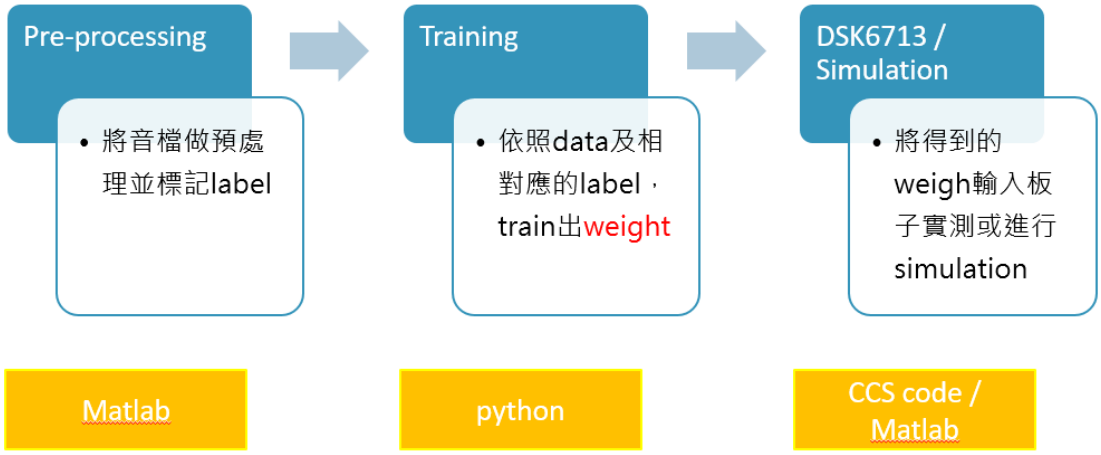
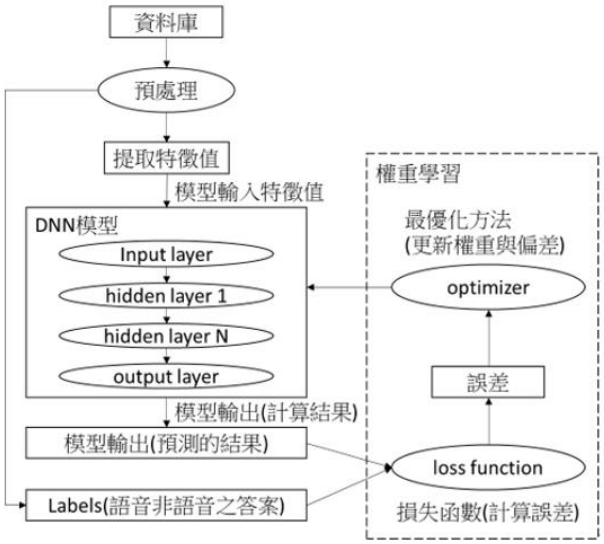
VAD coding tutorial

執行流程



Flow chart



所有流程中**最重要的步驟**，在 Run file 之前凡遇到**要儲存 data 或是 weight** 的地方，請記得重新命名，以免舊的資料被覆蓋。

Step0. Download data

目前 data 類型主要為.wav/.mp3 或是取自板子 memory 的.dat
.dat 的檔案要先將資料轉為.mat 檔以便後續在 matlab 上做 pre-processing

(on Matlab)

(file: transfer_dat_to_mat.m)

Editor - C:\Users\User\Desktop\VAD_code\transfer_dat_to_mat.m

transfer_dat_to_mat.m | data_label_generate.m | simulation.m

This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
1 -   clc;clear all;
2     % close all;
3
4 -   filelist_dat_6713 = dir('.\data\dat\DecRoomOption');
5
6 -   for no=1:length(filelist_dat_6713)-2
7 -       fid = fopen(['.\data\dat\DecRoomOption\',filelist_dat_6713(no+2).name],'r');
8
9 -       datacell = textscan(fid, '%f%f%f', 'HeaderLines', 1, 'Collect', 1);
10 -      fclose(fid);
11 -      A.data = datacell{1};
12 -      s = A.data(:,1);
13
14 -      save(['.\data\noise_mat\DecRoomOption\',filelist_dat_6713(no+2).name,'.mat'],'s');
15
16 -  end
17
18    %%
19
20    % load(['.\data\noise\record_6713\',filelist_dat_6713(no+2).name]);
21    % soundsc(s,16000);
```

.dat 檔案所在資料夾的相對路徑

.dat 檔案所在資料夾的相對路徑

轉成.mat 後要存放資料夾的相對路徑

Step1. Pre-processing (on Matlab)

(file: data_label_generate.m)

(a) 檔案路徑設定

Editor - C:\Users\User\Desktop\VAD_code\data_label_generate.m

transfer_dat_to_mat.m | data_label_generate.m | simulation.m

This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
clc;clear all;close all;
% noise
filelist_noise = dir('.\data\noise');
filelist_noise_mat = dir('.\data\noise_mat');

% clean speech
% filelist_speech = dir('.\data\clean speech');
filelist_data = dir('.\data\clean speech\MIR-1K');
filelist_data_new1 = dir('.\data\clean speech\交大VAD人聲training');
filelist_data_new2 = dir('.\data\clean speech\WeiFangVocal_20180220');
filelist_data_new3 = dir('.\data\clean speech\中英字音_20180730');

x_data = [];
y_data = [];
%% icfilter design
inband = 4;
bandnum = 10;

% midfre = [16 20 26 36 48 60 80 101];  %1024 point 8band
% midfre = [6 7 9 12 16 20 27 35 46 61 80 105 141 184 243 321];  %1024 point 16band
% midfre = [16 20 26 36 48 60 80 101 223];  %1024 point 8+1 3~4k
% midfre = [16 20 26 36 48 60 80 101 353];  %1024 point 8+1 5~6k
midfre = [16 20 26 36 48 60 80 101 256 353];  %1024 point 8+2

slope = 12; %1 2 4 6 8 10 12 14 16 18 20
```

Noise(.wav)所在的相對路徑

Noise(.dat)所在的相對路徑

speech 所在的相對路徑

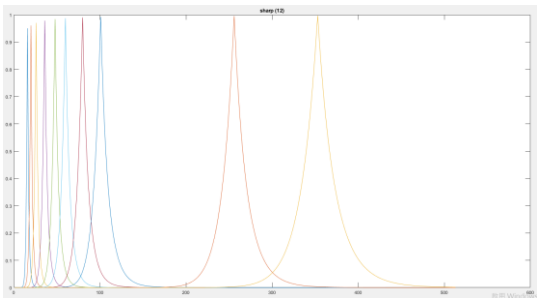
(b) Filter design (參考 VAD 演算法大綱_new p.4 p.5)

```
x_data = [];
y_data = [];
%% icfilter design
inband = 4;
bandnum = 10;

% midfre = [16 20 26 36 48 60 80 101];  %1024 point 8band
% midfre = [6 7 9 12 16 20 27 35 46 61 80 105 141 184 243 321];  %1024 point 16band
% midfre = [16 20 26 36 48 60 80 101 223];  %1024 point 8+1 3~4k
% midfre = [16 20 26 36 48 60 80 101 353];  %1024 point 8+1 5~6k
midfre = [16 20 26 36 48 60 80 101 256 353];  %1024 point 8+2

slope = 12; %1 2 4 6 8 10 12 14 16 18 20

for i=1:bandnum
    if i==9
        for j=1:513
            icfilter(i,j)=10^((-360*abs(log10(15.625*(j-1)+1)-log10(15.625*(midfre(i)-1))))/20);
        end

    elseif i==10
        for j=1:513
            icfilter(i,j)=10^((-360*abs(log10(15.625*(j-1)+1)-log10(15.625*(midfre(i)-1))))/20);
        end

    else
        for j=1:513
            icfilter(i,j)=10^((-20*slope*abs(log10(15.625*(j-1)+1)-log10(15.625*(midfre(i)-1))))/20);
        end
    end
end
```

中心頻

斜率

Filter design



(c) 生成 noise data 與 label (參考 VAD 演算法大鋼 new p.7 p.8)

Editor - C:\Users\User\Desktop\VAD_code\data_label_generate.m

transfer_dat_to_mat.m | data_label_generate.m | simulation.m

This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
43 -    end
44
45      %% noise for .wav / .mp3
46 -    for N=1:length(filelist_noise)-2
47 -        thisdir = filelist_noise(N+2).name;
48 -        new_filelist_noise = dir(['.\data\noise\',thisdir]);
49 -        for no=1:length(new_filelist_noise)-2
50 -            y = 0;
51 -            x_all = zeros(1,inband*bandnum);
52 -            [s fs] = audioread(['.\data\noise\',thisdir,'\',new_filelist_noise(no+2).name]);
53 -            s = resample(s(:,1),16000,fs);  %44100 to 16000
54 -            fs = 16000;
55 -            for i=1601:400:400*(floor(length(s)/400))+1  %0.25s
56                  %label
57 -                y = [y,0];
58                  %8bins
59 -                s_fft_1 = fft(s(i-1600:i-1201),1024);
60 -                s_fft_2 = fft(s(i-1200:i-801),1024);
61 -                s_fft_3 = fft(s(i-800:i-401),1024);
62 -                s_fft_4 = fft(s(i-400:i-1),1024);
63
64 -                x = [log(icfilter*(abs(s_fft_1(1:513)).^2)+0.0001)'...
65                      log(icfilter*(abs(s_fft_2(1:513)).^2)+0.0001)'...
66                      log(icfilter*(abs(s_fft_3(1:513)).^2)+0.0001)'...
67                      log(icfilter*(abs(s_fft_4(1:513)).^2)+0.0001)'];
68
69 -                x_all = [x_all;x];
70 -            end
71 -            x_data = [x_data;x_all(2:length(x_all(:,1)),:)];
72 -            y_data = [y_data,y(2:length(y))];
73 -        end
74 -    end
75
```

讀取 Noise 所在資料夾

讀取資料夾中的音檔

Noise 標記 label 為 0

每 25ms 分別做 fft

每 25ms 分別與 filter 相乘

```
    %% noise for .mat
77 -    for N=1:length(filelist_noise_mat)-2
78 -        thisdir = filelist_noise_mat(N+2).name;
79 -        new_filelist_noise = dir(['.\data\noise_mat\',thisdir]);
80 -        for no=1:length(new_filelist_noise)-2
81 -            y = 0;
82 -            x_all = zeros(1,inband*bandnum);
83 -            load(['.\data\noise_mat\',thisdir,'\',new_filelist_noise(no+2).name]);
84 -            for i=1601:400:400*(floor(length(s)/400))+1  %0.25s
85                  %label
86 -                y = [y,0];
87                  %8bins
88 -                s_fft_1 = fft(s(i-1600:i-1201),1024);
89 -                s_fft_2 = fft(s(i-1200:i-801),1024);
90 -                s_fft_3 = fft(s(i-800:i-401),1024);
91 -                s_fft_4 = fft(s(i-400:i-1),1024);
92
93 -                x = [log(icfilter*(abs(s_fft_1(1:513)).^2)+0.0001)'...
94                      log(icfilter*(abs(s_fft_2(1:513)).^2)+0.0001)'...
95                      log(icfilter*(abs(s_fft_3(1:513)).^2)+0.0001)'...
96                      log(icfilter*(abs(s_fft_4(1:513)).^2)+0.0001)'];
97
98 -                x_all = [x_all;x];
99 -            end
100 -           x_data = [x_data;x_all(2:length(x_all(:,1)),:)];
101 -           y_data = [y_data,y(2:length(y))];
102 -       end
103 -   end
```

(d) 生成 **speech data** 與 **label** (參考 VAD 演算法大鋼 new p.7 p.8)

Editor - C:\Users\User\Desktop\VAD_code\data_label_generate.m

transfer_dat_to_mat.m | data_label_generate.m | simulation.m

This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
105     %% clean speech MIR-1k
106 -   y = 0;
107 -   x = [];
108 -   x_all = zeros(1,inband*bandnum);
109
110 -   for da=1:length(filelist_data)-2
111 -       [s fs] = audioread(['.\data\clean speech\MIR-1K\',filelist_data(da+2).name]);
112 -       s = resample(s,16000,fs);
113 -       fs = 16000;
114
115     %     if length(s)>16000*40
116     %         s = s(1:16000*40);  % 110*8=880
117     %     end
118 -       t = s;
119
120 -       for i=1601:400:400*(floor(length(s)/400))+1  %0.25s
121 -           if(log(norm(t(i-1600:i-1))/norm(t)+0.0001)>-3)  %-3%
122                 %label
123 -               y = [y,1];
124
125 -               s_fft_1 = fft(s(i-1600:i-1201),1024);
126 -               s_fft_2 = fft(s(i-1200:i-801),1024);
127 -               s_fft_3 = fft(s(i-800:i-401),1024);
128 -               s_fft_4 = fft(s(i-400:i-1),1024);
129
130 -               x = [log(icfilter*(abs(s_fft_1(1:513)).^2)+0.0001)'...
131                     log(icfilter*(abs(s_fft_2(1:513)).^2)+0.0001)'...
132                     log(icfilter*(abs(s_fft_3(1:513)).^2)+0.0001)'...
133                     log(icfilter*(abs(s_fft_4(1:513)).^2)+0.0001)'];
134
135 -               x_all = [x_all;x];
136 -           end
137 -       end
```

讀取資料夾中的 speech 音檔並 resample 到 16kHz

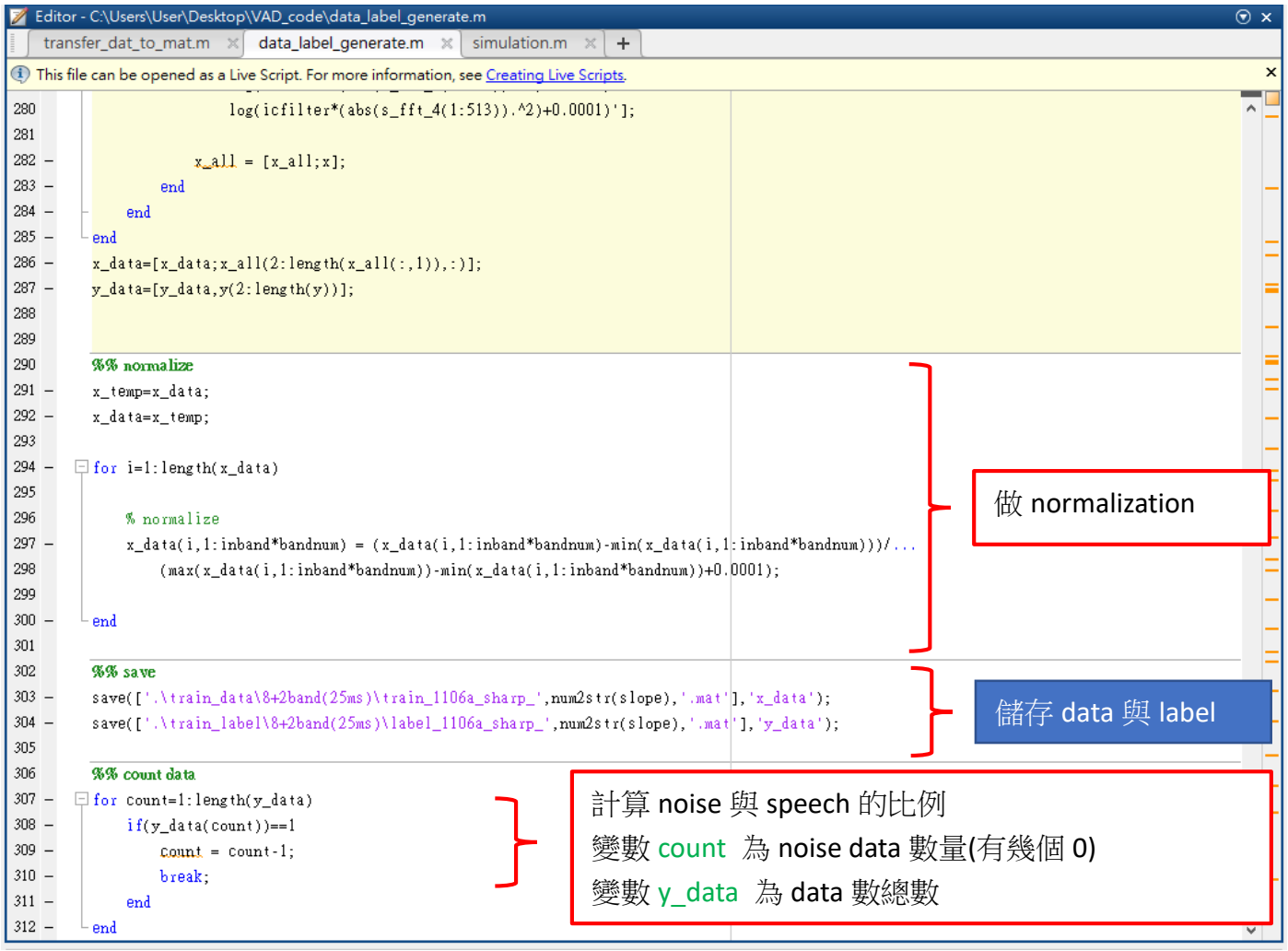
可選擇每首 speech 要使用多少秒 (ex:這裡是取 40 秒)

speech 標記 label 為 1

每 25ms 分別做 fft

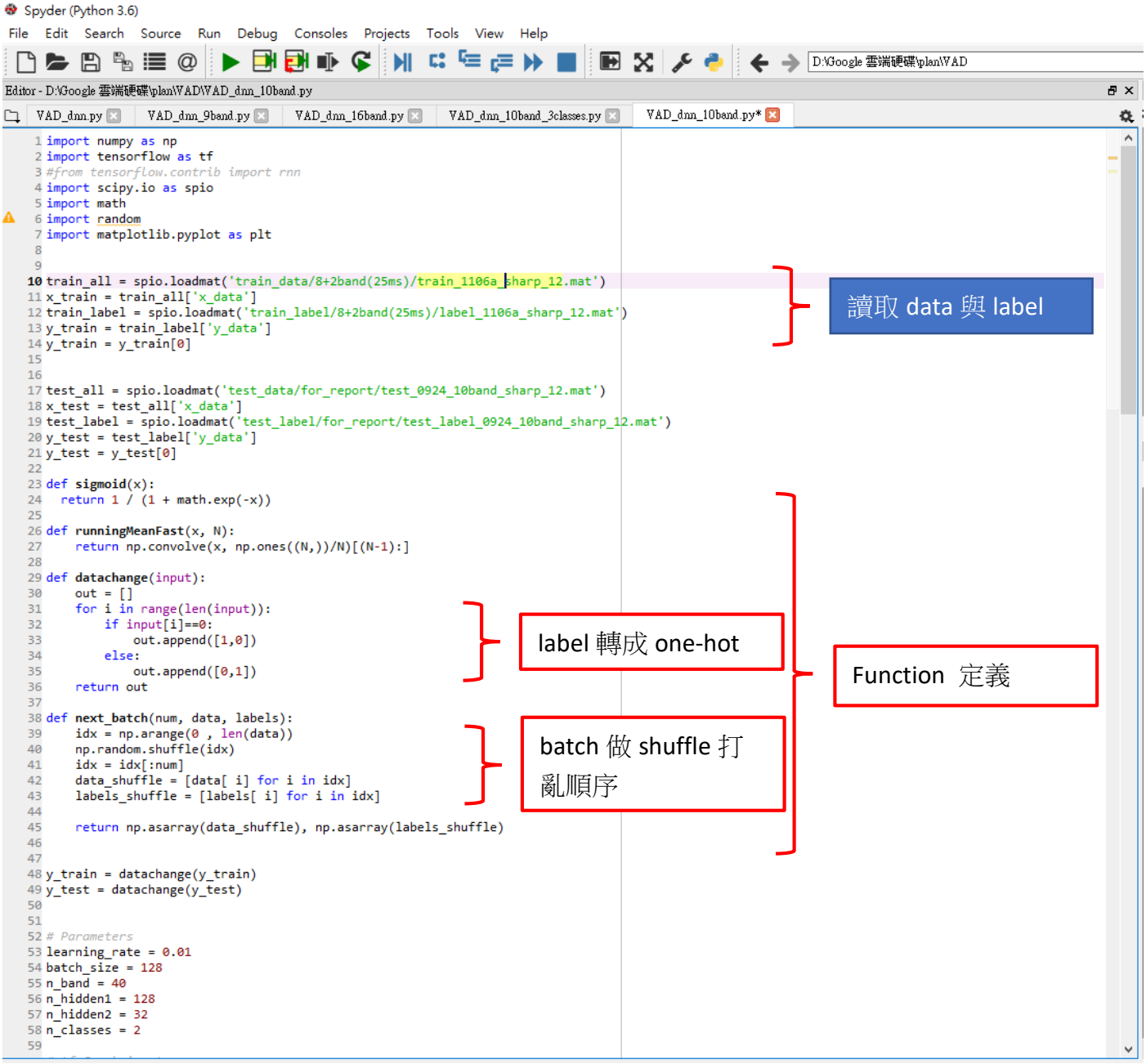
每 25ms 分別與 filter 相乘

(e) Normalization 與存檔 (參考 VAD 演算法大綱 new p.9)

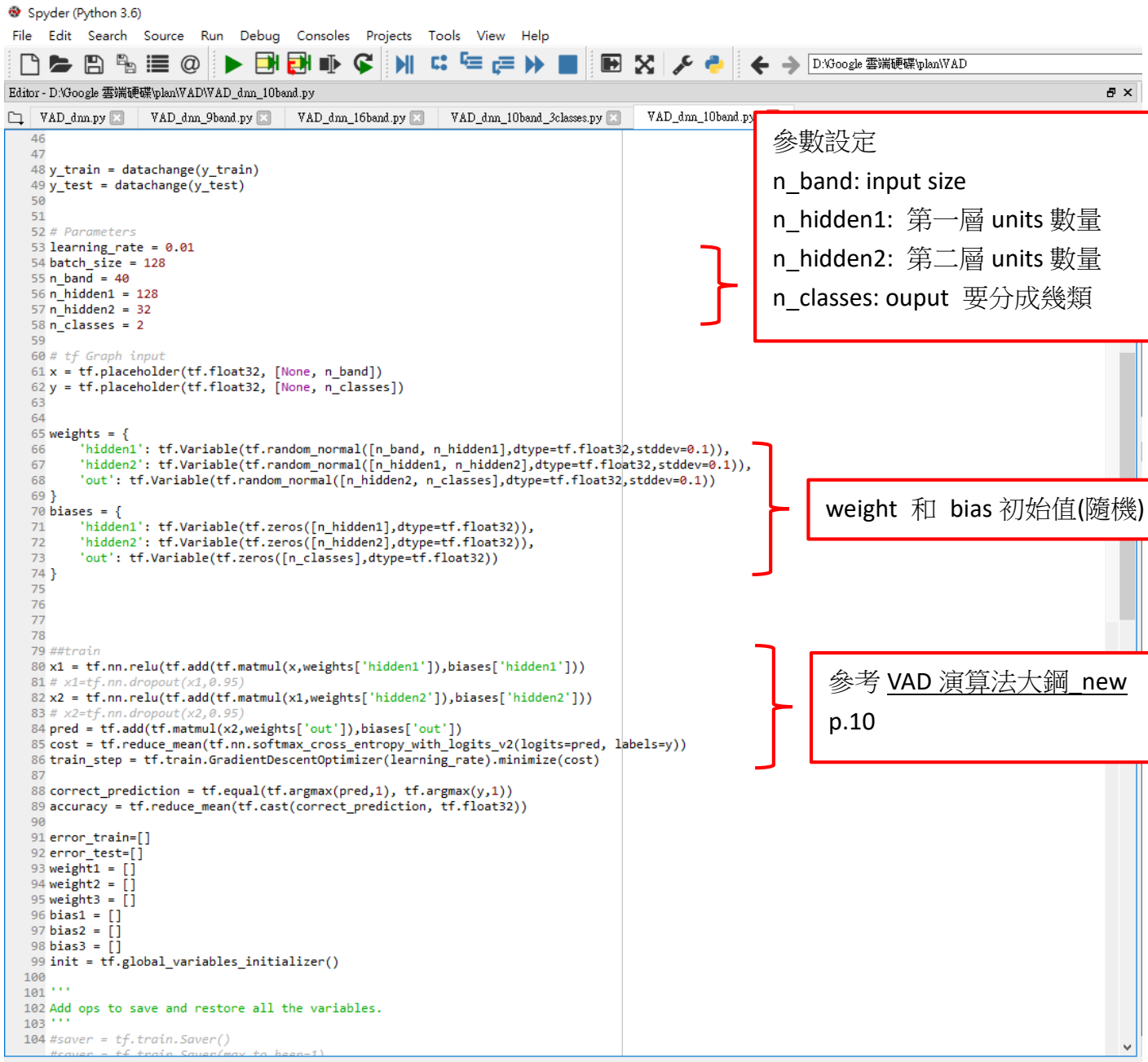


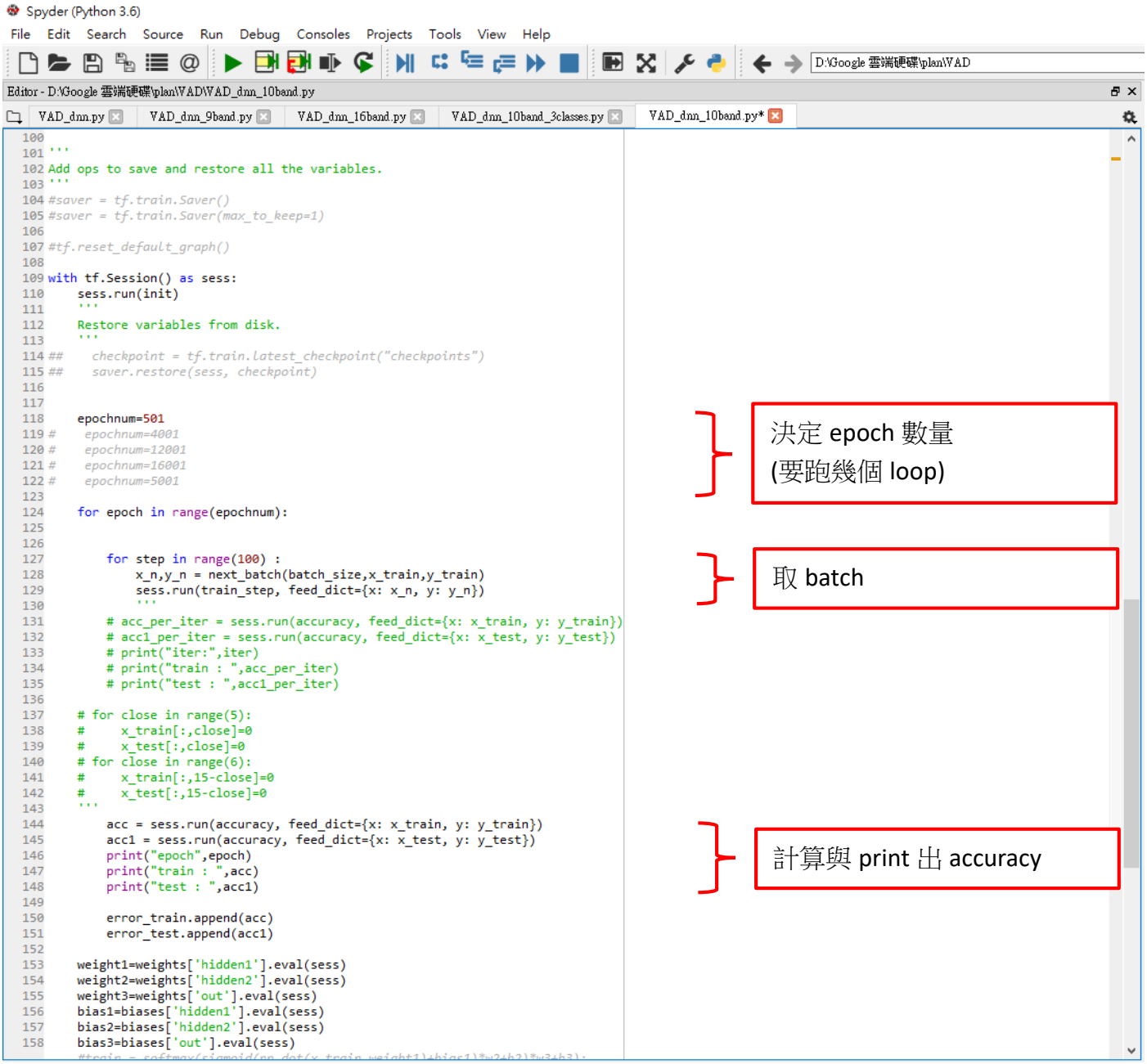
Step2. Training (on Python)

(file: VAD_dnn_10band.py)




```python
import numpy as np
import tensorflow as tf
#from tensorflow.contrib import rnn
import scipy.io as spio
import math
import random
import matplotlib.pyplot as plt


train_all = spio.loadmat('train_data/8+2band(25ms)/train_1106a_sharp_12.mat')
x_train = train_all['x_data']
train_label = spio.loadmat('train_label/8+2band(25ms)/label_1106a_sharp_12.mat')
y_train = train_label['y_data']
y_train = y_train[0]


test_all = spio.loadmat('test_data/for_report/test_0924_10band_sharp_12.mat')
x_test = test_all['x_data']
test_label = spio.loadmat('test_label/for_report/test_label_0924_10band_sharp_12.mat')
y_test = test_label['y_data']
y_test = y_test[0]

def sigmoid(x):
  return 1 / (1 + math.exp(-x))

def runningMeanFast(x, N):
    return np.convolve(x, np.ones((N,))/N)[(N-1):]

def datachange(input):
    out = []
    for i in range(len(input)):
        if input[i]==0:
            out.append([1,0])
        else:
            out.append([0,1])
    return out

def next_batch(num, data, labels):
    idx = np.arange(0 , len(data))
    np.random.shuffle(idx)
    idx = idx[:num]
    data_shuffle = [data[ i] for i in idx]
    labels_shuffle = [labels[ i] for i in idx]

    return np.asarray(data_shuffle), np.asarray(labels_shuffle)


y_train = datachange(y_train)
y_test = datachange(y_test)


# Parameters
learning_rate = 0.01
batch_size = 128
n_band = 40
n_hidden1 = 128
n_hidden2 = 32
n_classes = 2
```


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\Google 雲端硬碟\plan\VAD
Editor - D:\Google 雲端硬碟\plan\VAD\VAD_dnn_10band.py
VAD_dnn.py
VAD_dnn_9band.py
VAD_dnn_16band.py
VAD_dnn_10band_3classes.py
VAD_dnn_10band.py
y_train = datachange(y_train)
y_test = datachange(y_test)
# Parameters
learning_rate = 0.01
batch_size = 128
n_band = 40
n_hidden1 = 128
n_hidden2 = 32
n_classes = 2
# tf Graph input
x = tf.placeholder(tf.float32, [None, n_band])
y = tf.placeholder(tf.float32, [None, n_classes])
weights = {
'hidden1': tf.Variable(tf.random_normal([n_band, n_hidden1],dtype=tf.float32,stddev=0.1)),
'hidden2': tf.Variable(tf.random_normal([n_hidden1, n_hidden2],dtype=tf.float32,stddev=0.1)),
'out': tf.Variable(tf.random_normal([n_hidden2, n_classes],dtype=tf.float32,stddev=0.1))
}
biases = {
'hidden1': tf.Variable(tf.zeros([n_hidden1],dtype=tf.float32)),
'hidden2': tf.Variable(tf.zeros([n_hidden2],dtype=tf.float32)),
'out': tf.Variable(tf.zeros([n_classes],dtype=tf.float32))
}
##train
x1 = tf.nn.relu(tf.add(tf.matmul(x,weights['hidden1']),biases['hidden1']))
# x1=tf.nn.dropout(x1,0.95)
x2 = tf.nn.relu(tf.add(tf.matmul(x1,weights['hidden2']),biases['hidden2']))
# x2=tf.nn.dropout(x2,0.95)
pred = tf.add(tf.matmul(x2,weights['out']),biases['out'])
cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits_v2(logits=pred, labels=y))
train_step = tf.train.GradientDescentOptimizer(learning_rate).minimize(cost)
correct_prediction = tf.equal(tf.argmax(pred,1), tf.argmax(y,1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
error_train=[]
error_test=[]
weight1 = []
weight2 = []
weight3 = []
bias1 = []
bias2 = []
bias3 = []
init = tf.global_variables_initializer()
'''
Add ops to save and restore all the variables.
'''
#saver = tf.train.Saver()
參數設定
n_band: input size
n_hidden1: 第一層 units 數量
n_hidden2: 第二層 units 數量
n_classes: ouput 要分成幾類
weight 和 bias 初始值(隨機)
參考 VAD 演算法大綱_new
p.10


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\Google 雲端硬碟\plan\VAD
Editor - D:\Google 雲端硬碟\plan\VAD\VAD_dnn_10band.py
VAD_dnn.py
VAD_dnn_9band.py
VAD_dnn_16band.py
VAD_dnn_10band_3classes.py
VAD_dnn_10band.py*
100
101 '''
102 Add ops to save and restore all the variables.
103 '''
104 #saver = tf.train.Saver()
105 #saver = tf.train.Saver(max_to_keep=1)
106
107 #tf.reset_default_graph()
108
109 with tf.Session() as sess:
110 sess.run(init)
111 '''
112 Restore variables from disk.
113 '''
114 ## checkpoint = tf.train.latest_checkpoint("checkpoints")
115 ## saver.restore(sess, checkpoint)
116
117
118 epochnum=501
119 # epochnum=4001
120 # epochnum=12001
121 # epochnum=16001
122 # epochnum=5001
123
124 for epoch in range(epochnum):
125
126
127 for step in range(100) :
128 x_n,y_n = next_batch(batch_size,x_train,y_train)
129 sess.run(train_step, feed_dict={x: x_n, y: y_n})
130 '''
131 # acc_per_iter = sess.run(accuracy, feed_dict={x: x_train, y: y_train})
132 # acc1_per_iter = sess.run(accuracy, feed_dict={x: x_test, y: y_test})
133 # print("iter:",iter)
134 # print("train : ",acc_per_iter)
135 # print("test : ",acc1_per_iter)
136
137 # for close in range(5):
138 # x_train[:,close]=0
139 # x_test[:,close]=0
140 # for close in range(6):
141 # x_train[:,15-close]=0
142 # x_test[:,15-close]=0
143 '''
144 acc = sess.run(accuracy, feed_dict={x: x_train, y: y_train})
145 acc1 = sess.run(accuracy, feed_dict={x: x_test, y: y_test})
146 print("epoch",epoch)
147 print("train : ",acc)
148 print("test : ",acc1)
149
150 error_train.append(acc)
151 error_test.append(acc1)
152
153 weight1=weights['hidden1'].eval(sess)
154 weight2=weights['hidden2'].eval(sess)
155 weight3=weights['out'].eval(sess)
156 bias1=biases['hidden1'].eval(sess)
157 bias2=biases['hidden2'].eval(sess)
158 bias3=biases['out'].eval(sess)
決定 epoch 數量
(要跑幾個 loop)
取 batch
計算與 print 出 accuracy

Spyder (Python 3.6)

File Edit Search Source Run Debug Consoles Projects Tools View Help

D:\Google 雲端硬碟\plan\VAD

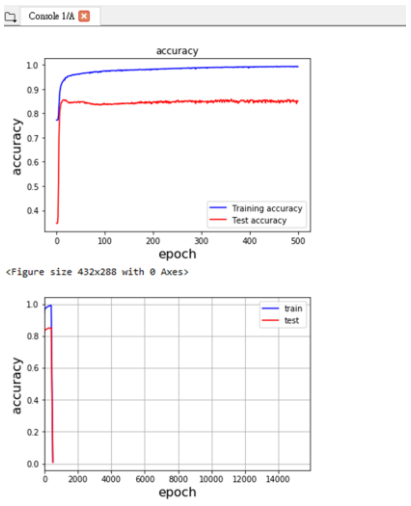
Editor - D:\Google 雲端硬碟\plan\VAD\VAD_dnn_10band.py

VAD_dnn.py | VAD_dnn_9band.py | VAD_dnn_16band.py | VAD_dnn_10band_3classes.py | VAD_dnn_10band.py*

```
    #     x_train[:,close]=0
    #     x_test[:,close]=0
    # for close in range(6):
    #     x_train[:,15-close]=0
    #     x_test[:,15-close]=0
    '''
        acc = sess.run(accuracy, feed_dict={x: x_train, y: y_train})
        acc1 = sess.run(accuracy, feed_dict={x: x_test, y: y_test})
        print("epoch",epoch)
        print("train : ",acc)
        print("test : ",acc1)

        error_train.append(acc)
        error_test.append(acc1)

    weight1=weights['hidden1'].eval(sess)
    weight2=weights['hidden2'].eval(sess)
    weight3=weights['out'].eval(sess)
    bias1=biases['hidden1'].eval(sess)
    bias2=biases['hidden2'].eval(sess)
    bias3=biases['out'].eval(sess)
    #train = softmax(sigmoid(np.dot(x_train,weight1)+bias1)*w2+b2)*w3+b3);

        # print("Optimization Finished!")
        # spio.savemat('accuracy_train.mat', {'accuracy_train': error_train})
        # spio.savemat('accuracy_test.mat', {'accuracy_test': error_test})
    spio.savemat('weight/8+2band(25ms)/w_1106a_sharp_12.mat', {'w1': weight1,'w2':weight2,'w3':weight3,'b1':bias1,'b2':bias2,'b3':bias3})

    '''
    # Save the variables to disk.
    '''
#    save_path = saver.save(sess, "checkpoints_1106a/model_1106a_sharp_12_{}.ckpt".format(epoch))


plt.plot(range(len(error_train)), error_train, 'b', label='Training accuracy')
plt.plot(range(len(error_test)), error_test, 'r', label='Test accuracy')
plt.title('accuracy')
plt.xlabel('epoch',fontsize=16)
plt.ylabel('accuracy',fontsize=16)
plt.legend()
plt.figure()
plt.show()


a = runningMeanFast(error_train, 100)
b = runningMeanFast(error_test, 100)
plt.plot(range(len(a)), a, 'b', label="train")
plt.plot(range(len(b)), b, 'r', label="test")
plt.grid()
plt.xlabel('epoch',fontsize=16)
plt.ylabel('accuracy',fontsize=16)
#plt.xlim(0,11900)
plt.xlim(0,15900)
#plt.xlim(0,9900)
#plt.xlim(0,4900)
plt.legend()
plt.figure()
plt.show()
```

儲存 weight 與 bias (.mat 檔案)

畫圖用 可看收斂趨勢 ex:



Step3. Simulation (on Matlab)

(file: simulation.m)

Editor - C:\Users\User\Desktop\VAD_code\simulation.m

transfer_dat_to_mat.m | data_label_generate.m | simulation.m

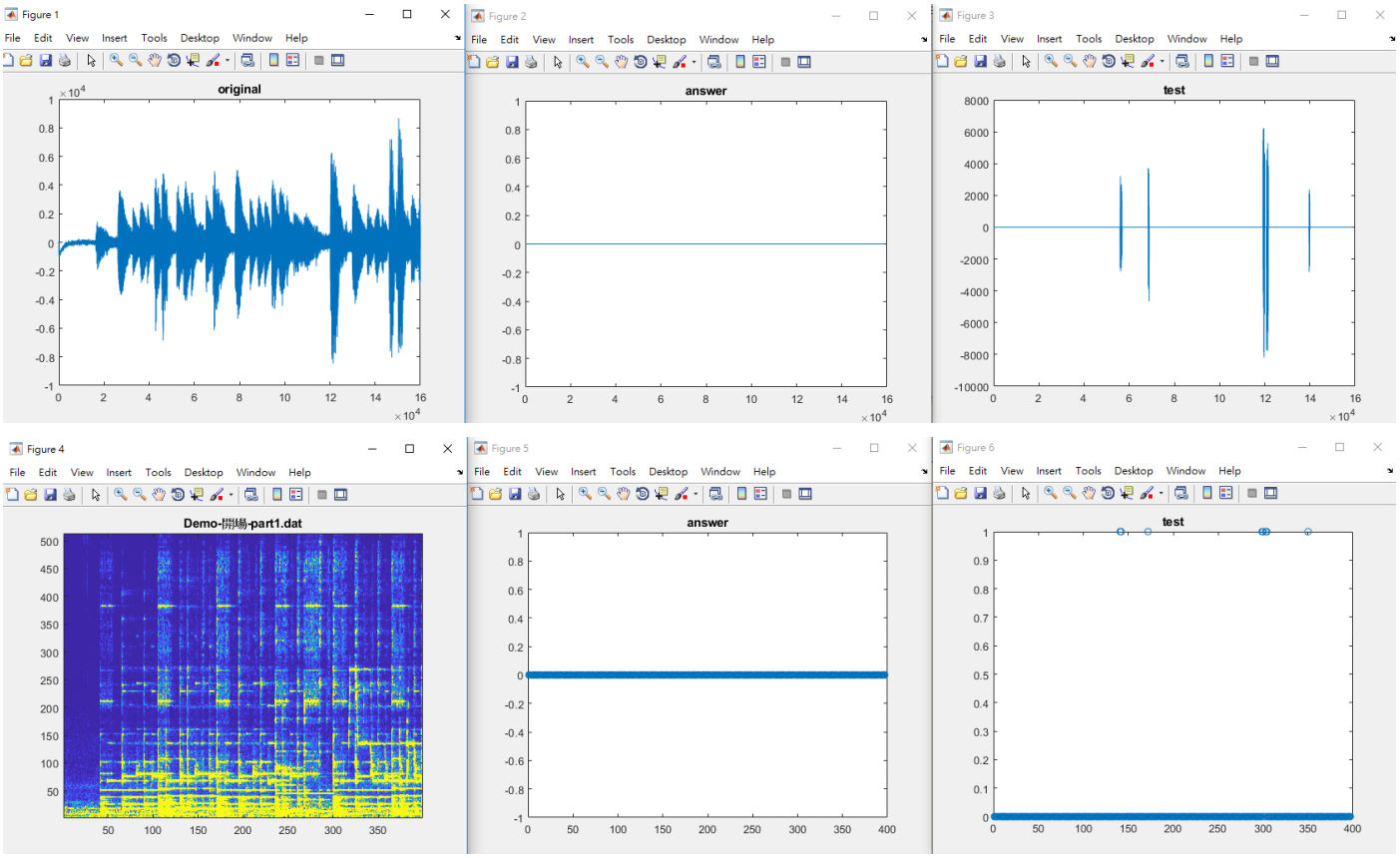
This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
1  clc;clear all;close all;
2
3  slope = 12;
4  % load weight
5  load(['.\weight\8+2band(25ms)\w_1106a_sharp_',num2str(slope)]);
6
7  % load sound (.wav /.mp3)
8  % [s f] = audioread('.\data\noise\Office Noise\mixnoise\快速敲桌子.mp3');
9  % s = resample(s(:,1),16000,f);
10
11 % load sound (.mat)
12 load('.\data\noise_mat\Speakers Train\Demo_開場_part1.dat.mat');
13
14 target = 0; % 0:for noise ; 1:for speech
15
16 t = s;
17
18 % threshold = -250;
19 threshold = -145;
20 % threshold = -10;
21
22 %% icfilter design
23 inband = 4;
24 bandnum = 10;
25
26 % midfre = [16 20 26 36 48 60 80 101];  %1024 point 8band
27 % midfre = [6 7 9 12 16 20 27 35 46 61 80 105 141 184 243 321];  %1024 point 16band
```

讀取 weight

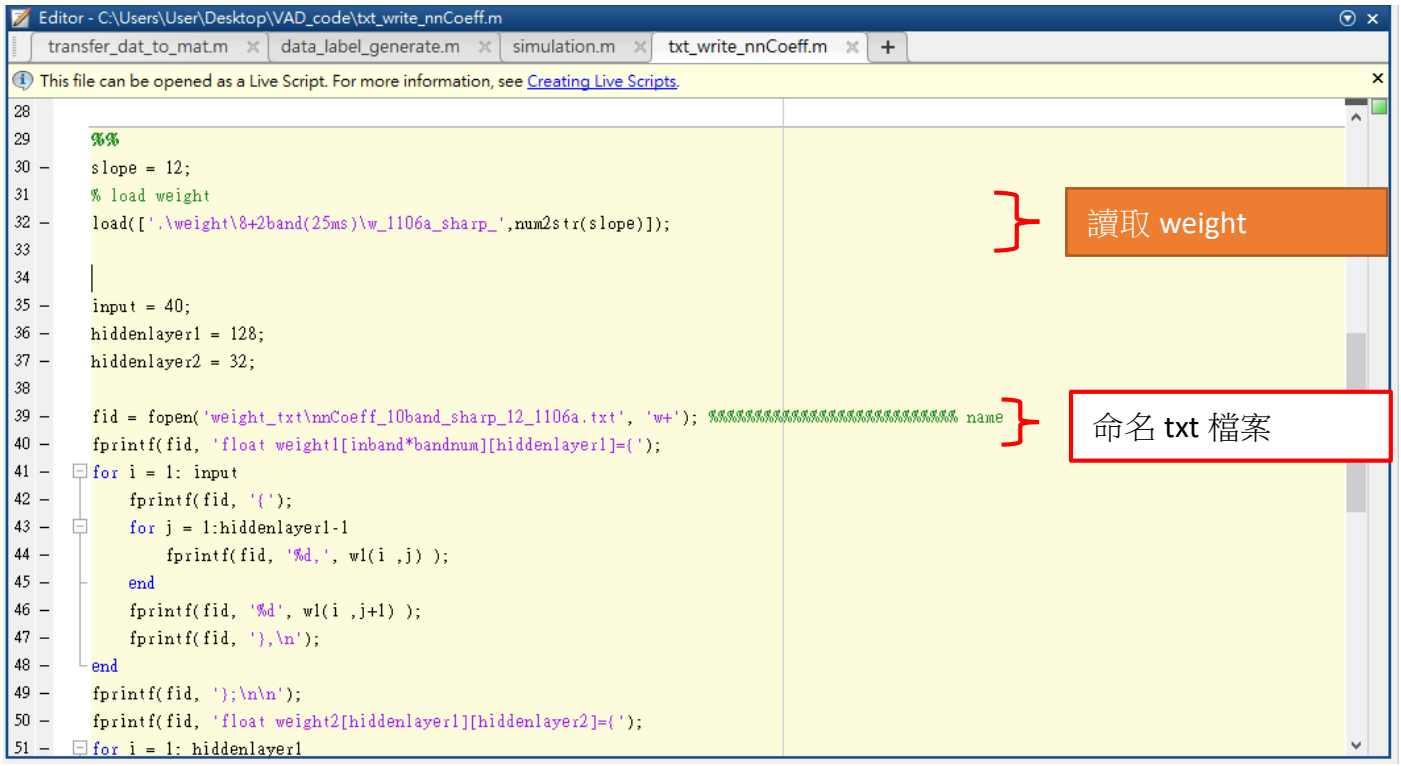
讀取要跑 simulation 的音檔

Simulation 結果



Step4. 輸入板子 DSK6713 前的 txt 轉換 (on Matlab)

(file: txt_write_nnCoeff.m)



命名完後，直接 Run

Step4. DSK6713 (on CCS code)

更新 nnCoeff.h 檔案裡的 weight
完成