**연구노트**

**(수학과)이 재 봉**  2017.07.03 (월)

|  |  |  |
| --- | --- | --- |
|  | | |
| **연 구**  **주 제** | **특허분석을 통한 미래기술 예측** | |
| **관련 논문 조사** | |
| **구 분** | **연구내용** | **진행률** |
|  | 김종영, 구글 TensorFlow 소개  ○ TensorFlow  -모델 확장성: TensorFlow에서 행렬 계산 및 comvolution은 심볼릭 그래프를 이용하여 수행하며 사용자 는 복잡한 신경망 구조를 역전파(back-propagation) 구현에 구애받지 않고 만드는 것이 가능. 현재 그래프 상에서 반복, 조건 분기와 같은 기능은 구현되어 있지 않은 상태. 이 점은 RNN (Recurrent Neural Network)을 구현할 때 파이썬 언 어가 제공하는 반복 메커니즘을 사용해야 한다는 점과 그래 프 컴파일러 최적화를 할 수 없다는 점에서 단점으로 작용  -인터페이스: 파이썬과 C/C++ 언어를 기본적으로 제공, 이는 네이티브 코드 또는 속도가 중요시되는 환경에서 장점 으로 작용  -범용성: ARM 아키텍처에서도 사용이 가능, 이는 단말에 서 서버에 이르기까지 다양하게 사용될 수 있음을 의미. 마이 크로소프트 윈도우는 현재 미지원 상태임.  -성능: cuDNN v2(NVIDIA) 기준 Torch에 비해 1.5배 느린 성능을 보이며 배치 사이즈 128 GoogleNet의 경우 메 모리 부족 현상을 보임.  -다중 GPU 지원: GPU에서 CPU로 메모리 전송, 다수의 GPU상에서 얻어진 결과 취합 등이 비교적 용이함. 분산 트 레이닝은 향후 포함될 예정임  -모델 디버깅: TensorBoard룰 사용하여 변수 추적이 용이 하도록 설계되어 있음.  ○ Theano  -모델 확장성: 거의 대부분의 학습 알고리즘이 구현되어 있으며 심볼릭 그래프를 계산에 도입한 흐름을 만든 사례에 해당. scan이라는 반복 메커니즘을 제공하는데 이는 RNN 구현에 장점으로 작용. 사용자는 이미 만들어져 있는 tensor 연산 프레임워크를 사용하면 되기 때문에 모델 정의와 학습 이 단순함  -인터페이스: 파이썬 언어만을 지원  -범용성: 네이티브 언어를 지원하지 않으며 파이썬 인터프 리터의 비효율성으로 인해 산업계에서 그다지 많이 사용되지 않음.  -성능: 대규모 네트워크의 경우 Torch7과 동일한 성능을 보임, 하지만 C/CUDA 코드를 컴파일해서 바이너리 코드롤 12 한국컴퓨터정보학회지 만들어내는 시간이 오래 걸림  -다중 GPU 지원: 네이티브 코드 상에서 지원은 하지 않음. 개발자는 로우레벨 프로그래밍 관점에서 직접 구현해야 함  -모델 디버깅: 없음  ○ Torch  -모델 확장성: convolutional 신경망 구현이 우수함, 특히 temporal convolution의 경우 직관적 사용이 용이함. 새로운 레이어 타입을 정의하는 경우 사용자는 포워드, 백워드, 그래 디언트(gradient) 입력의 갱신 모듈을 모두 구현해야 함.  -인터페이스: LuaJIT 환경을 기반으로 실행되며 사용자는 심볼릭 프로그래밍에 대한 부담이 없음.  -범용성: LuaJIT 환경에서 실행되므로 API 수분에서 다른 요소와 통합 작업이 용이하지 않음  -성능: 가장 나은 성능을 보임 -  다중 GPU 지원: TensorFlow와 Theano의 중간 단계임  -모델 디버깅: 없음 |  |
| **차 주**  **계 획** | ■ 계획  - 관련 논문 조사 | |
| **이 슈** | ■ **해당사항 없음** | |

|  |  |
| --- | --- |
|  |  |