

*XPS Analyzer*



KHU SpondLab

최정훈

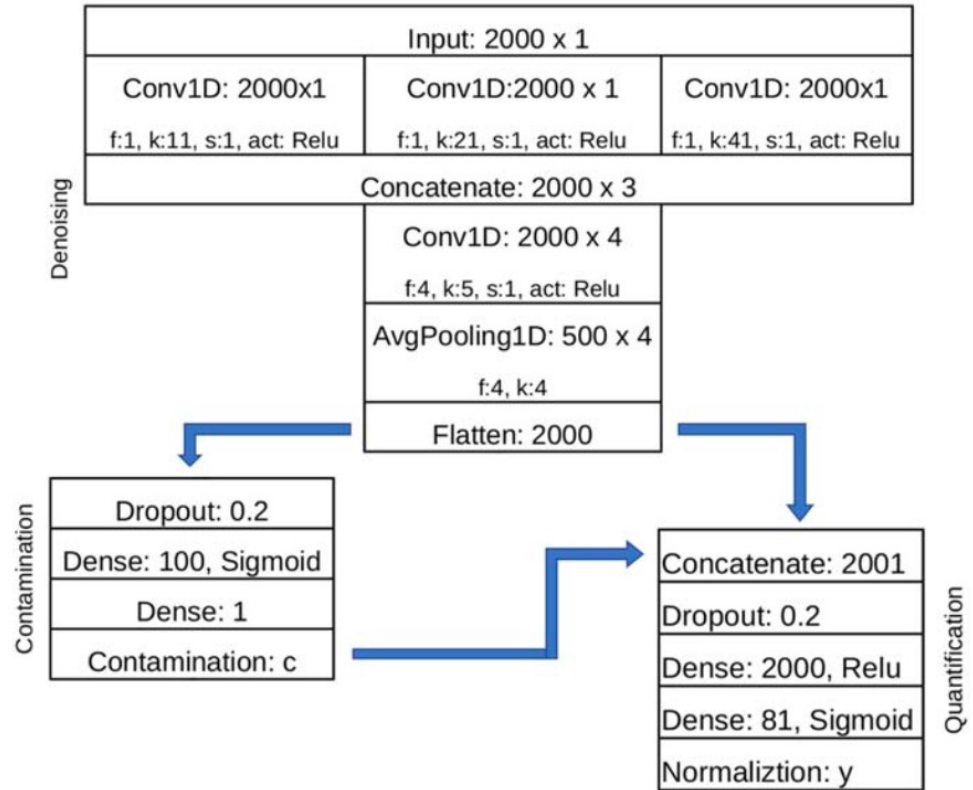
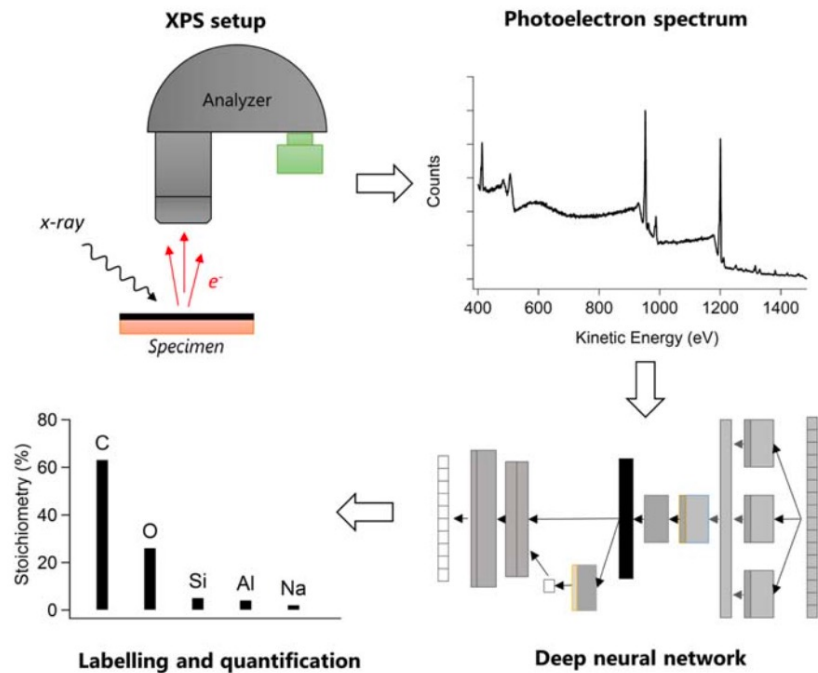
# *Index*

- 01 About **Project**
- 02 About **Process**
- 03 About **Programs**

# 01 About Project

[Deep neural network for x-ray photoelectron spectroscopy data analysis; 2020]

Ph.D. G Drera



Li(3)-Bi(83) 원소 총 81개 중  
2-5개의 조합



C1O5 contamination  
0-40 Å



Training: 100k x (2000 x 1)  
Test: 534 x (2000 x 1)

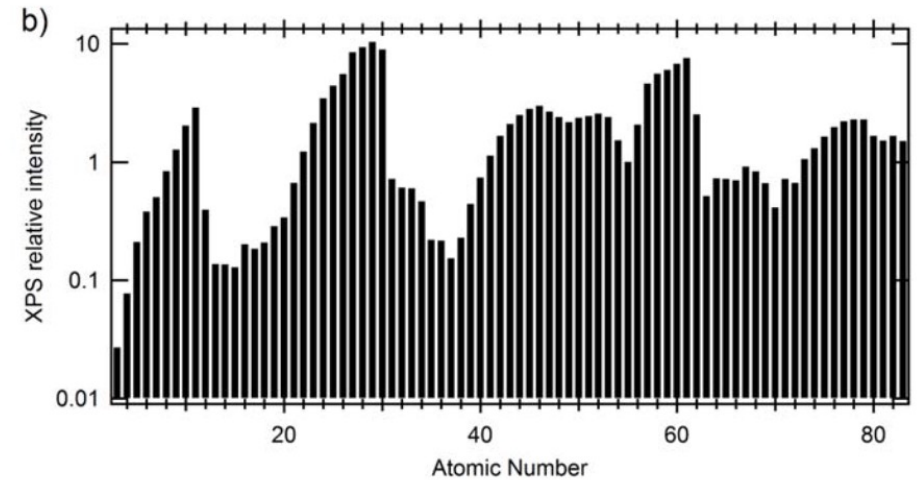
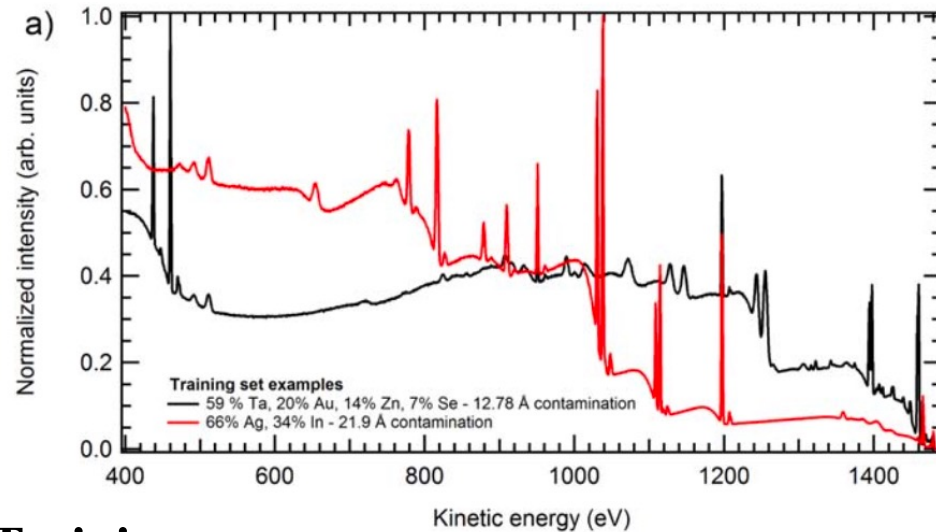
# 01 About Project

Training: 100k x (2000 x 1)

Test: 534 x (2000 x 1)

KE range 400-1486.6eV,

with source Al  $\kappa\alpha$



## - Training

Monte-Carlo simulations, as described by Werner,  
including IMFP and TMFP

- Peak position randomly shifted up to **10eV**
- About carbon rich layer (5:1 carbon to oxygen) **10%** intensity noisy + random shift up to **0.5eV**

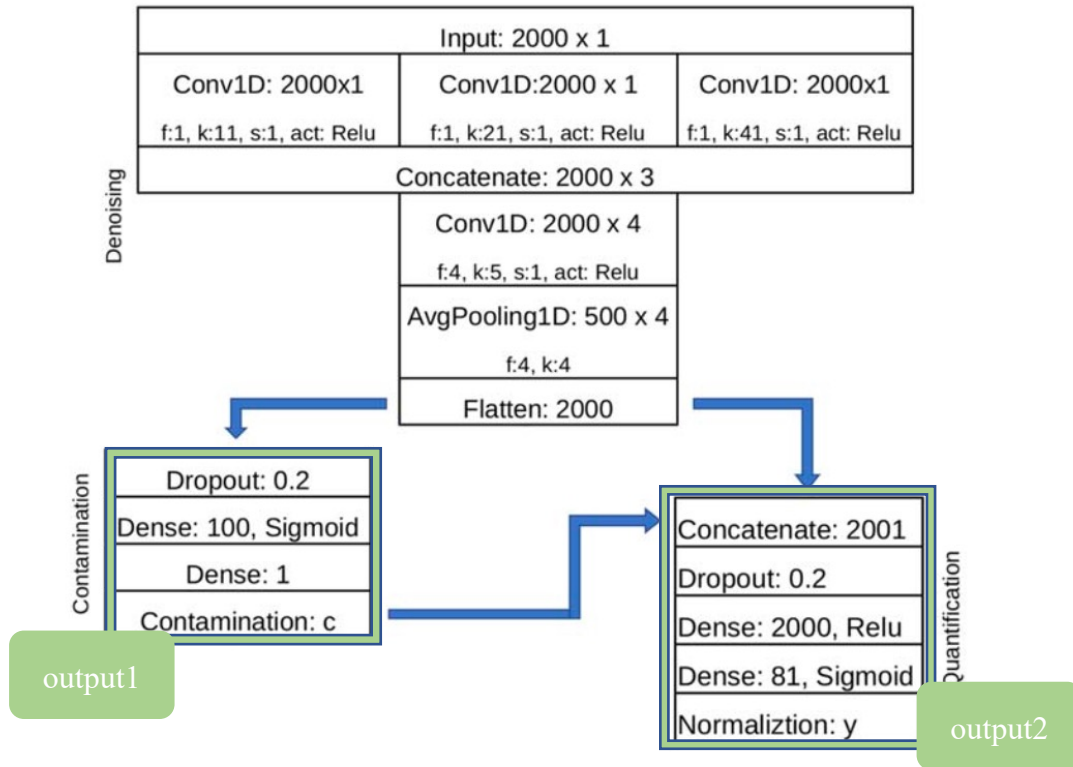
Convolved with a Gaussian peak to reproduce the experimental resolution

Relative elemental quantification ( $T_i$ )

Contribution of the element,  $y_i$  
$$\bar{y}_i = \bar{q}_i T_i \left( \sum_i \bar{q}_i T_i \right)^{-1},$$

Actual quantification,  $q_i$  
$$q_i = y_i / T_i \left( \sum_i y_i \right)^{-1}.$$

# 01 About Project



**Figure 3.** Deep neural network layout for XPS data identification; for each convolution layer,  $f$  is the filter number,  $k$  is the convolution kernel size,  $s$  is the stride length. Total number of trainable parameters is 4366 410.

## - Contamination

$$L2LossFunction = \sum_{i=1}^n (y_{true} - y_{predicted})^2$$

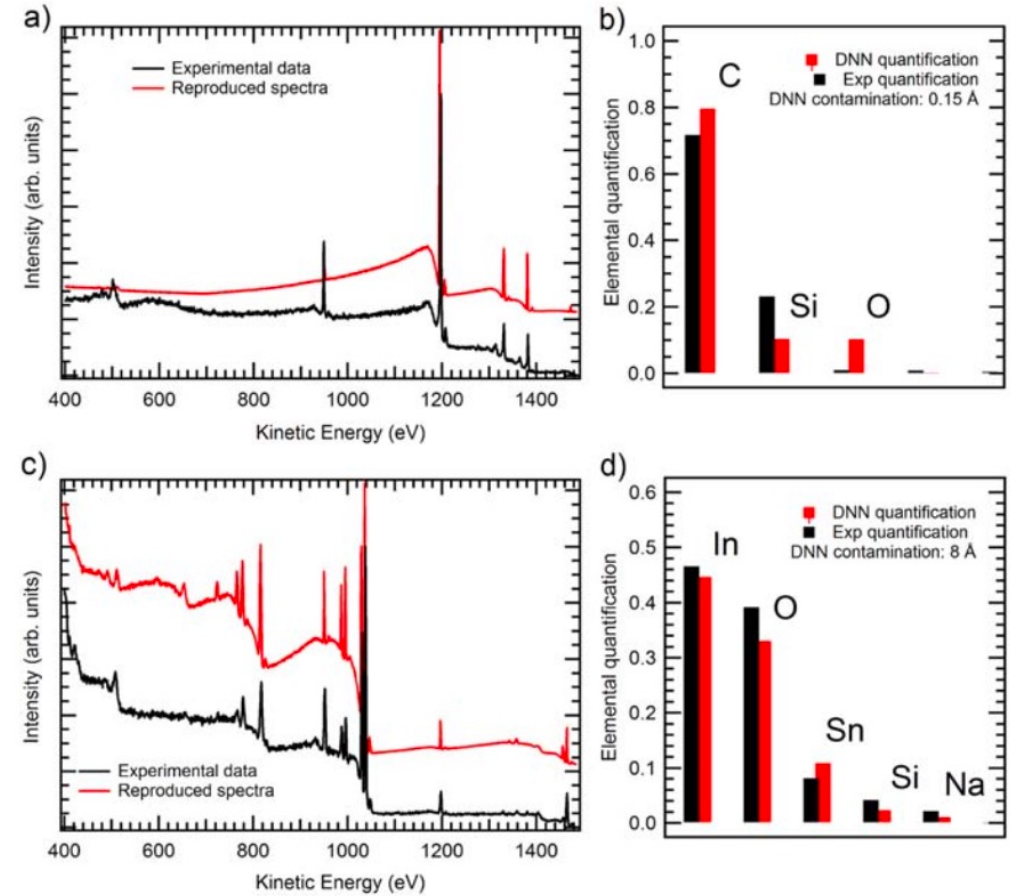
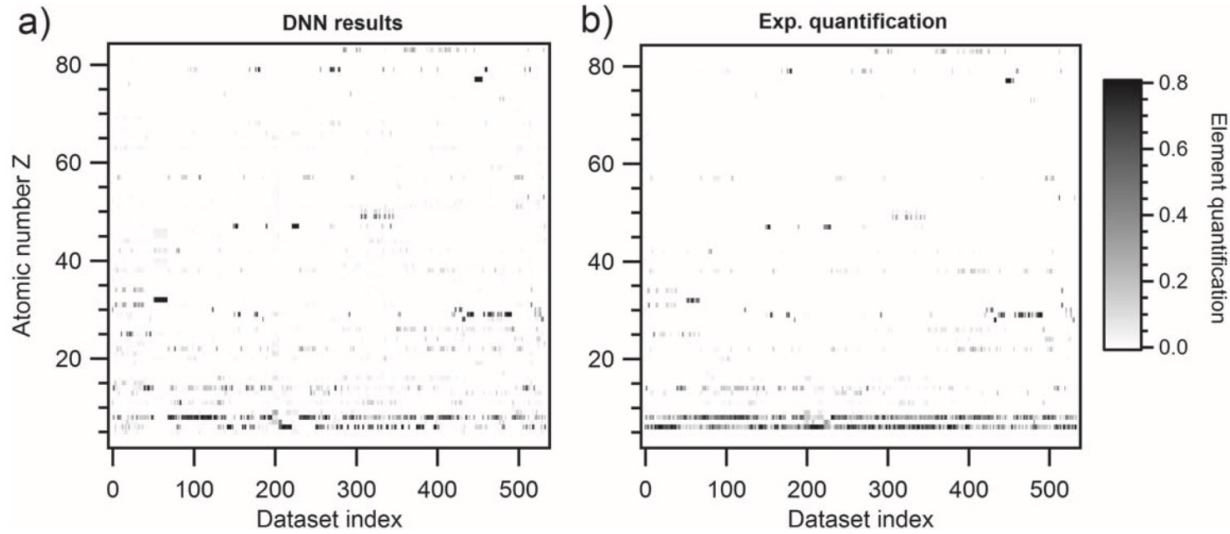
## - Quantification

$$L(y, \bar{y}) = \sum_{i=1}^{81} y_i^2 (y_i - \bar{y}_i)^2,$$

Custom loss function;  
“High-Pass Filter” loss

**ADAM optimizer**

# 01 About Project



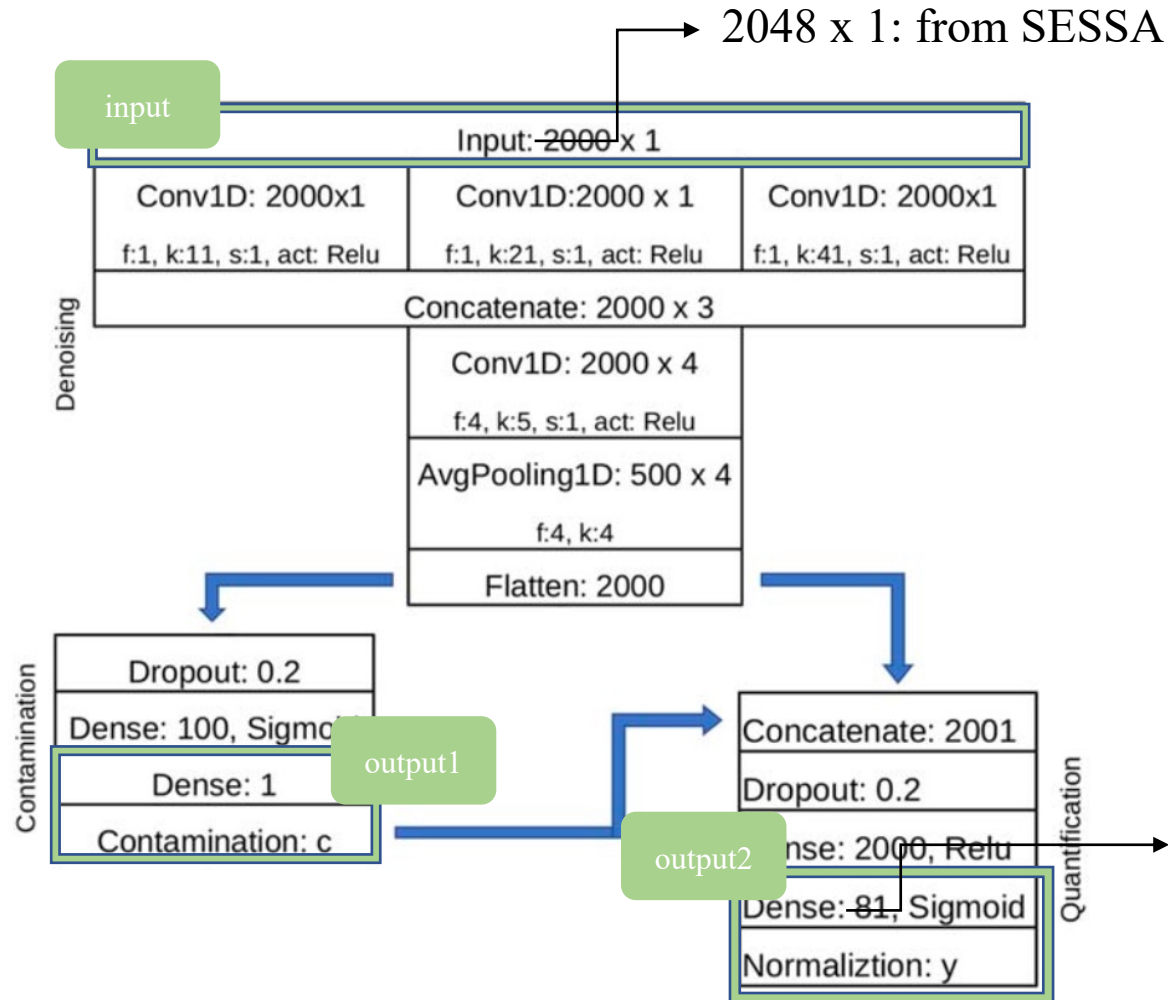
# 02 About Process

Training: 100k x (2000 x 1)  
 Test: 534 x (2000 x 1)



**SESSA**  
 Simulation of Electron Spectra for Surface Analysis

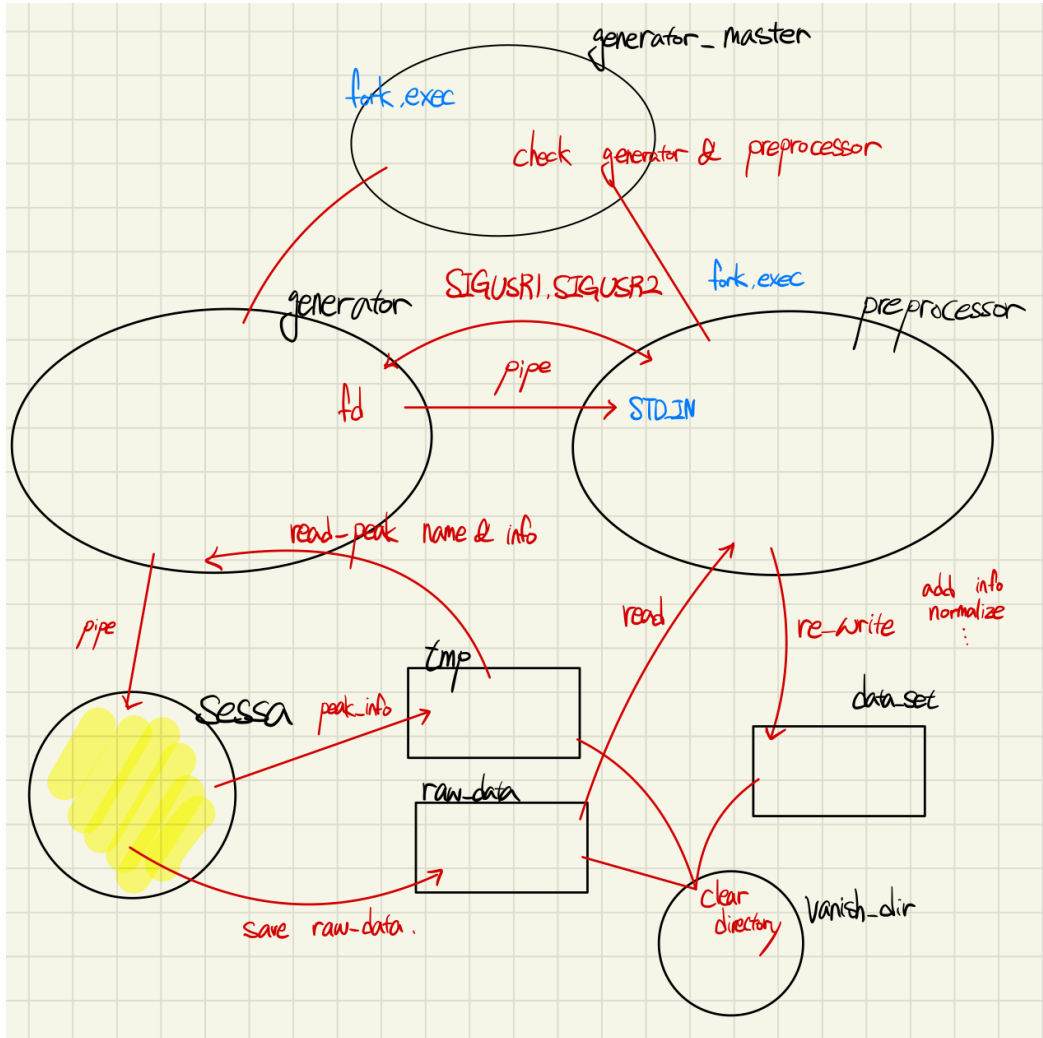
Training: 100k x (2048 x 1)  
 Test: 600 x (2048 x 1)



Li(3)-Bi(83) ; SESSA 제공하지 않는 Pm(61) 제외  
 총 80개

# 02 About Process

## - XPS\_analysis/Generator/



```
~/Desktop/CP/KHU_project/Photoelectron_Spectroscopy/XPS_Analysis/Generator > tree -L 2 .codes
.codes
├── 1d_gaussian.py
├── c_files
│   ├── 1d_gaussian.py
│   ├── generator.c
│   ├── generator_master.c
│   ├── makefile
│   ├── preprocessor.c
│   └── vanish_dir.c
├── generator_master
├── generator_master.c
├── preprocessor
└── preprocessor.c
```

```
~/Desktop/CP/KHU_project/Photoelectron_Spectroscopy/XPS_Analysis/Generator > tree -L 2 relative
relative
├── T.png
├── T.txt
├── T_check.txt
├── relative_intensity
├── relative_intensity.R
└── relative_intensity.c
```

**data\_merge** 3.95 GB  
Modified: October 14, 2022 12:34

Add Tags...

General:

- Kind: Folder
- Size: 3,950,505,420 bytes (4.25 GB on disk) for 74,228 items
- Where: Macintosh HD > Users > csian > Desktop > CP > KHU\_project > Photoelectron\_Spectroscopy > XPS\_Analysis > Generator
- Created: Friday, October 14, 2022 12:34
- Modified: Friday, October 14, 2022 12:34

**data\_600** 27.6 MB  
Modified: Tuesday, July 19, 2022 4:09

Add Tags...

General:

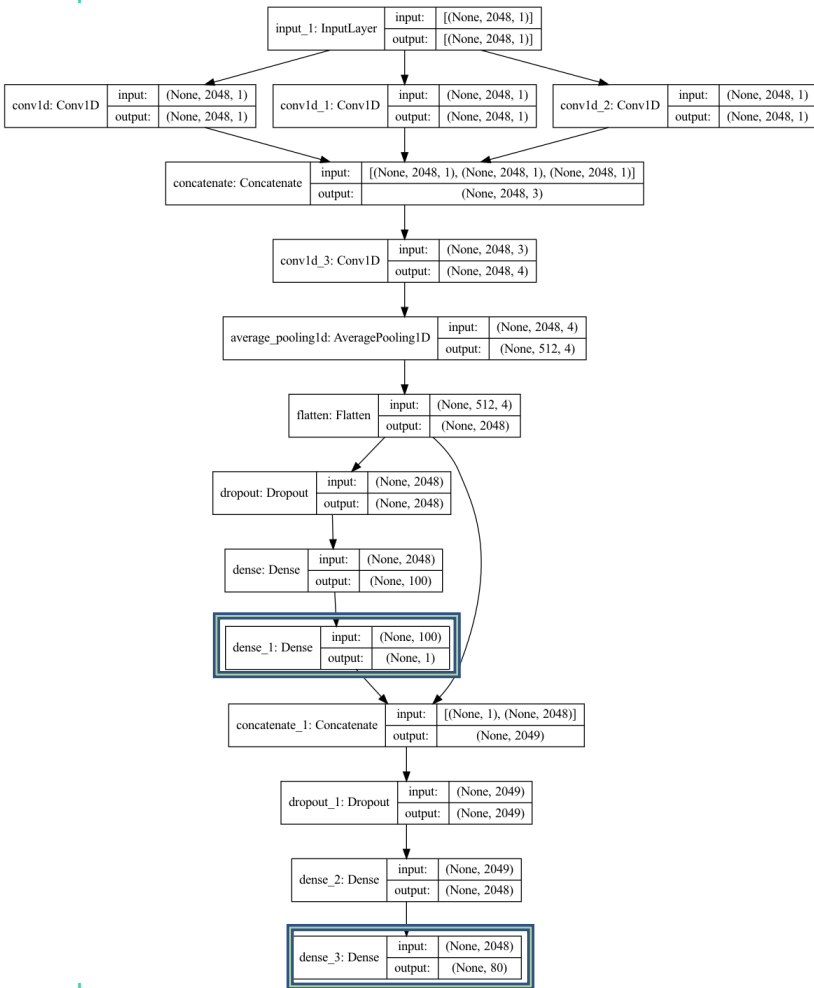
- Kind: Folder
- Size: 27,599,631 bytes (29.5 MB on disk) for 601 items
- Where: Macintosh HD > Users > csian > Desktop > CP > KHU\_project > Photoelectron\_Spectroscopy > XPS\_Analysis > Generator
- Created: Thursday, July 14, 2022 5:51
- Modified: Tuesday, July 19, 2022 4:09

Train data: 74152(74.1k)

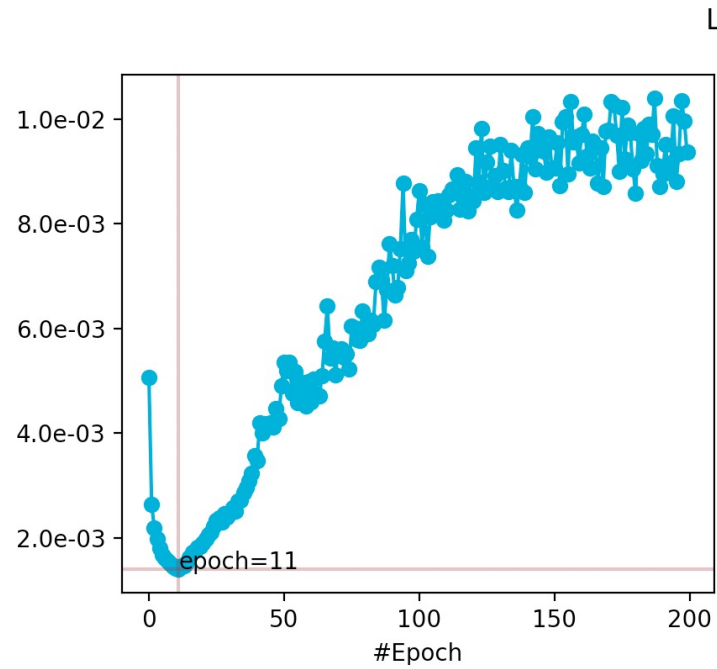
Validation data: 600



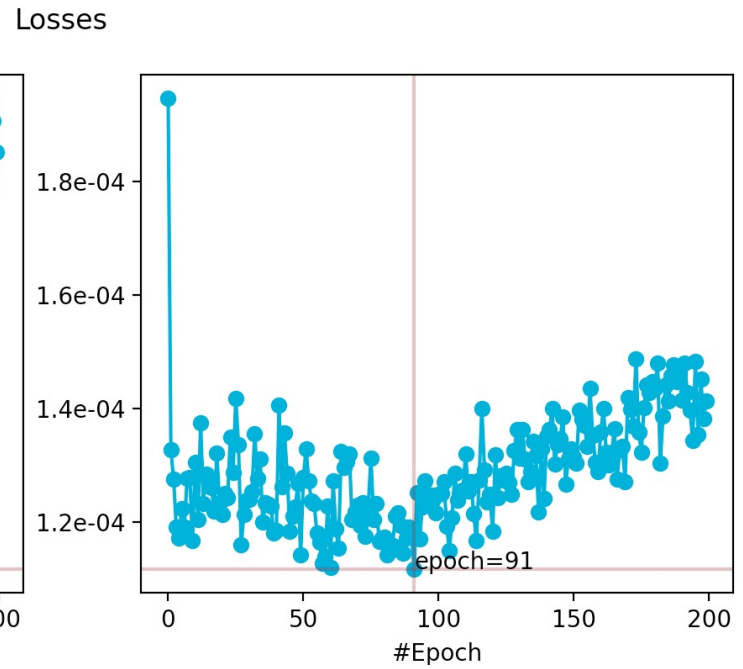
# 02 About Process



## Validation data(600) 대한 Loss



Quantification

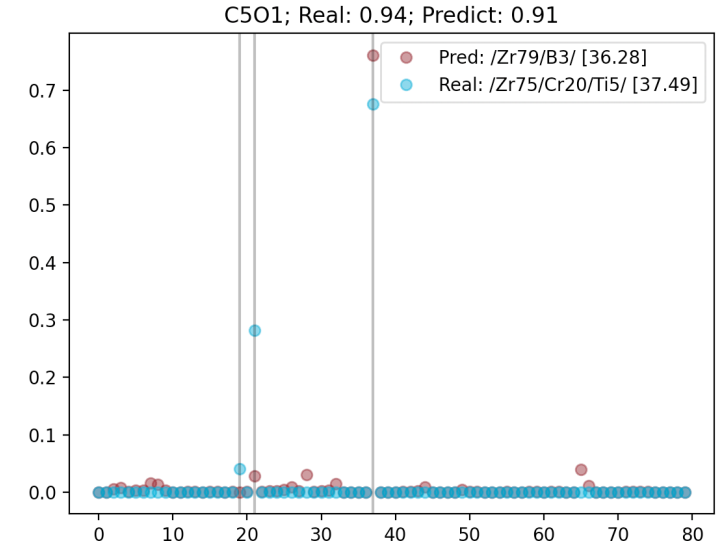
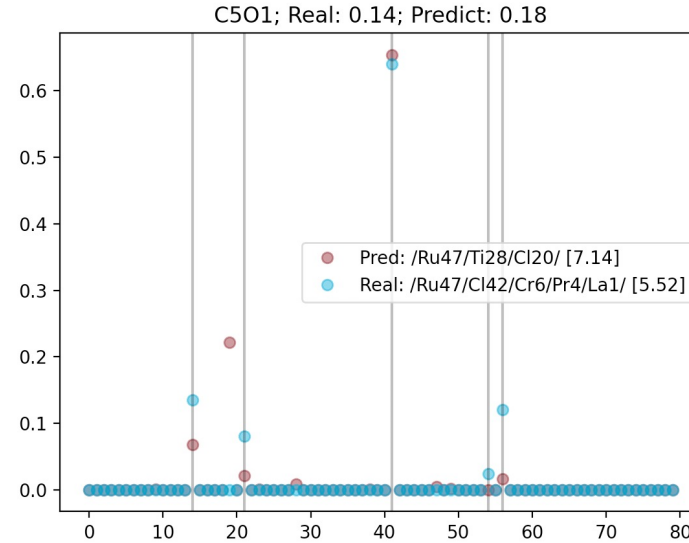
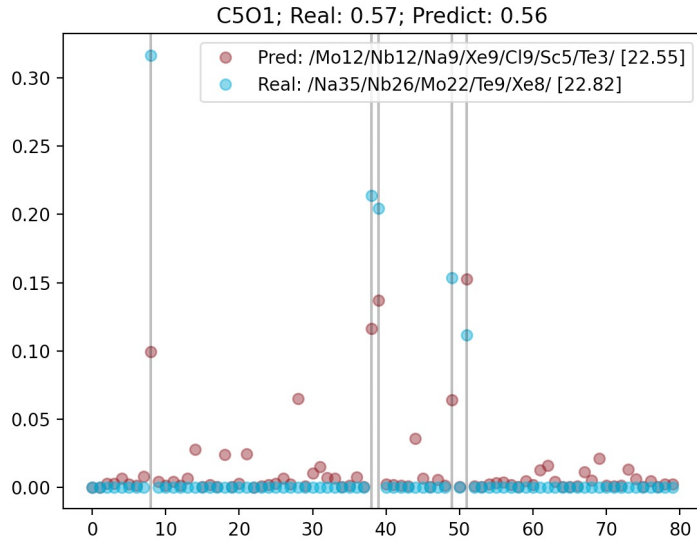


Contamination

Total params: 4,366,415  
 Trainable params: 4,366,415  
 Non-trainable params: 0  
 XPS\_Analysis/DNN/.codes/dnn.py  
 XPS\_Analysis/DNN/.codes/fit.py

# 02 About Process

## XPS\_weights\_11.h5



Pred: /Bi27/Re25/S9/Y8/As6/C4/Ru4/Ga3/ [16.46]

Real: /Bi36/Re26/F26/Co12/ [15.93]

Pred: /Ir45/Ce19/Ru6/Kr4/Ti3/I3/ [6.43]

Real: /Ir31/Ru28/Ce26/I11/Ag4/ [6.34]

Pred: /Sn36/Eu26/Rb18/Sr8/Na4/ [6.49]

Real: /Rb50/Sn17/Eu15/Mo11/Na7/ [6.05]

Pred: /Cd83/Na6/ [32.65]

Real: /B48/Cd42/Li10/ [32.52]

Pred: /Mg80/Ni12/Rh5/ [18.28]

Real: /Mg86/Ni11/B1/Dy1/Hf1/ [17.74]

Pred: /Mg64/Hg33/ [5.03]

Real: /Mg71/Hg29/ [5.33]

Pred: /Y88/K5/Dy3/ [7.76]

Real: /Y93/Cs5/Tc2/ [7.91]

Pred: /Pb83/Gd6/Sm3/ [36.52]

Real: /Pb87/Mo7/Fe3/Pr2/F1/ [38.37]

Pred: /Sr58/Pb14/Gd9/P6/Hg3/ [9.01]

Real: /Sr60/Pb21/Cr11/Hg8/ [9.65]

Pred: /Ci65/B19/C3/ [33.02]

Real: /Ci82/Tm9/Cr7/Ni1/Cd1/ [32.75]

Pred: /V67/Tc24/ [29.19]

Real: /V67/Tc33/ [30.86]

Pred: /S19/Na11/Y10/Rh9/V8/Mg8/Bi7/ [35.15]

Real: /Cl34/Rh26/V24/Bi16/ [34.54]

Pred: /Hg75/Nb10/Eu5/ [29.59]

Real: /Hg84/Ne13/Er1/Pd1/Tb1/ [29.56]

Pred: /Tb72/Y14/Ga8/ [23.17]

Real: /Tb78/S11/Ho10/Ti1/ [22.95]

Pred: /Ci99/ [1.03]

Real: /Li51/Ci47/N2/ [1.14]

Pred: /Rh43/Y23/Eu18/Mn8/Sc4/ [7.00]

Real: /Rh55/Mn24/Eu17/P4/ [6.49]

Pred: /N59/Au13/Ci12/Cd4/ [23.59]

Real: /N78/Au18/Ci4/ [23.05]

Pred: /Ci41/Fe41/O4/Mg3/ [30.02]

Real: /Fe52/Ci48/ [28.77]

Pred: /Gd58/La35/ [8.82]

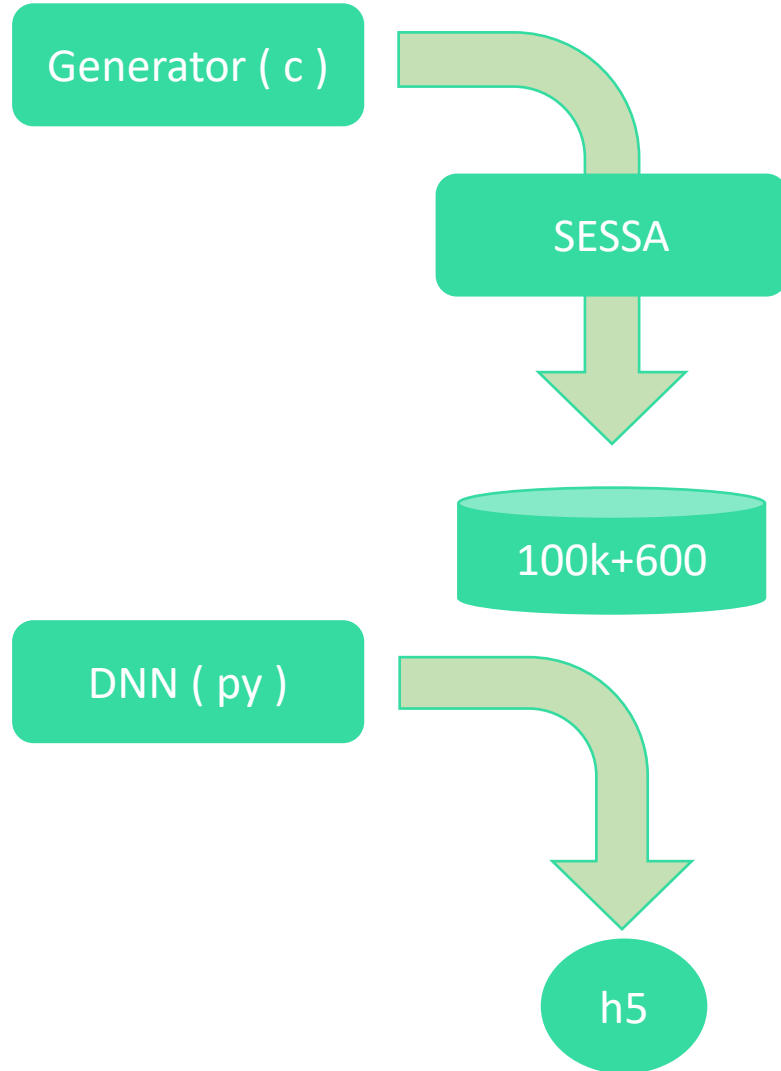
Real: /Gd50/La45/Br5/ [9.39]

Pred: /Os47/Se31/Pb5/Mg3/Ga3/ [32.01]

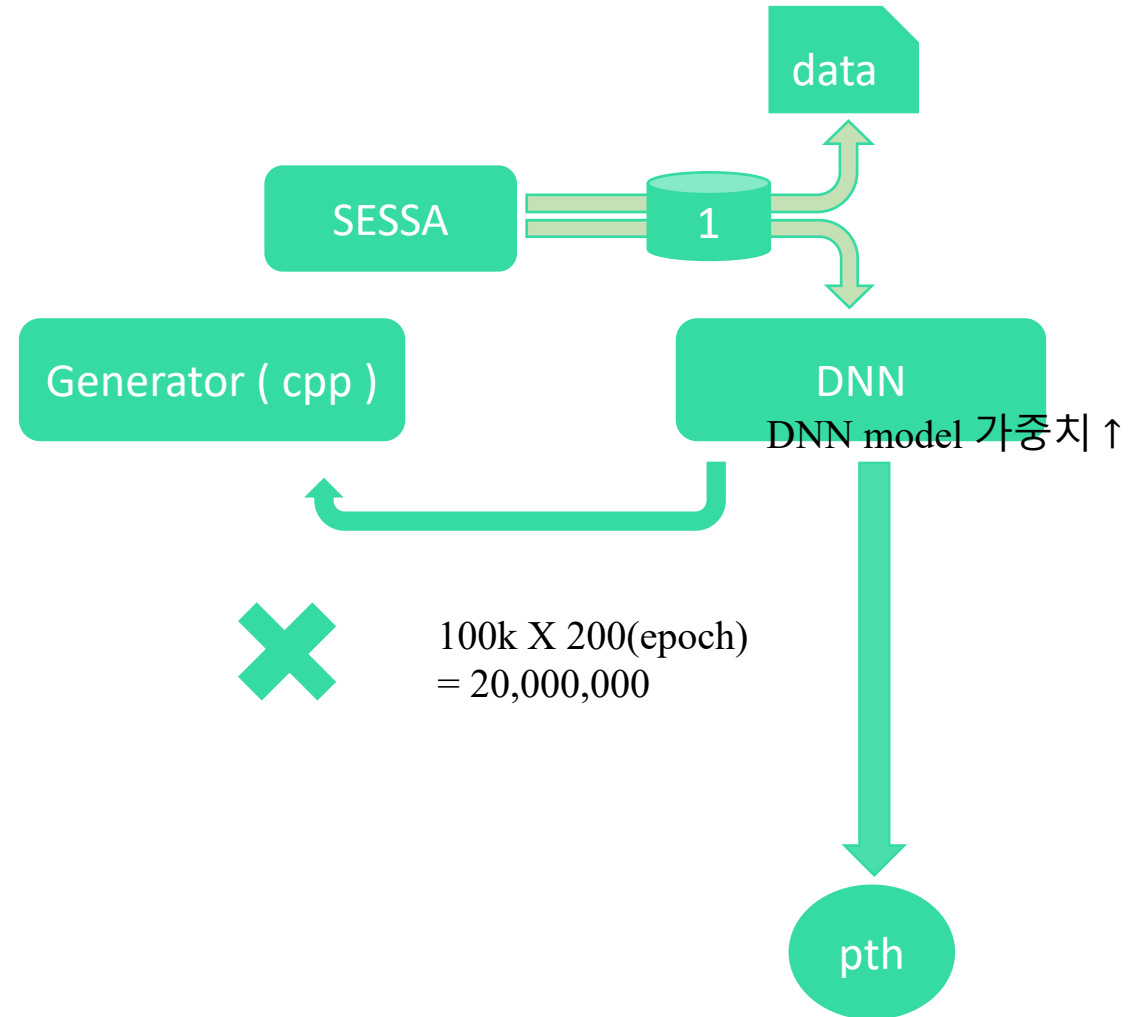
Real: /Os61/Pb15/Na12/V8/Eu4/ [31.89]

# 02 About Process

## Before; XPS\_Analysis



## After; XPSDNN



# 03 About Programs

## install/

```

├── README
├── Sessa-2.2.0-Linux.sh
├── aptInstall.sh
├── code_DNNXPS.tar
├── config
├── makeDir.sh
├── requirements.txt
├── valid_data.tar
└── xpsControl

```

wget <https://csian98.github.io/assets/code/XPSDNN/install.tar>

tar -xvf install.tar

sudo install/aptInstall.sh

## install/aptInstall.sh

\$USER가 root로 지정되기에 수정!

XPSDNN directory 생성 경로

\$HOME/XPSDNN/  
하위 디렉토리 생성

```

1: aptInstall.sh+
1  #!/bin/bash
2  USER="spondlab"
3  HOME="/home/"$USER
4
5  while [ 1 ]; do
6      echo -e "\n\tHello Welcome to installing XPSDNN\nBefore U
7      read -sn1 ask
8      if [ $ask = 'y' ]; then
9          break
10     else if [ $ask = 'n' ]; then
11         echo -e "'aptInstall.sh line number: 3'"
12         sleep 3
13         exit
14     fi
15 done
16
17 if [ $EUID -ne 0 ]; then
18     echo -e "\n\taptInstall require executed with 'sudo'"
19     echo -e "\t\tsudo ./aptInstall.sh\n"
20     exit
21 fi
22
23 # install Path
24 installPath=$HOME"/install/"`dirname $0`
25
26
27 if [ -d $HOME/XPSDNN/ ]; then
28     echo -e "\n\tAlready XPSDNN directory exist\n"
29     sleep 1
30 else
31     mkdir $HOME/XPSDNN
32 fi
33
34 if [ -f $installPath/makeDir.sh ]; then
35     echo -e "\n\tNow build directory structure...\n"
36     cp -p $installPath/makeDir.sh $HOME/XPSDNN/
37     cd $HOME/XPSDNN/
38     ./makeDir.sh $USER
39     rm ./makeDir.sh
40 else
41     echo -e "\n\tThere are no makeDir.sh shell script\n"
42     sleep 3
43     exit
44 fi

```

# 03 About Programs

## install/aptInstall.sh

코드 압축 해제  
\$HOME/XPSDNN/.codes

Validation data 600 압축 해제  
\$HOME/XPSDNN/validation

Sessa-2.2.0-Linux.sh  
Shell script 실행:  
\$HOME/XPSDNN/.SESSA/ 설치

Utility install (apt-get | yum | pkgm | rpm) 수정!

- gcc/g++/make
- python3-pip
- dialog
- **xvfb**

SESSA는 GUI 환경을 필요로 하는 소프트웨어, Xvfb는 GUI 가상화 용도

```

1: aptInstall.sh+
52 if [ -e $installPath/code_DNNXPS.tar ]; then
51   echo -e "\n\t\tNow moving codes...\n"
50   cp -p $installPath/code_DNNXPS.tar $HOME/XPSDNN/
49   cd $HOME/XPSDNN/
48   tar -xvf ./code_DNNXPS.tar 2>&1 1>/dev/null
47   rm ./code_DNNXPS.tar
46 else
45   echo -e "\n\t\tThere are no code_DNNXPS.tar archive file\n"
44   sleep 3
43   exit
42 fi
41
40 if [ -e $installPath/valid_data.tar ]; then
39   echo -e "\n\t\tNow moving validation data...\n"
38   cp -p $installPath/valid_data.tar $HOME/XPSDNN/
37   cd $HOME/XPSDNN/
36   tar -xvf ./valid_data.tar 2>&1 1>/dev/null
35   rm ./valid_data.tar
34 else
33   echo -e "\n\t\tThere are no valid_data.tar archive file\n"
32   sleep3
31   exit
30 fi
29
28 if [ -f $installPath/Sessa-2.2.0-Linux.sh ]; then
27   echo -e "\n\t\tNow Started to load SESSA\n"
26   cp -p $installPath/Sessa-2.2.0-Linux.sh $HOME/XPSDNN/.SESSA/
25   cd $HOME/XPSDNN/.SESSA/
24   sleep 2
23   ./Sessa-2.2.0-Linux.sh
22
21   echo -e "\n\t\tSESSA installed complete"
20   chown -R $USER ./
19   chgrp -R $USER ./
18   cd $HOME/XPSDNN/.SESSA/
17   rm Sessa-2.2.0-Linux.sh
16
15   cd $installPath
14 else
13   echo -e "\n\t\tThere are no Sessa-2.2.0-Linux.sh shell script\n"
12   sleep 3
11   exit
10 fi
9
8 echo -e "\n\t\tNow started to download compile utilities and python3\n"
7 apt-get install gcc g++ make python3-pip dialog xvfb
6 echo -e "\n\t\tDownload Completed\n"

```

# 03 About Programs

\$HOME/XPSDNN/.codes  
컴파일  
- \$HOME/XPSDNN/.codes/main  
- \$HOME/XPSDNN/.codes/cleaner

SESSA 구동시 필요한  
libgfortran.so.3 설치

Python pip3 필요 라이브러리 설치  
- GPU 연동 설치 X  
- Python Library 가상환경 X

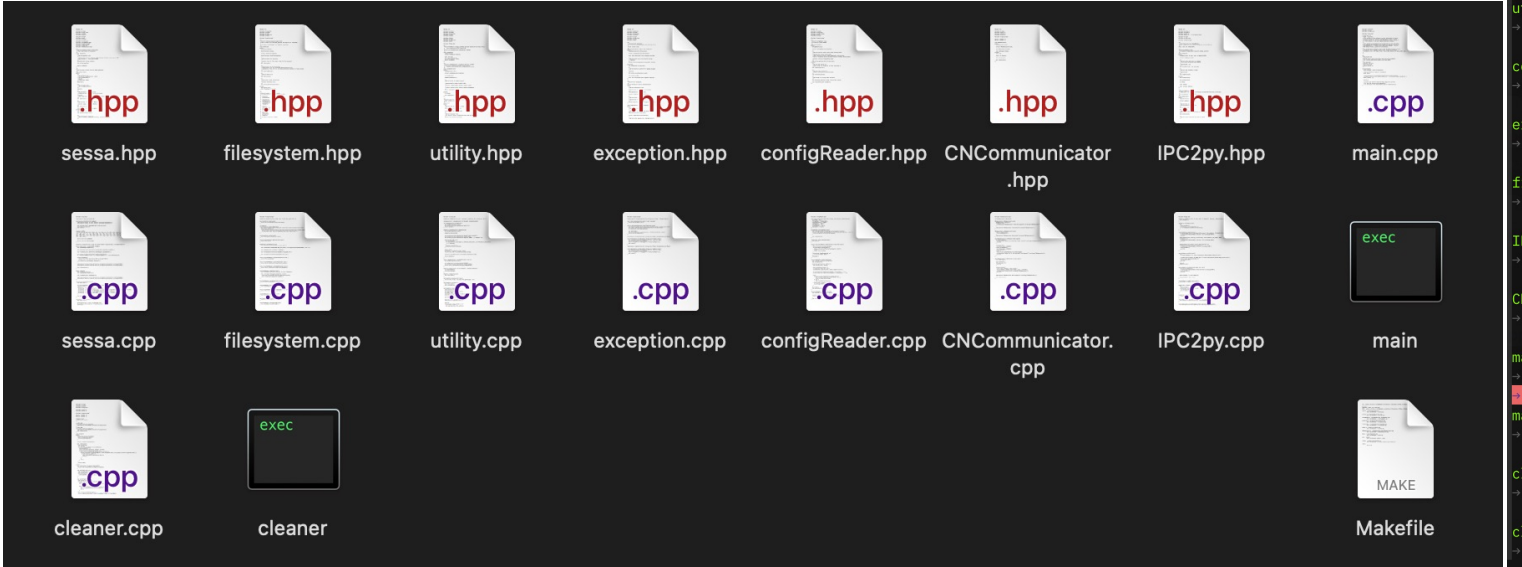
```

1: aptInstall.sh+
52 echo -e "\n\t\tNow Compile XPSDNN codes..."
51 cd $HOME/XPSDNN/.codes
50 echo -e "\n\t\tCompiling...\n"
49 make
48 make clean
47 echo -e "\n\t\tCompile Completed\n"
46
45 echo -e "\n\t\tNow Started to download libgfortran.so.3"
44
43 echo "Downloading gcc-6-base" && \
42 cd /tmp/ && wget http://archive.ubuntu.com/ubuntu/pool/universe/g/gcc
41 echo "Downloading libgfortran3" && \
40 cd /tmp/ && wget http://archive.ubuntu.com/ubuntu/pool/universe/g/gcc
39 echo "Installing gcc-6-base" && \
38 cd /tmp/ && sudo dpkg -i gcc-6-base_6.4.0-17ubuntu1_amd64.deb && \
37 echo "Installing libgfortran3" && \
36 cd /tmp/ && sudo dpkg -i libgfortran3_6.4.0-17ubuntu1_amd64.deb
35
34 echo -e "\n\t\t"
33
32 echo -e "\n\t\tNow collecting above pyLibraries\n"
31
30 REQUIRE=$installPath/requirements.txt
29 cat $REQUIRE
28 echo -e "\n\t\tEND\n"
27 sleep 2
26
25 if [ -f $REQUIRE ]; then
24 pip3 install -r requirements.txt 2>&1 1>/dev/null
23 fi
22
21 echo -e "\n\t\tAll of the Requirements installed and ready\n"
20 sleep 2
19
18 echo -e "\n\t\tCopy other needed shell scripts...\n"
17 cd $installPath
16 cp xpsControl $HOME/XPSDNN/
15 cp config $HOME/XPSDNN/
14 cp README $HOME/XPSDNN/
13
12 echo "All installing finished"
11 sleep 2
10
9 read -n1 -p "Do you want to remove installing files [y/n]? " answer
8 case $answer in
7 T | y)
6 echo
5 echo "removing all files..."
4 rm -rf $installPath;;
3 F | f)
2 echo
1 exit;;
147 esac

```

# 03 About Programs

## XPSDNN/.codes



```

Makefile
all : sessa.o utility.o configReader.o exception.o filesystem.o IPC2py.o CNCommunicator.o main.o main cleaner

CC = g++
CXXFLAGS = -Wall -O2 -std=c++2b
OBJJS = sessa.o utility.o configReader.o exception.o filesystem.o IPC2py.o CNCommunicator.o main.o

sessa.o : sessa.hpp sessa.cpp
-> $(CC) $(CXXFLAGS) -c sessa.cpp

utility.o : utility.hpp utility.cpp
-> $(CC) $(CXXFLAGS) -c utility.cpp

configReader.o : configReader.hpp configReader.cpp
-> $(CC) $(CXXFLAGS) -c configReader.cpp

exception.o : exception.hpp exception.cpp
-> $(CC) $(CXXFLAGS) -c exception.cpp

filesystem.o : filesystem.hpp filesystem.cpp
-> $(CC) $(CXXFLAGS) -c filesystem.cpp

IPC2py.o : IPC2py.hpp IPC2py.cpp
-> $(CC) $(CXXFLAGS) -c IPC2py.cpp

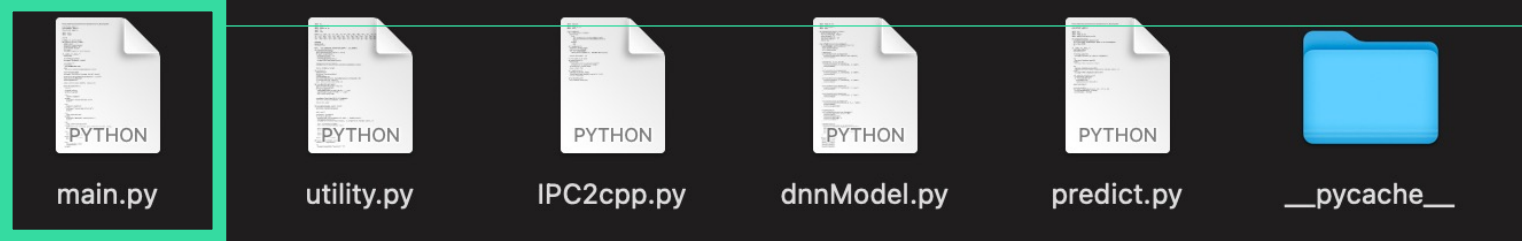
CNCommunicator.o : CNCommunicator.hpp CNCommunicator.cpp
-> $(CC) $(CXXFLAGS) -c CNCommunicator.cpp

main.o : sessa.hpp main.cpp
-> $(CC) $(CXXFLAGS) -c main.cpp

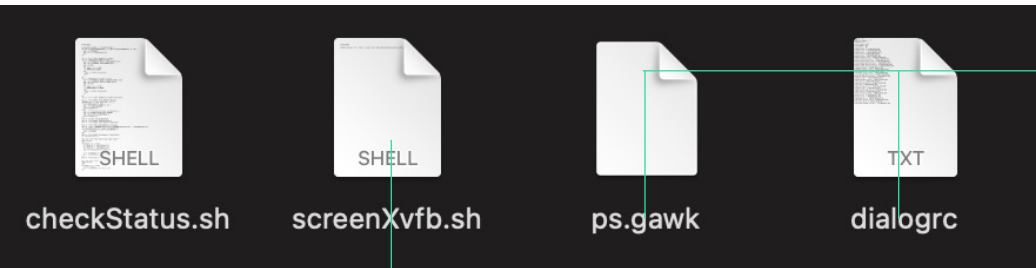
main : $(OBJJS)
-> $(CC) $(CXXFLAGS) $(OBJJS) -o main

cleaner : cleaner.cpp exception.o
-> $(CC) $(CXXFLAGS) -o cleaner cleaner.cpp exception.o

clean :
-> rm -f *.o
  
```



Python 사용하는 #! `which python3` 으로 변경!



dialog 운용 용도

Jupyter Server에서 운영시 목적으로 만들 Xvfb 화면 캡처 저장(사용시, 저장 위치 변경 필요)

# 03 About Programs

## XPSDNN/.codes/checkStatus.sh

서브셸로 수행하면 Xvfb 동작 X

. XPSDNN/.codes/checkStatus.sh

dialog로 프로그램 실행 시, 자동으로 checkStatus를 수행함  
수동으로 프로그램 실행 시, 동작 환경 설정

SESSA 수행 시, 생성되는 \$HOME/ sessa-\$USER 삭제  
SESSA 실행 중, \$HOME/ sessa-\$USER 생성되며, 정상 종료 시 삭제됨  
삭제되지 않은 경우 SESSA 실행 시 GUI 환경으로 확인 알림 발생

```

1: checkStatus.sh
44 #!/bin/bash
43
42 possibleName=$(SHELL "-`basename $SHELL`)
41 if [ $0 == ${possibleName[0]} ] || [ $0 == ${possibleName[1]} ]; then
40     echo -e "\t[USAGE]"
39     echo -e "\t. ./checkStatus.sh"
38     exit
37 fi
36
35 echo -e "\t+1) CHECK '$HOME/ sessa-$USER'"
34 if [ -e "$HOME/ sessa-$USER" ]; then
33     echo -e "\t\tNeed to delete ' sessa-$USER' file"
32     echo -en "\t\tDELETE sessa-$USER [y/n]: "
31     read -sn1 ask
30     echo
29     if [ $ask = 'y' ]; then
28         rm $HOME/ sessa-$USER
27     else
26         echo -e "\tEXIT checkStatus"
25         exit
24     fi
23 fi
22 if [ -e "$HOME/ SESSA v2-$USER" ]; then
21     echo -e "\t\tNeed to delete 'SESSA v2-$USER' file"
20     echo -en "\t\tDELETE SESSA v2-$USER [y/n]: "
19     read -sn1 ask
18     echo
17     if [ $ask = 'y' ]; then
16         rm $HOME/ SESSA\ v2-$USER
15     else
14         echo -e "\tEXIT checkStatus"
13         exit
12     fi
11 fi
10 echo -e "\t-1) CHECK '$HOME/ sessa-$USER' Completed\n"
  
```



# 03 About Programs

## XPSDNN/.codes/checkStatus.sh

Xvfb 실행 확인 후, export DISPLAY 지정

```

51 echo -e "\t+2) CHECK 'Xvfb DISPLAY' Setting"
50 lineNum=`ps -ef | grep 'Xvfb :99' | wc -l`
49 if [ $lineNum -eq 2 ]; then
48   echo -e "\t\tDetect Xvfb with 99..."
47   echo -e "\t\tset DISPLAY"
46   export DISPLAY=:99
45 else
44   echo -e "\t\tXvfb with 99 was not detected..."
43   echo -e "\t\tBuild Xvfb & set DISPLAY"
42   Xvfb :99 -screen 0 1024x768x24 &
41   export DISPLAY=:99
40 fi
39 echo -e "\t\tYou should declare\n"
38 echo -e "\t\t\texport DISPLAY=:99\n\n"
37 echo -e "\t-2) CHECK 'Xvfb DISPLAY' Complted\n"

```

cpp - py IPC에 사용되는 shared Memory 제거  
프로세스 정상 종료시에는 제거되지만,  
정상 종료되지 않은 경우 남아 있음

```

35 echo -e "\t+3) CHECK 'SharedMemory' Allocated"
34 ipcs -m | gawk '/0x00000cca/{print $2} /0x00000ccb/{print $2}' > .tmp/dump2/ipcs.txt
33 cat .tmp/dump2/ipcs.txt | while read ipcNum; do
32   echo -e "\t\tremove $ipcNum shared Memory"
31   ipcrm -m $ipcNum
30 done
29 echo -e "\t-3) CHECK 'SharedMemory' Completed\n"
28 rm .tmp/dump2/ipcs.txt

```

shrKey1 : 3274  
shrKey2 : 3275

# 03 About Programs

## XPSDNN/.codes/checkStatus.sh

\$HOME/XPSDNN/.tmp/log

\$HOME/XPSDNN/.tmp/dump  
\$HOME/XPSDNN/.tmp/dump2

\$HOME/XPSDNN/data/  
\$HOME/XPSDNN/.tmp/losses/

제거 여부 확인

```

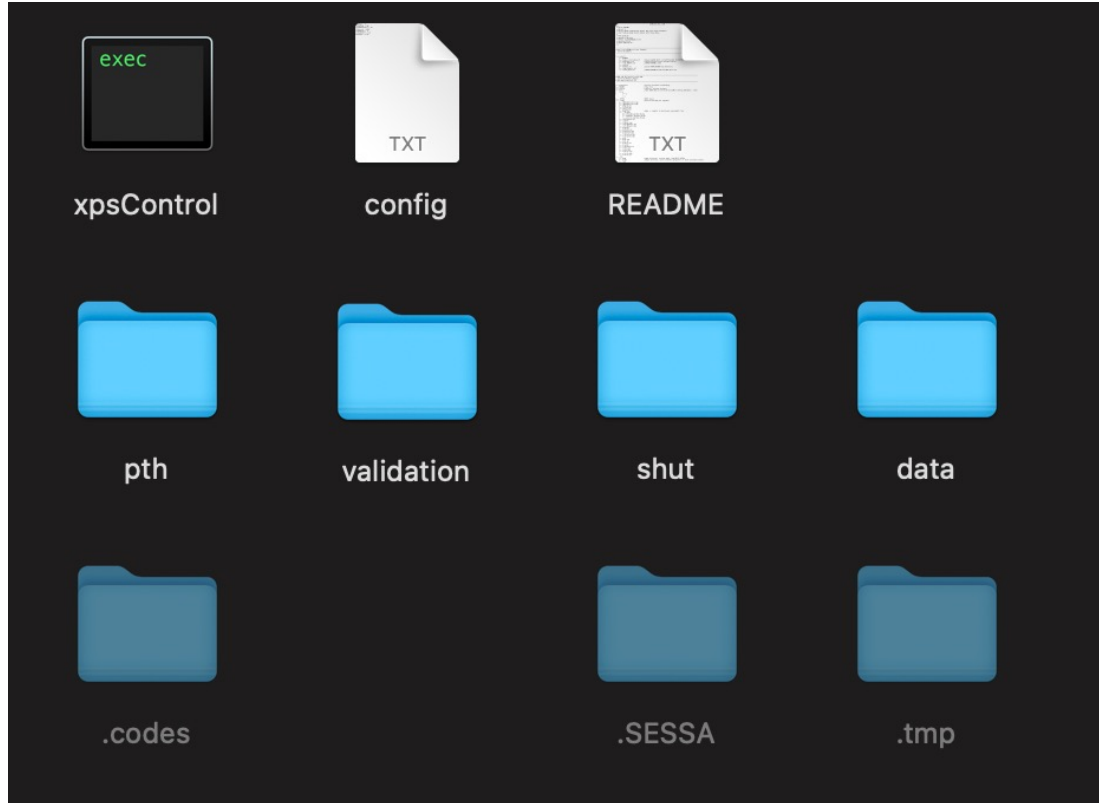
32 echo -en "\t+4) Clear the ex log files? [y/n]: "
31 read -sn1 ask
30 echo
29 if [ $ask = 'y' ]; then
28   cat /dev/null > .tmp/log/log.txt
27   cat /dev/null > .tmp/log/err.txt
26   cat /dev/null > .tmp/log/pyLog.txt
25   echo -e "\t\tclear all log files"
24   ..
23   rm -f .tmp/dump/* .tmp/dump2/*
22   echo -e "\t\tclear all dump files"
21 fi
20 echo -e "\t-4) Clear the ex log files Completed\n"
19
18 echo -en "\t+5) Clear the ex Data files? [y/n]: "
17 read -sn1 ask
16 echo
15 if [ $ask = 'y' ]; then
14   rm -rf ./data/*
13   echo -e "\t\tremove all data files"
12 fi
11 echo -e "\t-5) Clear the ex Data files Completed\n"
10
9   echo -en "\t+6) Clear the ex losses files? [y/n]: "
8   read -sn1 ask
7   echo
6   if [ $ask = 'y' ]; then
5     rm -f ./losses/*
4     echo -e "\t\tremove losses file from PY"
3   fi
2   echo -e "\t-6) Clear the ex losses files Completed\n"

```

# 03 About Programs

XPSDNN/

other directories & files



- pth

.codes/utility.py::validSpan(default1000) 주기로  
pth 생성(weight)

- data

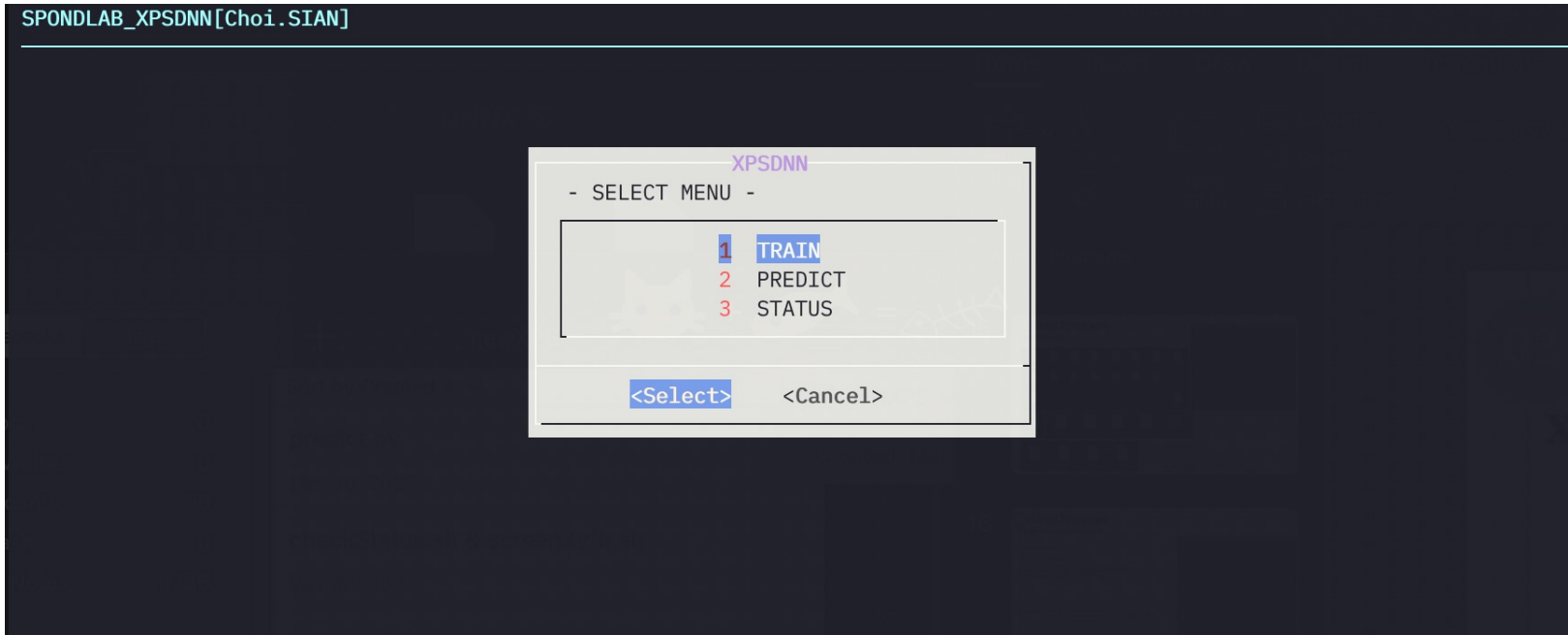
config::DataSave : true인 경우,  
학습한 데이터 저장

- shut

config::TargetNum 도달 시,  
pth 저장

# 03 About Programs

## XPSDNN/xpsControl

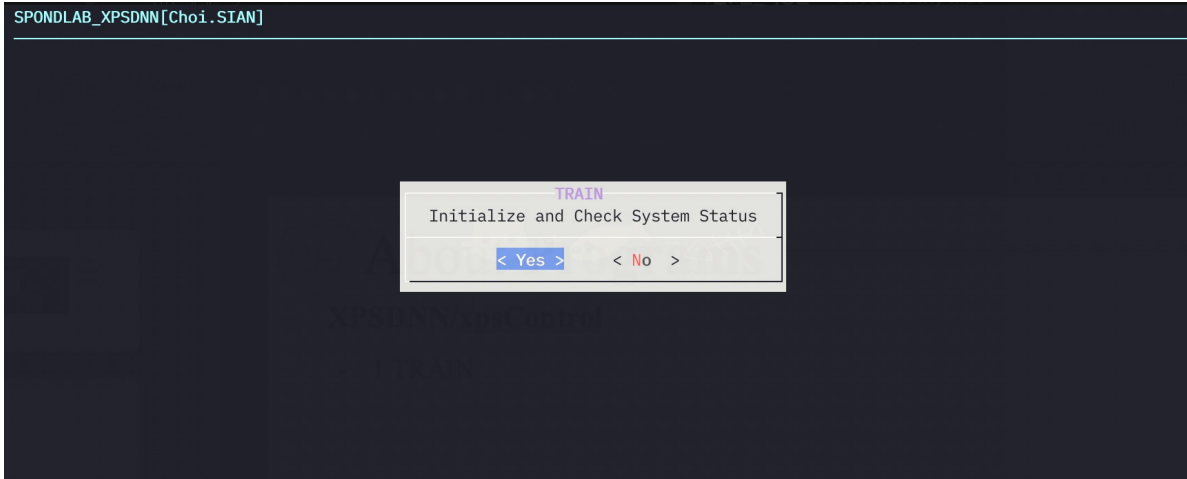


- **1 TRAIN**
- **2 PREDICT**
- **3 STATUS**

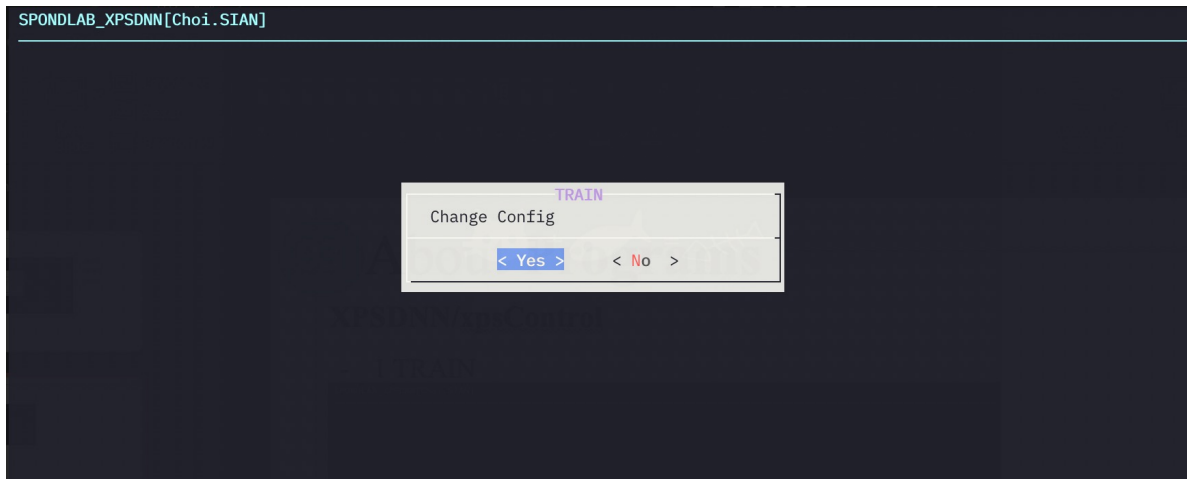
# 03 About Programs

## XPSDNN/xpsControl

### 1 TRAIN

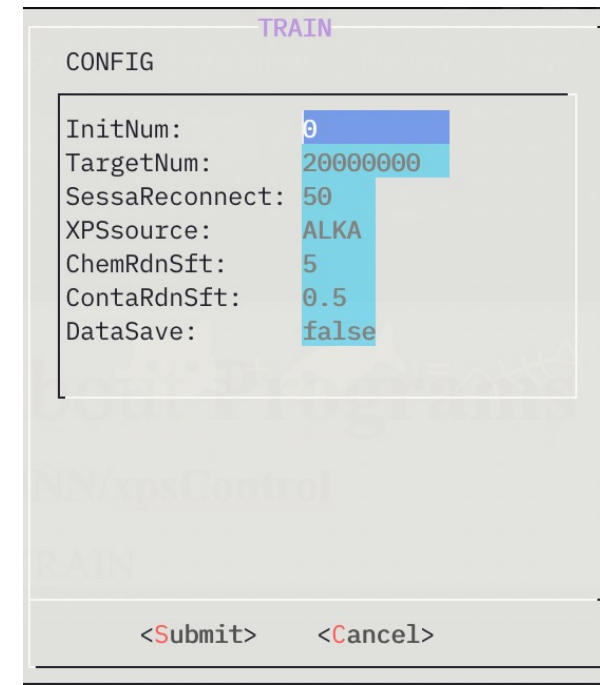


checkStatus.sh 수행



XPSDNN/config 수정 여부

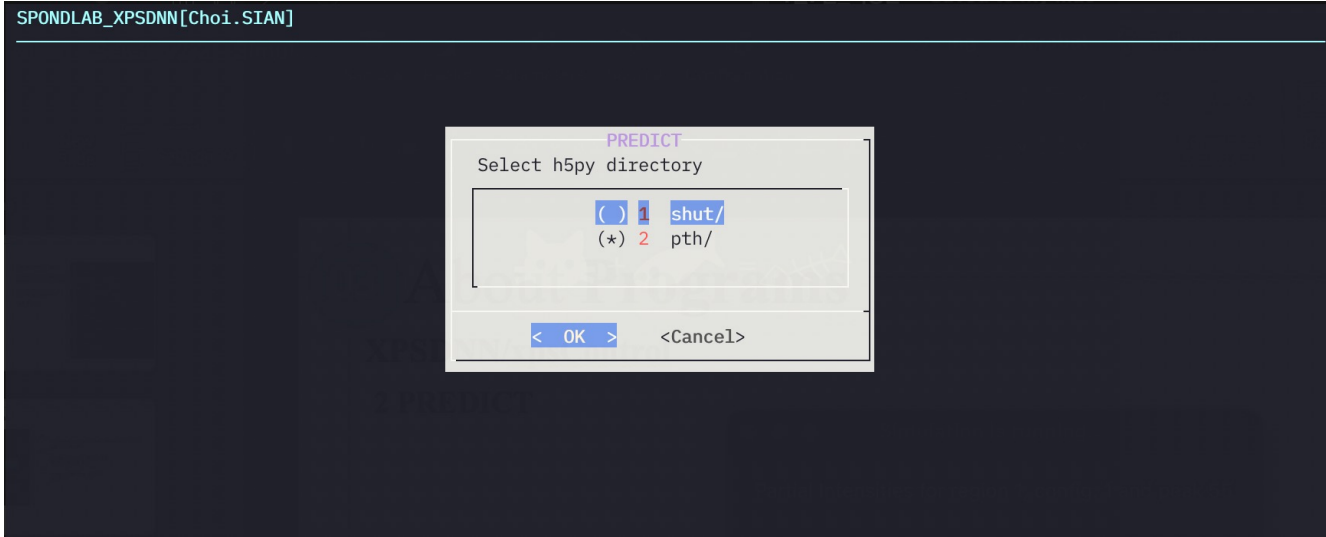
수정 하지 않을 시,  
XPSDNN/config 수행



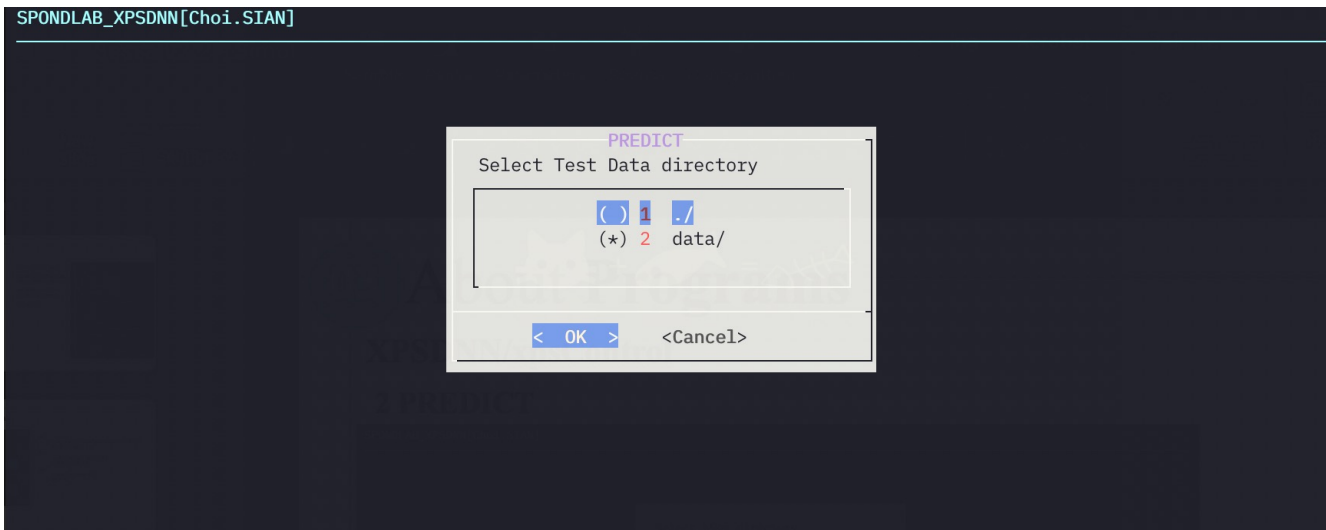
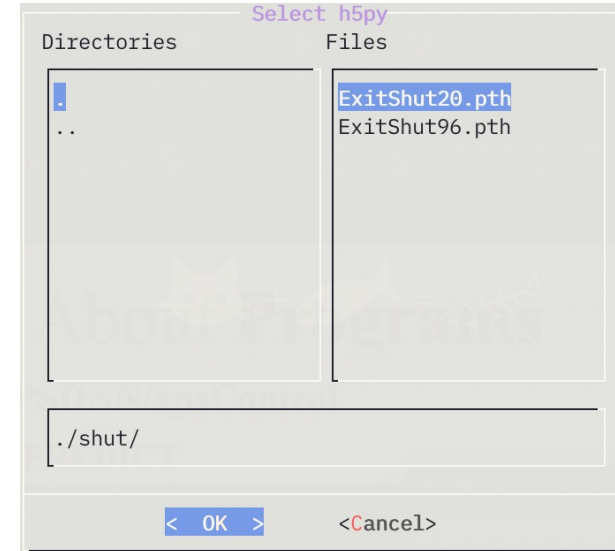
# 03 About Programs

## XPSDNN/xpsControl

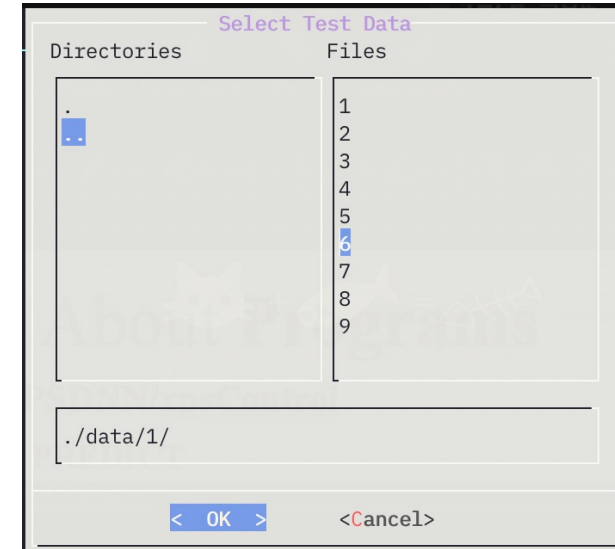
### 2 PREDICT



pth(weight) 선택



Data 선택



# 03 About Programs

## XPSDNN/xpsControl

### 2 PREDICT

```

PREDICT
/Xe0.393805/Ni0.225664/Yb0.212389/Co.168142/[31.560000]

(tensor([0.5125], grad_fn=<SigmoidBackward0>), tensor([0.0135, 0.0140, 0.013
0.0139, 0.0130, 0.0134, 0.0133, 0.0097, 0.0125, 0.0136, 0.0138, 0.01
0.0129, 0.0121, 0.0120, 0.0133, 0.0132, 0.0133, 0.0135, 0.0110, 0.01
0.0126, 0.0137, 0.0108, 0.0108, 0.0132, 0.0134, 0.0137, 0.0089, 0.01
0.0100, 0.0138, 0.0136, 0.0133, 0.0138, 0.0134, 0.0133, 0.0128, 0.01
0.0130, 0.0112, 0.0115, 0.0134, 0.0135, 0.0099, 0.0134, 0.0137, 0.01
0.0130, 0.0137, 0.0112, 0.0132, 0.0132, 0.0129, 0.0136, 0.0055, 0.00
0.0133, 0.0132, 0.0131, 0.0136, 0.0136, 0.0128, 0.0126, 0.0114, 0.01
0.0110, 0.0124, 0.0131, 0.0133, 0.0131, 0.0135, 0.0137, 0.0116],
↓(+)
96%
< EXIT >

```

#### - 주의 사항 -

XPSDNN에서 config::DataSave=true 로 생성된 데이터 XPSDNN/data 와  
Validation 데이터 XPSDNN/validation/ 는 형식이 다름

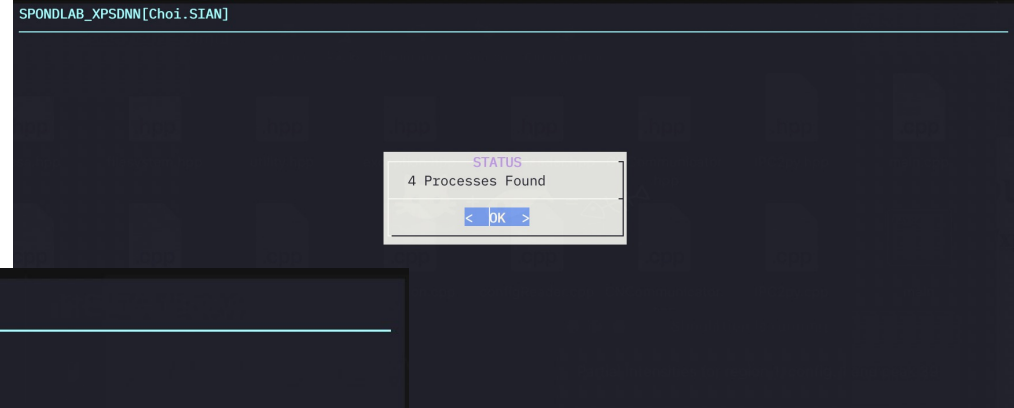
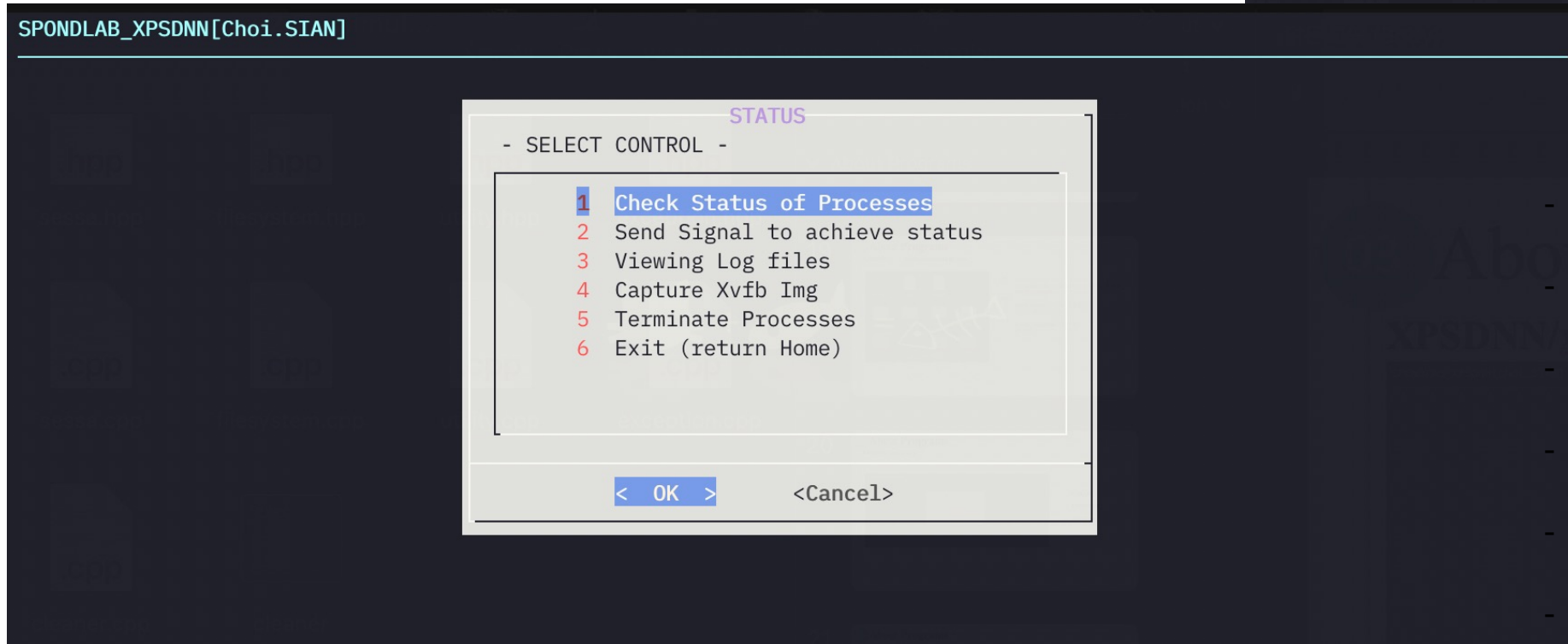
PREDICT에서 내부적으로 실행되는 XPSDNN/.codes/predict.py는 XPSDNN/data 형식을 입력으로 받음  
(아직 예외 처리 적용 되지 않음)

XPSDNN/.codes/main.py에서 validation 과정에서는 XPSDNN/validation 형식을 입력으로 사용함

# 03 About Programs

## XPSDNN/xpsControl

### 3 STATUS



- 1 Check Status of Process

- 2 Send Signal to achieve status

- 3 Viewing Log files

- 4 Capture Xvfb Img

- 5 Terminate Process

- 6 Exit (return Home)



# 03 About Programs

## XPSDNN/xpsControl

### 3 STATUS

```

PROCESS
501 5809 0 .codes/main
501 5811 0 /opt/homebrew/Caskroom/miniforge/base/envs/tf_38/bin/python
501 5812 0 .codes/cleaner
501 5813 0 .SESSA/SESSA.MAC_v2.2/sessa

```

< EXIT >

SPONDLAB\_XPSDNN[Choi.SIAN]

```

PROGRESS
CPP main is up to 3
PY main is up to 2
Target is 10

```

30%

### 1 Check Status of Process

.codes/ps.gawk로 ps -ef 정보 추출(시스템 마다 다를 수 있음)  
리눅스는 5개 표시

- .codes/main [spin lock]
- /usr/bin/python3 .codes/main.py [spin lock]
- .codes/cleaner [mutex lock]
- .SESSA/ ~~~~/sessa

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
75733	spondlab	20	0	79876	4196	3932	R	194.7	0.0	0:55.74	main
75734	spondlab	20	0	3997300	524004	178192	R	112.0	1.6	0:32.04	main.py
75737	spondlab	20	0	813216	193584	84472	R	99.3	0.6	0:28.53	sessa
73268	spondlab	20	0	201580	69992	51188	S	0.7	0.2	0:01.71	Xvfb
14	root	20	0	0	0	0	I	0.3	0.0	8:36.16	rcu_sched
75000	root	20	0	0	0	0	I	0.3	0.0	0:00.68	kworker/u64:0-events_power_efficient
1	root	20	0	168284	13868	8516	S	0.0	0.0	0:16.52	systemd

### 2 Send Signal to achieve status

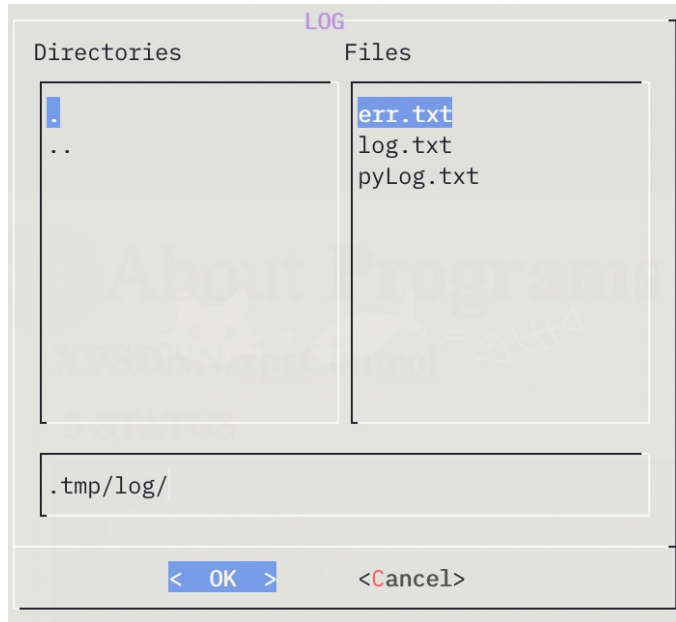
SIGUSR1 to main & main.py

q 입력 시, 상태 종료

# 03 About Programs

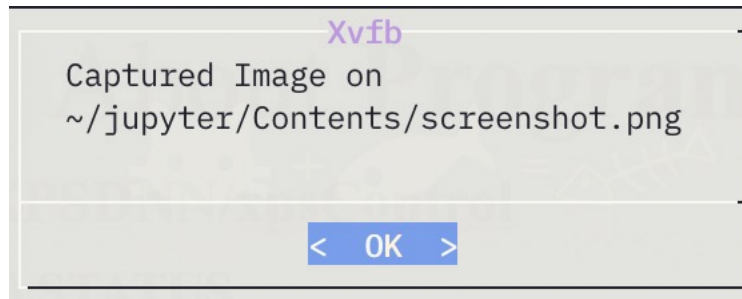
## XPSDNN/xpsControl

### 3 STATUS



### 3 Viewing Log files

- log.txt : cpp main 기록
- err.txt : cpp main 기록
- pyLog.txt : py main.py 기록



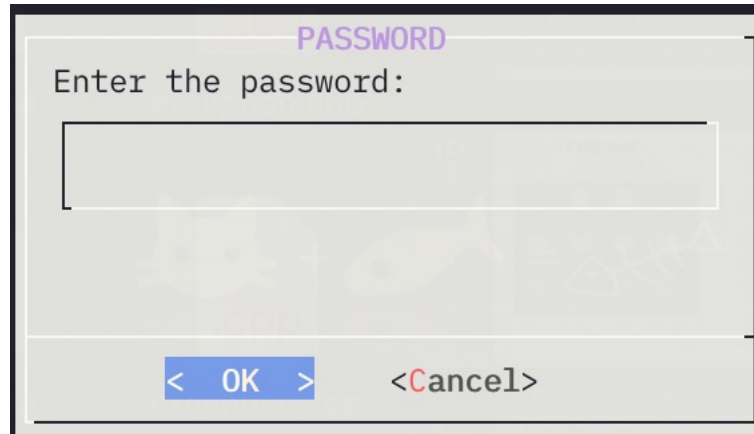
### 4 Capture Xvfb Img

XPSDNN/.codes/screenXvfb.sh 실행

# 03 About Programs

## XPSDNN/xpsControl

### 3 STATUS



### 5 Terminate Process

`kill -9 <all processes>`

PID는 실행 중인 프로세스가 기록하는 것이 아닌,  
`ps -ef | gawk` 를 통해 추출한 PID를 사용함(주의!)

### 그 외 주의 사항

1. SESSA에 명령 입력 과정에서 오류는 일반적으로 쉽게 복구(60sec 대기 후, 회생 시도) 되지만,  
 저장 경로 전달과정에서 입력 오류 발생 시 저장 경로 중간에 파일이 저장됨

cleaner는 초기 상태를 저장하고 회생 절차 때 전달 받는 SIGUSR1을 통해서 변경된 파일을 삭제함

2. MacOS에서 압축 및 코드 작성하였기 때문에 .DS\_Store와 같은 Spotlight 파일이 포함되어 있을 수 있음