**학번: 20205107**

**학과: 콘텐츠IT**

**이름: 강지현**

**파이썬과학프로그래밍기초 기말프로젝트 보고서**

**소스설명**

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

Tkinter, numpy 사용을 위해 임포트한다. 그래프 사용을 위해 matplotlib.pyplot 모듈을 임포트한다. Data1 버튼이 눌릴 시 실행하는 함수인 data1을 정의한다. "data1.txt" 파일을 열고 데이터를 추출하여 각각의 변수에 저장한다. 우선, file1 변수를 사용하여 "data1.txt" 파일을 읽기 모드로 열었다. Data1 버튼을 누르면 각각의 데이터를 저장하는 빈리스트(gan\_loss1, val\_mse1, psnr1, mse1, epoch1)를 생성하고file1.readlines()를 사용하여 파일의 모든 라인을 읽고, 각 라인을 요소로 갖는 리스트를 반환한다. for 루프를 사용하여 각 라인에 대해 데이터를 추출한다. split() 메소드를 사용하여 각 라인을 공백으로 분리하여 리스트로 만든 후 lstrip() 메소드를 사용하여 epoch 값에서 "epoch" 문자를 제거하고, int() 함수를 사용하여 정수로 변환하여 epoch1 리스트에 추가한다. rstrip() 메소드를 사용하여 gan\_loss1, val\_mse1, psnr1, mse1 값에서 ";" 문자를 제거하고, float() 함수를 사용하여 실수로 변환하여 각각의 리스트에 추가한다. 마지막으로 파일을 닫기 위해 file1.colse()를 호출했다.

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명이 코드는 Data2 버튼이 눌렸을 때, 실행하는 함수인 data2를 정의한다. "data2.txt" 파일에서 데이터를 추출하여 리스트에 저장하는 기능을 구현한 것으로 "data2.txt" 파일을 열고 데이터를 추출하여 각각의 변수에 저장한다. 우선, file2 변수를 사용하여 "data2.txt" 파일을 읽기 모드로 열었다. 각각의 데이터를 저장할 빈 리스트(gan\_loss2, val\_mse2, psnr2, mse2, epoch2)를 생성한후file2.readlines()를 사용하여 파일의 모든 라인을 읽고, 각 라인을 요소로 갖는 리스트를 반환한다. for 루프를 사용하여 각 라인에 대해 데이터를 추출한다. split() 메소드를 사용하여 각 라인을 공백으로 분리하여 리스트로 만든 후 lstrip() 메소드를 사용하여 epoch 값에서 "epoch" 문자를 제거하고, int() 함수를 사용하여 정수로 변환하여 epoch2 리스트에 추가했다. 이후 rstrip() 메소드를 사용하여 gan\_loss2, val\_mse2, psnr2, mse2 값에서 ";" 문자를 제거하고, float() 함수를 사용하여 실수로 변환하여 각각의 리스트에 추가한다. 마지막으로 파일을 닫기 위해 file2.close()를 호출한다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명이 코드는 그래프를 그리는 함수 plotting()과 그래프를 그리기 위한 각각의 데이터를 설정하는 함수들을 포함한 것이다. plotting(num, name, Data1, Data2) 함수는 그래프를 그리기 위한 함수로 num은 그래프의 순번을 나타내며, 첫 번째 그래프는 num\*2-1, 두 번째 그래프는 num\*2로 설정했다. name은 y축의 이름을 나타내며 Data1과 Data2는 사용할 데이터를 나타낸다. 우선, plt.figure() 함수를 사용하여 그래프의 Figure 객체를 생성했다. plt.plot() 함수를 사용하여 데이터를 이용해 그래프를 그리는데 plt.xlabel(), plt.ylabel(), plt.title() 함수를 사용하여 각각의 x축, y축, 그래프의 제목을 설정한다. plt.legend() 함수는 범례를 표시하는 것이고 plt.grid(True) 함수를 사용하여 그래프 격자를 표시했다. plt.subplot() 함수를 사용하여 두 번째 그래프를 그릴 때 subplot을 설정한다. plotting\_data1(), plotting\_data2(), plotting\_data3(), plotting\_data4() 함수들은 각각 특정 데이터를 사용하여 그래프를 그리기 위한 함수들이다. plotting() 함수를 호출하여 해당 데이터와 그래프를 그린 것 이다. 즉, 함수의 이름에 따라 어떤 데이터를 사용하여 그래프를 그릴지 결정하기 위하여 정의한 것이다.

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명이 코드를 실행하면, plotting\_data1(), plotting\_data2(), plotting\_data3(), plotting\_data4() 함수를 호출하여 각각의 그래프를 생성하고, 설정된 데이터를 이용하여 그래프를 그린다. 각 그래프는 서로 다른 데이터를 비교하여 표시하고, 각각의 그래프는 하나의 Figure 객체 내에 subplot으로 표시된다.

이 코드는 개인정보를 파일에 저장하고, 해당 파일에서 개인정보를 읽어와서 텍스트 위젯에 표시하는 기능을 한다.

Data\_Read() 함수는 개인정보 불러오기 버튼이 눌렸을 때 myInfo.txt 파일을 읽어 들인다. 파일에서 데이터를 읽고 각각의 정보를 변수에 저장한다. Name(이름), Number(학번), Grade(학년), Major(전공)이 저장된다. t1.insert() 함수를 사용하여 텍스트 위젯 첫 번째 줄에는 성명, 두 번째 줄에는 학번, 세 번째 줄에는 학년, 네 번째 줄에는 전공으로 개인정보를 표시한다. 이후 파일을 닫는다. Data\_Save() 함수는 Entry 위젯에 입력된 내용을 가져오는 함수로 개인정보저장 버튼이 눌렸을 때 실행 된다. 가져온 데이터를 myInfo.txt 파일에 저장한후 f1.write() 함수를 사용하여 파일에 개인정보를 저장한다. 이후 Entry 위젯의 내용을 처음부터 끝까지 지워버린다. 즉, Data\_Read() 함수는 myInfo.txt 파일에서 개인정보를 읽어와서 텍스트 위젯에 표시하고, Data\_Save() 함수는 Entry 위젯에 입력된 개인정보를 myInfo.txt 파일에 저장한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명이 코드는 Tkinter를 사용하여 GUI 창을 생성하고, 위젯들을 배치하는 역할을 한다. Tkinter의 Tk() 함수를 사용하여 root라는 Tk 객체를 생성한다. 또한, root.geometry()를 사용하여 윈도우 창의 크기와 위치를 설정했고 root.title()로 윈도우 창의 제목을 설정했다. 이후 Frame(root)를 사용하여 frame1이라는 프레임을 생성한다. frame1.pack()을 사용하여 frame1을 root 윈도우에 배치하고 Frame(frame1)를 사용하여 window1이라는 프레임을 생성한다. window1.pack()을 사용하여 window1을 frame1에 배치한다. 마찬가지로 window2와 window3도 처리해주었다. 이제 라벨 위젯으로 윈도우 창에 표시해주어야 하기 때문에 Label위젯을 만들어준다. Label(window1, text="개인정보 입력", fg="blue")을 사용하여 L1이라는 레이블을 생성했다. L1.grid()를 사용하여 L1을 window1에 배치한다. Label(window1, text="성명")을 사용하여 L2라는 레이블을 생성합니다.L2.grid()를 사용하여 L2를 window1에 배치합니다.Label(window1, text="학번")을 사용하여 L3라는 레이블을 생성합니다.L3.grid()를 사용하여 L3를 window1에 배치합니다.L4와 L5도 마찬가지로 각각 학년과 전공을 넣어서 배치해준다. 이 소스 코드는 root 윈도우와 여러 프레임을 생성하고, 각각의 위젯들을 해당 프레임에 배치하는 것이다. 또한, Label 위젯을 사용하여 텍스트를 표시한다.

이 소스 코드는 Entry 박스와 버튼을 생성하고 해당 위젯들을 window1 프레임에 배치하는 기능으로 이름과 학번, 학년, 전공을 입력하는 Entry박스를 생성하고 설정하는 것이다. Entry(window1,relief="solid", bd=1)을 사용하여 e1이라는 Entry 박스를 생성한다. 마찬가지로 e2와 e3, e4라는 Entry 박스를 생성한다. e1.grid()는 e1을 window1에 배치하는 것인데 옵션을 설명하자면 row는 위젯이 배치될 행 번호로 row=1로 설정했기 때문에 e1은 window1의 첫 번째 행에 배치된다. column은 위젯이 배치될 열 번호이며 column=1로 설정했기 때문에 e1은 window1의 두 번째 열에 배치된다. padx는 위젯의 좌우 측면에 적용되는 가로 여백을 지정한다. 즉, e1의 좌우에 5px의 여백이 적용된 것이다. Pady는 위젯의 상하 측면에 적용되는 세로 여백을 지정하는 것으로 e1의 상하에 5px의 여백이 적용된다. e1과 동일하게 e2, e3, e4를 배치해준다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명Button(window1, text='개인정보 저장', command=Data\_Save, width=18, height=1,relief="solid", bd=1)을 사용하여 B1이라는 버튼을 생성하는데 텍스트 이름은 개인정보 저장으로 하고 버튼이 눌리면 Data\_Save함수를 실행한다. B2는 개인정보 불러오기라는 이름의 버튼으로 버튼이 눌렸을 때 Data\_Read()함수를 수행한다. B3버튼은 Data1의 이름의 버튼으로 버튼이 눌렸을 때 data1함수를 수행하고 B4버튼은 Data2 이름으로 버튼이 눌리면 data2함수를 실행한다. Width는 폭 height는 높이 solid설정은 실선으로 하겠다는 의미이다. 즉, 이 소스 코드는 Entry 박스를 사용하여 사용자로부터 데이터를 입력 받을 수 있도록 설정하고 버튼을 클릭하면 지정된 함수가 실행되도록 설정한 것이다.

위 소스 코드는 window2 프레임에 개인정보 요약을 위한 라벨, 텍스트 박스, 그리고 데이터 플롯을 생성하는 버튼들을 배치하는 기능이다. L7은 개인정보 요약의 텍스트 라벨로 색은 파란색으로 설정한다. t1은 window2 프레임에 표시되는 텍스트 박스로 height=11, width=20으로 텍스트 박스의 크기를 지정하며, 실선으로 설정했다.

B5버튼은 window2 프레임에 생성한 버튼으로 "Gan Loss Plot"으로 지정되며, 해당 버튼을 클릭할 경우 plotting\_data1 함수가 실행된다. B6 버튼은 "PSNR Plot"으로 지정되며, 버튼을 클릭할 경우 plotting\_data2 함수가 실행된다. B7 버튼은 "Val MSE Plot"으로 지정되며, 해당 버튼을 클릭할 경우 plotting\_data3 함수가 실행되는 것이다. B8은 마지막 버튼으로 텍스트는 "MSE Plot"으로 지정되며, 버튼을 클릭할 경우 plotting\_data4 함수가 실행된다. 즉, 개인정보 요약을 표시하는 텍스트 박스와 각종 데이터 플롯을 생성하기 위한 버튼들을 window2 프레임에 배치하는 코드로 버튼을 클릭하면 해당하는 함수가 실행되어 그래프가 생성된다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

data\_processing\_and\_save() 함수는 데이터 처리와 저장을 수행하는 함수로 title\_save리스트에는 각 데이터에 대한 제목"Gan Loss", "Val\_mse", "PSNR", "Mse"를 저장한다. save1은 첫 번째 데이터에 해당하는 변수들을 저장하는 리스트로 "gan\_loss1", "val\_mse1", "psnr1", "mse1"이다. save2는 두 번째 데이터에 해당하는 변수들을 저장하는 리스트로 "gan\_loss2", "val\_mse2", "psnr2", "mse2"로 정의되어 있다.

이후 "DataProcessing.txt"라는 파일을 열고 데이터 처리 결과를 파일에 저장한 뒤, 해당 내용을 텍스트 박스에도 출력한다. 파일은 with open('DataProcessing.txt', 'w') as file\_save: 구문을 통해 열리며, 파일을 열고 작업을 마친 뒤 자동으로 닫힌다.

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명title\_save에 있는 각 데이터 제목에 대해 최대값, 최소값, 평균값을 계산하여 파일과 텍스트 박스에 작성해야 하는데 for 루프를 사용하여 각 데이터에 대한 정보를 파일과 텍스트 박스에 작성하는 부분을 반복한다. 두 번째 데이터 또한 같은 방식으로 이루어지며, 구분을 위해 "=========================================\n"과 [data2.txt 파일의 최대값, 최소값, 평균값]를 파일과 텍스트 박스에 추가로 작성한다. 즉, 이 소스 코드는 데이터 처리 결과를 파일로 저장하고, 해당 내용을 텍스트 박스에 표시하는 것이다.

위 소스 코드 중 L8은 "데이터"라는 텍스트를 포함하는 레이블이며 글자색은 파란색으로 설정한다. t2는 데이터를 표시하는 텍스트 박스로 높이는 22, 너비는 43으로 설정했다. B8 버튼은 "Data Processing & Save"라는 텍스트를 가진 버튼으로 이 버튼이 눌렸을 때 data\_processing\_and\_save() 함수를 실행하여 데이터 처리 및 저장 작업을 수행한다. img는 "hallym\_logo.gif" 파일로부터 이미지를 생성하는 객체로 한림로고를 나타낸다. Hallym\_Logo는 한림로고 이미지를 표시하는 레이블이며 이미지는 (390, 500)에 배치시켰다. 마지막으로 root.mainloop()로 윈도우 창이 종료되기 전까지 프로그램이 이벤트 루프에 진입하여 사용자 입력을 처리한다.

**실행결과**

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

프로그램 실행 결과

텍스트, 스크린샷, 도표, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

현재 myInfo.txt파일에는 아무것도 적혀 있지 않은 상태이고 개인정보 입력을 한 뒤 개인정보 저장 버튼을 클릭한다.

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

입력된 개인정보들은 지워지고 myInfo.txt파일에 입력했던 정보들이 저장된다.

텍스트, 스크린샷, 디스플레이, 도표이(가) 표시된 사진

자동 생성된 설명

개인정보 불러오기 버튼을 누르면 myInfo.txt파일에서 읽어들어와 개인정보 요약에 띄운다.

텍스트, 스크린샷, 소프트웨어, 도표이(가) 표시된 사진

자동 생성된 설명

Plot버튼을 누르면 리스트가 생성되지 않았으므로 아무것도 뜨지않고 Data1버튼과 Data2버튼을 눌러야 리스트가 생성돼서 저장된다.

텍스트, 도표, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 도표, 번호이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 그래프, 번호이(가) 표시된 사진

자동 생성된 설명

Gan Loss Plot 버튼을 누르면 gan\_loss 그래프가 그려지고, PSNR Plot 버튼이 눌리면 psnr 그래프가 그려진다. 또한 Val MSE Plot와 MSE Plot버튼을 누르면 각각 val mse 그래프와 mse 그래프가 그려진다.

텍스트, 스크린샷, 도표, 번호이(가) 표시된 사진

자동 생성된 설명

Data Processing & Save 버튼을 누르면 data1.txt 파일에서 gan\_loss, val\_mse, psnr, mse의 각각의

최대값과 최소값 평균값을 구해서

데이터에 표시하고

텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명========라인을 그어서 data2.txt 파일과 구분해준다. 마찬가지로 data2.txt 파일에서 gan\_loss, val\_mse,

Psnr, mse의 최대값과 최소값, 평균값을 구해서 표시해 주었다.

DataProcessing.txt파일에도 생성과 동시에 똑같이 작성된 것을 확인할 수 있다.

실행결과를 보면 알 수 있듯이 우측 아래에 한림 로고를 배치하였다.

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명