

CNN : 2D

1.使用model:

右圖是我使用的model，是keras的cnn，是一層Convolution(64,4,4)，activation為relu，加上三層dense，dense的out_dim分別為256 256 2，前面兩層activation為relu，最後一層為softmax。

我嘗試過Convolution兩層三層四層，效果都不是很好。也有嘗試過convolution加maxpooling交錯各兩層三層，效果不彰。

```
model.add(Convolution2D(64,4,4,
input_shape=x_train.shape[1:], activation='relu'))
model.add(Flatten())
model.add(Dense(output_dim=256))
model.add(Activation('relu'))
model.add(Dense(output_dim=256))
model.add(Activation('relu'))
model.add(Dense(output_dim=2))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam',metrics=[recall,precision])
model.summary()
model.fit(x_train, y_train, nb_epoch=5, batch_size=100, class_weight = class_weight)
```

2.其他細節

首先，我有調class_weight，是使用compute_class_weight所產生的。

再來是因為過了softmax之後，predict出來的值會是該筆資料為0 or 1的probability，我有設threshold，這個threshold為training data中label為1的資料，經過model.predict後為1的機率，取最小的那個，並除以50，只要testing data predict出來的機率大於這個機率，我們就判定他為1，否則為0。

除以50是調出來的，有試過不除，除以10除以100，最好的是除以50。

3.不足

這個model對三份的predict結果都可以蠻高的（recall precision都0.8 up），但偶爾會突然跑出一個偏低的（recall precision都突然降至0.4），所以我還在繼續調參數跟試不同的model。

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 18, 37, 64)	1088
Flatten_1 (Flatten)	(None, 42624)	0
dense_1 (Dense)	(None, 256)	10912000
activation_1 (Activation)	(None, 256)	0
dense_2 (Dense)	(None, 256)	65792
activation_2 (Activation)	(None, 256)	0
dense_3 (Dense)	(None, 2)	514
activation_3 (Activation)	(None, 2)	0

```
class_weight = class_weight.compute_class_weight('balanced',
                                                    np.unique(y_train),
                                                    y_train)
```

```
label_array=np.where(y_train==1)
thresh= min(model.predict(x_train)[label_array][1])/50
```

CNN: 1D

1.使用model:

如右圖，我做的調整是convolution1D中的filter數，分別試過32 64 128 256 512，每一種結果都跑了20次，做平均。

```
def create_model(input_shape):
    model = Sequential()
    model.add(Convolution2D(128, 3, 3, input_shape=input_shape[1:], activation='relu'))
    model.add(Convolution2D(512, 3, 3, input_shape=input_shape[1:], activation='relu'))

    ...

    model.add(Convolution2D(128, 3, 3))

    model.add(Activation('relu'))
    model.add(MaxPooling2D((3, 3)))
    model.add(Convolution2D(128, 3, 3))
    model.add(Activation('relu'))
    model.add(MaxPooling2D((3, 3)))
    ...

    model.add(Flatten())

    model.add(Dense(output_dim=128))
    model.add(Activation('relu'))
    model.add(Dense(output_dim=256))
    model.add(Activation('relu'))
    model.add(Dense(output_dim=256))
    model.add(Activation('relu'))
    model.add(Dense(output_dim=2))
    model.add(Activation('softmax'))

    return model
```

2.結果

	pre_0	rec_0	pre_1	rec_1	pre_2	rec_2	total_pre	total_rec
1D一層								
32	0.778	0.875	0.714	0.417	0.727	0.889	0.741	0.69
64	0.67	0.25	0.714	0.417	0.875	0.778	0.778	0.482
128	0.53	0.9	0.83	0.417	0.72	0.89	0.656	0.724
256	0.5	0.125	0.83	0.417	0.57	0.444	0.667	0.344
512	0.889	0.9	0.77	0.83	0.727	0.889	0.788	0.9

3.分析

在filter數不多的情況下，常常會有某一組的recall超低，大部分是第1組的data。上下浮動較大，之後會使用ensemble降低浮動狀況。

filter數多的時候雖然比較穩定，但會出現全部判斷成0的狀況，還不知道為什麼會發生這種狀況。

Ensemble

1.使用model

我挑的是1D filter是32的，model的架構跟上面提到的一樣

2.參數

我做的實驗包括train 5次或7次，而判斷為1的票數分別有2 3 4 票，結果如下

1D ensemble	pre_0	rec_0	pre_1	rec_1	pre_2	rec_2	total_pre	total_rec
32 1/5	0.53	1	0.83	0.417	0.75	0.7	0.69	0.7
32 2/5	0.89	1	0.83	0.417	0.8	0.89	0.84	0.724
32 3/5	0.67	0.25	0.83	0.417	0.8	0.89	0.79	0.517
32 1/7	0.42	1	0.77	0.417	0.69	0.89	0.6	0.77
32 2/7	0.53	1	0.83	0.417	0.667	0.89	0.63	0.724
32 3/7	0.889	1	0.778	0.58	0.818	0.89	0.828	0.818
32 4/7	0.875	0.875	0.83	0.417	0.8	0.89	0.83	0.7

比上面只train一次的穩定很多，每一次ensemble出來的都差不多。

可以發現5票取2票的performance不錯，取3票的就會變得很差，recall會變低。

而7票最好的是取3票，performance也很高

我覺得最怪的是第1組資料的recall，不管我怎麼調，ensemble也做了，recall總是很低。