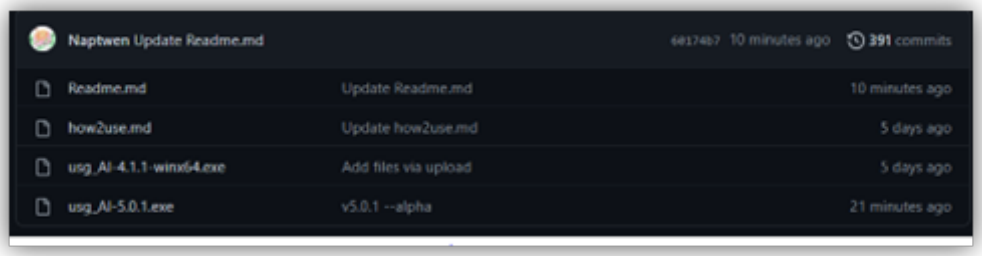
USG AI 엔진 프로그램 사용설명서

본 프로그램의 소스코드는 모두 본인이 제작하였으며, GNU AFFERO LICENSE V3 를 통해 보호됨과 동시에 특별조항을 명시합니다. 또한 특정 저작권에 속하는 OpenGL, OpenCL, OpenCV, base64, FreeType코드는 프로그램에 첨부된 저작권을 확인 하여주시기 바랍니다.

프로그램 설치법

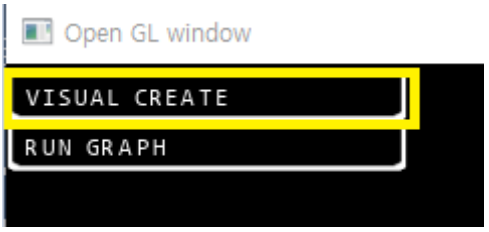
1. 프로그램의 원활한 실행을 위한 사용 소프트웨어 OpenGL, OpenCV, OpenCL을 설치합니다.
2. https://github.com/Naptwen/usgAI 에 접속하여 가장 최신버전의 프로그램 설치파일 받습니다.



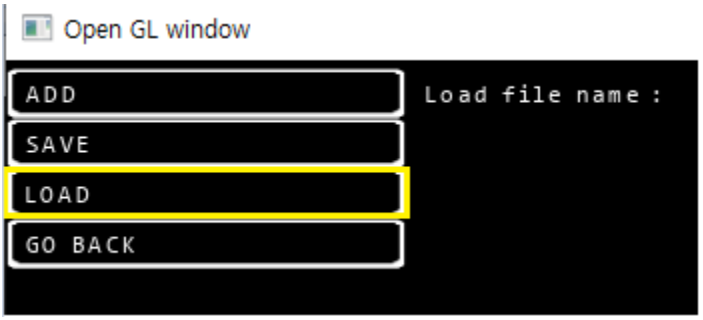
3. 설치된 폴더의 usg_AI 실행파일을 실행합니다.

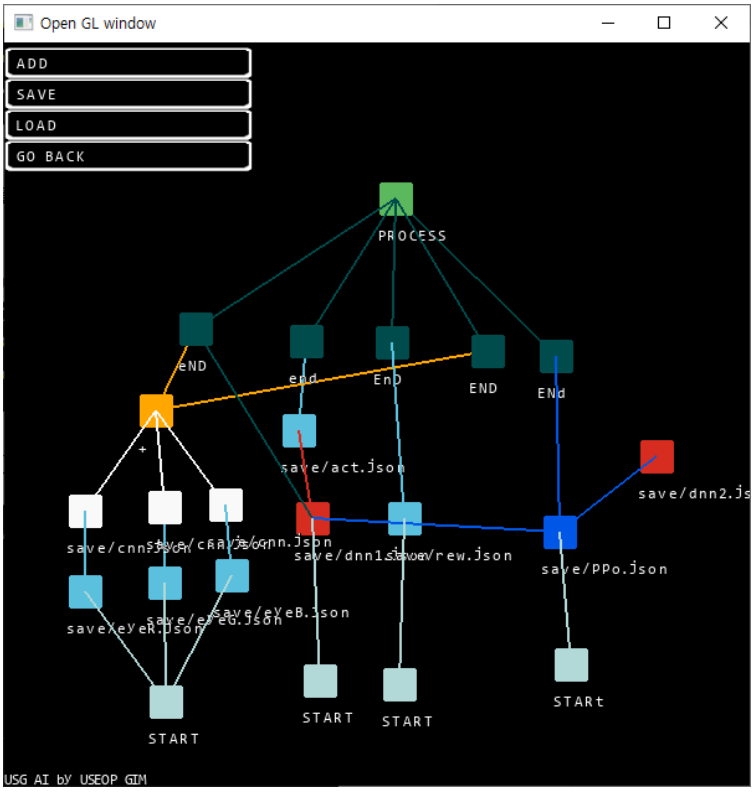
- 위와 같은 화면이 실행된다면 프로그램 설치가 완료되었음을 의미합니다.

4. 프로그램의 테스트를 위하여 VISUAL CREATE 버튼을 클릭합니다.



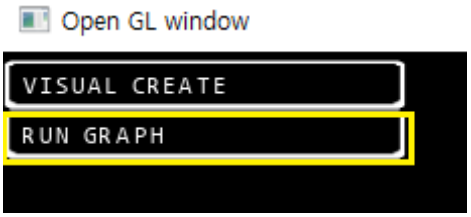
5. LOAD 버튼을 클릭후 “save/test.txt’ 를 입력한후 엔터키를 누릅니다.



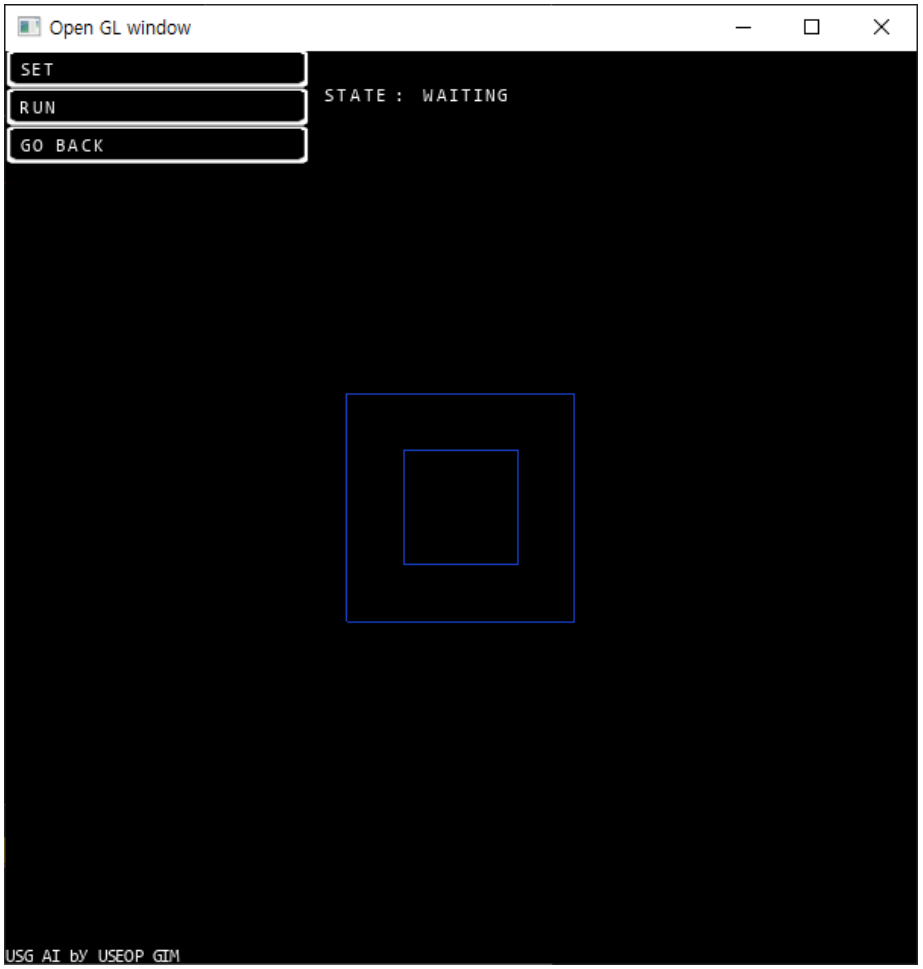


- 위와 같은 화면이 나오지 않는다면, save 폴더안에 예제 파일들이 존재하는지 확인합니다.

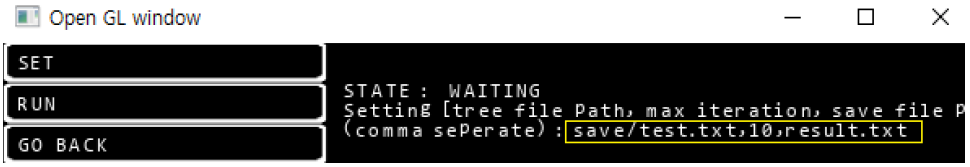
7. RUN GRAPH 버튼을 클릭합니다.



8. 밑의 화면과 같은 화면이 나오면 SET버튼을 클릭합니다.



9. 버튼을 클릭한후 “save/test.txt, 10, result.txt” 를 입력합니다.

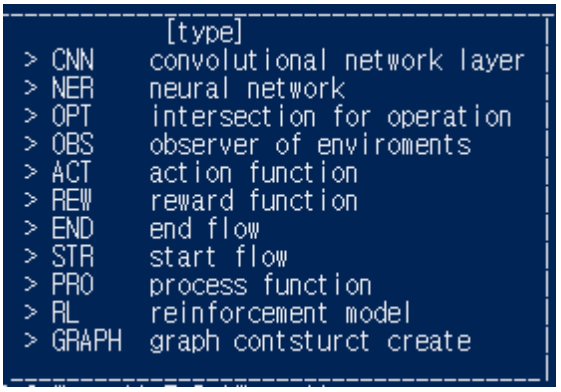


10. 다음으로 RUN 버튼을 클릭합니다.
11. 중앙의 사각형이 움직이면서 방향키가 자동으로 움직이면 프로그램이 작동하는 것입니다.
12. 중앙의 사격형의 움직임이 멈추면 프로그램을 종료하셔도 됩니다.

기본 콘솔 인터페이스 설명

각모델들은 편집파일을 통해 직접 작성 및 편집 혹은 기본 제공되는 콘솔 인터페이스 프로그램을 통해서 작성 할 수 있습니다.

--help : 기본 제공되는 기능을 살펴 볼 수 있습니다.
--license : 라이센스 정보를 표시합니다.
--create "type": 콘솔인터페이스를 통해서 모델들을 작성할 수 있습니다.



CNN : Convolutional Neural Network 블록을 작성합니다.
NER : Deep Neural Network 블록을 작성합니다.
OPT : 사칙연산 블록을 작성합니다. (version 5 는 합과 곱만을 제공합니다.)
OBS : 화면에 관한 공유 라이브러리 블록을 작성합니다.
ACT : 행동에 관한 공유 라이브러리 블록을 작성합니다.
REW: 보상에 관한 공유 라이브러리 블록을 작성합니다.
END: 그래프 흐름의 끝을 의미하는 블록을 작성합니다.
STR: 그래프 흐름의 시작을 의미하는 블록을 작성합니다.
PRO: 그래프 흐름을 제어할 블록을 작성합니다.
RL: 강화학습 모델 블록을 작성합니다.
GRAPH: 그래프를 작성합니다.

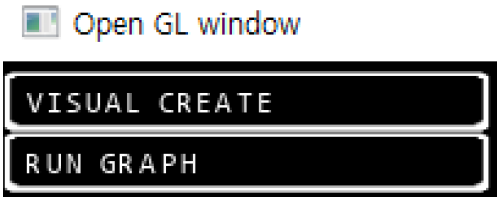
프로그램에 실제 사용되는 것은 CNN, NEW, OBS, ACT, REW 들입니다.
GRAPH, OPT, END, STR 은 GUI 프로그램을 통해서 작성가능합니다.

--setting : 각종 설정을 조절합니다.
gpgpuon(off) : gpgpu 계산을 on off합니다 (1만개 이상일때 효과가 더 높습니다)
verboseon(off) : 자세한 설명을 추가로 보여줍니다. (터미널은 ASC 설정을 해주세요)
multion(off) : 멀티 쓰레드 환경을 허용합니다.
나머지 설정은 아직 테스트 중입니다.

--run "불러올 파일 이름**" "**최대 실행 횟수**" "**저장 파일이름**"** : 프로그램을 실행합니다.

불러올 파일 이름 : 반드시 그래프 흐름 파일이어야합니다.
최대 실행 횟수 : 프로그램의 최대 실행 횟수입니다.
저장 파일 이름 : 저장할 파일 이름을 의미합니다.

그래픽 인터페이스 설명



VISUAL CREATE : 제어흐름도를 작성하기 위한 프로그램을 실행합니다.
RUN GRAPH : 그래프 파일을 동작하기 위한 프로그램을 실행합니다.

VISUAL CREATE

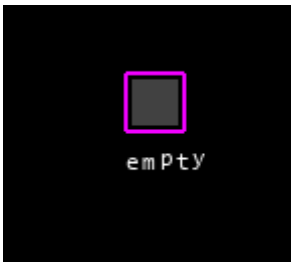
먼저 블록이란 각 알고리즘을 실행할 단위입니다.
본 머신러닝 엔진프로그램은 이러한 블록들의 관계를 통해 제어흐름을 조절하고 보다 쉽게 인공지능의 다양한 알고리즘 구현에 있어서 관계도를 통하여 작성하기 쉽도록 구성되어있습니다.

ADD : 블록을 추가합니다.
SAVE : 현제 선택된 블록과 그 블록의 자식블록들을 모두 저장합니다.
LOAD : 그래프 파일을 불러옵니다.

사용법

1. ADD를 클릭하면 화면에 empty라는 블록이 생성됩니다.

2. 블록을 왼쪽 클릭하면 블록주변에 테두리가 나타나며 현제 선택된 블록을 나타냅니다.



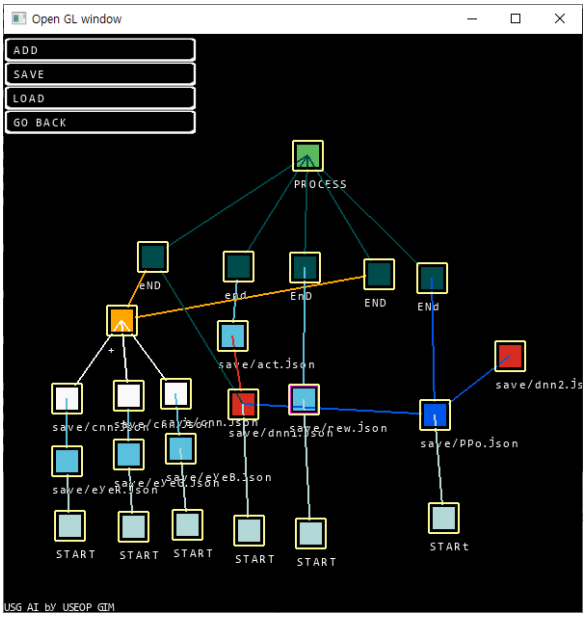
3. 블록 오른쪽 클릭하면 파일이름을 입력할 수 있습니다.

4. 파일의 종류에 따라 블록의 색이 적용되며 아래와 같습니다.

파일명 입력시 “+”,”x”, “END”,” START” 들은 파일이 없어도 각각의 블록들로 자동으로 변하게됩니다.

5. 왼쪽 클릭 후 다른 블록을 클릭하게 되면 각 블록끼리 연결되며 선택했던 블록이 자식 블록이 되고 마지막에 클릭한 블록이 부모 블록이 됩니다.

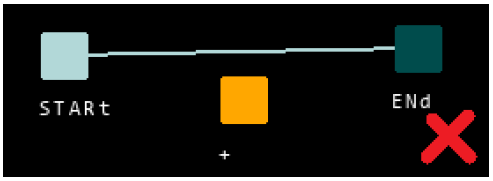
6. 부모 블록을 클릭시 연결된 모든 블록을 표시해줍니다 (v5.0.2버전 이후)

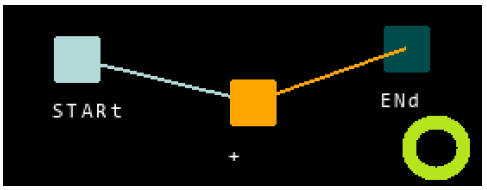


7. 같은 블록을 두번 왼쪽 클릭시 해당 블록에 연결된 모든 관계가 제거됩니다.

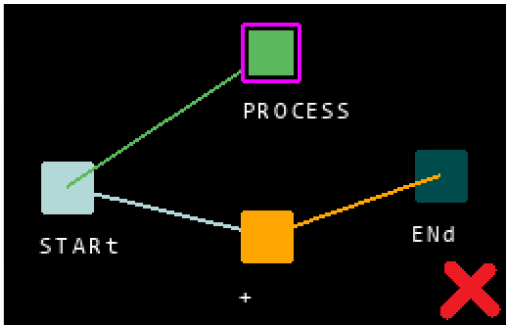
관계도 설명

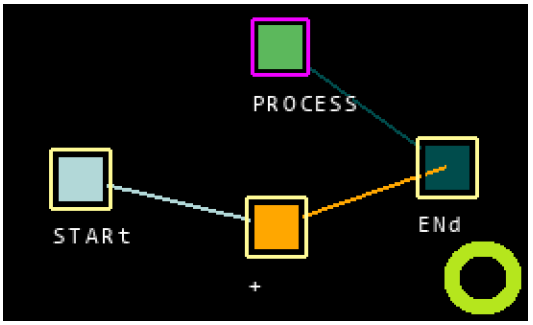
1. 프로그램의 실행과 끝을 의미하는 START 와 END 블록은 반드시 각 블록관계의 처음과 끝에 존재하며, 쌍을 이루어여합니다.



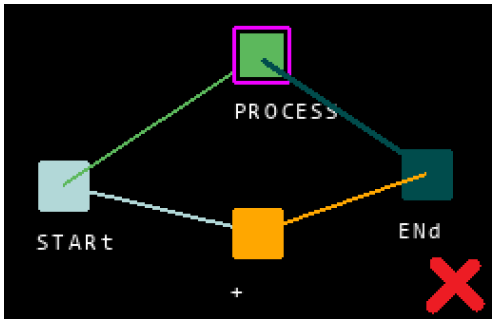


2. 프로그램의 실행순서를 제어할 PROCESS블록이 END블록의 부모블록으로 존재해야합니다.

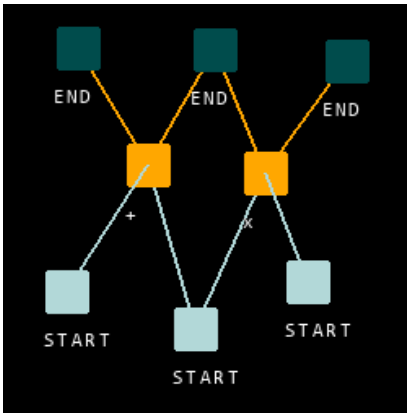




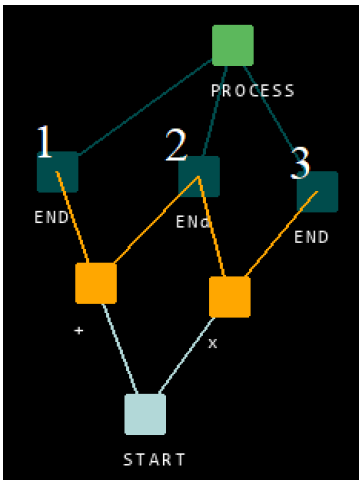
주의 : 부모가 다시 자식으로 연결되는 순환구조는 프로그램을 강제종료시킵니다.



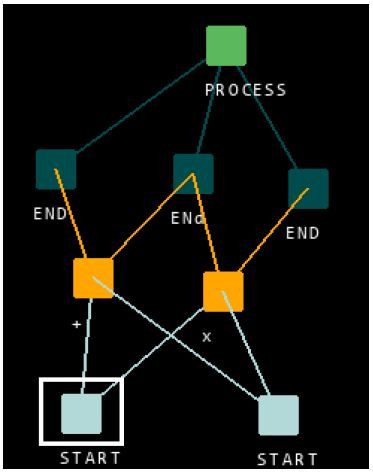
3. 한블록이 가질 수 있는 부모와 자식의 수는 제한이 없습니다.



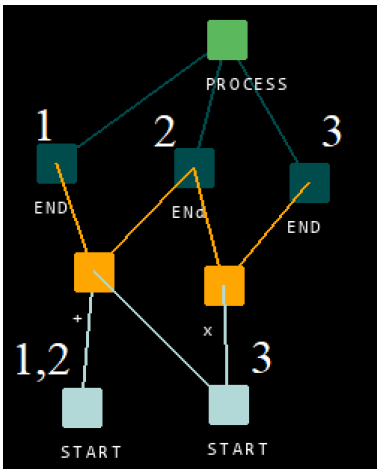
4. 프로그램의 실행 순서는 PROCESS에 연결된 END블록중 가장 왼쪽부터 실행됩니다.



5. 마찬가지로 END블록은 자식 블록중 가장 왼쪽의 START블록으로부터 시작합니다

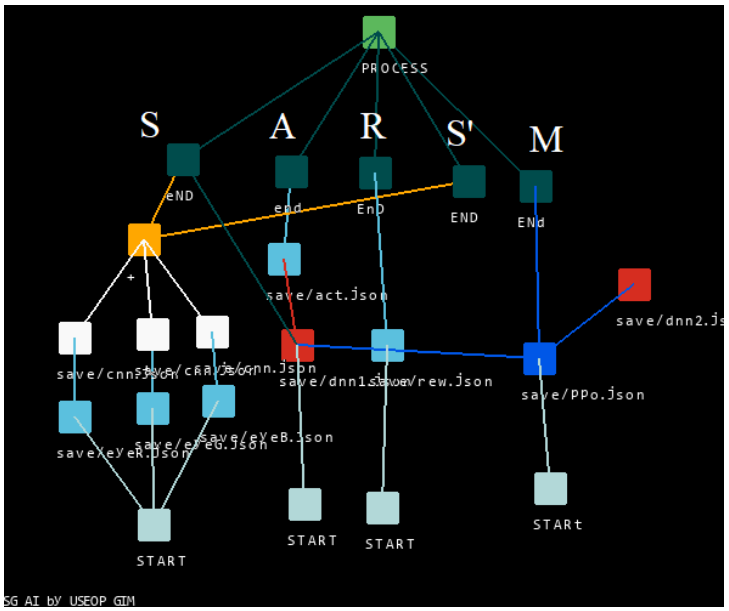


위와 같은경우는 가장 왼쪽의 START만 작동합니다.



위와 같은경우는 1번째 2번째 END 블록은 가장 왼쪽의 START와 연결되며 3번째 가장 오른쪽 START와 연결됩니다.

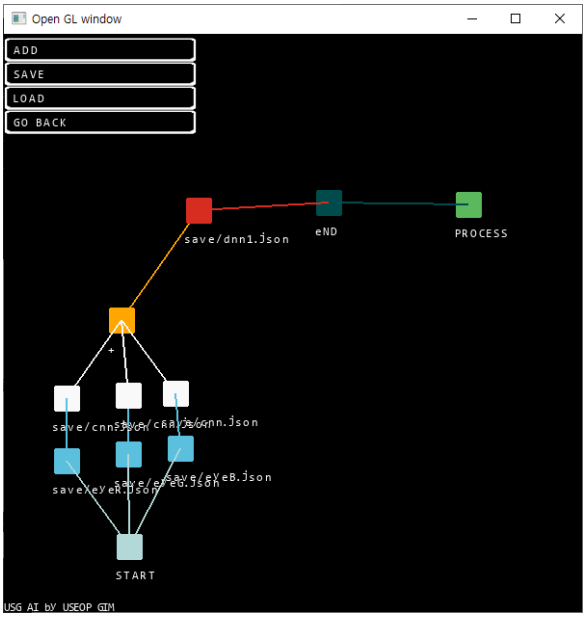
강화학습 블럭
(v5.0.2) 새로운 버전 부터 프로세스 블록이 모든 블록을 관리합니다.



- S : 상태 값을 가져옵니다.
- A : 행동 값을 가져옵니다.
- R: 보상 값을 가져옵니다.
- S': 변화된 상태값을 가져옵니다.
- M : 모델에게 명령을 내립니다.
- 강화학습 블럭의 뉴럴네트워크 중 가장 왼쪽의 뉴럴 네트워크블럭이 메인모델이됩니다. (실제 행동을 결정하는 네트워크)

예제) save/test.txt

실제 시뮬레이션시 작성 모양

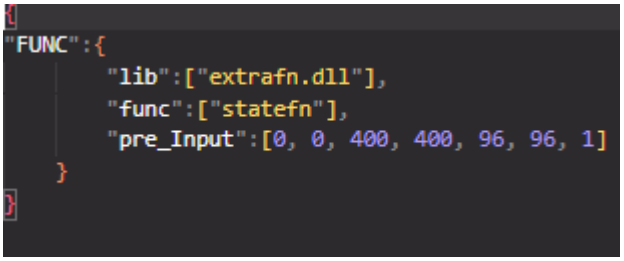


위와 같이 학습후 저장된 뉴럴네트워크 블럭을 연결하여 process를 직접 실행하시면 됩니다.
예제) save/simul.txt

공유 라이브러리 함수 모양

```
extern "C" __declspec(dllexport) void statefn(std::deque<std::vector<float>> &src, std::vector<float> &dist)
```

공유함수의 모양은 위와 같아야하며.



```
{
"FUNC":{
        "lib":["extrafn.dll"],
        "func":["statefn"],
        "pre_Input":[0, 0, 400, 400, 96, 96, 1]
    }
}
```

실행 파일 포맷은

1. FUNC : 태그 이름
2. LIB : 공유라이브러리 위치 이름 (Linux 는 확장자명 a)
3. FUNC : 공유 라이브러리 속 함수이름
4. PRE_INPUT : 초기설정값(만약 가장 처음의 블럭이면 변하지 않는 초기 값)

과 같이 작성하시면 됩니다.
위와 같은 형식의 동적 라이브러리 함수는 무엇이는 실행이 됩니다.
예제) src/extrafn.cpp : v5.x,x 에서는 윈도우와 리눅스만 지원합니다.

텍스트 파일 작성 요령
첨부된 save폴더속 파일와 같은 형식으로 작성하시면 됩니다.
Neural network 의 *weight* 와 *basis*는 *base64*의 규칙에 따라 작성되어있으므로 직접 작성시 혹은 수정시 *Console Interface Editor* 로 작성하는 것을 추천드립니다.