

Width: 1024

Stride: 256

Only use channels LEFT\_TA, LEFT\_TS, RIGHT\_TA, RIGHT\_TS.

No scale on raw signal. Not shuffle and pick the last 20% data of each labels out for validation.

Shuffle and split the rest to 80% for training and 20% for test.

Use empirical mode decomposition on raw data. Then get data with dimension [N,C,L,I].

N: number of the data

I: number of the components from decomposition

L: length of the window width

C: number of channels

Feed the data to 1D CNN model.

No detrend

rate=0.2

kernel\_size=7

kernel\_size2=5

stride=1

acti='relu'

with averagepooling

Train (acc 1.000)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	277	0	0
Actual 2	0	605	0
Actual 6	0	0	392

Test (acc 0.974)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	66	2	0
Actual 2	4	139	1
Actual 6	1	0	106

Last 20% (acc 0.801)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	45	15	28
Actual 2	8	177	10
Actual 6	6	14	106

```
input_ = layers.Input(shape=imfs[:,[0,1,3,5],:,:].shape[1:])
```

```
cnn1 = layers.Conv1D(256,kernel_size,strides=stride,
```

```

padding='same')(input_)
cnn1 = layers.Activation(acti)(cnn1)
cnn1 = layers.AveragePooling2D((1,2))(cnn1)
cnn1 = layers.Dropout(rate)(cnn1)

cnn2 = layers.Conv1D(128,kernel_size,strides=stride,
padding='same')(cnn1)
cnn2 = layers.Activation(acti)(cnn2)
cnn2 = layers.BatchNormalization(momentum=0.8)(cnn2)
cnn2 = layers.AveragePooling2D((1,2))(cnn2)
cnn2 = layers.Dropout(rate)(cnn2)

cnn3 = layers.Conv1D(64,kernel_size,strides=stride,
padding='same')(cnn2)
cnn3 = layers.Activation(acti)(cnn3)
cnn3 = layers.BatchNormalization(momentum=0.8)(cnn3)
cnn3 = layers.AveragePooling2D((1,2))(cnn3)
cnn3 = layers.Dropout(rate)(cnn3)

cnn4 = layers.Conv1D(32,kernel_size2,strides=stride,
padding='same')(cnn3)
cnn4 = layers.Activation(acti)(cnn4)
cnn4 = layers.BatchNormalization(momentum=0.8)(cnn4)
cnn4 = layers.AveragePooling2D((1,2))(cnn4)
cnn4 = layers.Dropout(rate)(cnn4)

cnn5 = layers.Conv1D(16,kernel_size,strides=stride,
padding='same')(cnn4)
cnn5 = layers.Activation(acti)(cnn5)
cnn5 = layers.BatchNormalization(momentum=0.8)(cnn5)
cnn5 = layers.AveragePooling2D((1,2))(cnn5)
cnn5 = layers.Dropout(rate)(cnn5)

cnn6 = layers.Conv1D(8,kernel_size2,strides=stride,
padding='same')(cnn5)
cnn6 = layers.Activation(acti)(cnn6)
cnn6 = layers.BatchNormalization(momentum=0.8)(cnn6)
cnn6 = layers.AveragePooling2D((1,2))(cnn6)
cnn6 = layers.Dropout(rate)(cnn6)

cnn7 = layers.Conv1D(4,kernel_size2,strides=stride,
padding='same')(cnn6)
cnn7 = layers.Activation(acti)(cnn7)
cnn7 = layers.BatchNormalization(momentum=0.8)(cnn7)

```

```
cnn7 = layers.AveragePooling2D((1,2))(cnn7)
```

```
cnn7 = layers.Dropout(rate)(cnn7)
```

```
flatten = layers.Flatten()(cnn7)
```

```
dropout = layers.Dropout(rate)(flatten)
```

```
output = layers.Dense(3,activation = 'softmax')(dropout)
```

```
model = Model(inputs=[input_],outputs=[output])
```

without pooling:

Train (acc 1.000)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	277	0	0
Actual 2	0	605	0
Actual 6	0	0	392

Test (acc 0.990)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	67	1	0
Actual 2	2	142	0
Actual 6	0	0	107

Last 20% (acc 0.748)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	55	22	11
Actual 2	25	162	8
Actual 6	29	8	89

With maxpooling:

Train (acc 1.000)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	277	0	0
Actual 2	0	605	0
Actual 6	0	0	392

Test (acc 0.990)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	67	1	0
Actual 2	2	142	0
Actual 6	0	0	107

Last 20% (acc 0. 831)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	59	16	13
Actual 2	16	167	12
Actual 6	2	10	114

Detrend(lambda=300):

Maxpooling:

Train (acc 1.000)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	277	0	0
Actual 2	0	605	0
Actual 6	0	0	392

Test (acc 0.981)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	65	2	1
Actual 2	1	142	1
Actual 6	1	0	106

Last 20% (acc 0.777)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	50	24	14
Actual 2	25	166	4
Actual 6	8	16	102

Averagepooling:

Train (acc 0.992)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	275	0	2
Actual 2	3	600	2
Actual 6	2	0	390

Test (acc 0.962)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	63	3	2
Actual 2	3	140	1
Actual 6	3	0	104

Last 20% (acc 0.816)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	55	14	19
Actual 2	21	167	7
Actual 6	4	10	112

No detrend:

rate=0.2

kernel\_size=7

kernel\_size2=7

stride=1

acti='relu'

averagepooling

Train (acc 0.995)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	275	0	1
Actual 2	2	602	1
Actual 6	2	0	390

Test (acc 0.937)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	62	2	4
Actual 2	4	137	3
Actual 6	6	1	100

Last 20% (acc 0.814)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	65	9	14
Actual 2	11	166	18
Actual 6	19	5	102

rate=0.2

kernel\_size=14

kernel\_size2=7

stride=1

acti='relu'

averagepooling

Train (acc 0.997)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	276	0	0
Actual 2	1	604	0
Actual 6	0	1	391

Test (acc 0.937)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	62	3	3
Actual 2	4	133	4
Actual 6	3	3 1	101

Last 20% (acc 0.804)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	50	18	20
Actual 2	11	175	9
Actual 6	17	5	104

rate=0.2  
 kernel\_size=7  
 kernel\_size2=7  
 stride=1  
 acti='relu'  
 averagepooling:  
 [N,I,L,C]

Train (acc 0.967)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	263	13	1
Actual 2	9	595	1
Actual 6	11	7	374

Test (acc 0.896)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	57	9	2
Actual 2	6	137	1
Actual 6	11	4	92

Last 20% (acc 0.821)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	52	20	16
Actual 2	13	180	2
Actual 6	8	14	104

Using LayerNormalization:

rate=0.2

kernel\_size=7

kernel\_size2=7

stride=1

acti='relu'

averagepooling:

layer normalization axis: 1

Train (acc 0.996)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	274	2	1
Actual 2	0	605	0
Actual 6	1	0	391

Test (acc 0.962)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	63	1	4
Actual 2	3	138	3
Actual 6	1	0	106

Last 20% (acc 0.711)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	24	21	43
Actual 2	16	167	12
Actual 6	10	16	100

Raw data(channel[0,1,4,5]):

Using WaveNet:

```
model = keras.models.Sequential()
```

```
model.add(layers.InputLayer(input_shape=feature[:,[0,1,3,5],:].shape[1:]))
```

```
for rate in (1, 2, 4, 8,16,32,64):
```

```
    model.add(layers.Conv1D(filters=20, kernel_size=2, padding="causal",  
                           activation="relu", dilation_rate=rate))
```

```
model.add(layers.Conv1D(filters=10, kernel_size=1))
```

```
model.add(layers.Conv1D(filters=16, kernel_size=5,padding='same'))
```

```
model.add(layers.BatchNormalization(momentum=0.8))
```

```
model.add(layers.Activation('relu'))
```

```
model.add(layers.AveragePooling2D((1,2)))
```

```
model.add(layers.Dropout(0.2))
```

```
model.add(layers.Conv1D(filters=8, kernel_size=5,padding='same'))
```

```

model.add(layers.BatchNormalization(momentum=0.8))
model.add(layers.Activation('relu'))
model.add(layers.AveragePooling2D((1,2)))
model.add(layers.Dropout(0.2))
model.add(layers.Conv1D(filters=4, kernel_size=5,padding='same'))
model.add(layers.BatchNormalization(momentum=0.8))
model.add(layers.Activation('relu'))
model.add(layers.AveragePooling2D((1,2)))
model.add(layers.Dropout(0.2))
model.add(layers.Flatten())
model.add(layers.Dense(3,activation='softmax'))

```

Train (acc 1 000)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	277	0	0
Actual 2	0	605	0
Actual 6	0	0	392

Test (acc 0.868)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	48	8	12
Actual 2	7	131	6
Actual 6	6	3	98

Last 20% (acc 0.782)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	44	22	22
Actual 2	9	170	16
Actual 6	5	15	106

```
input_ = layers.Input(shape=feature[:, :, [0,1,3,5]].shape[1:])
```

```
cnn1 = layers.Conv1D(16,kernel_size,strides=stride,
                    padding='same')(input_)
```

```
cnn1 = layers.BatchNormalization()(cnn1)
```

```
cnn1 = layers.Activation(acti)(cnn1)
```

```
cnn1 = layers.AveragePooling2D((1,2))(cnn1)
```

```
cnn1 = layers.Dropout(rate)(cnn1)
```

```
cnn2 = layers.Conv1D(32,kernel_size,strides=stride,
                    padding='same')(cnn1)
```

```
cnn2 = layers.BatchNormalization(momentum=0.8)(cnn2)
```

```
cnn2 = layers.Activation(acti)(cnn2)
```



```

cnn2 = layers.AveragePooling2D((1,2))(cnn2)
cnn2 = layers.Dropout(rate)(cnn2)

cnn3 = layers.Conv1D(64,kernel_size, strides=stride,
                    padding='same')(cnn2)
cnn3 = layers.BatchNormalization(momentum=0.8)(cnn3)
cnn3 = layers.Activation(acti)(cnn3)
cnn3 = layers.AveragePooling2D((1,2))(cnn3)
cnn3 = layers.Dropout(rate)(cnn3)

cnn4 = layers.Conv1D(128,kernel_size2, strides=stride,
                    padding='same')(cnn3)
cnn4 = layers.BatchNormalization(momentum=0.8)(cnn4)
cnn4 = layers.Activation(acti)(cnn4)
cnn4 = layers.AveragePooling2D((1,2))(cnn4)
cnn4 = layers.Dropout(rate)(cnn4)

cnn5 = layers.Conv1D(64,kernel_size, strides=stride,
                    padding='same')(cnn4)
cnn5 = layers.BatchNormalization(momentum=0.8)(cnn5)
cnn5 = layers.Activation(acti)(cnn5)
cnn5 = layers.AveragePooling2D((1,2))(cnn5)
cnn5 = layers.Dropout(rate)(cnn5)

cnn6 = layers.Conv1D(32,kernel_size2, strides=stride,
                    padding='same')(cnn5)
cnn6 = layers.BatchNormalization(momentum=0.8)(cnn6)
cnn6 = layers.Activation(acti)(cnn6)
cnn6 = layers.AveragePooling2D((1,2))(cnn6)
cnn6 = layers.Dropout(rate)(cnn6)

cnn7 = layers.Conv1D(16,kernel_size2, strides=stride,
                    padding='same')(cnn6)
cnn7 = layers.BatchNormalization(momentum=0.8)(cnn7)
cnn7 = layers.Activation(acti)(cnn7)
cnn7 = layers.AveragePooling2D((1,2))(cnn7)
cnn7 = layers.Dropout(rate)(cnn7)

flatten = layers.Flatten()(cnn7)
dropout = layers.Dropout(rate)(flatten)
output = layers.Dense(3,activation = 'softmax')(dropout)
model = Model(inputs=[input_],outputs=[output])

```

Train (acc 999)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	277	0	0
Actual 2	0	605	0
Actual 6	1	0	391

Test (acc 0.937)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	59	6	3
Actual 2	2	139	3
Actual 6	6	0	101

Last 20% (acc 0.787)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	47	17	24
Actual 2	15	171	9
Actual 6	14	8	104

```
input_ = layers.Input(shape=feature[:, :, [0, 1, 3, 5]].shape[1:])
```

```
cnn1 = layers.Conv1D(16, kernel_size, strides=stride,
                    padding='same')(input_)
```

```
cnn1 = layers.BatchNormalization()(cnn1)
```

```
cnn1 = layers.Activation(acti)(cnn1)
```

```
cnn1 = layers.AveragePooling2D((1, 2))(cnn1)
```

```
cnn1 = layers.Dropout(rate)(cnn1)
```

```
cnn2 = layers.Conv1D(32, kernel_size, strides=stride,
                    padding='same')(cnn1)
```

```
cnn2 = layers.BatchNormalization(momentum=0.8)(cnn2)
```

```
cnn2 = layers.Activation(acti)(cnn2)
```

```
cnn2 = layers.AveragePooling2D((1, 2))(cnn2)
```

```
cnn2 = layers.Dropout(rate)(cnn2)
```

```
cnn3 = layers.Conv1D(64, kernel_size, strides=stride,
                    padding='same')(cnn2)
```

```
cnn3 = layers.BatchNormalization(momentum=0.8)(cnn3)
```

```
cnn3 = layers.Activation(acti)(cnn3)
```

```
cnn3 = layers.AveragePooling2D((1, 2))(cnn3)
```

```
cnn3 = layers.Dropout(rate)(cnn3)
```

```
cnn4 = layers.Conv1D(128, kernel_size2, strides=stride,
```

```

padding='same')(cnn3)
cnn4 = layers.BatchNormalization(momentum=0.8)(cnn4)
cnn4 = layers.Activation(acti)(cnn4)
cnn4 = layers.AveragePooling2D((1,2))(cnn4)
cnn4 = layers.Dropout(rate)(cnn4)

cnn5 = layers.Conv1D(64,kernel_size, strides=stride,
padding='same')(cnn4)
cnn5 = layers.BatchNormalization(momentum=0.8)(cnn5)
cnn5 = layers.Activation(acti)(cnn5)
cnn5 = layers.AveragePooling2D((1,2))(cnn5)
cnn5 = layers.Dropout(rate)(cnn5)

cnn6 = layers.Conv1D(32,kernel_size2, strides=stride,
padding='same')(cnn5)
cnn6 = layers.BatchNormalization(momentum=0.8)(cnn6)
cnn6 = layers.Activation(acti)(cnn6)
cnn6 = layers.AveragePooling2D((1,2))(cnn6)
cnn6 = layers.Dropout(rate)(cnn6)

cnn7 = layers.Conv1D(16,kernel_size2, strides=stride,
padding='same')(cnn6)
cnn7 = layers.BatchNormalization(momentum=0.8)(cnn7)
cnn7 = layers.Activation(acti)(cnn7)
cnn7 = layers.AveragePooling2D((1,2))(cnn7)
cnn7 = layers.Dropout(rate)(cnn7)

output = layers.Dense(3,activation = 'softmax')(cnn7)
model = Model(inputs=[input_],outputs=[output])

rate=0.2
kernel_size=7
kernel_size2=7
stride=1
acti='relu'

```

Train (acc 998)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	275	2	0
Actual 2	0	605	0
Actual 6	0	0	392

Test (acc 0.931)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	54	12	2
Actual 2	3	140	1
Actual 6	4	0	103

Last 20% (acc 0.806)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	51	19	18
Actual 2	11	182	2
Actual 6	14	15	97