

Use rectified signals. (rectify_emg_moving_average(X,20))

use cost sensitive learning(1:10) 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.

Model consists of rnn and conv1d neural network.

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

Class 0 : others

Train (acc 0.992)

	Predicted 0	Predicted others
Actual 0	5516	46
Actual others	0	525

Valid (acc 0.984)

	Predicted 0	Predicted others
Actual 0	1808	32
Actual others	0	190

Test (acc 0.916)

	Predicted 0	Predicted others
Actual 0	286	40
Actual others	4	197

Class 1 : 2 : 6

Train (acc 1.000)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	122	0	0
Actual 2	0	257	0
Actual 6	0	0	156

Valid (acc 0.899)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	24	2	5
Actual 2	4	92	2
Actual 6	2	3	45

Test (acc 0.597)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	24	19	9
Actual 2	18	39	0
Actual 6	31	4	57

Class 2 : 6

Train (acc 0.985)

	Predicted 2	Predicted 6
Actual 2	253	6
Actual 6	0	161

Valid (acc 0.964)

	Predicted 2	Predicted 6
Actual 2	93	3
Actual 6	2	43

Test (acc 0.966)

	Predicted 2	Predicted 6
Actual 2	56	1
Actual 6	4	88

Class 1 : 6

Train (acc 1.000)

	Predicted 1	Predicted 6
Actual 1	122	0
Actual 6	0	147

Valid (acc 0.944)

	Predicted 1	Predicted 6
Actual 1	27	4
Actual 6	1	58

Test (acc 0.833)

	Predicted 1	Predicted 6
Actual 1	46	6
Actual 6	18	74

Class 1 : 2

Train (acc 0.997)

	Predicted 1	Predicted 2
Actual 1	121	0
Actual 2	1	259

Valid (acc 0.913)

	Predicted 1	Predicted 2
Actual 1	25	7
Actual 2	4	91

Test (acc 0.779)

	Predicted 1	Predicted 2
Actual 1	34	18
Actual 2	6	51

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

Class 0 : others

Train (acc 0.993)

	Predicted 0	Predicted others
Actual 0	5558	41
Actual others	0	437

Valid (acc 0.987)

	Predicted 0	Predicted others
Actual 0	1833	20
Actual others	5	154

Test (acc 0.969)

	Predicted 0	Predicted others
Actual 0	272	4
Actual others	14	306

Class 1 : 2 : 6

Train (acc 1.000)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	64	0	0
Actual 2	0	232	0
Actual 6	0	0	151

Valid (acc 0.926)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	9	4	2
Actual 2	1	92	2
Actual 6	2	0	37

Test (acc 0.655)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	49	26	51
Actual 2	17	60	8
Actual 6	5	3	100

Class 2 : 6

Train (acc 0.981)

	Predicted 2	Predicted 6
Actual 2	239	5
Actual 6	2	141

Valid (acc 0.953)

	Predicted 2	Predicted 6
Actual 2	79	4
Actual 6	2	45

Test (acc 0.932)

	Predicted 2	Predicted 6
Actual 2	73	12
Actual 6	1	107

Class 1 : 6

Train (acc 1.000)

	Predicted 1	Predicted 6
Actual 1	59	0
Actual 6	0	142

Valid (acc 0.955)

	Predicted 1	Predicted 6
Actual 1	19	1
Actual 6	2	46

Test (acc 0.739)

	Predicted 1	Predicted 6
Actual 1	65	61
Actual 6	0	108

Class 1 : 2

Train (acc 0.993)

	Predicted 1	Predicted 2
Actual 1	56	2
Actual 2	0	246

Valid (acc 0.882)

	Predicted 1	Predicted 2
Actual 1	13	8
Actual 2	4	77

Test (acc 0.779)

	Predicted 1	Predicted 2
Actual 1	106	20
Actual 2	28	57

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

Class 0 : others

Train (acc 0.993)

	Predicted 0	Predicted others
Actual 0	5511	60
Actual others	0	591

Valid (acc 0.987)

	Predicted 0	Predicted others
Actual 0	1821	25
Actual others	8	201

Test (acc 0.969)

	Predicted 0	Predicted others
Actual 0	301	10
Actual others	12	104

Class 1 : 2 : 6

Train (acc 1.000)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	146	0	0
Actual 2	0	297	0
Actual 6	0	0	156

Valid (acc 0.905)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	38	3	5
Actual 2	4	100	2
Actual 6	4	1	43

Test (acc 0.344)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	2	11	51
Actual 2	1	7	1
Actual 6	30	33	31

Class 2 : 6

Train (acc 0.997)

	Predicted 2	Predicted 6
Actual 2	291	1
Actual 6	0	163

Valid (acc 0.986)

	Predicted 2	Predicted 6
Actual 2	109	2
Actual 6	0	41

Test (acc 0.572)

	Predicted 2	Predicted 6
Actual 2	8	1
Actual 6	43	51

Class 1 : 6

Train (acc 1.000)

	Predicted 1	Predicted 6
Actual 1	145	0
Actual 6	0	152

Valid (acc 0.909)

	Predicted 1	Predicted 6
Actual 1	40	7
Actual 6	2	50

Test (acc 0.439)

	Predicted 1	Predicted 6
Actual 1	13	0
Actual 6	60	34

Class 1 : 2

Train (acc 1.000)

	Predicted 1	Predicted 2
Actual 1	152	0
Actual 2	0	294

Valid (acc 0.892)

	Predicted 1	Predicted 2
Actual 1	34	6
Actual 2	10	99

Test (acc 0.409)

	Predicted 1	Predicted 2
Actual 1	2	11
Actual 2	2	7

Model:

```
kernel_size=3
reg=regularizers.l2(1e-4)
drop_rate = 0.
kernel_initializer = 'glorot_normal'
mo = 0.8
st = 1
axis = 2
model = keras.models.Sequential()
model.add(layers.InputLayer(input_shape=X[:, :, :].shape[1:]))
model.add(layers.Bidirectional(layers.LSTM(32, return_sequences=True,
recurrent_regularizer=reg)))
model.add(layers.Conv1D(filters=32, kernel_size=kernel_size,
                        strides=st,
                        padding='same',
                        kernel_regularizer=reg,
                        ))
model.add(layers.BatchNormalization(momentum=mo))
model.add(layers.LeakyReLU(0.1))
model.add(layers.MaxPooling1D(2))
model.add(layers.Dropout(drop_rate))
model.add(layers.Conv1D(filters=16, kernel_size=kernel_size,
                        strides=st,
                        padding='same',
                        kernel_regularizer=reg,
                        ))
model.add(layers.BatchNormalization(momentum=mo))
model.add(layers.LeakyReLU(0.1))
model.add(layers.MaxPooling1D(2))
model.add(layers.Dropout(drop_rate))
model.add(layers.Conv1D(filters=8, kernel_size=kernel_size,
                        strides=st,
                        padding='same',
                        kernel_regularizer=reg,
                        ))
model.add(layers.BatchNormalization(momentum=mo))
```

```
model.add(layers.LeakyReLU(0.1))
model.add(layers.MaxPooling1D(2))
model.add(layers.Dropout(drop_rate))
model.add(layers.Conv1D(filters=4, kernel_size=kernel_size,
                        strides=st,
                        padding='same',
                        kernel_regularizer=reg,
                        ))
model.add(layers.BatchNormalization(momentum=mo))
model.add(layers.LeakyReLU(0.1))
model.add(layers.MaxPooling1D(2))
model.add(layers.GlobalAveragePooling1D())
model.add(layers.Dropout(drop_rate))
model.add(layers.Dense(2,activation='softmax',kernel_regularizer=reg))
```