Use rectified signals. (rectify\_emg\_moving\_average(X,20))

Use cost sensitive learning(1:5) for binary classification(0:others)

Drop some files out for test. Shuffle and split the rest data as 75% for training and 25% for validation.

Use residual block in conv1d structure.

## Drop Files[6,30,31,32,33,34,35]:

Class 0 : others

Train (acc 0.992)

|  |  |  |
| --- | --- | --- |
|  | Predicted 0 | Predicted others |
| Actual 0 | 5515 | 47 |
| Actual others | 1 | 524 |

Valid (acc 0.984)

|  |  |  |
| --- | --- | --- |
|  | Predicted 0 | Predicted others |
| Actual 0 | 1811 | 29 |
| Actual others | 2 | 188 |

Test (acc 0.920)

|  |  |  |
| --- | --- | --- |
|  | Predicted 0 | Predicted others |
| Actual 0 | 288 | 38 |
| Actual others | 4 | 197 |

Class 1 : 2 : 6

Train (acc 0.897)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted 1 | Predicted 2 | Predicted 6 |
| Actual 1 | 82 | 9 | 31 |
| Actual 2 | 6 | 245 | 6 |
| Actual 6 | 0 | 3 | 153 |

Valid (acc 0.854)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted 1 | Predicted 2 | Predicted 6 |
| Actual 1 | 20 | 4 | 7 |
| Actual 2 | 7 | 88 | 3 |
| Actual 6 | 0 | 5 | 45 |

Test (acc 0.761)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted 1 | Predicted 2 | Predicted 6 |
| Actual 1 | 25 | 20 | 7 |
| Actual 2 | 10 | 46 | 1 |
| Actual 6 | 8 | 2 | 82 |

Class 2 : 6

Train (acc 0.971)

|  |  |  |
| --- | --- | --- |
|  | Predicted 2 | Predicted 6 |
| Actual 2 | 250 | 9 |
| Actual 6 | 3 | 158 |

Valid (acc 0.964)

|  |  |  |
| --- | --- | --- |
|  | Predicted 2 | Predicted 6 |
| Actual 2 | 92 | 4 |
| Actual 6 | 1 | 44 |

Test (acc 0.919)

|  |  |  |
| --- | --- | --- |
|  | Predicted 2 | Predicted 6 |
| Actual 2 | 53 | 4 |
| Actual 6 | 8 | 84 |

Class 1 : 6

Train (acc 0.825)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 6 |
| Actual 1 | 80 | 42 |
| Actual 6 | 5 | 142 |

Valid (acc 0.877)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 6 |
| Actual 1 | 22 | 9 |
| Actual 6 | 2 | 57 |

Test (acc 0.833)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 6 |
| Actual 1 | 36 | 16 |
| Actual 6 | 8 | 84 |

Class 1 : 2

Train (acc 0.871)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 2 |
| Actual 1 | 97 | 24 |
| Actual 2 | 25 | 235 |

Valid (acc 0.874)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 2 |
| Actual 1 | 23 | 9 |
| Actual 2 | 7 | 88 |

Test (acc 0.770)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 2 |
| Actual 1 | 42 | 10 |
| Actual 2 | 15 | 42 |

## Drop Files[7,30,31,32,33,34,35]:

Class 0 : others

Train (acc 0.978)

|  |  |  |
| --- | --- | --- |
|  | Predicted 0 | Predicted others |
| Actual 0 | 5478 | 121 |
| Actual others | 7 | 430 |

Valid (acc 0.974)

|  |  |  |
| --- | --- | --- |
|  | Predicted 0 | Predicted others |
| Actual 0 | 1808 | 45 |
| Actual others | 7 | 152 |

Test (acc 0.973)

|  |  |  |
| --- | --- | --- |
|  | Predicted 0 | Predicted others |
| Actual 0 | 271 | 5 |
| Actual others | 11 | 309 |

Class 1 : 2 : 6

Train (acc 0.890)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted 1 | Predicted 2 | Predicted 6 |
| Actual 1 | 33 | 22 | 9 |
| Actual 2 | 10 | 221 | 1 |
| Actual 6 | 7 | 0 | 144 |

Valid (acc 0.885)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted 1 | Predicted 2 | Predicted 6 |
| Actual 1 | 7 | 5 | 3 |
| Actual 2 | 3 | 90 | 2 |
| Actual 6 | 4 | 0 | 35 |

Test (acc 0.655)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted 1 | Predicted 2 | Predicted 6 |
| Actual 1 | 39 | 37 | 50 |
| Actual 2 | 14 | 62 | 9 |
| Actual 6 | 0 | 0 | 108 |

Class 2 : 6

Train (acc 0.968)

|  |  |  |
| --- | --- | --- |
|  | Predicted 2 | Predicted 6 |
| Actual 2 | 242 | 2 |
| Actual 6 | 10 | 133 |

Valid (acc 0.961)

|  |  |  |
| --- | --- | --- |
|  | Predicted 2 | Predicted 6 |
| Actual 2 | 81 | 2 |
| Actual 6 | 3 | 44 |

Test (acc 0.953)

|  |  |  |
| --- | --- | --- |
|  | Predicted 2 | Predicted 6 |
| Actual 2 | 77 | 8 |
| Actual 6 | 1 | 107 |

Class 1 : 6

Train (acc 0.965)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 6 |
| Actual 1 | 53 | 6 |
| Actual 6 | 1 | 141 |

Valid (acc 0.882)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 6 |
| Actual 1 | 15 | 5 |
| Actual 6 | 3 | 45 |

Test (acc 0.756)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 6 |
| Actual 1 | 71 | 55 |
| Actual 6 | 2 | 106 |

Class 1 : 2

Train (acc 0.940)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 2 |
| Actual 1 | 53 | 5 |
| Actual 2 | 13 | 233 |

Valid (acc 0.901)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 2 |
| Actual 1 | 17 | 4 |
| Actual 2 | 6 | 75 |

Test (acc 0.715)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 2 |
| Actual 1 | 98 | 28 |
| Actual 2 | 32 | 53 |

## Drop Files[5,30,31,32,33,34,35]:

Class 0 : others

Train (acc 0.977)

|  |  |  |
| --- | --- | --- |
|  | Predicted 0 | Predicted others |
| Actual 0 | 5433 | 138 |
| Actual others | 0 | 591 |

Valid (acc 0.974)

|  |  |  |
| --- | --- | --- |
|  | Predicted 0 | Predicted others |
| Actual 0 | 1797 | 49 |
| Actual others | 3 | 206 |

Test (acc 0.946)

|  |  |  |
| --- | --- | --- |
|  | Predicted 0 | Predicted others |
| Actual 0 | 300 | 11 |
| Actual others | 12 | 104 |

Class 1 : 2 : 6

Train (acc 0.871)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted 1 | Predicted 2 | Predicted 6 |
| Actual 1 | 97 | 21 | 28 |
| Actual 2 | 18 | 278 | 1 |
| Actual 6 | 6 | 3 | 147 |

Valid (acc 0.835)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted 1 | Predicted 2 | Predicted 6 |
| Actual 1 | 31 | 8 | 7 |
| Actual 2 | 10 | 93 | 3 |
| Actual 6 | 5 | 0 | 43 |

Test (acc 0.456)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted 1 | Predicted 2 | Predicted 6 |
| Actual 1 | 3 | 10 | 0 |
| Actual 2 | 3 | 5 | 1 |
| Actual 6 | 29 | 20 | 45 |

Class 2 : 6

Train (acc 0.980)

|  |  |  |
| --- | --- | --- |
|  | Predicted 2 | Predicted 6 |
| Actual 2 | 288 | 4 |
| Actual 6 | 5 | 158 |

Valid (acc 0.973)

|  |  |  |
| --- | --- | --- |
|  | Predicted 2 | Predicted 6 |
| Actual 2 | 108 | 3 |
| Actual 6 | 1 | 40 |

Test (acc 0.601)

|  |  |  |
| --- | --- | --- |
|  | Predicted 2 | Predicted 6 |
| Actual 2 | 6 | 3 |
| Actual 6 | 38 | 56 |

Class 1 : 6

Train (acc 0.902)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 6 |
| Actual 1 | 123 | 22 |
| Actual 6 | 7 | 145 |

Valid (acc 0.858)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 6 |
| Actual 1 | 38 | 9 |
| Actual 6 | 5 | 47 |

Test (acc 0.514)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 6 |
| Actual 1 | 13 | 0 |
| Actual 6 | 52 | 42 |

Class 1 : 2

Train (acc 0.946)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 2 |
| Actual 1 | 141 | 11 |
| Actual 2 | 13 | 281 |

Valid (acc 0.845)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 2 |
| Actual 1 | 30 | 10 |
| Actual 2 | 13 | 95 |

Test (acc 0.363)

|  |  |  |
| --- | --- | --- |
|  | Predicted 1 | Predicted 2 |
| Actual 1 | 0 | 13 |
| Actual 2 | 1 | 8 |

## Model:

def residual\_block(x, i):

tanh\_out = layers.Conv1D(filters,

2,

dilation\_rate = 2\*\*i,

padding='causal',

name='dilated\_conv\_%d\_tanh' % (2 \*\* i),

activation='tanh'

)(x)

sigm\_out = layers.Conv1D(filters,

2,

dilation\_rate = 2\*\*i,

padding='causal',

name='dilated\_conv\_%d\_sigm' % (2 \*\* i),

activation='sigmoid'

)(x)

z = layers.Multiply(name='gated\_activation\_%d' % (i))([tanh\_out, sigm\_out])

skip = layers.Conv1D(filters, 1, name='skip\_%d'%(i))(z)

res = layers.Add(name='residual\_block\_%d' % (i))([skip, x])

return res, skip

x = layers.Input(shape=(1024,8), name='original\_input')

skip\_connections = []

out = layers.Conv1D(filters, 2, dilation\_rate=1, padding='causal', name='dilated\_conv\_1')(x)

for i in range(1, 6):

out, skip = residual\_block(out,i)

skip\_connections.append(skip)

out = layers.Add(name='skip\_connections')(skip\_connections)

out = layers.Activation('elu')(out)

out = layers.Conv1D(32, 3, strides = 1, padding='same', activation = 'relu')(out)

out = layers.MaxPooling1D(2, padding='same')(out)

out = layers.Conv1D(16, 7, padding='same', activation='elu')(out)

out = layers.MaxPooling1D(2, padding='same)(out)

out = layers.Conv1D(8, 3, activation='elu', padding='same')(out)

out = layers.GlobalAveragePooling1D()(out)

out = layers.Dense(2,activation='softmax')(out)

model = Model(x, out)