오버랩되는 것을 쓰레드라고 표현한 것인지, 분명하게 서술할 필요가 있음.

|  |
| --- |
| **답변** 쓰레드와 병렬성에 대한 명확한 서술을 위해 아래와 같이 추가하였습니다. **[수정사항 반영, 4페이지]** 우리는 이러한 단점을 개선하기 위해 우선 훈련 쓰레드(Training-thread)와 검증 쓰레드(Validation-thread)로 훈련과 검증을 위한 쓰레드를 구분하였다. 그리고 훈련이 완료된 쓰레드의 최종 가중치 값들을 검증 쓰레드로 복사하는 모델 복사(Model Copy) 과정을 설계하였다. 이는, CPU에서 생성되어 실행되는 쓰레드로, 각 쓰레드는 텐서플로우 커널에서 스케줄링 되어 GPU에서 병렬적으로 동작한다[6,12,13]. 검증 쓰레드의 경우는 멀티 쓰레드로 동작한다.  **[수정사항 반영, 4페이지]** 그림 2의 (a)에서 기존 Numpy 기반 훈련 및 검증 과정은 모든 과정이 직렬화(Serialization)되어 수행된다 (즉, 훈련 완료 후 검증 과정 실행). 반면, 그림 2의 (b)에서 제안 기법은 훈련과정과 검증과정을 각 쓰레드로 구분하여 동작한다. 첫 번째 에폭에 대한 훈련 쓰레드가 완료된 시점에서 검증 쓰레드는 쓰레드풀(Threadpool)을 통해 여러 개의 자식 쓰레드를 생성하고 학습모델을 복사하여 모델 복제본과 분할한 데이터를 각 자식 쓰레드에 할당한다. 자식 쓰레드가 요청하는 추론 연산은 텐서플로우 커널에 의해 스케줄링 되어 GPU에서 병렬적으로 검증과정을 수행한다. 이와 동시에 훈련 쓰레드는 유휴 CPU 리소스를 사용하여 다음 에폭 훈련에 대한 데이터 전처리를 수행한다. 훈련 쓰레드는 검증 쓰레드의 종료 유무에 관계없이 전처리 과정이 끝나면 다음 에폭 훈련을 수행한다. 검증 쓰레드는 자식 쓰레드의 검증 과정이 종료되면 다음 에폭에 대한 훈련 쓰레드가 종료될 때까지 대기한다. |

연구의 방법과 결과가 분명히 서술되었는가? 수정필요

연구 결과 중 모델 복사 오버헤드 측정 및 예측 결과가 필요함. MNIST와 같은 작은 모델인 경우 모델 복사 오버헤드가 미미할 수 있으나, 현실적인 모델의 경우에는 이 오버헤드가 중첩으로 얻은 성능 향상을 상쇄할 수 있음.

|  |
| --- |
| **답변** 전체 훈련 시간에 대해서 모델 복사에 의한 오버헤드가 미치는 영향을 측정하는 실험을 추가로 진행하고 관련 내용을 그래프와 함께 아래와 같이 추가하였습니다.  **[수정사항 반영, 8페이지]** 마지막으로, 우리는 제안 기법에서 검증 쓰레드 실행 전 모델 복사가 훈련시간에 미치는 영향을 확인하기 위해 전체 훈련시간 중 1에폭 수행시간에서 모델복사가 차지하는 비율을 계산하였다. 그림 14는 합성곱 신경망(CNN)모델과 1GB MNIST 데이터셋을 사용하여 검증 쓰레드 개수는 6개에서 48개로 설정했을 때 측정한 결과를 보여준다. 즉, 검증 쓰레드가 48개이면 모델 객체복사를 48번 수행한다. 그림 14에서 보여주듯이, 검증 쓰레드가 6개일 때, 1에폭 훈련시간에 대한 모델복사 수행시간에 대한 비율이 0.01%만큼 차지하고 검증 쓰레드의 개수를 48개로 확장했을 때, 0.41%만큼만 차지하는 것을 확인할 수 있다.  위의 실험결과에서 보여주듯이 검증 쓰레드 개수를 시스템 CPU 코어 개수보다 8배 많은 48개까지 확장하여도 1에폭 훈련시간에서 차지하는 비율이 0.5% 이하로, 전체 훈련시간에 대해 미치는 영향이 작다고 할 수 있다. 향후 VGG16, ResNet50과 같은 대규모 학습모델과 대용량 데이터셋을 사용하여 제안기법의 확장성에 대한 연구를 진행하고자 한다.  그림 14 에폭 훈련시간에 대한 모델 복사 오버헤드 비율  Fig. 14 The ratio of model copy overhead to epoch training time. |

과거의 연구와 비교가 되었는가? 수정필요

Background만 있고 Related work이 없음. 기존의 훈련/검증을 병렬화하는 연구들, parallel validation 연구의 차별성을 분명히 서술할 필요가 있음.

|  |
| --- |
| **답변** 본 연구와 유사한 관련 연구를 아래와 같이 추가하였습니다.  **[수정사항 반영, 2페이지]** Downpour SGD는 GPU 클러스터 환경에서 훈련데이터와 학습모델의 복제본을 각 GPU에 분배하고 개별적으로 파이프라인을 구성하여 수행한다. 갱신된 모든 모델의 가중치는 공유가중치 서버에 전송하여 가중치 평균값을 각 GPU의 모델에 재분배하여 가중치를 반영한다. 이를 통해 모델복사를 사용하지 않는 기존 방식보다 일정 정확도에 도달하는데 60% 빠른 수행시간을 보였다[11]. Nimble은 단일 GPU 환경에서 모델 학습에 필요한 연산을 여러 개의 GPU 스트림(Stream)[12]을 생성하여 분산 처리하는 스케줄링 기법을 제안하여 파이토치에서 기존대비 1.88배만큼 처리속도가 향상되는 결과를 보였다[13]. |

참고 문헌의 인용이 적합한가? 적합

(지침사항없음)

논문의 독창성이 있는가? 불분명

(지침사항없음)

|  |
| --- |
| **답변** 본 연구의 독창성을 제고하기 위해 아래와 같은 내용을 추가하였습니다.   1. **[수정사항 반영, 3페이지]** 지금까지의 병렬화 연구들은 학습모델의 훈련 과정을 분산 처리하여 처리속도를 높이는 방법을 제시하였다. 반면, 본 연구에서는 CPU 멀티 쓰레드로 학습모델의 훈련과정과 검증과정을 중첩하여 CPU와 GPU의 활용률을 높임으로써 전체 훈련시간을 개선하는 새로운 기법을 제안한다. |

논문의 내용이 많은 독자의 관심이 되는가? 보통

(지침사항없음)

서술식 심사평

CPU/GPU pipeline stages를 overlap하여 훈련/검증 시간을 줄일 수 있음을 보여주었으나, 현실적인 모델에 대해서도 performance gain과 model copy overhead가 scale할 지 보일 필요가 있음.

1. training prepration과 이전 epoch의 validation을 중첩하기 위한 모델 복사 오버헤드에 대한 디테일한 evaluation이 필요함. MNIST보다 현실적인 크기의 모델을 고려할 필요가 있음.

2. 위와 연관하여, performance gain이 scalable한지, CPU/GPU utilization 증가가 그대로 실행 시간 감소로 translate 되는지 등에 대한 분석이 유용할 것임.

3. 쓰레드의 의미를 분명히 해야 함. 독립적인 execution flow를 뜻하는 것인지, 실제 CPU나 GPU의 쓰레드를 지칭하는 것인지를 설명해야 함.

4. Parallel validation에 대한 이전 연구들과의 비교가 필요함.

|  |
| --- |
| **답변** 1. 모델 복사 오버헤드에 대한 디테일한 evaluation을 위해 1GB로 용량을 늘린 MNIST 데이터셋을 분류하는 모델에 대한 copy overhead를 측정하는 실험을 추가로 진행하고 관련 내용과 그래프 및 대규모 모델 실험에 대한 향후 연구계획을 아래와 같이 추가하였습니다.  **[수정사항 반영, 8페이지]** 마지막으로, 우리는 제안 기법에서 검증 쓰레드 실행 전 모델 복사가 훈련시간에 미치는 영향을 확인하기 위해 전체 훈련시간 중 1에폭 수행시간에서 모델복사가 차지하는 비율을 계산하였다. 그림 14는 합성곱 신경망(CNN)모델과 1GB MNIST 데이터셋을 사용하여 검증 쓰레드 개수는 6개에서 48개로 설정했을 때 측정한 결과를 보여준다. 즉, 검증 쓰레드가 48개이면 모델 객체복사를 48번 수행한다. 그림 14에서 보여주듯이, 검증 쓰레드가 6개일 때, 1에폭 훈련시간에 대한 모델복사 수행시간에 대한 비율이 0.01%만큼 차지하고 검증 쓰레드의 개수를 48개로 확장했을 때, 0.41%만큼만 차지하는 것을 확인할 수 있다.  위의 실험결과에서 보여주듯이 검증 쓰레드 개수를 시스템 CPU 코어 개수보다 8배 많은 48개까지 확장하여도 1에폭 훈련시간에서 차지하는 비율이 0.5% 이하로, 전체 훈련시간에 대해 미치는 영향이 작다고 할 수 있다. 향후 VGG16, ResNet50과 같은 대규모 학습모델과 대용량 데이터셋을 사용하여 제안기법의 확장성에 대한 연구를 진행하고자 한다.  그림 14 에폭 훈련시간에 대한 모델 복사 오버헤드 비율  Fig. 14 The ratio of model copy overhead to epoch training time.  2. CPU/GPU utilization 증가와 실행시간 감소의 연관성을 서술하기 위해 제안기법의 구현내용에 대한 서술을 아래와 같이 추가하였습니다.  **[수정사항 반영, 7페이지]** 본 제안 기법은 텐서플로우 커널이나 케라스 라이브러리의 기존 코드를 수정하지 않고 사용자 수준에서 Python3.8 버전을 사용하여 구현되었다. 중첩되어 동작하는 훈련 쓰레드와 검증 쓰레드는 Python에서 제공하는 Threading 라이브러리를 사용하였으며 에폭에 대한 훈련 쓰레드가 종료되는 시점에 검증 쓰레드가 동작한다. 검증 쓰레드는 모델복사가 완료되는 시점에 다음 에폭에 대한 훈련 쓰레드를 호출하고 쓰레드풀(Threadpool)을 통해 복사된 모델과 분할된 검증데이터를 분배하여 멀티 쓰레드로 동작한다. 따라서 훈련 쓰레드는 CPU에서 훈련데이터 전처리를 수행하는 동시에 검증 쓰레드는 멀티 쓰레딩을 통해 개별 쓰레드의 작업을 텐서플로우 커널으로 전달한다. 텐서플로우 커널은 요청받은 작업을 여러 개의 GPU 스트림에 할당하여 검증과정을 수행한다[6,12,13]. CPU에서 동작중인 각 쓰레드가 동시에 작업을 요청함에 따라, 이 시점에서 CPU 활용률이 증가하고 검증 과정에서 여러 개의 모델이 병렬적으로 GPU 리소스를 활용함으로써 GPU 활용률 또한 증가하여 전체 수행시간이 단축되는 효과를 보인다.  3. 쓰레드의 의미와 병렬성에 대한 명확한 서술을 위해 아래와 같이 추가하였습니다. **[수정사항 반영, 3페이지]** 우리는 이러한 단점을 개선하기 위해 우선 훈련 쓰레드(Training-thread)와 검증 쓰레드(Validation-thread)로 훈련과 검증을 위한 쓰레드를 구분하였다. 그리고 훈련이 완료된 쓰레드의 최종 가중치 값들을 검증 쓰레드로 복사하는 모델 복사(Model Copy) 과정을 설계하였다 이는, CPU에서 생성되어 실행되는 쓰레드로, 각 쓰레드는 텐서플로우 커널에서 스케줄링 되어 GPU에서 병렬적으로 동작한다[6,12,13]. 검증 쓰레드의 경우는 멀티 쓰레드로 동작한다.  **[수정사항 반영, 4페이지]** 훈련과정과 검증과정을 각 쓰레드로 구분하여 동작한다. 첫 번째 에폭에 대한 훈련 쓰레드가 완료된 시점에서 검증 쓰레드는 쓰레드풀(Threadpool)을 통해 여러 개의 자식 쓰레드를 생성하고 학습모델을 복사하여 모델 복제본과 분할한 데이터를 각 자식 쓰레드에 할당한다. 자식 쓰레드가 요청하는 추론 연산은 텐서플로우 커널에 의해 스케줄링 되어 GPU에서 병렬적으로 검증과정을 수행한다. 이와 동시에 훈련 쓰레드는 유휴 CPU 리소스를 사용하여 다음 에폭 훈련에 대한 데이터 전처리를 수행한다. 훈련 쓰레드는 검증 쓰레드의 종료 유무에 관계없이 전처리 과정이 끝나면 다음 에폭 훈련을 수행한다. 검증 쓰레드는 자식 쓰레드의 검증 과정이 종료되면 다음 에폭에 대한 훈련 쓰레드가 종료될 때까지 대기한다.  **[수정사항 반영, 2페이지]** Downpour SGD는 GPU 클러스터 환경에서 훈련데이터와 학습모델의 복제본을 각 GPU에 분배하고 개별적으로 파이프라인을 구성하여 수행한다. 갱신된 모든 모델의 가중치는 공유가중치 서버에 전송하여 가중치 평균값을 각 GPU의 모델에 재분배하여 가중치를 반영한다. 이를 통해 모델복사를 사용하지 않는 기존 방식보다 일정 정확도에 도달하는데 60% 빠른 수행시간을 보였다[11]. Nimble은 단일 GPU 환경에서 모델 학습에 필요한 연산을 여러 개의 GPU 스트림(Stream)[12]을 생성하여 분산 처리하는 스케줄링 기법을 제안하여 파이토치에서 기존대비 1.88배만큼 처리속도가 향상되는 결과를 보였다[13].  **[수정사항 반영, 3페이지]** 지금까지의 병렬화 연구들은 학습모델의 훈련 과정을 분산 처리하여 처리속도를 높이는 방법을 제시하였다. 반면, 본 연구에서는 CPU 멀티 쓰레드로 학습모델의 훈련과정과 검증과정을 중첩하여 CPU와 GPU의 활용률을 높임으로써 전체 훈련시간을 개선하는 새로운 기법을 제안한다. |