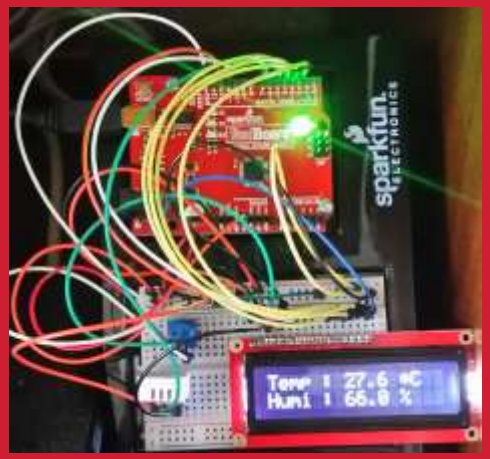
Arduino-IOT

[wk15]



nano33 BLE sensor IMU Project

Visualization of Signals using Arduino, Node.js & storing signals in MongoDB & mining iot data using Python

Drone-IoT-Comsi, INJE University

2nd semester, 2022

Email : chaos21c@gmail.com

My ID

ID를 확인하고 github에 repo 만들기

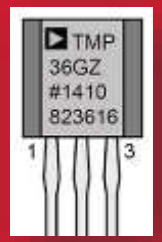
AA01	강대진	AA13	박제홍
		AA14	심준혁
AA03	김성우	AA15	이상혁
AA04	김정헌	AA16	이승무
		AA17	이승준
AA06	김창연	AA18	이준희
AA07	김창욱	AA19	이현준
AA08	김태화	AA20	임태형
AA09	남승현	AA21	정동현
AA10	류재환		
AA11	박세훈	AA23	정희서
AA12	박신영	AA24	최재형

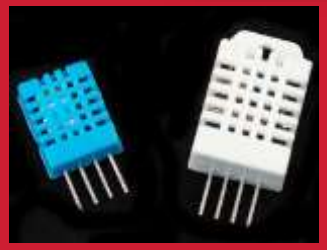
위의 id를 이용해서 github에 repo를 만드시오.

Option: 아두이노응용 실습 과제 – AAnn

Public, README.md check

[Practice]





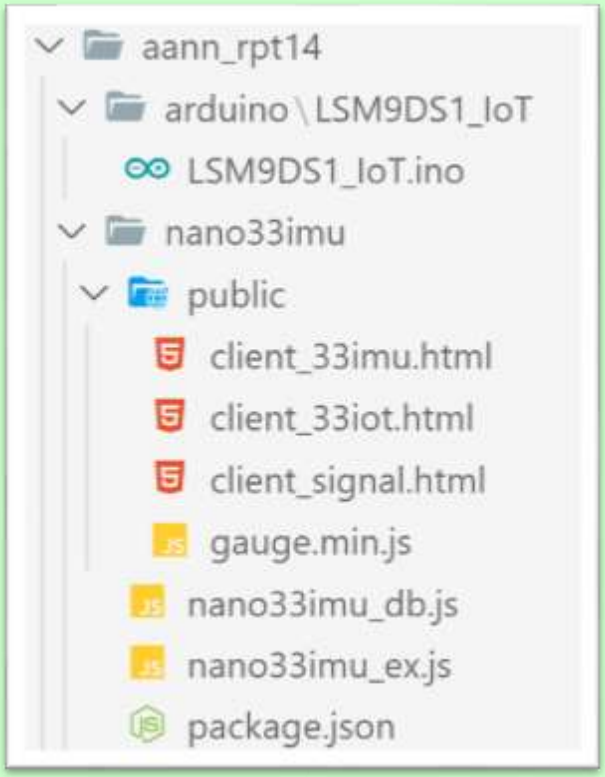
- [wk14]
  - IoT Project: nano33ble
  - Multi-sensor circuits : IMU
  - Complete your project
  - Upload folder: aann-rpt14
  - Use repo “aann” in github

wk14 : Practice : aann-rpt14

- **[Target of this week]**
  - **Complete your works**
  - **Save your outcomes and upload outputs in github**

제출폴더명 : **aann-rpt14**

- 제출할 파일들



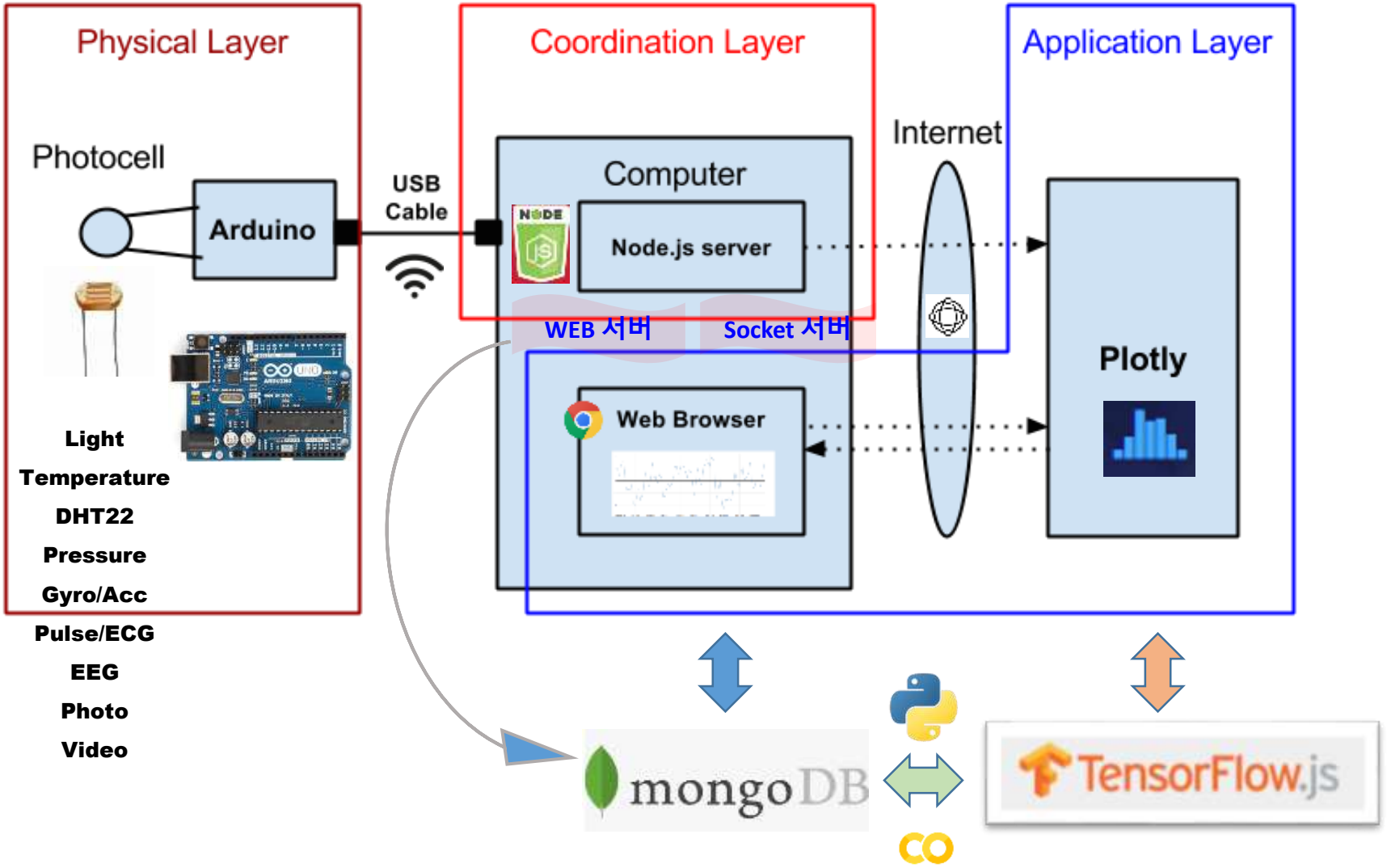
Purpose of AA

주요 수업 목표는 다음과 같다.

1. Node.js를 이용한 아두이노 센서 신호 처리
2. Plotly.js를 이용한 아두이노 센서 신호 시각화
3. MongoDB에 아두이노 센서 데이터 저장 및 처리

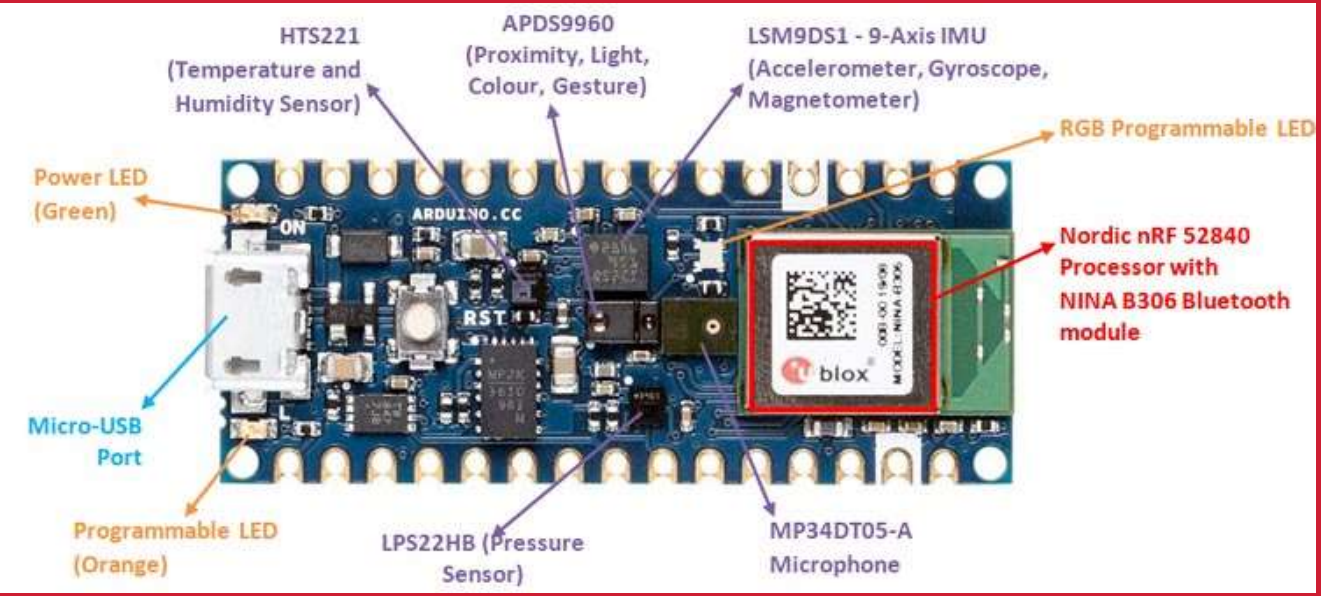
4. 저장된 IoT 데이터의 마이닝 (파이썬 코딩)

Layout [H S C]

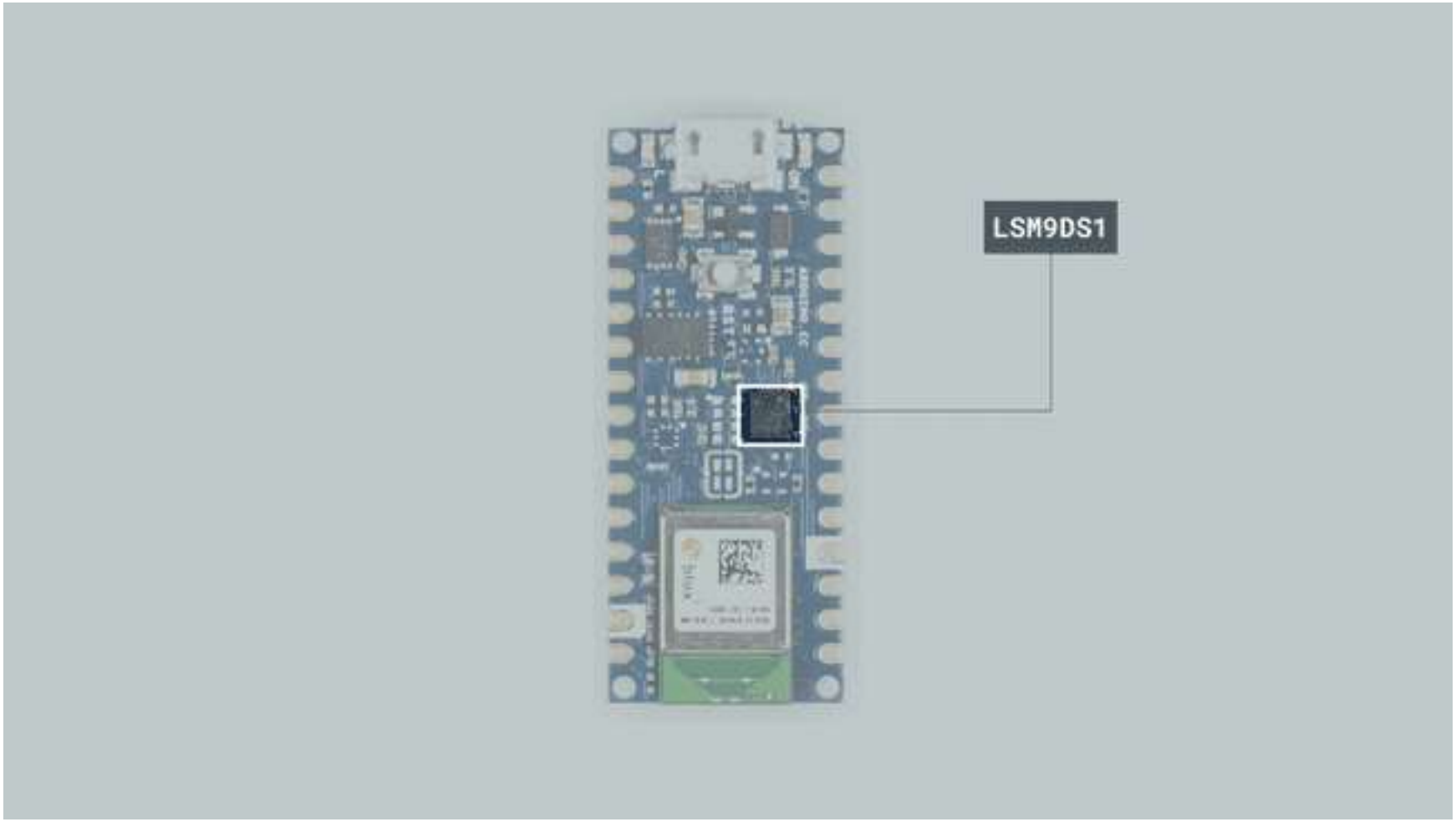

Physical Layer
Photocell
Arduino
USB Cable
Light
Temperature
DHT22
Pressure
Gyro/Acc
Pulse/ECG
EEG
Photo
Video
Coordination Layer
Computer
Node.js server
WEB 서버
Socket 서버
Web Browser
Internet
Application Layer
Plotly
mongoDB
TensorFlow.js

2021/22 AA project

nano33BLE sensor



LSM9DS1, 9축 IMU센서: acc, gyro, mag



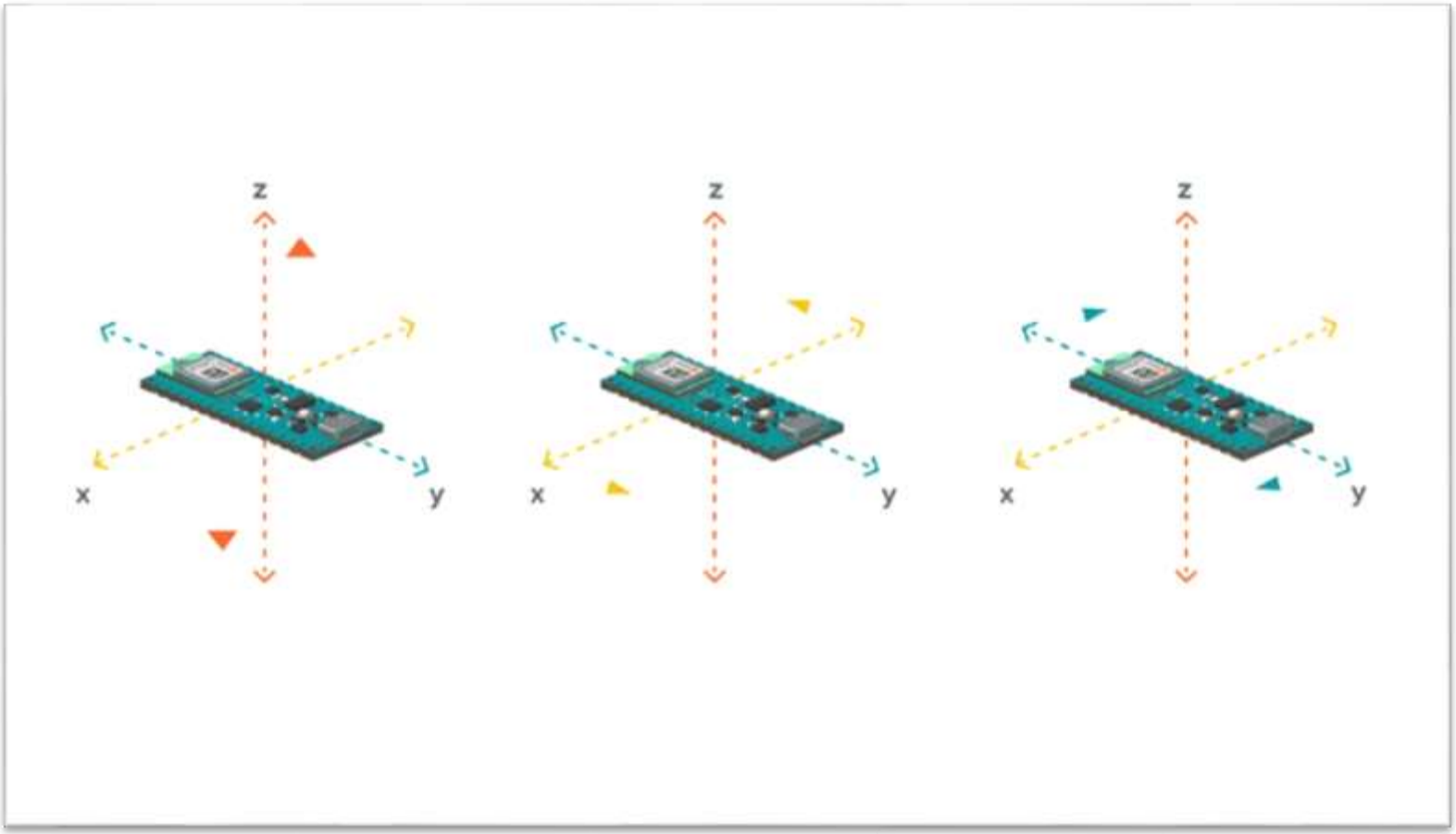
LSM9DS1, 9축 IMU센서: acc, gyro, mag

The LSM9DS1 Library

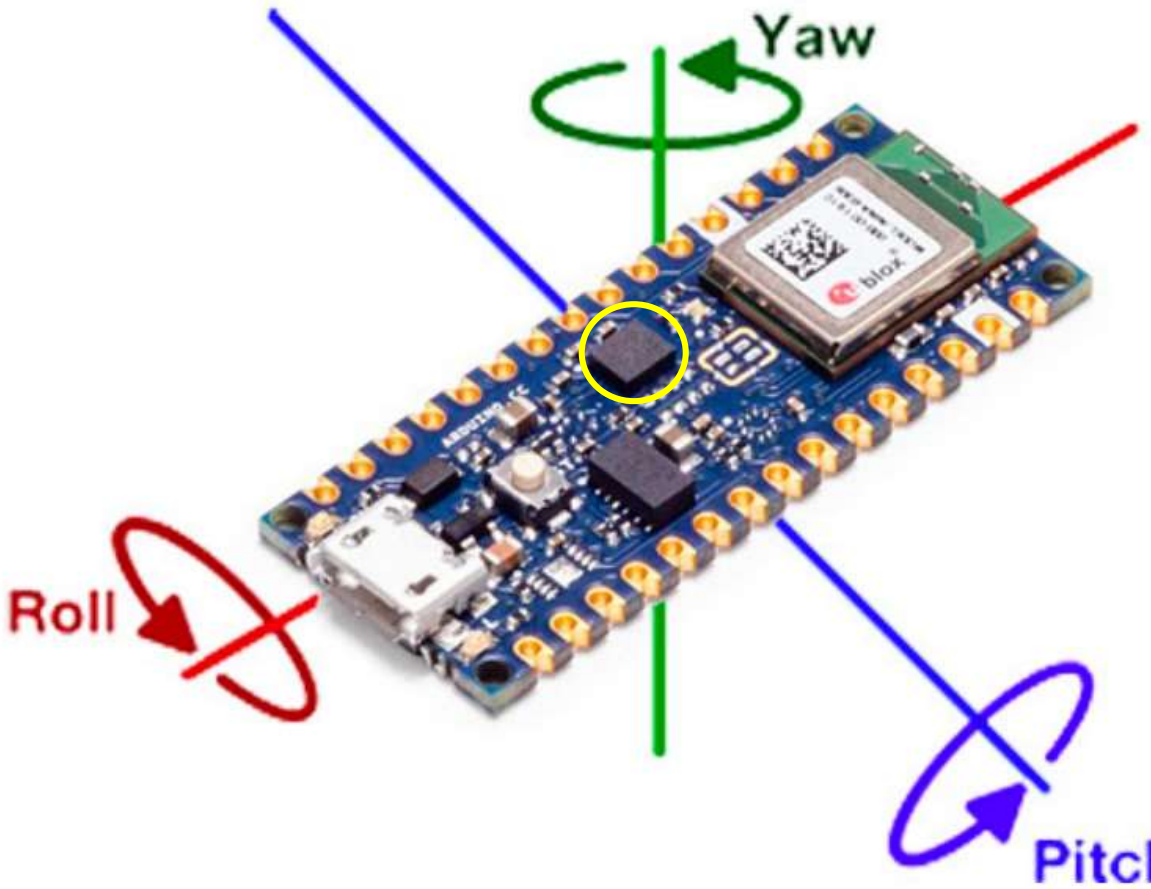
The Arduino LSM9DS1 library allows us to use the Arduino Nano 33 BLE IMU module without having to go into complicated programming. The library takes care of the sensor initialization and sets its values as follows:

- **Accelerometer** range is set at [-4, +4]g -/+0.122 mg.
- **Gyroscope** range is set at [-2000, +2000] dps +/-70 mdps.
- **Magnetometer** range is set at [-400, +400] uT +/-0.014 uT.
- **Accelerometer** output data rate is fixed at 104 Hz.
- **Gyroscope** output data rate is fixed at 104 Hz.
- **Magnetometer** output data rate is fixed at 20 Hz.

LSM9DS1, 9축 IMU센서: acc , gyro, mag



LSM9DS1, 9축 IMU센서: acc, gyro, mag



자이로 스코프 측정, 이미지 출처 https://www.mauroalfieri.it/elettronica/arduino-nano-33-ble-giroscopio-lsm9ds1.html

https://fishpoint.tistory.com/4438

LSM9DS1, 9축 IMU센서: acc, gyro, mag

LSM9DS1_IoT.ino

```
1  #include <Arduino_LSM9DS1.h>
2
3  void setup() {
4    Serial.begin(9600);    // 19200
5    while (!Serial);
6
7    if (!IMU.begin()) { // IMU센서를 초기화합니다. 초기화중 문제가 발생하면 오류를 발생시킵니다.
8      Serial.println("Failed to initialize IMU!");
9      while (1);
10   }
11
12 }
13
```

LSM9DS1, 9축 IMU센서: acc, gyro, mag

```
14  float aX, aY, aZ, gX, gY, gZ, mX, mY, mZ;
15
16  void loop() {
17
18    delay(500);
19    // 가속도, 자이로, 지자기 센서의 값이 모두 정상 출력되면 데이터 수집 시작.
20    if (IMU.accelerationAvailable() && IMU.gyroscopeAvailable() && IMU.magneticFieldAvailable()) {
21      // read the acceleration, gyroscope, and magnetic data
22      IMU.readAcceleration(aX, aY, aZ);
23      IMU.readGyroscope(gX, gY, gZ);
24      IMU.readMagneticField(mX, mY, mZ);
25
26      Serial.print("AA00,");   // Change to your ID
27      Serial.print(aX);
28      Serial.print(',');
29      Serial.print(aY);
30      Serial.print(',');
31      Serial.print(aZ);
32      Serial.print(',');
33      Serial.print(gX);
34      Serial.print(',');
35      Serial.print(gY);
36      Serial.print(',');
37      Serial.print(gZ);
38      Serial.print(',');
39      Serial.print(mX);
40      Serial.print(',');
41      Serial.print(mY);
42      Serial.print(',');
43      Serial.println(mZ);
44    }
45  }
```

LSM9DS1, 9축 IMU센서: acc, gyro, mag

IoT 데이터 수집 형태로 출력을 변경하시오.

ax,ay,az,gx,gy,gz,mx,my,mz

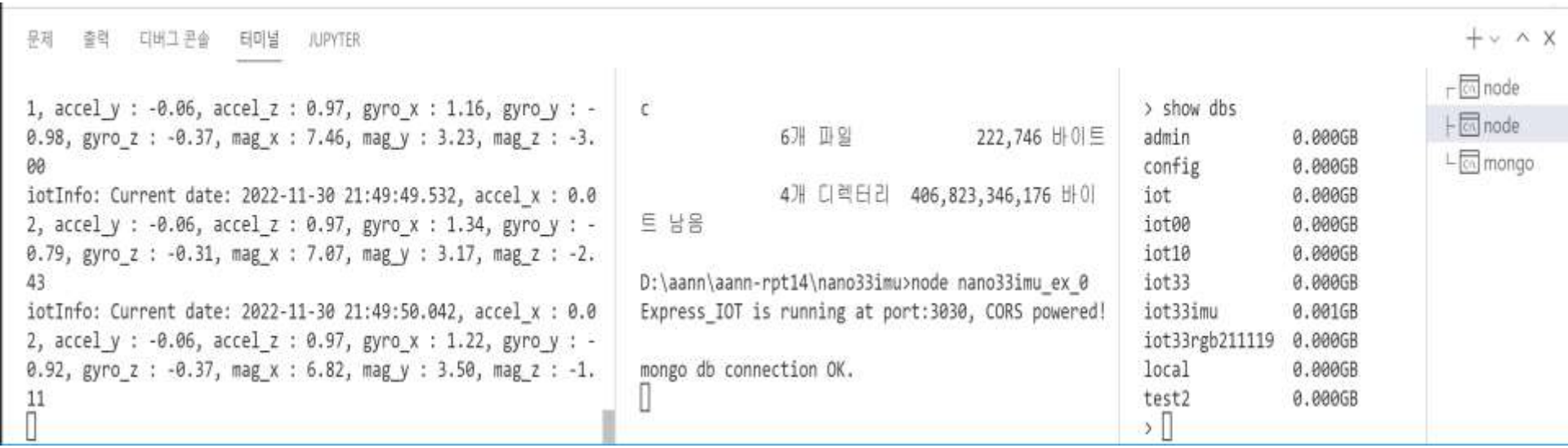
```
0.17,-0.60,1.43,17.70,134.40,-209.05,12.52,12.59,0.34
0.53,0.40,0.88,-52.73,-38.51,-230.77,16.10,11.32,3.06
0.91,1.58,1.04,-62.19,-116.94,-66.41,18.52,11.22,8.23
-0.18,0.82,-0.29,77.21,-88.44,190.61,16.74,11.05,8.46
-0.55,-1.39,0.99,5.43,-71.72,180.30,14.15,16.38,2.43
-0.39,-0.68,0.75,-19.96,-192.44,-118.53,14.09,13.23,4.27
0.31,0.30,-0.75,-47.55,10.56,-24.17,16.70,-1.49,29.46
-0.75,-0.89,-0.06,65.55,-186.58,-158.57,6.42,20.53,2.88
-0.59,0.44,0.48,-338.13,159.85,-424.26,8.85,-13.66,15.32
0.75,0.37,0.46,288.39,-52.19,-362.00,20.43,7.52,11.69
0.47,0.77,1.98,-304.93,353.21,-110.60,8.02,-2.36,-4.66
-0.24,2.32,-0.79,192.26,-132.08,-271.85,22.35,-6.70,18.77
-0.65,-0.02,0.51,-0.79,-47.85,-60.00,8.63,9.52,0.02
0.30,-0.40,1.47,35.83,-155.82,-67.26,10.94,10.00,-3.55
-0.75,-0.39,3.63,-34.73,-219.54,6.53,9.24,11.16,-3.72
0.61,0.08,3.10,-125.43,118.04,-36.44,11.18,14.27,-1.40
-0.74,-0.47,0.50,688.78,-650.76,-144.29,19.58,-2.94,17.32
-0.12,-1.01,0.68,-319.52,398.25,761.60,13.33,19.10,2.84
0.13,-0.12,1.05,5.31,-0.92,13.18,15.00,8.51,-0.85
```

LSM9DS1, 9축 IMU센서: acc, gyro, mag

* 9축 IMU 센서 신호 마이닝

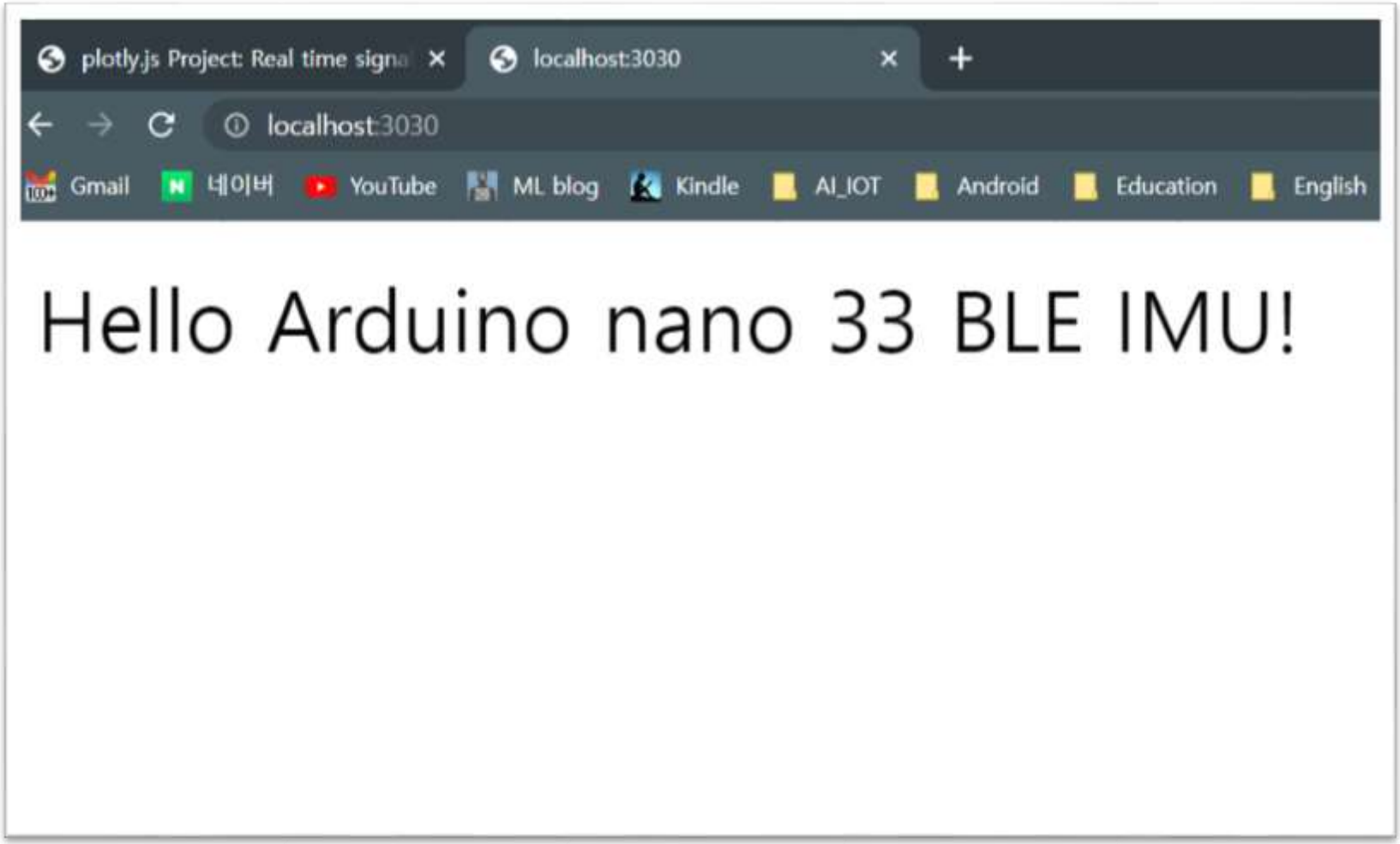
1. MongoDB
2. Express server
   - 실시간 모니터링
   - DB 모니터링
3. Gauges: mACC, mGYRO, mMAG
4. data mining using Colab
5. Deep learning ?

LSM9DS1, 9축 IMU센서: All servers



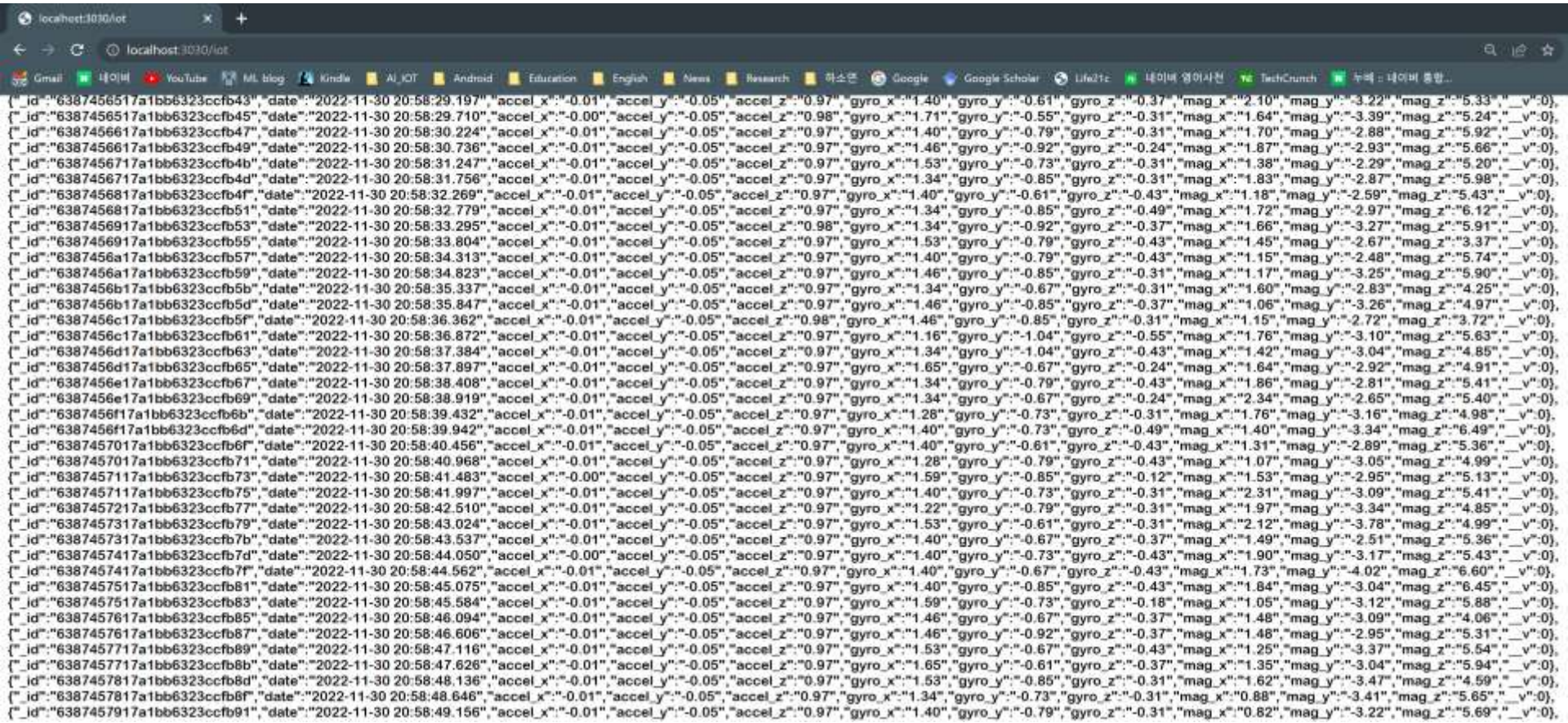
LSM9DS1, 9축 IMU센서:

express server routing: localhost:3030



LSM9DS1, 9축 IMU센서:

express server routing: localhost:3030/iot



LSM9DS1, 9축 IMU센서:
localhost:3030/client_signal.html

IoT Signals from nano33ble IMU sensor

Real-time Signals

on Time: 2022-12-14 21:18:22.835

IMU signals (ax,ay,az),(gx,gy,gz),(mx,my,mx) : (0.03,-0.01,0.99) , (-0.31,-4.03,-0.55) , (-19.68,22.90,-39.88)

RMS of IMU signals (ma, mg, mm) : 0.99,4.08,50.02

LSM9DS1, 9축 IMU센서:

localhost:3030/client_33imu.html

LSM9DS1, 9축 IMU센서:

localhost:3030/client_33imu_gauge.html

LSM9DS1, 9축 IMU센서:

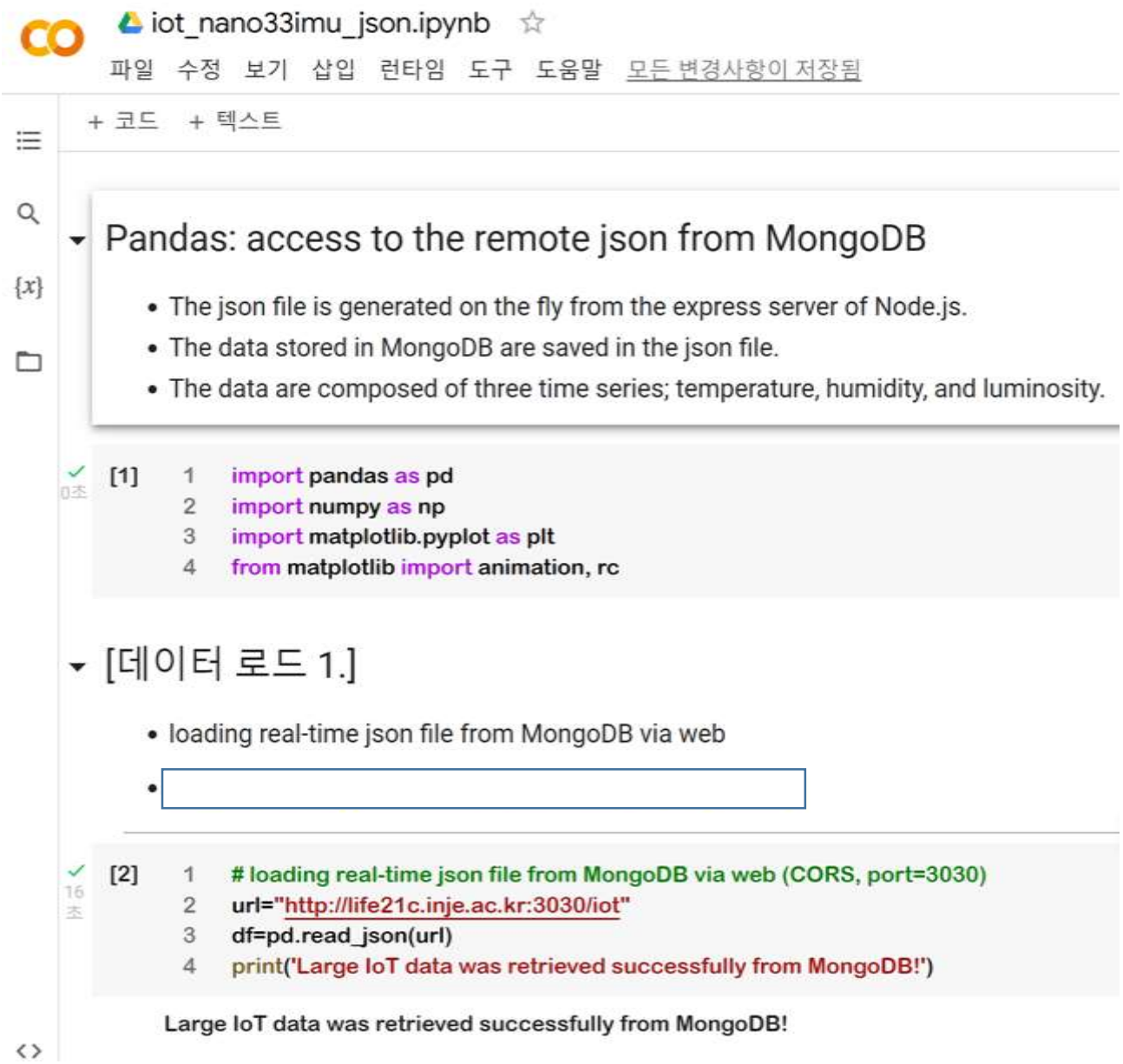
localhost:3030/client_33imu_gauge.html

Real-time IMU from nano 33 BLE sensor

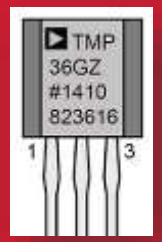
on Time: 2022-12-15 09:42:44.434

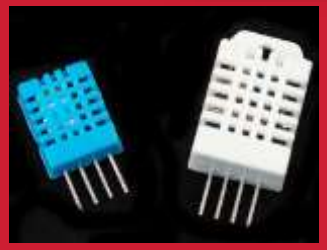
LSM9DS1, 9축 IMU센서: data mining

iot_nano33imu_json.ipynb



[Practice]





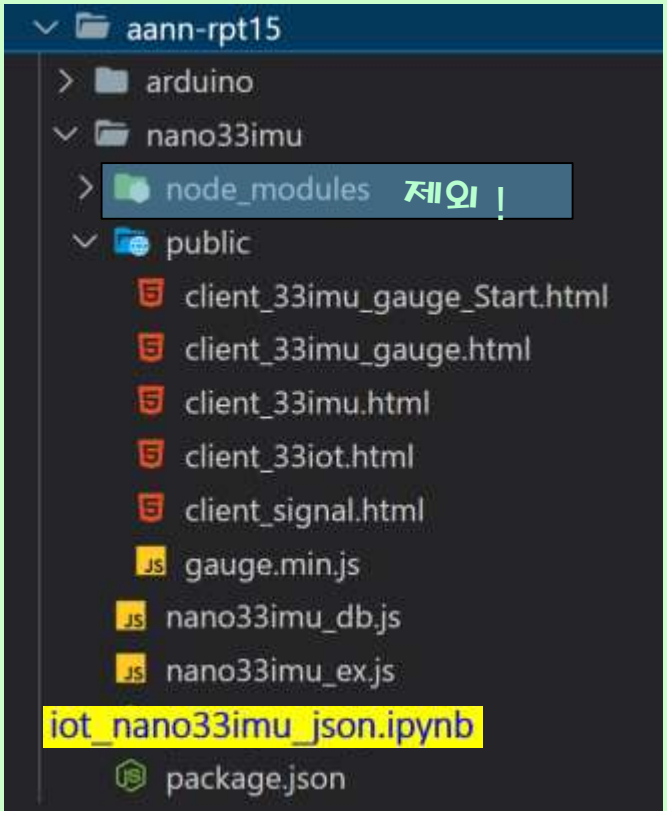
- [wk15]
  - IoT Project: nano33ble
  - Multi-sensor circuits : IMU
  - Complete your project
  - Upload folder: aann-rpt15
  - Use repo “aann” in github

wk15 : Practice : aann-rpt15

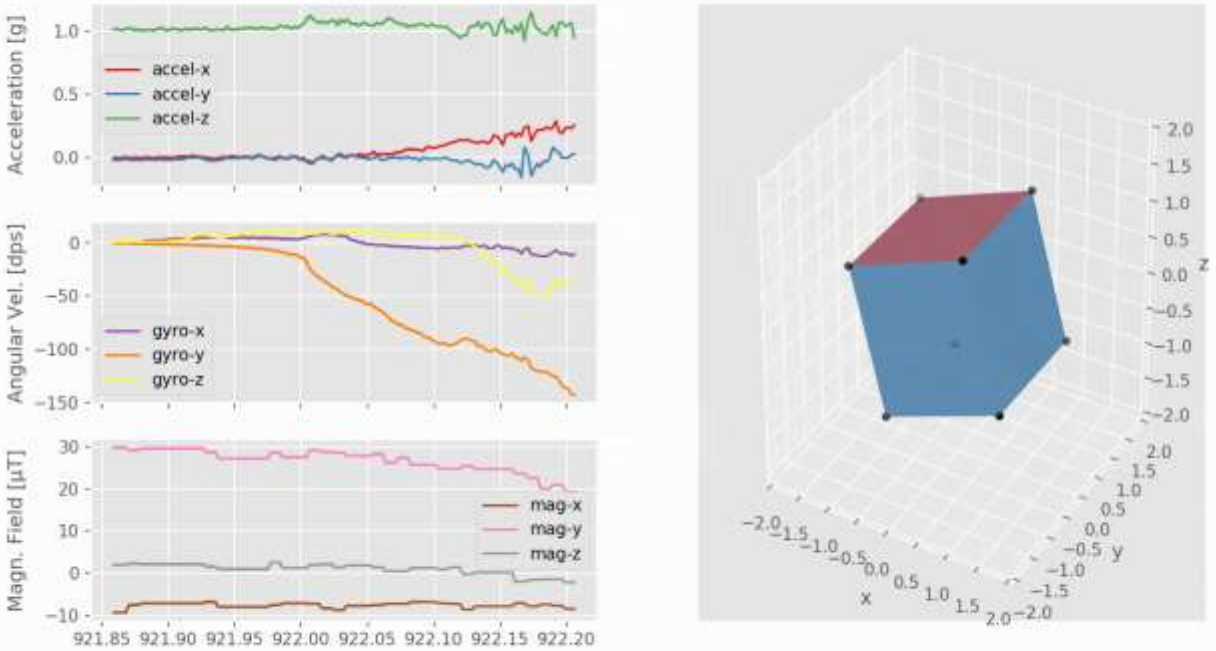
- **[Target of this week]**
  - **Complete your works**
  - **Save your outcomes and upload outputs in github**

제출폴더명 : **aann-rpt15**

- 제출할 파일들



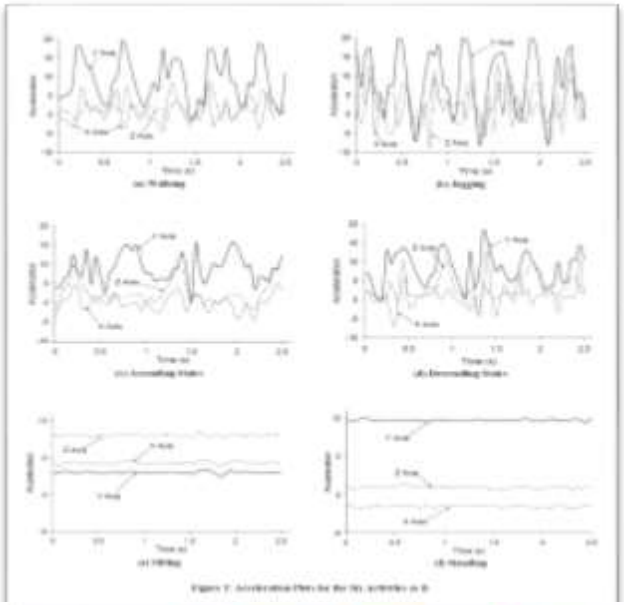
모션 인식(9-축 IMU)

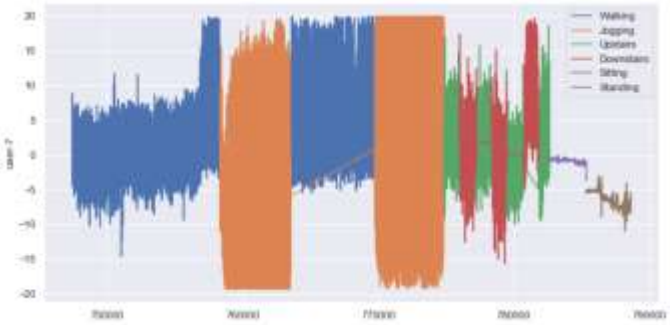


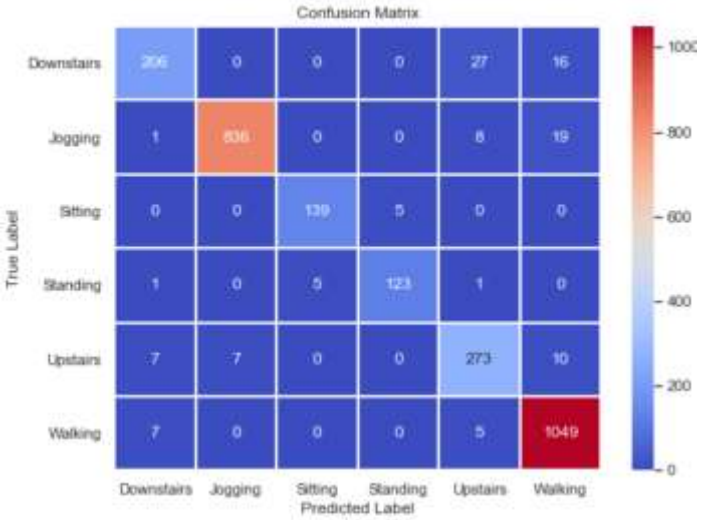
https://images.squarespace-cdn.com/content/v1/59b037304c0dbfb092fbe894/1573836927118-IS5CS61OW9XH9HSRCMA1/ke17ZwdGBTodd8pDm48kGbFogdxZzB1B7PQq3zm9xl7gQa3H78H3Y0txjaiv_0fDoOvxcdMmMKkDsyUqMSsMWxHk725yiiHCCLfrh8O1z5QPOohDIaIeljMHgDF5CVlOqpeNLcJ80NK65_fV7S1UQupMlr7Z9cq9PZkRYtzEu3SbZmkCxOjksrEup4_K2kPH3bqxw7fF48mhrq5Ulr0Hg/mpu9250_cube_rotation_compressed.gif

일상활동 인식(3축 가속도)

['Downstairs',
'Jogging',
'Sitting',
'Standing',
'Upstairs',
'Walking']

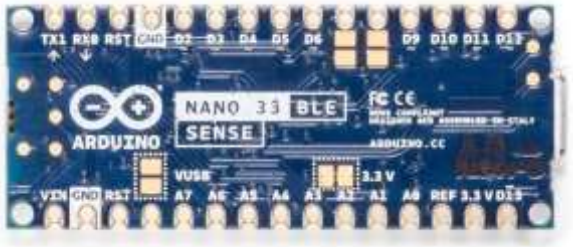






--- ACC_XYZ, 4s: classification report for test data ---

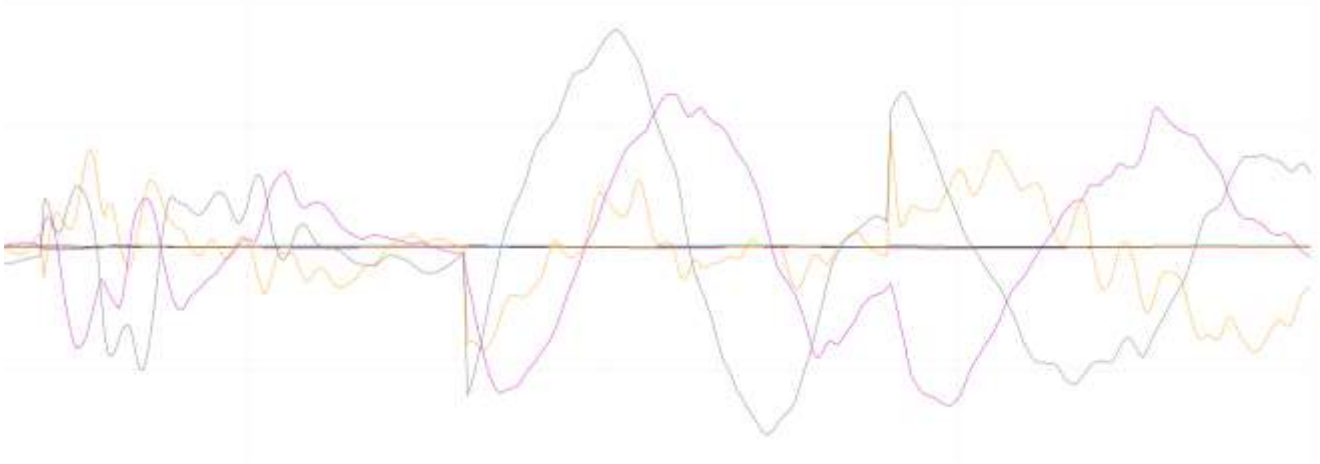
	precision	recall	f1-score	support
0	0.93	0.83	0.87	249
1	0.99	0.97	0.98	864
2	0.97	0.97	0.97	144
3	0.96	0.95	0.95	130
4	0.87	0.92	0.89	297
5	0.96	0.99	0.97	1061
accuracy			0.96	2745
macro avg	0.95	0.94	0.94	2745
weighted avg	0.96	0.96	0.96	2745



Arduino nano33 BLE

Classification of gestures using ACC in Tensorflow 2.x & TinyML/TF-Lite

Data 모으기



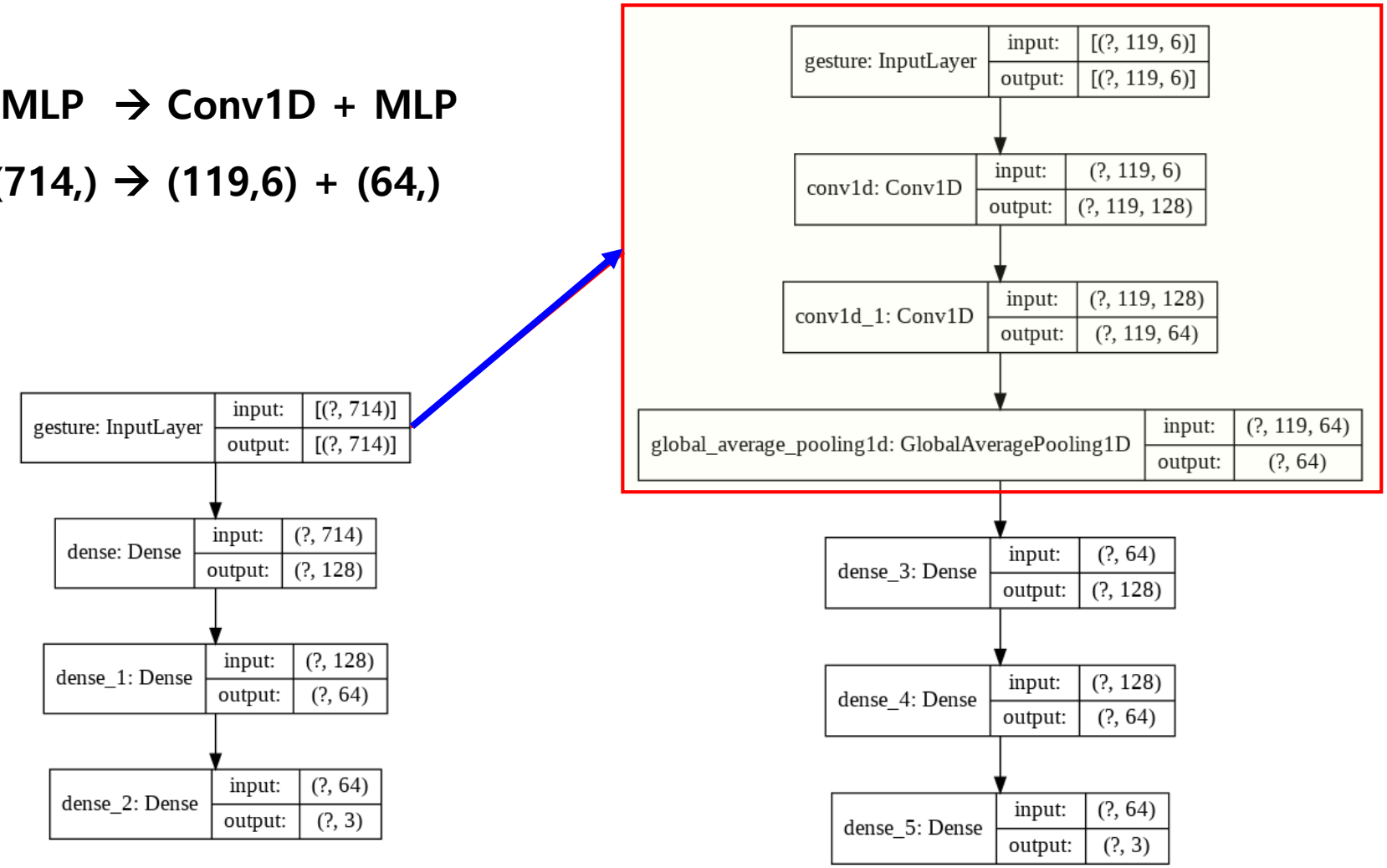
PUNCH CCW CW

119 X 6 (ax,ay,az, gx,gy,gz)

DL architecture

MLP → Conv1D + MLP

(714,) → (119,6) + (64,)



DL-model

```
from tensorflow.keras import layers
```

```
# TF2 functional API
# CONV1D & MLP
inputs = keras.Input(shape=(119,6), name='gesture')
x = layers.Conv1D(128, 3, padding='causal',activation='relu')(inputs)  # 32,
x = layers.Conv1D(64, 3, padding='causal', activation='relu')(x)      # 16
x = layers.GlobalAveragePooling1D()(x)  # New features (714 => 16 or 64)
x = layers.Dense(128, activation='relu')(x)
x = layers.Dense(64, activation='relu')(x)
outputs = layers.Dense(NUM_GESTURES, activation='softmax')(x)

model_conv = keras.Model(inputs=inputs, outputs=outputs, name='gesture_model2')

model_conv.compile(optimizer='rmsprop', loss='mse', metrics=['accuracy'])
```

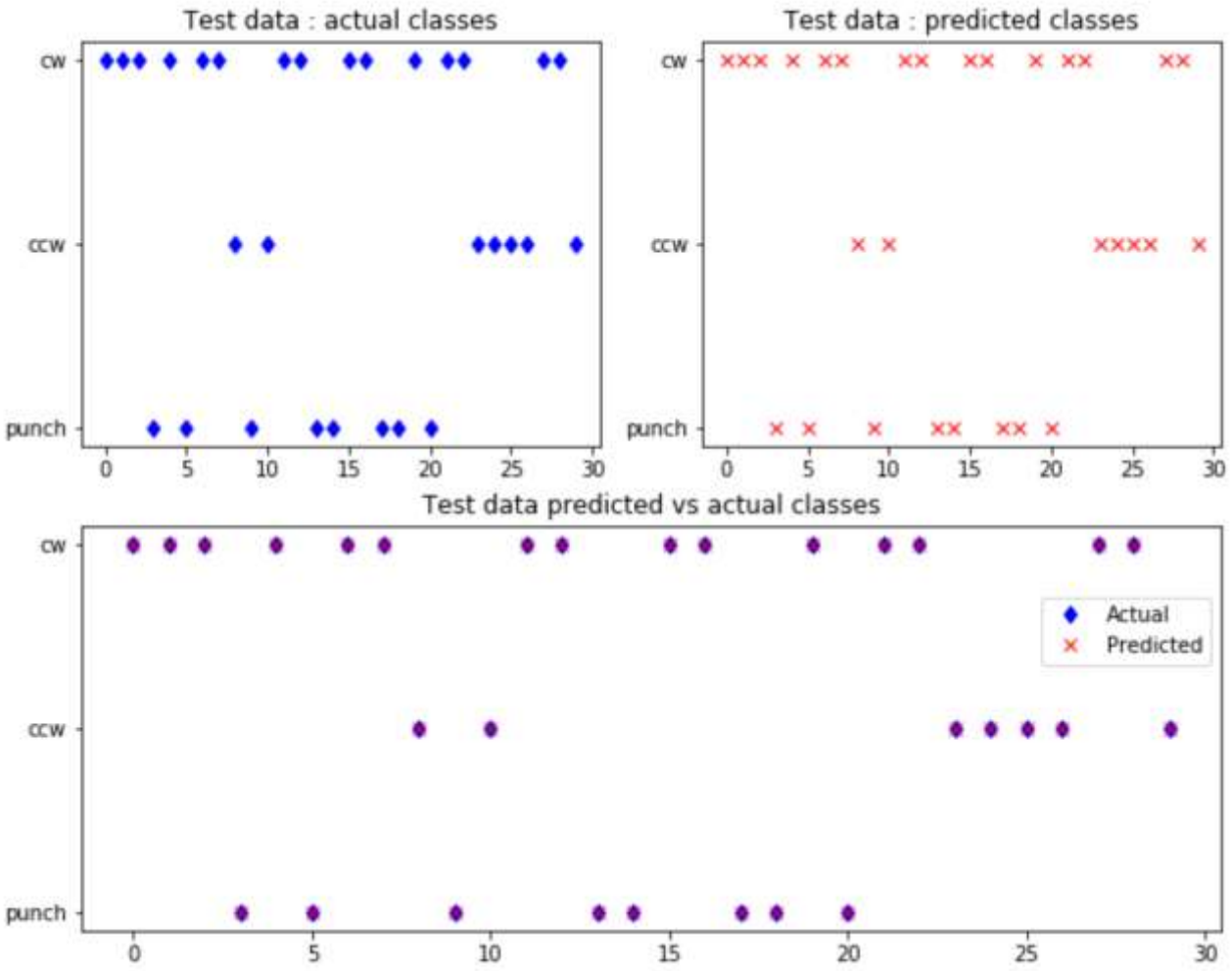
```
# train the model
history = model_conv.fit(inputs_train2, outputs_train, epochs=500, batch_size=16
                         validation_data=(inputs_validate2, outputs_validate))
```

```
model_conv.summary()
```

```
Model: "gesture_model2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gesture (InputLayer)         [(None, 119, 6)]          0
_________________________________________________________________
conv1d (Conv1D)              (None, 119, 128)          2432
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 119, 64)           24640
_________________________________________________________________
global_average_pooling1d (Gl (None, 64)                0
_________________________________________________________________
dense_3 (Dense)              (None, 128)               8320
_________________________________________________________________
dense_4 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_5 (Dense)              (None, 3)                 195
=================================================================
Total params: 43,843
Trainable params: 43,843
Non-trainable params: 0
```

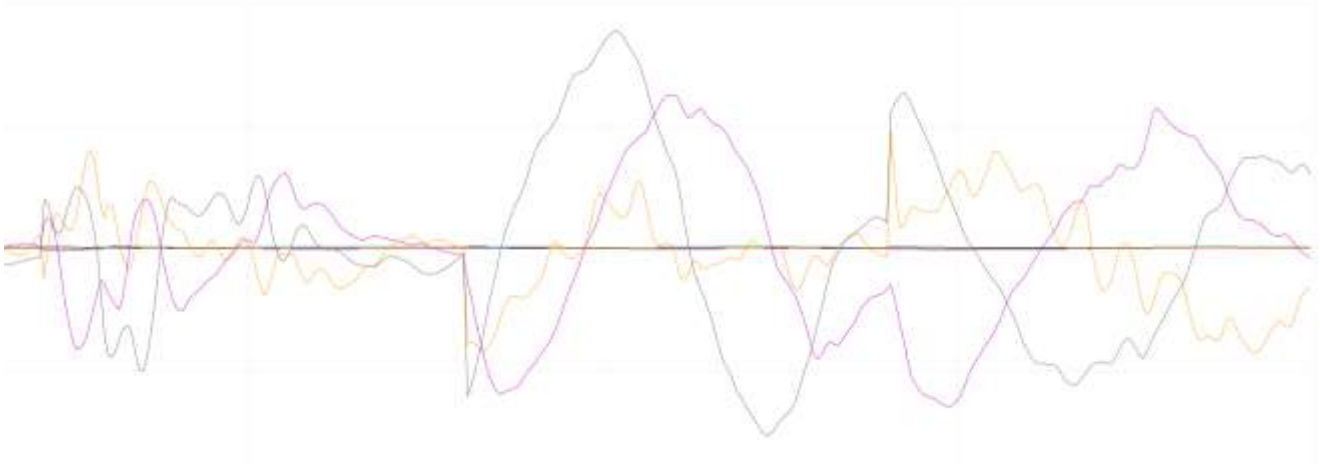
DL-model testing

actual =
[[0. 0. 1.]
[0. 0. 1.]
[0. 0. 1.]
[1. 0. 0.]
[0. 0. 1.]
[1. 0. 0.]
[0. 0. 1.]
[0. 0. 1.]
[0. 1. 0.]
[1. 0. 0.]
[0. 1. 0.]
[0. 0. 1.]
[0. 0. 1.]
[1. 0. 0.]



predictions =
[[0. 0.001 0.999]
[0.001 0.001 0.998]
[0.005 0.001 0.994]
[0.999 0. 0.001]
[0. 0.001 0.999]
[1. 0. 0.]
[0. 0.001 0.999]
[0.002 0.001 0.997]
[0. 1. 0.]
[1. 0. 0.]
[0. 1. 0.]
[0. 0.001 0.999]
[0.001 0. 0.999]
[0.997 0. 0.003]

Real-time testing



PUNCH CCW CW

COM8

```
13:24:06.140 -> punch: 0.996470
13:24:06.140 -> ccw: 0.000000
13:24:06.140 -> cw: 0.003530
13:24:06.140 ->
13:24:10.197 -> punch: 0.000095
13:24:10.197 -> ccw: 0.000000
13:24:10.197 -> cw: 0.999905
13:24:10.197 ->
13:24:13.193 -> punch: 0.000000
13:24:13.193 -> ccw: 1.000000
13:24:13.193 -> cw: 0.000000
```

Lecture materials

References & good sites

- ✓ http://www.arduino.cc Arduino Homepage
- ✓ http://www.nodejs.org/ko Node.js
- ✓ https://plot.ly/ plotly
- ✓ https://www.mongodb.com/ MongoDB
- ✓ http://www.w3schools.com By w3schools
- ✓ http://www.github.com GitHub

Target of this class