

2024년 전기 졸업과제 중간보고서

# IMU 기반 경량 재활운동 자세 추론 및 보조 시스템 개발



팀명	패트와 매트랩
분과	D(하드웨어 · 보안)
학번	201924603
이름	하규승
학번	201924446
이름	김지훈
지도교수	백윤주 교수님

## 목차

<b>1. 요구조건 및 제약 사항 분석에 대한 수정사항.....</b>	<b>3</b>
1) 기존 요구조건.....	3
2) 수정 후 요구조건.....	3
3) 요구 조건에 따른 신규 제약 사항.....	3
<b>2. 설계 상세화 및 변경내역.....</b>	<b>4</b>
1) IMU 데이터 수집.....	4
2) IMU 데이터를 게이트웨이에 전송.....	4
3) 머신러닝 모델 학습 수행.....	5
4) 재활운동 자세별 모듈 개수 최적화.....	5
<b>3. 갱신된 과제 추진 내역.....</b>	<b>6</b>
<b>4. 구성원별 진척도.....</b>	<b>6</b>
<b>5. 보고 시점까지의 과제 수행내용 및 중간결과.....</b>	<b>7</b>
1) 자세 선정.....	7
2) 센싱 데이터 수신 & 모델 학습.....	7

## 1. 요구조건 및 제약 사항 분석에 대한 수정사항

기존의 개발 주제는 임베디드 환경에서 **HPE**를 수행하고, 이를 운동재활치료에 도입하는 것이었다. 하지만 단순히 **IMU**를 이용한 **HPE**는 그 범위가 상당히 방대하다. 따라서 미리 재활운동을 6개 정도 선정하여, 사용자가 선정된 재활운동 중 하나를 선택하면 그 운동 자세에 대해 시스템에게서 피드백 받을 수 있도록 주제를 조금 더 세분화 하였다.

### 1) 기존 요구조건

- 임베디드 환경에서 **IMU** 데이터를 이용해 재활운동 자세를 추론

### 2) 수정 후 요구조건

- 임베디드 환경에서 **IMU** 데이터를 이용해 재활운동 자세를 추론
- 사용자가 미리 제공된 재활운동 자세들 중 하나를 선택하면, 시스템은 해당 자세에 필요한 센서 부착 개수와 위치를 알림
- 사용자가 센서를 부착 후 재활 운동을 수행시, 시스템은 수신된 **IMU** 데이터와 미리 학습된 모델을 이용해 사용자의 자세를 인식하여 행동 교정 등 피드백

### 3) 요구 조건에 따른 신규 제약 사항

#### A. 공개 데이터셋의 부족

기존의 **IMU**를 이용한 연구를 살펴보면 모델 학습에 이용할만한 공개 데이터셋을 찾아보았으나, 대부분의 데이터셋이 신체 일부의 특정 자세를 추론하기보단 전신 **pose**를 추론하는 데이터셋이었다. 또한 특정 재활 자세를 추론한 연구의 경우, 자체적으로 피험자를 모집한 후 데이터셋을 얻어 연구한 경우가 많았다. 따라서 재활운동 자세를 따로 선정한 뒤 자세 추론을 위해, 그 자세에 맞는 데이터셋을 직접 제작하기로 하였다.

#### B. 데이터셋을 직접 제작할 경우 생기는 문제

사용자가 목표한 자세를 달성했을때, **ground truth** 또한 수집되어야 올바른 모델 학습을 수행할 수 있다. 이에 따라, 데이터셋을 수집시 실제 동작을 수행하는 사람 옆에서 올바른 자세에 도달했는지 검사할 검사자가 필요하다. 이 검사자는 사용자가 지금 자세를 달성하고 있는지, 아닌지에 대해서만 체크를 하고 이를 시스템에

알린다. 또한 관절의 움직임에 따른 실시간 관절 각도가 필요한 경우 **kinect**와 같이 비전을 이용해 각을 얻어내거나 디지털 각도계와 같은 센서를 활용한다.

## 2. 설계 상세화 및 변경내역

### 1) IMU 데이터 수집

기존에 **BLE** 통신을 구현하고 **IMU** 센서로부터 데이터를 받아올 **MCU**로 **Nordic Semiconductor** 사의 **nRF52832** 모듈을 이용할 계획이었으나, 개발보드를 구하기 어렵게 되어 **ESP32-C3** 기반의 개발 보드와 **MPU9250** 9축(가속도, 자이로, 지자기) **IMU**센서를 이용하였다. 이후 감바랩스에서 6축(가속도, 자이로) **IMU**센서인 **ICM-42670-P**가 내장된 **ESP32-S3** 개발보드를 10개 제공받아 이를 이용해 과제를 수행하게 되었다.

내장된 **IMU**센서는 **ESP32-S3**와 **I2C** 통신을 할 수 있도록 **SCL**, **SDA**선이 연결되어 있다. **ESP32**에서는 정해진 시간(현재 **25ms**)마다 **I2C**를 통해 센서로부터 6개의 **IMU** 데이터(**x,y,z** 축 각각의 가속도, 자이로)를 가져오게 하였다.

### 2) IMU 데이터를 게이트웨이에 전송

다양한 신체부위에 부착될 **ESP32** 개발보드들은 **BLE**를 이용해 게이트웨이로 **IMU** 데이터를 전송하게 된다. 이를 위해 게이트웨이는 다수의 보드 연결을 관리할 수 있어야 하고 동시에 들어오는 센싱 값을 처리할 수 있어야 한다. 또한 같은 시간에 측정된 **IMU** 데이터를 이용해야 정확한 모델 학습과 추론이 가능하므로 게이트웨이에 전송되는 데이터의 **sync**를 맞출 수 있어야 한다.

게이트웨이에서는 파이썬에서 **BLE**를 지원하는 **bleak** 모듈을 통해 개발 보드와 무선 연결을 수립하였다. 또한 비동기 라이브러리인 **asyncio**를 이용해 오래 걸리는 작업을 순차적으로 수행하지 않고 동시에 수행하며 보드를 제어하게 하였다.

각자 개발 보드에서는 **IMU** 데이터와 더불어 자신이 어떤 보드인지에 대한 정보와, 현재 **timestamp** 정보를 게이트웨이에 보내게 하였다. 이때 **timestamp**는 새롭게 **IMU** 데이터를 가져올 때마다 증가하는 시간 값인데, 게이트웨이에서 연결된 모든 보드의

timestamp를 동시에 초기화할 수 있다. 게이트웨이는 이 방식으로 모든 보드의 측정 시작 시간을 같게 해 보드들 간 데이터의 sync를 맞춘다.

### 3) 머신러닝 모델 학습 수행

게이트웨이에서 보드로부터 IMU 데이터를 sync에 맞게 잘 수신하였다면, 이것을 모델 학습에 사용해야 한다. 이때, 재활 운동 자세마다 추론 방법과 학습 모델이 달라진다. 다음과 같이 두 가지 경우로 나누어 볼 수 있다.

- A. 무릎 굽히기와 같이 적은 수의 관절이 동반되는 운동의 경우
- B. 스쿼트, 브릿지 자세와 같이 전신을 움직이는 운동의 경우

A와 같은 경우는 관절 각도를 추론해 사용자가 목표 자세를 달성하는지 확인할 수 있도록 한다. 예시로, 무릎을 굽혔다 펴는 경우에 실시간으로 각도를 알려 주면서, 어느정도 더 피거나 굽혀야 목표 자세에 도달하는지 알 수 있게 된다. B와 같은 경우는 목표하는 자세를 달성했는지, 하지 못했는지만 추론하면 된다.

이때, ground truth 또한 수집되어야 모델 학습을 수행할 수 있다. B와 같은 경우는 데이터셋을 수집하는 사람 옆에서 자세를 검사할 검사자가 필요하다. 이 검사자는 사용자가 자세를 달성했는지, 달성하지 못했는지에 대해서만 체크를 하면 되는데, 이 정보는 ESP32 보드를 하나 더 사용해 BLE 통신으로 넘겨주면 된다. A의 경우 관절의 움직임에 따른 실시간 관절 각도가 필요한데 kinect와 같이 비전을 이용해 각을 얻어내거나 디지털 각도계와 같은 센서를 활용할 수 있다.

재활 운동 자세의 연속성, 데이터셋의 가변성을 고려하여 각 동작마다 가장 잘 어울리는 학습 모델(SVM, LSTM 등등)의 선정 또한 필요하다.

### 4) 재활운동 자세별 모듈 개수 최적화

선정된 재활운동 동작별로, 학습 모델이 자세를 올바르게 추론할 수 있는 최소의 센서 모듈의 개수와 적정 위치를 찾는다.

### 3. 갱신된 과제 추진 내역

세부 항목	8		9					10		
	4	5	1	2	3	4	5	1	2	3
데이터셋 제작										
모델학습										
서비스 구현										
서비스 최적화										
디버깅										
최종 보고서 발표준비										

### 4. 구성원별 진척도

이름	역할
하규승	1. 측정 모듈-게이트웨이간 BLE 통신 구현(완료) 2. 게이트웨이 - 유저간 Feedback 구현
김지훈	1. 센서값 수신 및 데이터 전처리(완료) 2. 수신값 sync 구현(완료)
공통	1. 관련자료 조사 및 문서 작성(완료) 2. 데이터셋 수집 및 가공 3. 모델 학습 및 평가 4. 서비스 구현 5. 최적화 및 디버깅

## 5. 보고 시점까지의 과제 수행내용 및 중간결과

### 1) 자세 선정

처음엔 전반적인 HPE를 주제로 잡았으나, 범위가 너무 넓은 주제에다 데이터셋마다 센서 부착 위치가 다 다른 등 애로사항이 있었다. 따라서 특정 재활운동 자세를 지정하고, 이에 맞게 데이터셋을 수집하고 모델학습을 해보기로 했다.

선정한 재활운동은 총 6가지 종류로, 다음과 같다.

상지 : 팔 교차 근력 강화

하지 : 앉아서 다리 올리기(대퇴사두근 강화), 다리 들어올려 유지하기(단하지 거상)

경추 : 머리 측면으로 굽히기(측방굴곡)

요추 : 엉덩이 들어 올리기(브릿지 자세) / 누워서 한쪽 다리 넘기기

### 2) 센싱 데이터 수신 & 모델 학습

#### A. 개발보드 변경 이전

두 쌍의 ESP32C3기반 개발보드와 MPU9250 IMU를 이용해 과제를 진행하였다. 먼저 ESP32와 IMU를 브레드보드에 끼운 뒤 I2C 통신이 가능하게 점퍼 선을 연결하였고, 유선 Serial 통신을 통해 IMU 데이터가 잘 수집되는지 테스트하였다.

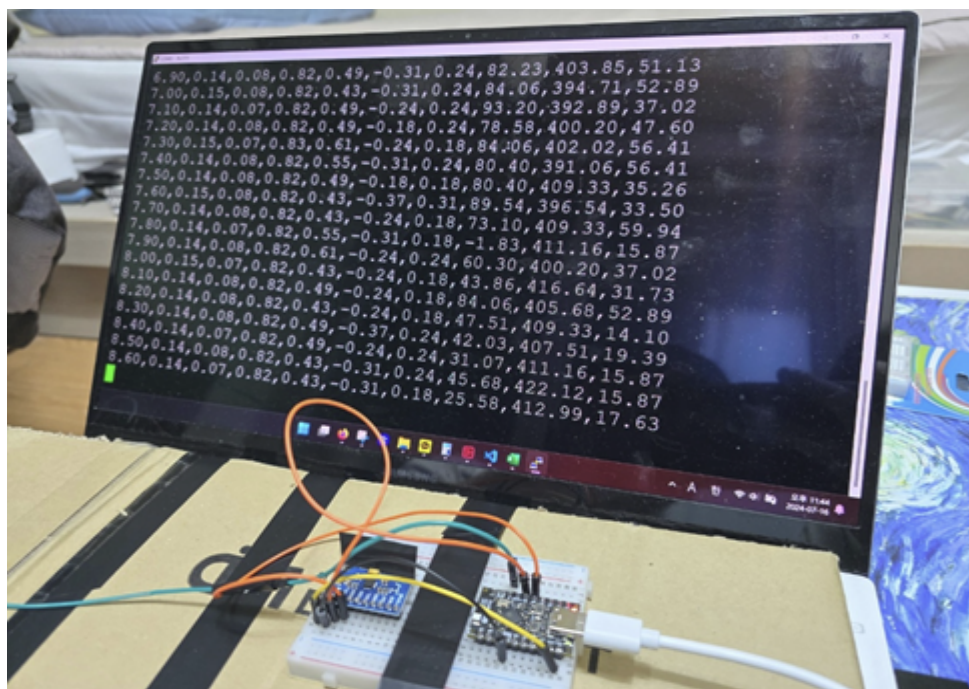


그림 1. 하나의 보드에서 시리얼 통신으로 IMU 센싱 값을 수신

다른 한쌍의 보드에서도 같은 작업을 거친다.

다음으로, 아직 인체에 부착하기에는 어려운 점이 많아 팔의 상부와 하부의 움직임을 단순하게 구현한 팔꿈치 관절 모형을 박스로 제작한 뒤, 두 센서를 상부와 하부에 부착하여 박스를 움직여가며 데이터셋을 생성하였다. 이때 IMU 값은 0.1초마다, 즉 10Hz의 샘플링 속도로 데이터를 수집하였으며 따라서 4분 동안 2400개의 관절각-IMU 센싱 데이터를 얻을 수 있었다.

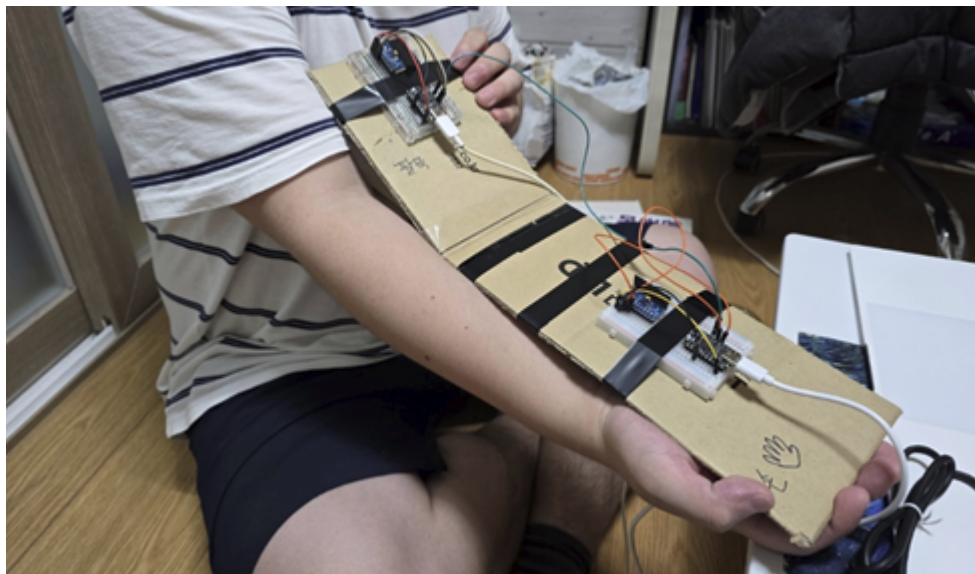


그림 2. 박스로 제작한 팔꿈치 관절 모형에 IMU와 MCU 장착

scikit-learn에서 Ridge regression을 통해 모델을 학습시켜 보았고, 결과는 다음 그림과 같이 훈련 데이터셋에 대해 0.9923, 테스트 데이터셋에 대해 0.9897의 정확도를 보였음을 확인할 수 있다. (전처리로 입력 데이터를 PolynomialFeatures를 이용해 다항화하였고, 따라서 모델의 coefs가 입력 데이터의 특성 수인 18개보다 더 많이 나오게 되었다)



```
PS D:\대학\졸과\중간>
PS D:\대학\졸과\중간> d:; cd 'd:\대학\졸과\중간'; & 'c:\Users\jihoon\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\jihoon\.vscode\extensions\ms-python.debugpy-2024.8.0-win32-x64\bundle\lib\debugpy\adapter\..\..\debugpy\launcher' '52729' '--' 'd:\대학\졸과\중간\DegreeConv.py'
input data size : (2404, 18)
target data size : (2404, 1)
train score : 0.9923003111680782
test score : 0.9897850696635325
coefs : [[-1.07402418e-02  5.12710410e-03  1.24683834e-02  6.24106315e-02
-1.48797396e-01 -4.29377335e-02  6.13723107e-02 -4.59334366e-02
-5.41160770e-02 -1.12593922e-02 -9.63256302e-03  2.42135062e-02
 9.54423411e-03  1.49847606e-01  2.59269335e-02 -7.95149682e-02
-1.85371544e-01 -9.38520533e-02 -1.49432167e-01  6.39696334e-02
 2.49887020e-02  1.05635536e-01 -9.77070358e-02  9.22289945e-02
 3.49531208e-02 -3.62768617e-02 -3.60601367e-02 -1.47369626e-01
-4.14253560e-02 -4.23450361e-02  1.69336793e-02  9.10802617e-04
-2.45181449e-02 -9.17102838e-03  1.83077653e-02  5.08126063e-04
```

그림 3. 팔꿈치 관절 모형에 대한 데이터셋 학습 결과

마지막으로 실제 팔꿈치모형을 굽혀보면서 각도 추론이 잘 이루어지고 있는지 아래와 같이 확인해 보았다.

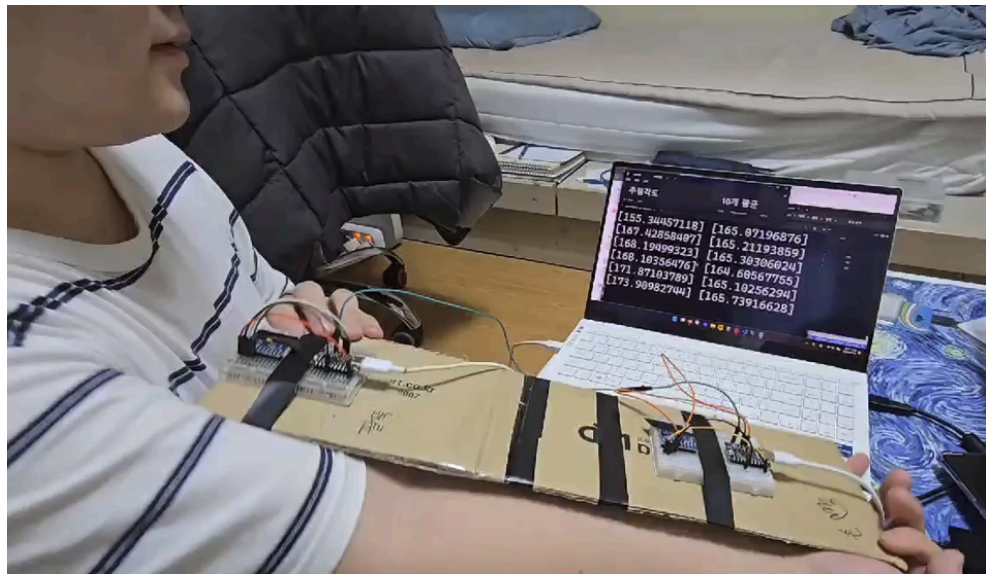


그림 4. 팔꿈치 관절 모형의 실시간 각도 추론

추론은 어느 정도 잘 이루어졌으나, 학습 모델이 선형이고 학습에 사용한 데이터셋도 현저하게 작아 정확한 추론은 이루어지지 않았다. 실제로 IMU 센서의 지자기 데이터 특성에 대한 **weight**가 꽤 컸는지, 바닥에서 수직 방향 축을 기준으로 회전하니 추론 각도가 들쭉날쭉 바뀌었다.

## B. 개발보드 변경 이후

새로운 임베디드 보드를 감바랩스에서 제공받아 과제를 진행하였다.

ESP32S3기반 개발보드이고, 기판에 6축 IMU 센서가 내장되어 있어 따로 브레드보드에 IMU 모듈을 연결할 필요 없이 보드 하나로 센싱과 BLE 통신이 가능하게 되었다.

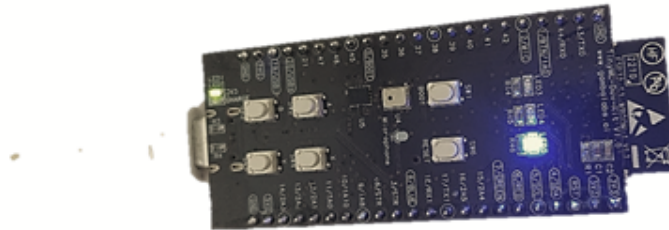


그림 5. ESP32-S3 기반 개발 보드

또한, 게이트웨이(PC)에서 BLE 통신으로 여러 보드의 연결을 동시에 제어할 수 있도록 파이썬의 **bleak**, **asyncio** 라이브러리를 이용해 데이터 수신 프로그램을 제작하였다.

```
>>>scan
센서 검색 중..
현재 센서 목록
  Name=A Address=7C:DF:A1:EA:7C:4E ONLINE
  Name=B Address=7C:DF:A1:EA:45:F2 ONLINE
  Name=C Address=7C:DF:A1:EA:42:F6 offline
  Name=D Address=7C:DF:A1:EA:60:AE offline
  Name=E Address=7C:DF:A1:EA:44:5E ONLINE
  Name=F Address=7C:DF:A1:EA:60:5E offline
  Name=G Address=7C:DF:A1:EA:7C:62 ONLINE
  Name=H Address=7C:DF:A1:EA:42:FA ONLINE

>>>scan
센서 검색 중..
현재 센서 목록
  Name=A Address=7C:DF:A1:EA:7C:4E ONLINE
  Name=B Address=7C:DF:A1:EA:45:F2 ONLINE
  Name=C Address=7C:DF:A1:EA:42:F6 ONLINE
  Name=D Address=7C:DF:A1:EA:60:AE ONLINE
  Name=E Address=7C:DF:A1:EA:44:5E ONLINE
  Name=F Address=7C:DF:A1:EA:60:5E ONLINE
  Name=G Address=7C:DF:A1:EA:7C:62 ONLINE
  Name=H Address=7C:DF:A1:EA:42:FA ONLINE
```

그림 6. 게이트웨이(PC)에서 BLE를 이용해 사용 가능한 보드(장치) 스캔

```
>>get
IMU 데이터를 받을 센서 이름을 공백으로 구분하여 입력
>>A B C D E F G H
측정 시간을 입력(단위:s)
>>3
데이터 수신 중 실시간 feedback? (y/N)
>>y
센서와 연결 시작
    센서 연결됨:Name=A      Address=7C:DF:A1:EA:7C:4E
    센서 연결됨:Name=B      Address=7C:DF:A1:EA:45:F2
    센서 연결됨:Name=C      Address=7C:DF:A1:EA:42:F6
    센서 연결됨:Name=D      Address=7C:DF:A1:EA:60:AE
    센서 연결됨:Name=E      Address=7C:DF:A1:EA:44:5E
    센서 연결됨:Name=F      Address=7C:DF:A1:EA:60:5E
    센서 연결됨:Name=G      Address=7C:DF:A1:EA:7C:62
    센서 연결됨:Name=H      Address=7C:DF:A1:EA:42:FA
```

그림 7. 다수의 센서를 선택해 연결한 후 IMU 데이터 수신

이제 보드를 여러 개 연결하여, BLE를 통해 IMU 값을 가져올 수 있게 되었다. 이미 선정된 재활운동 자세와는 관계가 많이 없지만, 딥러닝 모델 학습을 익히고 학습의 정확도를 알아보기 위해 보드 변경 이전과 같이 팔꿈치 관절에 대해 관절 각도 추론을 위한 학습을 간단히 수행해보았다. 구성한 모델은 다음과 같이 LSTM을 이용한 것 2개와 Fully connected layer만을 이용한 것 하나이다.

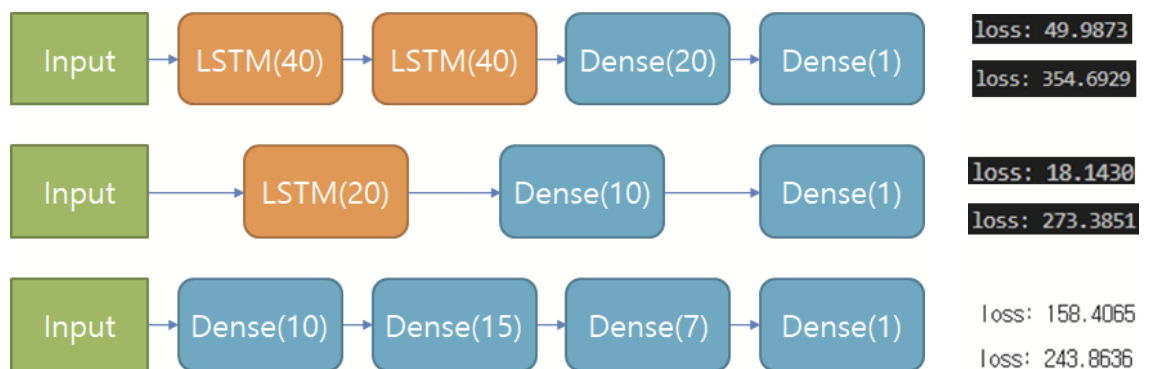


그림 8. 좌) 사용한 3가지 딥러닝 모델의 구조. 우) 모델마다의 train, validation loss

입력 데이터를 train set과 validation set으로 나누어 학습을 진행하였고, 우측에 MSE loss값을 확인할 수 있다. (위:train loss, 아래:validation loss)

validation set의 loss를 확인하였을 때 300 전후로 정도로 나타났는데, MSE(Mean Squared Error)인것을 감안하면 실제 각도 오차는 최대 20도 가량 나타나는 것을 알 수 있다. 이는 데이터셋을 수집할 당시에 아직 보드에 필요한 케이스나 배터리가 있지 않아 유선으로 전원을 공급하면서 각도를 측정하였고, 이런 제한적인 환경에서 데이터를 수집하기 어려워 짧은 시간에 데이터 수집을 끝냈기 때문이다. 따라서 딥러닝 모델을 학습시킬 때 사용한 데이터셋이 질적/양적으로 너무 빈약해 상당히 큰 각도 오차를 얻게 되었다.

이후 리튬폴리머 배터리가 도착하게 되면, 케이스를 제작해 개발 보드를 완전히 무선으로 사용하며 데이터셋을 만들고 이를 여러 모델에 학습시킬 예정이다.