

First I used data with label2, then I tried to use data, whose label1 and label2 are same.
 There are no data with label 5 in files.
 Shuffling and split the data to 20% for test and 80% for training.

Label2:

label	1	2	3	4	6
number	3899	6563	363	29	3401

The number of data with label 4 is too less, so I didn't use it.

XGBoost:

Train (acc 1.000)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	3142	0	0	
Actual 2	0	5273	0	0
Actual 3	0	0	284	0
Actual 6	0	0	0	2681

Test (acc 0.867)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	607	103	0	47
Actual 2	70	1197	1	22
Actual 3	0	38	41	0
Actual 6	60	38	0	622

ANN:

Train (acc 0.986)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	3040	69	0	33
Actual 2	36	5226	1	10
Actual 3	0	0	284	0
Actual 6	7	0	0	2674

Test (acc 0.878)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	602	93	1	61
Actual 2	64	1194	5	27
Actual 3	0	15	64	0
Actual 6	59	22	0	639

Same label:

label	1	2	3	6
number	2370	5348	47	2941

XGBoost:

Train (acc 1.000)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	1901	0	0	
Actual 2	0	4265	0	0
Actual 3	0	0	40	0
Actual 6	0	0	0	2358

Test (acc 0.915)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	377	61	0	31
Actual 2	28	1027	0	28
Actual 3	1	6	0	0
Actual 6	18	10	0	555

ANN (without class weight):

Train (acc 0.994)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	1874	27	0	0
Actual 2	5	4258	0	2
Actual 3	0	14	26	0
Actual 6	0	0	0	2358

Test (acc 0.934)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	407	48	0	14
Actual 2	35	1027	0	21
Actual 3	1	4	2	0
Actual 6	14	4	0	565

ANN (with class weight {1:2, 2:1, 3:10, 6:2}):

Train (acc 0.997)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	1893	8	0	0
Actual 2	11	4253	0	1
Actual 3	0	0	40	0
Actual 6	0	0	0	2358

Test (acc 0.941)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	426	28	0	15
Actual 2	49	1022	0	12
Actual 3	0	1	6	0
Actual 6	20	2	0	561

CNN (without class weight):

Train (acc 0.997)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	1900	1	0	0
Actual 2	7	4257	0	1
Actual 3	0	1	39	0
Actual 6	12	3	0	2343

Test (acc 0.951)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	435	26	0	8
Actual 2	22	1055	0	6
Actual 3	1	2	4	0
Actual 6	26	12	0	545

CNN (with class weight {1:2, 2:1, 3:10, 6:2}):

Train (acc 0.993)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	1894	6	0	1
Actual 2	32	4232	0	1
Actual 3	0	0	40	0
Actual 6	4	11	0	2343

Test (acc 0.941)

	Predicted 1	Predicted 2	Predicted 3	Predicted 6
Actual 1	425	38	0	6
Actual 2	51	1022	0	10
Actual 3	0	1	6	0
Actual 6	9	12	0	562

Parameters:

XGBoost:

```
model = xgb.XGBClassifier(max_depth=8,
                           learning_rate=0.3,
                           n_estimators=1000,
                           objective='multi:softmax',
                           seed=100,
                           subsample=0.7,
                           reg_lambda = 21,
                           )
```

ANN:

```
l1 = layers.Dense(128,activation='relu')(input_)
drop1 = layers.Dropout(0.2)(l1)
l2 = layers.Dense(64,activation='relu')(drop1)
drop2 = layers.Dropout(0.2)(l2)
l3 = layers.Dense(32,activation='relu')(drop2)
drop3 = layers.Dropout(0.2)(l3)
output = layers.Dense(4,activation='softmax')(drop3)
```

CNN:

```
max_pool = layers.MaxPooling2D((2,4))(input_)
conv1 = layers.Conv2D(16,3,strides=(1,1),
                      kernel_initializer=TruncatedNormal(),
                      activation='elu',padding='same')(max_pool)
max_pool1 = layers.MaxPooling2D((2,2))(conv1)
conv2 = layers.Conv2D(32,3,strides=(1,1),
                      kernel_initializer=TruncatedNormal(),
                      activation='elu',padding='same')(max_pool1)
max_pool2 = layers.MaxPooling2D(2)(conv2)
conv3 = layers.Conv2D(64,3,strides=(1,1),
                      kernel_initializer=TruncatedNormal(),
                      activation='elu',padding='same')(max_pool2)
```

```
max_pool3 = layers.MaxPooling2D(2)(conv3)
conv4 = layers.Conv2D(128,3, strides=(1,1),
                      kernel_initializer=TruncatedNormal(),
                      activation='elu',padding='same')(max_pool3)
flatten = layers.Flatten()(conv4)
dropout = layers.Dropout(0.5)(flatten)
layer1 = layers.Dense(256,activation = 'elu')(dropout)
dropout1 = layers.Dropout(0.5)(layer1)
layer2 = layers.Dense(128,activation = 'elu')(dropout1)
dropout2 = layers.Dropout(0.5)(layer2)
output = layers.Dense(4,activation = 'softmax')(dropout2)
```