# SNU 2021-1 DS Project 3

Team #4

2019-19835 김민영

2018-14070 김채린

2019-12863 문서윤

2016-10556 이재현

# 팀 소개

**"행복은 성적순이 아니야! 졌지만 잘 싸웠다!"**

- CNN model
- Hyper Parameter Tuning
- 다양한 models로 데이터 학습
- labels handling

이재현

김채린

- CNN model
- Hyper Parameter Tuning
- 다양한 models search / trial
- ECG data 다룬 기준 모델 조사

- Feature extraction & application
- ResNet 1D model
- Classification

김민영

문서윤

- 선행연구 조사
- EDA
- Data preprocessing
- ResNet

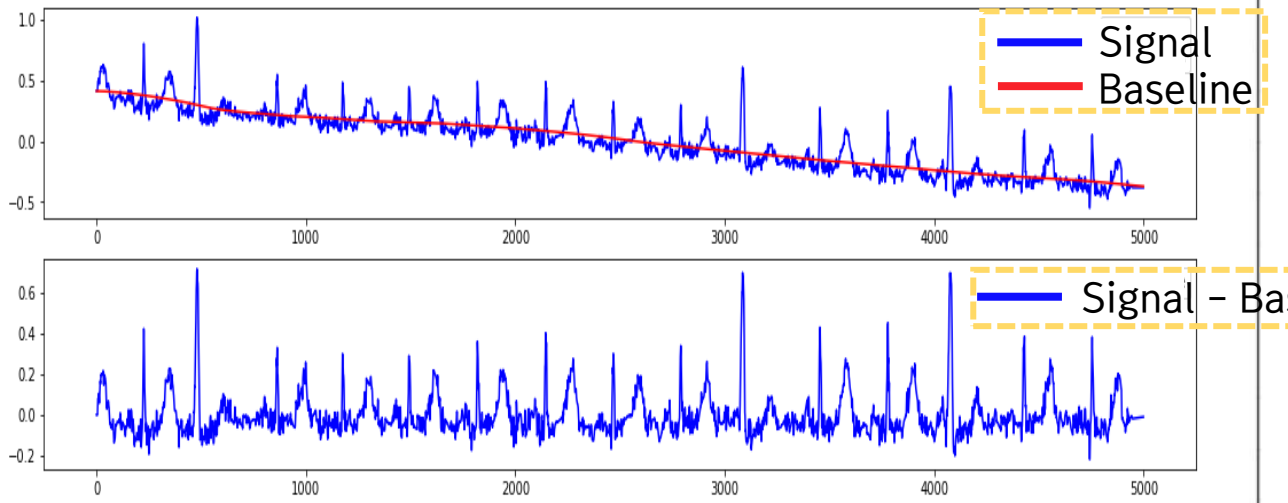
/ 4번의 Zoom 회의, 2번의 대면 회의 /

최종 성적 : 0.61024

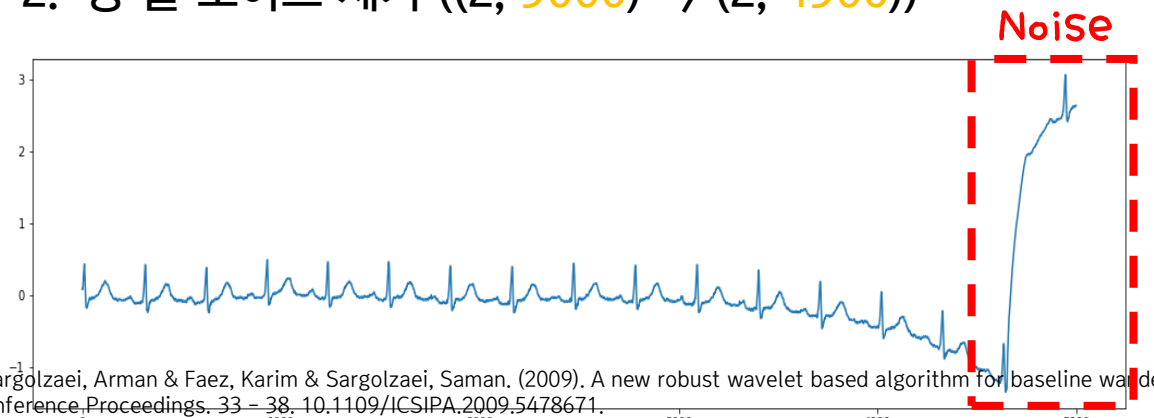
# Data pre-processing & feature engineering

## 1. Baseline wander 제거* (codes from github)

/ discrete wavelet transform /




## 2. 양 끝 노이즈 제거 ((2, 5000) -〉 (2, 4500))




## 3. Feature Extraction

/ A Wide and Deep Transformer Nerual Network for 12-Lead ECG Classification** /

| Rank | Feature |
|---|---|
| 1 | $HR_{min}$ |
| 2 | T wave multiscale permutation entropy $\sigma$ |
| 3 | $HR_{max}$ |
| 4 | T wave multiscale permutation entropy median |
| 5 | RMSSD |
| 6 | P wave correlation coefficient |
| 7 | RR interval median |
| 8 | Heart rate $\mu$ |
| 9 | RR interval intra-cluster distance, cluster 3 |
| 10 | RR interval Fisher information |
| 11 | [illegible] |
| 12 | SWT decomposition level 4 entropy |
| 13 | [illegible], cluster 2 |
| 14 | Heart rate activity |
| 15 | $\Delta RR_{min}$ |
| 16 | T wave permutation entropy $\sigma$ |
| 17 | P wave sample entropy $\sigma$ |
| 18 | SWT decomposition level 3 entropy |
| 19 | Median p wave approximate entropy |
| 20 | R peak approximate entropy |

: feature extraction

/ 방법 /

1) scipy.find_peaks(distance=200, prominence = 0.25)
(*preprocessed data 사용)

2) QRS detection -〉 R peak 찾기
(참고 : Electrocardiograms:QRS Detection Using Wavelet Analysis_Andrew Tan)

/ 결과 /

Private score 기준 0.50620(cnn, recordings, impute X)
-〉 0.52178(cnn, recordings + 5 features + age, sex, impute '8')

***결론적으로 Raw data만을 사용했을 때 가장 성능이 잘 나왔습니다.***

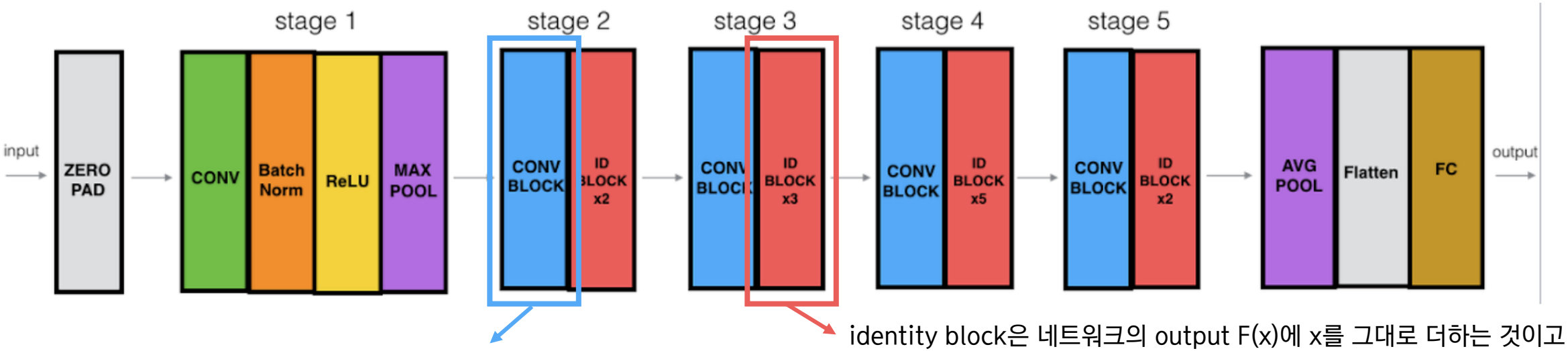
*Sargolzaei, Arman & Faez, Karim & Sargolzaei, Saman. (2009). A new robust wavelet based algorithm for baseline wandering cancellation in EC[illegible] Conference Proceedings. 33 - 38. 10.1109/ICSIPA.2009.5478671.

** A. Natarajan *et al.*, "A Wide and Deep Transformer Neural Network for 12-Lead ECG Classification," *2020 Computing in Cardiology*, 2020, pp. 1-4, doi: 10.22489/CinC.2020.107

# Modeling

## - Resnet

/ Resnet Structure /




identity block은 네트워크의 output F(x)에 x를 그대로 더하는 것이고

convolution block은 역시 1x1 convolution연산을 거친 후 F(x)에 더해주는 것

✚ skip connection을 이용한 residual learning을 통해 layer가 깊어짐에 따른 gradient vanishing 문제를 해결

- 이번 competition data는 1D, Resnet은 주로 2D data에 쓰임 ▶ 데이터를 2D로 변형 vs 1D Resnet 코드 찾기
- Github 참고, 변형하여 Resnet1D 모델 사용 ▶ PhysioNet/CinC Challenge 2017 SOTA
- epoch = 25, batch_size = 32, learning_rate = 0.001, kernel_size = 16, stride = 2, n_block = 48, downsample_gap = 6, increasefilter_gap = 12, base_filters = 128 등 ▶ 0.59859(epoch 24) => 0.61024(epoch 25)

  *epoch 1만 늘려도 0.01상승 => 한 epoch 50까지 돌리지 못한 아쉬움..

# 기타

- **Imputing**
  : null 값 “8”로 〉 0.59859 => 0.59812, 0.61024 => 0.60979 대체적으로 성능 악화

- **Three models with ResNet**

  1. Model1 (8인지 아닌지) / Model2 (3인지 10인지 아닌지) / Model3 (나머지)
  2. ResNet으로 각각 학습시킨 후 마지막 submission file 만들 때 합침
  3. 각 모델은 F1이 0.9, 0.7, 0.6까지 나왔지만 test submission 결과 F1=0.15가 나옴.

- **Three models with CNN**

  1. Model1 (12개의 labels을 정상 학습) / Model2 (label “8”만 학습) / Model3 (label “8”이 아닌 것들만 학습)
  2. 각각 변형된 CNN으로 학습 후, 세 가지 모델을 활용하여 test data 예측 ▶ 0.41

- Classification : Using **h2o automl**

  1. normal(includes 8) vs abnormal
  2. only 8 vs 8+ others vs others
  3. Including 3 vs including 10 vs others

▶ / 3가지 시도 결과 /
normal 7388 | abnormal 1
only_8 5543 | others 1846
includes_3 14 | others 7375

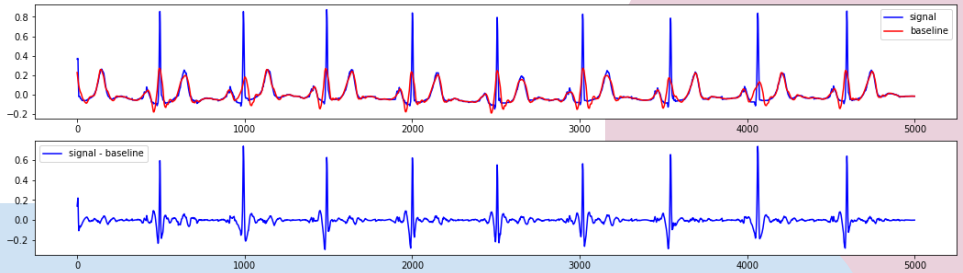
▶ train data에서 각 그룹에 크기가 달라서 그런가?
-〉 랜덤 추출로 그룹 크기 맞춰도 결과가
한 그룹에 편중되는 현상 발생

▼

모든 binary classification 결과 8 포함으로 나오는 400여개, abnormal으로 나오는 1000여개에 대해 CNN label 결과로 수정

0.50620 => 0.50500, 0.50416 (근소한 차이로 성능 악화)

# 팀원 1: 문서윤




## Data pickle로 정리 (데이터 불러오는 시간 단축)

## CNN 구조 변형

- BN, Dropout, Pooling 추가 (0.47 -> 0.49)

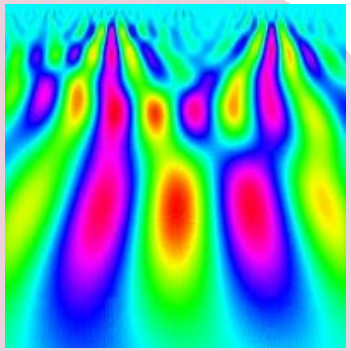
## Preprocessing (0.49 -> 0.50)

- Baseline wander, 양 끝 노이즈 제거
- 문제점 : 일부 데이터에서 P나 T wave도 사라지게 함
- 실제로 최종 ResNet에 넣어 본 결과
  - ▶ test score= 0.58

**도움이 된 것**

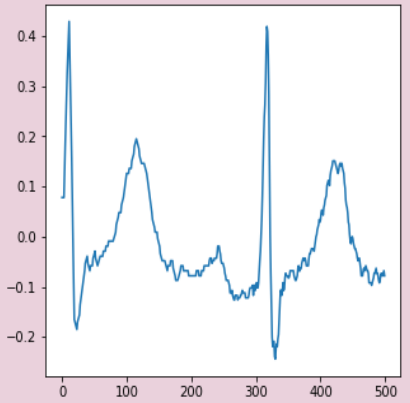
## CWT Spectrum 이용, 2D CNN 모델로 학습

- 데이터가 너무 커서 Kaggle, Google colab이 감당불가



## 500 단위로 자르기

- 학습이 안 됨



## 모델 세 개로 나눠 각각 학습

- (8), (3, 10), (나머지)
- ResNet이용, 각 class의 비율을 다 맞췄지만 test score = 0.15...

## TPU 사용

- ResNet이 너무 커 Google colab에서 TPU를 사용해봤지만 왠지 모르게 GPU 사용했을 때보다 학습시간이 더 늘어남

**도움이 안 된 것**

# 팀원 2: 이재현

**1.**

CNN을 조정하여 Preprocessed data (5000 -> 4500) 을 학습시킨 후 예측 ▶ 0.444

**2.**

test data

**/ Model_1 /**
: CNN training 12 labels

**/ Model_2 /**
: CNN training only "8"

"8" with other labels

"8" prediction

"NOT 8" prediction

..?

**/ Model_3 /**
: CNN training except "8"

labels including "8"

labels except "8"

labels which contain "2" without "3" 에 "3"도 넣어 주기
empty labels에 "6" 넣기

0.41

**/ SeoYoon's discussion /**
"증상 6이 부정맥인데 부정맥이 워낙 불규칙하니까 분류가 잘 안될 것 같다."

**/ notiona's discussion /**
"증상 2가 있는 경우, 증상3이 있을 가능성이 굉장히 높을 것 같습니다. 실제로 training set에서 확인했을 때, 증상 2가 발생하는 1185번 중 912번은 증상 3도 있었습니다. 반대의 인과는 설명력이 낮았습니다."

# 팀원 3: 김채린

## Modeling : 3개의 모델 시도

```
self.softmax= torch.nn.LogSoftmax(dim=0)
```

Leaky ReLU function is an improved version of the ReLU activation function.

```
def forward(self, x):
    residual= self.conv(x)

    #block1
    x = F.relu(self.conv_pad(residual))
    x = self.conv_pad(x)
      x = self.batch1(x)
    x+= residual
    x = F.relu(x)
    residual = self.maxpool(x) #[512 32 90]
```

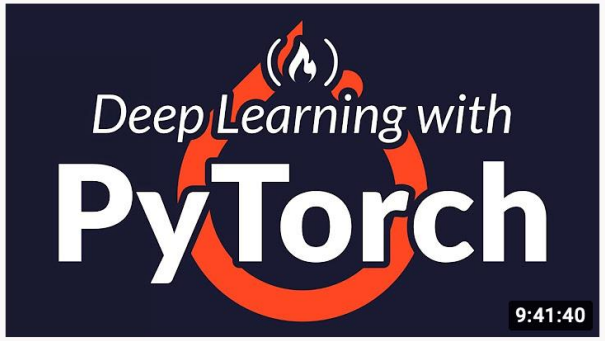
### / Adding layers /

- 0.43 -> 0.47 -> 0.50 -> 0.53
- 9 Conv layers + 5 FC layers `self.batch5 = torch.nn.BatchNorm1d(256)`
- dropout layer 점수 오히려 떨어짐
- Leaky ReLU 0.03 떨어짐

### / Hyper Parameter Tuning /

- learning rate = 0.0001 0.46820으로 떨어짐
- batch size = 32 (10 -> 0.50700 / 64 -> 0.50226)
- epoch = 200
- kernel_size : 2정도씩 줄인 것이 가장 좋았음

+ `replace(np.nan, '6')` 0.50955 -> 0.51331, + 0.002 이상



# 팀원 4: 김민영

(앞의 전체 ppt 슬라이드와 중복되는 내용이 많아 간략하게 설명하겠습니다!)

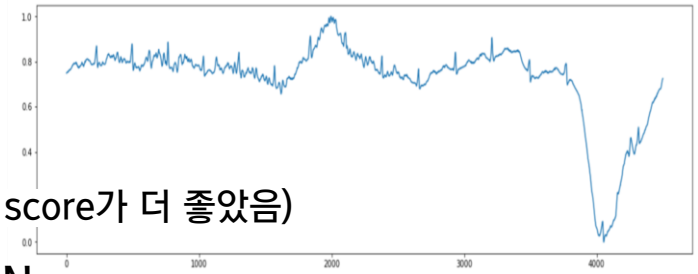
- Feature Extraction: Lead II로부터 HR_max, HR_min, RR interval median, HR_mean, min_RR interval 변화율 (ppt 4)

  find_peaks => r_peak 찾지 못한 그래프에 대해 QRS detection, 극대점 아니면 삭제 => 중간에 r점 몇개 찾지 못해 max,min에 영향 방지:  rr_interval outlier 제거(Q3값 3배 이상 -> 제거)

  scipy find_peaks 파라미터 근거=> 정상 rr interval 0.6~1.2 sec. r점과 t-wave 극대점 혼동하는 경우 있음 => distance 200 = 0.4sec. prominence 0.25(t-wave 보통 이보다 작음)

  QRS_detection window 600(1.2sec), max_bpm 220

  feature extraction 적용 전 팀 최고 점수: private 0.50620 public 0.52969



- CNN 모델 FC layer에 5 extracted features + age, sex 추가

  private: 0.52178(cnn, recordings + 5features + age, sex, impute '8') -- public 0.50899 (유일하게 private score가 더 좋았음)

- CNN 모델 결과물 수정 시도: H2o Automl 통한 Classification 시도(ppt 6), imputing NaN

  성능하락(0.50500, 0.50416) => 결론: 꼼수(?)는 잘 안통한다.. extracted_feature는 정상/비정상을 구분하는 데는 도움이 되지 않는다.

- Resnet1D 모델 시도(ppt 5)

  0.61024 => 최종결과!

  extra features(extracted features, age, sex) 연결 시도 -> forward 함수 고치는 데 실패.