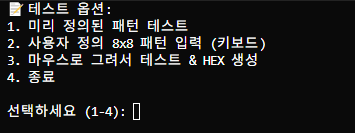
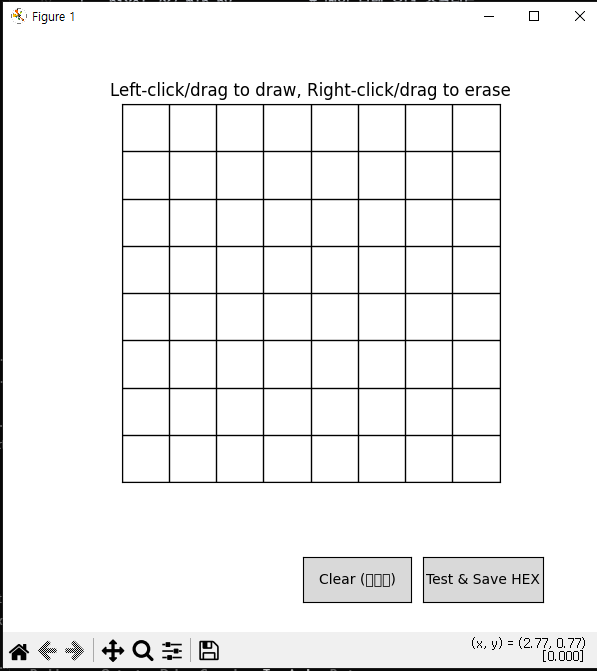
안녕하세요 교수님. 저희 논리회로는 로지심 파일 하나만으로 작동되는 것이 아니기 때문에,번거로우시겠지만 혹시 저희 논리회로를 테스트해보고 싶으시다면,

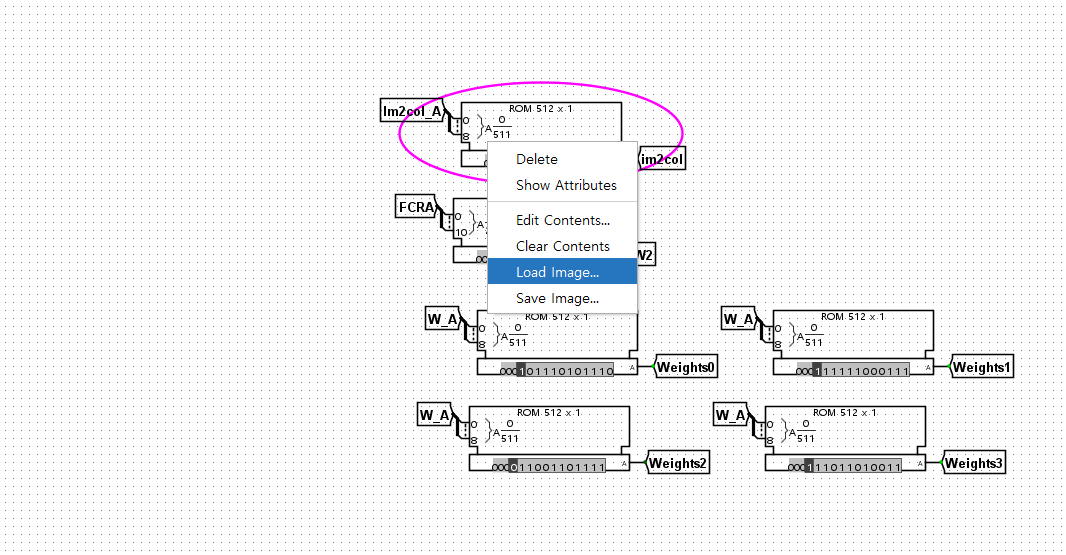
테스트 해보실 수 있게 파이썬코드와 함께 사용방법 가이드 작성해드립니다.

1. 저희가 첨부해드린 requirements.txt에 필요한 파이썬 라이브러리들이 적혀있습니다. python과 pip가 있으시다면, 터미널에 ‘pip install –r requirements.txt’ 명령어 작성하시면 필요한 라이브러리를 설치하실 수 있습니다.
2. 설치를 하셨다면 BNN\_test.py를 실행해주시면(터미널만 사용하시는 경우 ‘python BNN\_test.py’라는 명령어를 통해 실행이 가능합니다.) 터미널에 1~4번까지 선택지를 고를 수 있게 입력창이 나옵니다 3번을 선택하시면 8x8짜리 그림판이 나옵니다. 숫자 그림을 그려주시고 Test & Save Hex 버튼을 누르시면 숫자이미지가 hex파일로 생성됩니다.



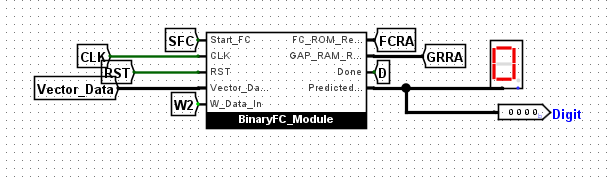


1. 이 hex 이미지 파일을 첨부해드린 회로의 BNN.circ에서 Quantized\_CNN 회로에 im2col 터널이 연결된 ROM에서 Load Image를 클릭하셔서 im2col\_matrix.hex을 업로드 하시면 준비가 끝이 납니다.



이제 Quantized\_CNN 모듈에서 Global\_Start를 누르시고 Auto-Tick frequency를 최대한 높게 설정해주시고 Auto-Tick Enabled 해주시면 계산이 시작됩니다.

이는 컴퓨터 환경에 따라 30초~1분 정도의 시간이 걸리는 것으로 확인이 돼서 조금 기다리시면 맨 오른쪽의 BinaryFC\_Module의 옆에 숫자 디스플레이에 AI가 예측한 숫자값이 나오게 됩니다.



예측성능이 그렇게 좋지는 못합니다.

매우 잘 동작하는 회로를 만들었었지만 연산이 상상 이상으로 오래걸려서 축소화 시키다보니, 숫자 그림의 선이 얇은 경우에만 예측이 잘 됩니다.

저희가 아무래도 논리회로만으로 테스트가 가능한 것이 아니라서 시연을 할 때에도 찜찜한 부분이 많았습니다. 많이 번거로우시겠지만 직접 테스트하셔서 평가에 반영해주신다면 정말 감사드리겠습니다.

감사합니다.