AI 컴파일러(MLIR/IREE) 학습/세미나 계획

1. 세미나 운영 방식

시간 및 장소

- **시간:** 매주 [월/금] [오전/오후] (1 ~ 1.5시간 소요 예상) / 교수님과 상의 후 결정 예정입니다
- **장소:** [누리라운지 / 정보마루 / 연구실 세미나실 중 택 1] / 교수님과 상의 후 결정 예정입니다
- **형식:** 오프라인 발표 및 토론

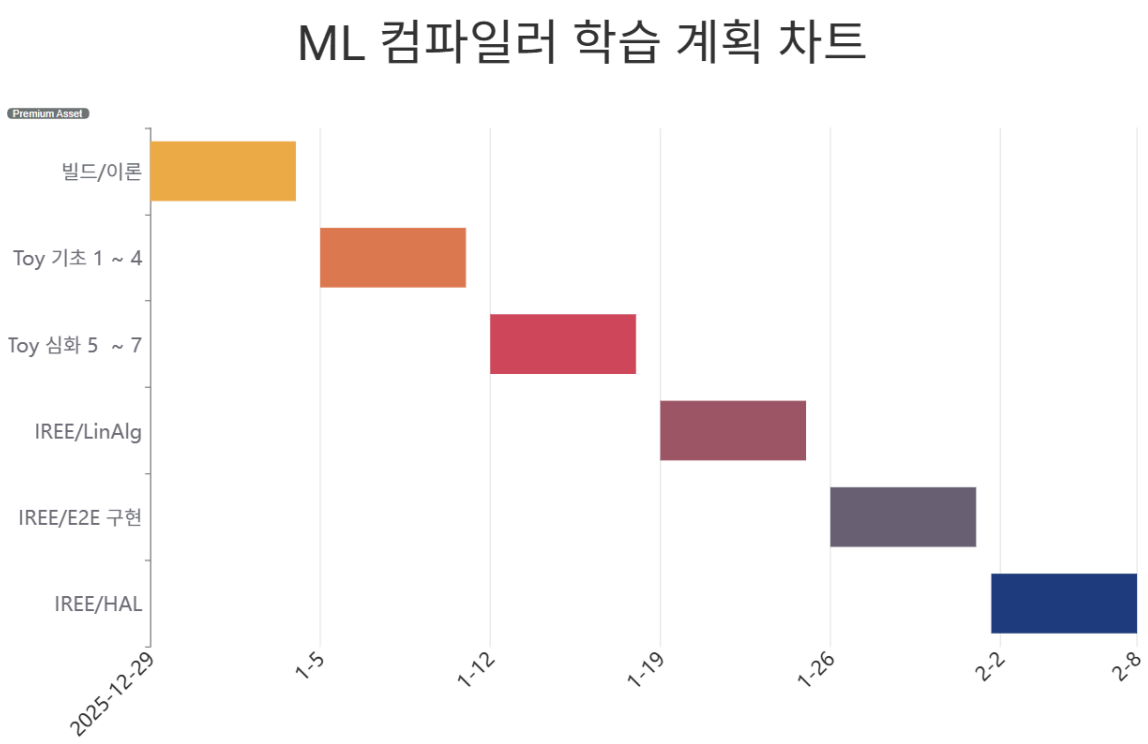
발표 형식

모든 발표자는 아래 4가지 항목을 필수로 포함하여 자료를 작성합니다.

1. **이번 주 목표:** 구현 목표 및 핵심 내용 요약
2. **핵심 개념 및 코드:** 이론 나열 지양, 실제 작성한 코드와 생성된 IR 결과물 위주 설명
3. **트러블 슈팅:** 발생했던 에러, 시도한 방법, 최종 해결책 공유 (가장 중요)
4. **인사이트 및 계획:** 튜토리얼 외 발견한 팁, 다음 주 계획 점검

코드 및 산출물 관리

- **GitHub Repository:** 개인 계정에 `study-mlir-iree` 공개 레포지토리 생성.
- **아카이빙**
  - 발표자료 - 구글 슬라이드
  - 각종 문서 - 구글 드라이브



2. 주차별 상세 계획 (25.12.29 ~ 26.02.08)

1주차: 환경 구축 및 컴파일러 기초 이론

- **기간:** 25.12.29 ~ 26.01.04
- **목표:** 리눅스 환경(Ubuntu 22.04.05)에서 LLVM/MLIR 소스코드 빌드 및 툴 실행
- **세부 활동:**
  - CMake, Ninja, Clang 필수 패키지 설치 및 환경 변수 설정
  - LLVM Project 클론 및 빌드 (Debug, Release 모드 비교)
  - **[세미나]** TableGen 파일이 C++로 변환되는 과정 시연 및 빌드 에러 공유
- **산출물:** MLIR 빌드 가이드 문서

2주차: MLIR Toy 튜토리얼 1~4 (Dialect 및 ODS)

- **기간:** 26.01.05 ~ 26.01.11
- **목표:** Toy language를 정의하고 AST를 MLIR로 변환
- **세부 활동:**
  - Toy 언어의 AST 구조 파악

    - **ODS**를 사용하여 .td 파일에 연산자 정의
    - toy-translate 툴 구현 및 테스트
    - **[세미나]** 내가 정의한 연산자 구조 발표 및 변환 시연 / ODS를 사용하는 이유
- **산출물:** 연산자가 정의된 .td 파일 코드 분석 보고서 & 세미나 발표 자료
- **참고 자료**
    - https://mlir.llvm.org/docs/Tutorials/Toy/

3주차: MLIR Toy 튜토리얼 5~7 (최적화 및 Lowering)

- **기간:** 26.01.12 ~ 26.01.18
- **목표:** MLIR의 핵심인 최적화 패스와 저수준 변환 이해
- **세부 활동:**
    - Canonicalizer 등을 이용해 불필요한 연산 제거 패스 구현
    - Toy Dialect를 Affine, Arith Dialect로 낮추는 로직 구현 및 JIT 실행
    - **[세미나]** 패스 적용 전후의 IR 코드 비교 및 패턴 리라이팅 매커니즘 분석
- **산출물:** 최적화 전후 성능 비교 리포트

4주차: IREE 빌드 및 LinAlg

- **기간:** 26.01.19 ~ 26.01.25
- **목표:** 행렬 연산이 컴파일러 내부에서 어떻게 쪼개지는지 확인
- **세부 활동:**
    - IREE 컴파일러 및 런타임 소스 빌드
    - LinAlg 튜토리얼 진행 및 행렬 곱 연산 구조 확인
        - Transform Dialect를 사용하여 타일링(Tiling) 스크립트를 작성하고, 루프가 쪼개지는 과정을 `mlir-opt`로 확인.
    - **[세미나]** 메모리 계층 구조와 타일링의 관계 설명 및 시연
- **산출물:** NPU를 위한 타일링 기초 발표 자료
- **참고자료**
    - https://iree.dev/community/blog/2024-01-29-iree-mlir-linalg-tutorial/

5주차: IREE End-to-End 파이프라인

- **기간:** 26.01.26 ~ 26.02.01
- **목표:** 외부 TFLite 모델을 가져와서 내 컴퓨터에서 IREE로 실행
- **세부 활동:**
  - MobileNet v2 또는 MNIST 모델 확보
  - iree-import-tflite, iree-compile, iree-run-module 파이프라인 수행
  - **[세미나]**
    - **[분석]** 컴파일 단계별 IR 변화 추적: TFLite → TOSA → Linalg 변환 과정 해부
    - **[시연]** Python 바인딩을 활용하여 실제 이미지(jpg)를 분류하는 추론 데모.
    - **[트러블 슈팅]**
- **산출물:** 세미나 자료
- **참고자료**
  - https://github.com/iree-org/iree/tree/main/samples/colab

6주차: IREE Hardware Abstraction Layer 분석

- **기간:** 26.02.02 ~ 26.02.08
- **목표:** IREE 하드웨어 추상화 계층(HAL) 분석
- **세부 활동:**
  - IREE HAL 코드 구조 분석 (메모리 할당자, 커맨드 버퍼 등) / 아키텍처 다이어그램
  - 벤치마크 툴을 활용한 지연 시간 측정
  - [세미나] HAL 아키텍처와 실제 코드 구성 분석
- **산출물:** HAL 분석 세미나 자료