Window size:1024
Stride: 512
Raw data with detrend(lambda:300)
Lowpass filter(300Hz)
Use all channels
Use rectified signal. No shuffle and split data into 20% for test, 20% for validation and 60% for training.

```
kernel_size=11
reg=regularizers.l2(1e-4)
drop_rate = 0.2
kernel_initializer = 'glorot_normal'
mo = 0.8
st = 1
model = keras.models.Sequential()
model.add(layers.InputLayer(input_shape=X[:,:,:].shape[1:]))
model.add(layers.Bidirectional(layers.LSTM(32,return_sequences=True,
                                                  #kernel_regularizer=reg,
                                                  recurrent_regularizer=reg)))
model.add(layers.Conv1D(filters=32, kernel_size=kernel_size,strides=st,
                         padding='same',
                         kernel_regularizer=reg,
                         kernel_initializer=kernel_initializer
                        ))
model.add(layers.BatchNormalization(momentum=mo))
model.add(layers.ELU())
model.add(layers.AveragePooling1D(2))
model.add(layers.Dropout(drop_rate))
model.add(layers.Conv1D(filters=16, kernel_size=kernel_size,strides=st,
                         padding='same',
                         kernel_regularizer=reg,
                         kernel_initializer=kernel_initializer
                        ))
model.add(layers.BatchNormalization(momentum=mo))
model.add(layers.ELU())
model.add(layers.AveragePooling1D(2))
model.add(layers.Dropout(drop_rate))
model.add(layers.Conv1D(filters=8, kernel_size=kernel_size,strides=st,
                         padding='same',
                         kernel_regularizer=reg,
                         kernel_initializer=kernel_initializer
                        ))
model.add(layers.BatchNormalization(momentum=mo))
model.add(layers.ELU())
model.add(layers.AveragePooling1D(2))
model.add(layers.Dropout(drop_rate))
model.add(layers.Conv1D(filters=4, kernel_size=kernel_size,strides=st,
                        padding='same',
                        kernel_regularizer=reg,
                        kernel_initializer=kernel_initializer
                       ))
model.add(layers.BatchNormalization(momentum=mo))
model.add(layers.ELU())
model.add(layers.GlobalAveragePooling1D())
model.add(layers.Dropout(drop_rate))
model.add(layers.Dense(2,activation='softmax',kernel_regularizer=reg))
```

Train (acc 0.988)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	125	0	1
Actual 2	2	291	0
Actual 6	2	2	180

Valid (acc 0.841)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	33	8	2
Actual 2	7	89	7
Actual 6	7	2	53

Test (acc 0.783)

	Predicted 1	Predicted 2	Predicted 6
Actual 1	28	7	10
Actual 2	8	94	6
Actual 6	7	9	48

Class 2:6

Train (acc 0.991)

	Predicted 2	Predicted 6
Actual 2	293	0
Actual 6	4	180

Valid (acc 0.957)

	Predicted 2	Predicted 6
Actual 2	98	5
Actual 6	2	60

Test (acc 0.912)

	Predicted 2	Predicted 6
Actual 2	105	3
Actual 6	12	52

Class 1:6

Train (acc 0.970)

	Predicted 1	Predicted 6
Actual 1	118	8
Actual 6	1	183

Valid (acc 0.866)

	Predicted 1	Predicted 6
Actual 1	29	14
Actual 6	0	62

Test (acc 0.669)

	Predicted 1	Predicted 6
Actual 1	20	25
Actual 6	11	53

Class 1:2

Train (acc 0.961)

	Predicted 1	Predicted 2
Actual 1	110	16
Actual 2	0	293

Valid (acc 0.863)

	Predicted 1	Predicted 2
Actual 1	28	15
Actual 2	5	98

Test (acc 0.836)

	Predicted 1	Predicted 2
Actual 1	35	10
Actual 2	15	93