COSE474-2022: FINAL PROJECT
멀티모달 데이터를 활용한 정신건강진단 모델 구현

2018320105 구재승

1. Introduction

- Motivation

과거에 비해 현대에 들어오면서 사람들은 신경 써야 할 것들이 늘어났고, 그에 따라서 정신적인 질환을 하나씩은 가지게 되었다. 사람이 감기에 걸리면 병원에 가듯, 정신적인 질환을 가지고 있다면 상담을 받아보는 것이 좋다. 그러나 자신이 정신 질환을 앓고 있다는 것이 타인에게 알려 졌을 때 그들이 바라보는 시선이 껄끄럽기 때문에, 사람들은 상담을 받기 주저한다. 필자 또한 상담을 받아 본 경험이 있는 사람으로서, 감정 상태를 판단해 볼 수 있는 도구가 있으면 유용할 것 같아 이를 딥 러닝을 이용해 구현해 보고자 했다.

-Problem definition & challenges

사람의 감정은 가변적인 것이기에, 우울증에 걸린 환자도 기쁜 상황일 때가 있고, 평범한 사람도 슬플 때가 있다. 또한 타인이 지금 어떤 감정을 느끼고 있는지 확실하게 알기는 어려움이 존재한다. 이런 모호성은 딥 러닝 모델에 사용될 데이터셋의 라벨이 명확하게 구분되지 않을 수 있는 위험을 야기한다.

이 문제를 해결하기 위해 사람이 우울감을 느낄 때 후두에서 발생하는 생리학적 변화를 이용, CNN 모델로 목소리를 학습시켜 우울증을 진단하는 이전 연구를 참고했다.[1] 이 논문을 통해서 사람의 감정 상태라는 모호했던 기준을 명확한 기준으로 구별할 수 있었다.

-Related works & Dataset

AI hub 웹 페이지에서 제공하는, "정신건강 진단 및 예측을 위한 멀티 모달 데이터" 를 사용할 예정이다. 전남대학교 병원에서 주관하여 데이터 수집, 데이터 정제 및 인공지능 모델 설계를 했다. 그 외 6 개의 기관이 데이터 수집, 가공, 모델링, 품질 관리 등을 담당했다. 전남대학교 병원에서 제공하는 모델은 Support-Vector Machine(SVM)이다[2]. 이를 baseline 으로 삼아서 같은 데이터 셋을 사용하되 Neural Network 형태를 차용, 직접 Layer 를 쌓아 올려 모델 학습을 시켜 비교를 했다.

2. Methods & Experiments

-Data processing

인공지능 모델을 학습시키는데 Pytorch 를 이용했고, 전처리 과정에서 Jupyter Notebook 을, 모델 학습 과정에서 Colab 을 사용했다.

인공지능에 학습시켜야 할 데이터는 단순 파일로 주어진 것이 아니라, 웹 기반 저작도구(web-based authoring tool)로서 주어졌기에 전처리 과정이 필요했다. MariaDB, Java, Apache Tomcat 을 이용하여 로컬 웹 서버를 구성하는 과정이 사전작업으로 필요했고, 이후 주어진 SQL 파일을 이용, Database 에 내용을 입력한 후 Chrome 으로 로컬 호스트에 접속해야 데이터를 확인 할 수 있었다.

서버에서 입력되어 있는 데이터만을 따로 다운받을 수 있는 방법이 없었기에, python 을 이용한 웹 크롤링을 통해 csv 파일로 데이터 전처리 과정을 진행했다. 이 과정에서 Colab 을 이용한다면 로컬 컴퓨터에 접속해서 해당 정보들을 불러 올 수 없었기에, 부득이하게 전처리 과정에서만 Jupyter Notebook 을 이용했다. 그 결과 약 2000 개의 데이터 셋을 확보할 수 있었다.

각 데이터는 No, 대상자 번호, 방문차수, PHQ-9 total, HADS-depression total, HADS-depression 중증도번호, 문항 1, 문항 2, 중증도 까지 총 10 개의 attribute 를 가졌고, 우울증 진단이 내려진 경우 중증도가 1, 아닌 경우 중증도가 0 으로 체크되었다.

-training

총 2060 개의 데이터 중에서 80%에 해당하는 1648 개를 training set 으로, 남은 416 개를 validation set 으로 사용했다. 모델의 경우 총 6 개의 layer 로 구성, 각 layer 를 Linear 한 형태로 두었고 activation function 으로 ReLU 를 사용했다. 처음 가중치는 Xavier Initialization 을 했다. Learning rate 는 0.002 로, epoch 는 30 으로 설정했다.

-validation

아래의 표는 baseline 의 test 결과이다.

Classification report

	Precision	Recall	F1-score	Support
0	0.7075	0.9036	0.7937	83
1	0.8667	0.6265	0.7273	83
accuracy			0.7651	166
macro avg	0.7871	0.7651	0.7605	166
weighted avg	0.7871	0.7651	0.7605	166

구성한 모델의 Validation 을 진행하였을 때 277/412 로 0.6723 의 accuracy 를 띄어, baseline 보다 낮은 정확성을 띄었다. 기존 baseline 보다 더 낮은 정확도가 나온 이유는 크게 2 가지 측면에서 분석할 수 있을 것 같다. 첫째로 표본이 너무 적었다. 둘째, overfitting 이 발생했다.

인공지능 모델의 output 은 오직 2 가지 종류다. 우울증에 걸렸거나 걸리지 않았거나. 그렇기에 복잡한 모델이 필요가 없다. 두 영역으로 구분만 하면 되기 때문에 Baseline 인 SVM 이 매우 적절하다. 두 영역을 나누는 구분선을 Neural Network 를 이용해서 모델을 구성하려고 하다 보니 overfitting 이 발생했다. 이를 방지하기 위해서는 많은 양의 데이터셋이 필요하다. 그러나 모델을 학습시키고 평가하는데 사용한 2080 개의 데이터 셋은 너무 적다. 이 두 가지 문제점이 합쳐져서 오히려 더 성능이 낮은 모델이 되었다.

3. Future direction

추가 연구를 해 볼만한 방향으로 두 가지 의견을 제시하고자 한다. 첫 번째는 데이터 셋을 확보하는 방법에 대한 관점이고, 두 번째는 우울증의 종류를 세분화 하는 것이다.

기존의 연구에서 데이터 셋을 확보할 때, 너무 많은 과정이 필요했다. 음성 데이터 뿐만 아니라 유전/혈액 데이터, 동반 질환 진단, 우울 증상 확인 등의 임상데이터를 사용했다. 때문에 많은 양의 데이터 셋을 확보하지 못했다. 따라서 우울증을 진단하기 위해 필요한 데이터 종류를 줄여야 할 필요가 있다.

또한, 우울증에서 여러 종류가 존재한다. 일반적으로 우리가 생각하는 우울증도 있지만, 조울증이나 정신증이 동반되는 우울증도 존재한다. 단순히 우울증이 있는지 없는지 판단하는 것이 아니라 종류까지 판별해 낼 수 있는 인공지능 모델을 구성하는 것은 상당히 흥미로운 연구가 될 것이라 생각한다.

Reference

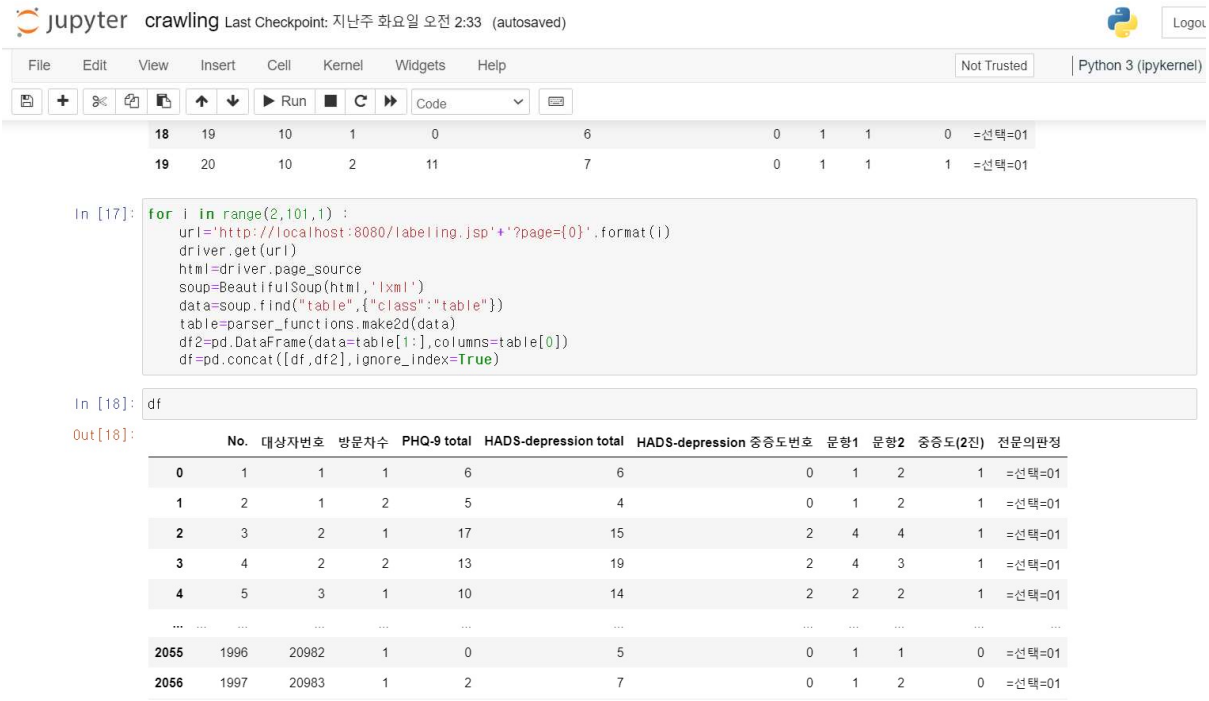
[1] Dubagunta, S. Pavankumar, Bogdan Vlasenko, and Mathew Magimai Doss. "*Learning voice source re7 lated information for depression detection.*" . ICASSP 2019-2019 IEEE International Conference on Acous┐ tics, Speech and Signal Processing (ICASSP). IEEE, 2019.

[2]데이터 분야 - AI 데이터찾기 - AI-Hub (aihub.or.kr) (2022.12.05)

https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=573

Appendix

a) Jupyter Notebook 을 이용한 데이터 크롤링



jupyter crawling Last Checkpoint: 지난주 화요일 오전 2:33 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

18	19	10	1	0	6	0	1	1	0	=선택=01
19	20	10	2	11	7	0	1	1	1	=선택=01

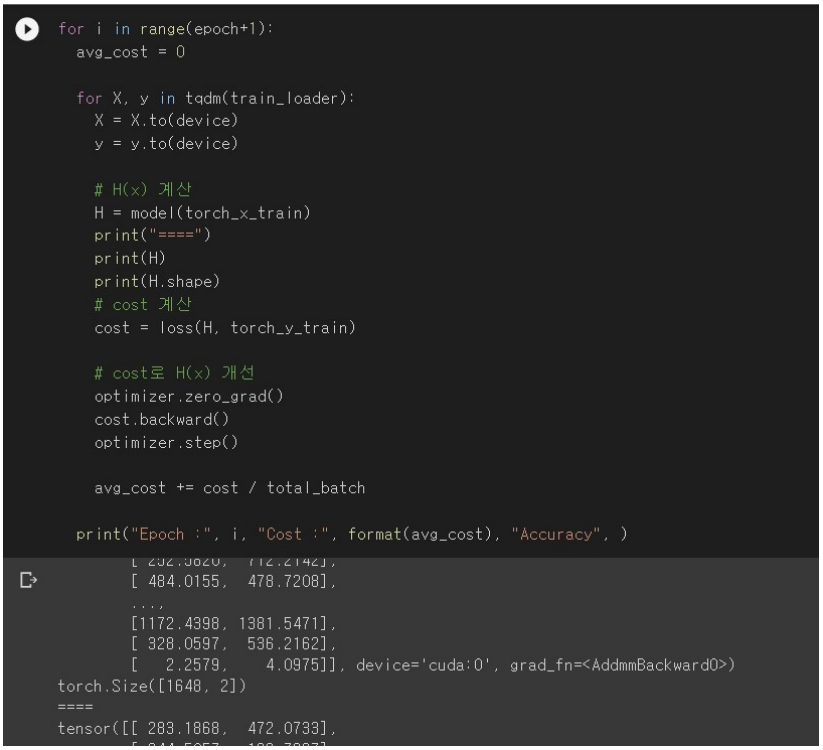
In [17]:
```
for i in range(2,101,1) :
    url='http://localhost:8080/labeling.jsp'+'?page={0}'.format(i)
    driver.get(url)
    html=driver.page_source
    soup=BeautifulSoup(html,'lxml')
    data=soup.find("table",{"class":"table"})
    table=parser_functions.make2d(data)
    df2=pd.DataFrame(data=table[1:],columns=table[0])
    df=pd.concat([df,df2],ignore_index=True)
```

In [18]:
```
df
```

Out[18]:

	No.	대상자번호	방문차수	PHQ-9 total	HADS-depression total	HADS-depression 중증도번호	문항1	문항2	중증도(2진)	전문의판정
0	1	1	1	6	6	0	1	2	1	=선택=01
1	2	1	2	5	4	0	1	2	1	=선택=01
2	3	2	1	17	15	2	4	4	1	=선택=01
3	4	2	2	13	19	2	4	3	1	=선택=01
4	5	3	1	10	14	2	2	2	1	=선택=01
...	...	...	...	...	...	...	...	...	...	...
2055	1996	20982	1	0	5	0	1	1	0	=선택=01
2056	1997	20983	1	2	7	0	1	2	0	=선택=01

b) Google Colab 을 이용한 모델 학습



```
for i in range(epoch+1):
  avg_cost = 0

  for X, y in tqdm(train_loader):
    X = X.to(device)
    y = y.to(device)

    # H(x) 계산
    H = model(torch_x_train)
    print("====")
    print(H)
    print(H.shape)
    # cost 계산
    cost = loss(H, torch_y_train)

    # cost로 H(x) 개선
    optimizer.zero_grad()
    cost.backward()
    optimizer.step()

    avg_cost += cost / total_batch

  print("Epoch :", i, "Cost :", format(avg_cost), "Accuracy", )
```

```
        [ 484.0155,  478.7208],
        ...,
        [1172.4398, 1381.5471],
        [ 328.0597,  536.2162],
        [   2.2579,    4.0975]], device='cuda:0', grad_fn=<AddmmBackward0>)
torch.Size([1648, 2])
====
tensor([[ 283.1868,  472.0733],
```

c) 개인 Github 페이지 : github/JaeseungGu

d) 저작도구 소개자료 : 별도 첨부