|  |  |
| --- | --- |
| 작성자 | 최종우 |
| 날짜 | 2019년 3월 24일 화요일 |
| 연구시간 | 총 시간 |
| 월6 화9 수9 목9 금8 토5 일0 |

그 동안 한 일.

1. ONNX for Midap
   * 동현 선배님의 업무 요청서를 받고 ONNX model을 직접 읽어볼 수 있는 방법, 또한 ONNX model을 layer by layer로 분해할 수 있는 방법을 연구 시작했습니다.
     + ONNX를 local(MACOS)과 server(LINUX)에 모두 깔고 caffe2 model -> ONNX -> tensorflow, YOLO 예제 등을 돌려봤습니다.
     + 현재 ONNX Model을 직접 읽는데는 ONNX runtime 이라는 프로젝트를 reference로 사용하고, Midap에서 Layer by Layer를 사용하기 위해서는 ONNX to Tensorflow를 사용하려고 하고 있습니다.
       - Midap Sim의 최신 버젼의 경우 Caffe2 Model과 Caffe2 Model에 붙였던 Tensorflow를 Depreciate 하여서 Tensorflow로 변환 뒤 다시 Midap Sim으로 붙이는 것의 타당성을 검토 중입니다. 기존 Tensorflow 모듈을 사용하지 않으면 ONNX Runtime으로 모든 것을 해결하는 것이 좋을지도 모르겠습니다.
       - 현재 Midap Sim은 Pytorch 기반으로 Network를 돌리는데, ONNX에서 바로 Pytorch로 전환되는 공식 코드는 없습니다.
       - ONNX-TF Supported Operators / ONNX Operators: 126 / 158
       - ONNX-Caffe2(Pytorch)Backend/ONNX 39 / 102
         * 왜인진 모르겠으나 서로 오퍼레이터의 갯수를 다르게 취급하고 있어서 정확한 이유를 체크중입니다.
2. Midap Sim 매뉴얼 점검 및 테스트 피드백
   * 어제 1차로 local 과 server에서 테스트 해서 1차 피드백을 드렸습니다. 공식 문서로 정리 중입니다.

다음 주에 해야 일.

1. Midap Sim 매뉴얼 점검 및 테스트 피드백 마무리
2. ONNX to Midap
   * Runtime / ONNX-TF / ONNX2-Caffe2 중 방향을 정하고 Operator를 직접 분석 시작
     + Operator 원형 확인, 가능하다면 양쪽 Backend가 왜 Operator 숫자를 다르게 세는지 이유 확인