In [1]:

```
import pandas as pd
```

In [3]:

```
train_df = pd.read_csv("wat-time-interval-1000.csv")
```

In [4]:

```
train_df.head()
```

Out[4]:

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_east	inport_west	outport_loca
0	1000	65	20	6	4	46	27	22	19	21	...	14	20	6
1	1000	66	86	22	12	47	70	29	26	27	...	33	34	13
2	2000	77	91	15	10	51	61	10	4	28	...	37	44	13
3	2000	68	24	10	7	43	12	8	0	21	...	15	16	6
4	3000	47	10	9	1	31	1	2	2	31	...	13	10	4

5 rows × 34 columns

In [5]:

```
train_X = train_df.drop(columns=['target'])
```

In [6]:

```
train_X
```

Out[6]:

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_south	inport_east	inport_v
0	1000	65	20	6	4	46	27	22	19	21	...	100	14	
1	1000	66	86	22	12	47	70	29	26	27	...	146	33	
2	2000	77	91	15	10	51	61	10	4	28	...	109	37	
3	2000	68	24	10	7	43	12	8	0	21	...	65	15	
4	3000	47	10	9	1	31	1	2	2	31	...	65	13	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
6298	3150000	77	24	18	18	41	12	6	12	35	...	47	54	
6299	3150000	81	39	30	18	42	26	24	24	24	...	96	70	
6300	3151000	73	36	30	18	36	12	18	6	30	...	66	66	
6301	3151000	69	15	6	6	42	16	6	12	36	...	30	62	
6302	3152000	78	42	36	18	36	12	22	6	30	...	66	77	

6303 rows × 33 columns

In [8]:

```
from keras.utils import to_categorical
```

```
Using TensorFlow backend.
```

In [12]:

```
train_Y = to_categorical(train_df.target)
```

In [13]:

```
train_Y
```

Out[13]:

```
array([[0., 1.],
       [1., 0.],
       [1., 0.],
       ...,
       [1., 0.],
       [0., 1.],
       [0., 1.]], dtype=float32)
```

In [15]:

```
from keras.models import Sequential
from keras.layers import Dense
```

In [16]:

```
model = Sequential()
```

In [17]:

```
n_cols = train_X.shape[1]
```

In [18]:

```
model.add(Dense(250, activation='relu', input_shape=(n_cols,)))
model.add(Dense(250, activation='relu'))
model.add(Dense(250, activation='relu'))
model.add(Dense(2, activation='softmax'))
```

In [19]:

```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

In [23]:

```
from keras.callbacks import EarlyStopping
```

In [24]:

```
early_stopping_monitor = EarlyStopping(patience=3)
```

In [25]:

```
model.fit(train_X, train_Y, epochs=30, validation_split=0.2, callbacks=[early_stopping_mon
itor])
```

```
Train on 5042 samples, validate on 1261 samples
Epoch 1/30
5042/5042 [==============================] - 2s 326us/step - loss: 6844.6024
- accuracy: 0.4897 - val_loss: 476.6540 - val_accuracy: 0.5004
Epoch 2/30
5042/5042 [==============================] - 1s 137us/step - loss: 1681.1147
- accuracy: 0.4859 - val_loss: 1528.6891 - val_accuracy: 0.5004
Epoch 3/30
5042/5042 [==============================] - 1s 138us/step - loss: 559.5375 -
accuracy: 0.4921 - val_loss: 108.3863 - val_accuracy: 0.4996
Epoch 4/30
5042/5042 [==============================] - 1s 128us/step - loss: 245.1320 -
accuracy: 0.4960 - val_loss: 407.7500 - val_accuracy: 0.4996
Epoch 5/30
5042/5042 [==============================] - ETA: 0s - loss: 127.3379 - accur
acy: 0.50 - 1s 149us/step - loss: 124.5260 - accuracy: 0.5065 - val_loss: 31
0.6542 - val_accuracy: 0.4996
Epoch 6/30
5042/5042 [==============================] - 1s 139us/step - loss: 17.9907 -
accuracy: 0.4988 - val_loss: 12.7944 - val_accuracy: 0.5004
Epoch 7/30
5042/5042 [==============================] - 1s 132us/step - loss: 5.9631 - a
ccuracy: 0.5028 - val_loss: 10.1529 - val_accuracy: 0.5004
Epoch 8/30
5042/5042 [==============================] - 1s 139us/step - loss: 3.1458 - a
ccuracy: 0.4940 - val_loss: 0.6940 - val_accuracy: 0.5004
Epoch 9/30
5042/5042 [==============================] - 1s 128us/step - loss: 0.6905 - a
ccuracy: 0.5054 - val_loss: 0.6933 - val_accuracy: 0.5004
Epoch 10/30
5042/5042 [==============================] - 1s 140us/step - loss: 0.6898 - a
ccuracy: 0.5065 - val_loss: 0.6932 - val_accuracy: 0.5004
Epoch 11/30
5042/5042 [==============================] - 1s 140us/step - loss: 0.6885 - a
ccuracy: 0.5077 - val_loss: 0.6932 - val_accuracy: 0.5004
Epoch 12/30
5042/5042 [==============================] - 1s 148us/step - loss: 0.6883 - a
ccuracy: 0.5079 - val_loss: 0.6932 - val_accuracy: 0.5004
Epoch 13/30
5042/5042 [==============================] - 1s 125us/step - loss: 0.6874 - a
ccuracy: 0.5089 - val_loss: 0.6932 - val_accuracy: 0.5004
Epoch 14/30
5042/5042 [==============================] - 1s 134us/step - loss: 2.1949 - a
ccuracy: 0.5093 - val_loss: 0.6932 - val_accuracy: 0.5004
Epoch 15/30
5042/5042 [==============================] - 1s 147us/step - loss: 0.6878 - a
ccuracy: 0.4986 - val_loss: 0.6932 - val_accuracy: 0.5004
Epoch 16/30
5042/5042 [==============================] - 1s 141us/step - loss: 0.6862 - a
ccuracy: 0.5091 - val_loss: 0.6932 - val_accuracy: 0.5004
Epoch 17/30
5042/5042 [==============================] - 1s 137us/step - loss: 0.6854 - a
ccuracy: 0.5097 - val_loss: 0.6932 - val_accuracy: 0.5004
```

Out[25]:

```
<keras.callbacks.callbacks.History at 0x2417ab366c8>
```