TCL 创意感动生活 The Creative Life

Topics on Face Detection and Recognition

Present by: Yixuan Hu (Eason)
TCL Research Co. Ltd (Hong Kong)

Content

1. Introduction to the project background

2. About the face detection

3. About the face recognition and BPNN

4. Demo and Interesting video sharing

5. Q&A

Background

- Application of face detection and recognition

✓ customs port

✓ crime investigation

✓ Camera assistant auto-focus

- What shall we do for our consumer

✓ Recommendation

✓ Authentication based application

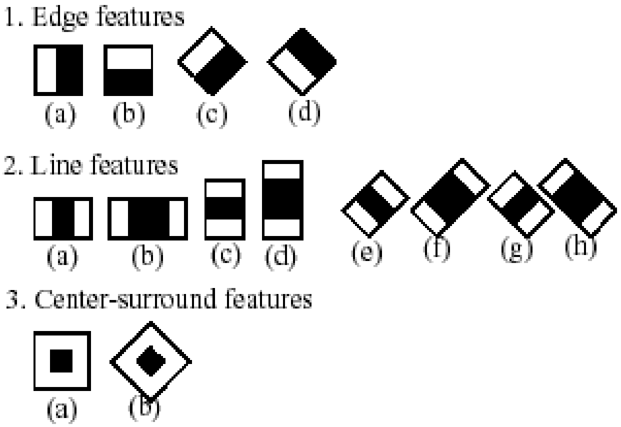
- Machine learning itself is a flexible tool for value add application

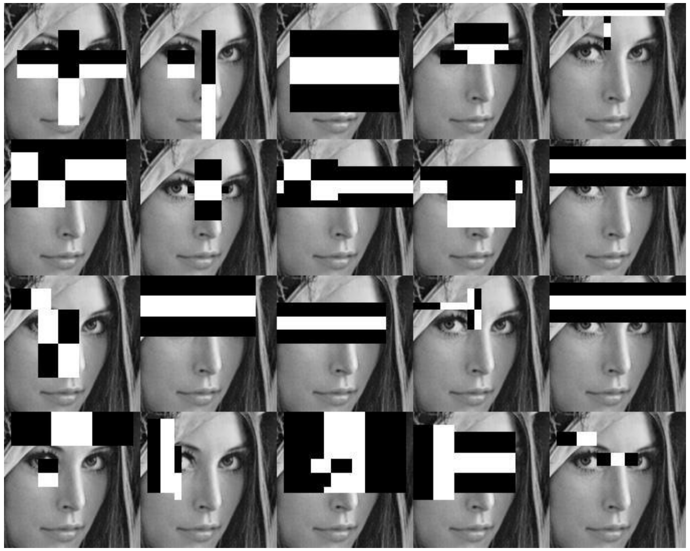
✓ Health related application

✓ Artificial intelligent

Face Detection

- A little bit more detail
  - The Haar-like feature vs. Local Binary Pattern feature



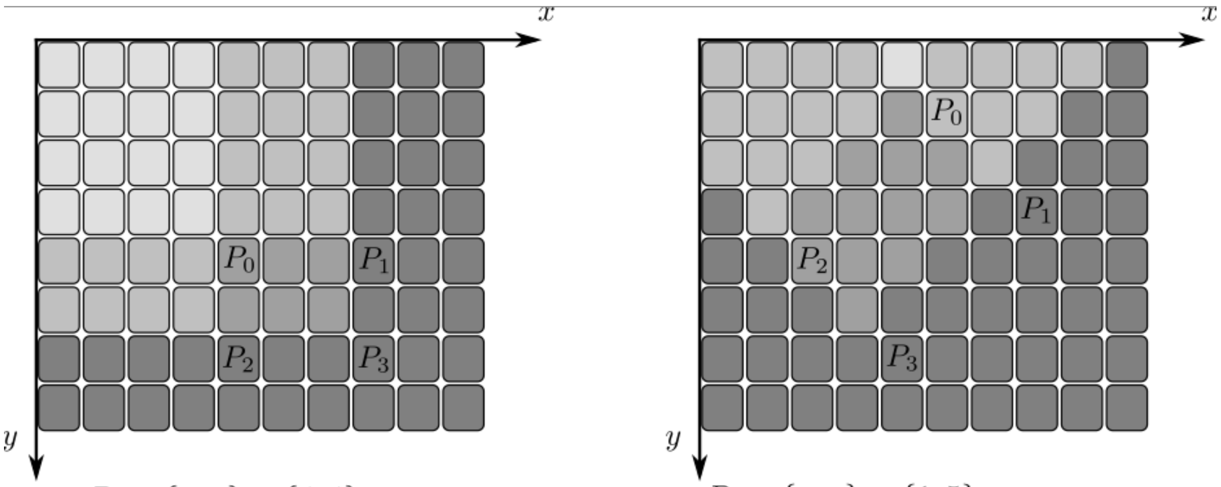


In our solution

- Minimum sliding window size: 30*30 changeable
- Haar feature window size: 20*20 fixed

Face Detection

- A little bit more detail
  - Integral image for acceleration

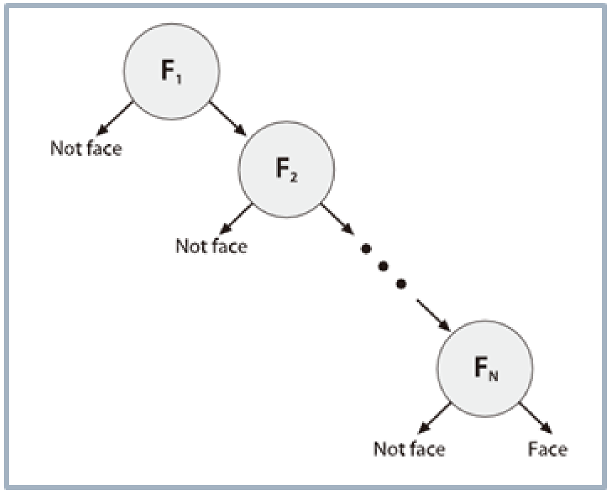


$$P_0 = \{y, x\} = \{4, 4\}$$
$$P_1 = \{y, x + w\} = \{4, 7\}$$
$$P_2 = \{y + h, x\} = \{6, 4\}$$
$$P_3 = \{y + h, x + w\} = \{6, 7\}$$

$$P_0 = \{y, x\} = \{1, 5\}$$
$$P_1 = \{y + w, x + w\} = \{3, 7\}$$
$$P_2 = \{y + h, x - h\} = \{4, 2\}$$
$$P_3 = \{y + w + h, x + w - h\} = \{6, 4\}$$

Face Detection

- A little bit more detail
  - ➢ Adaptive boost cascade classifier



Simplified explanation:

Reject rate = [1 - (1-P)^n]
P: probability of being reject
n: number of stage

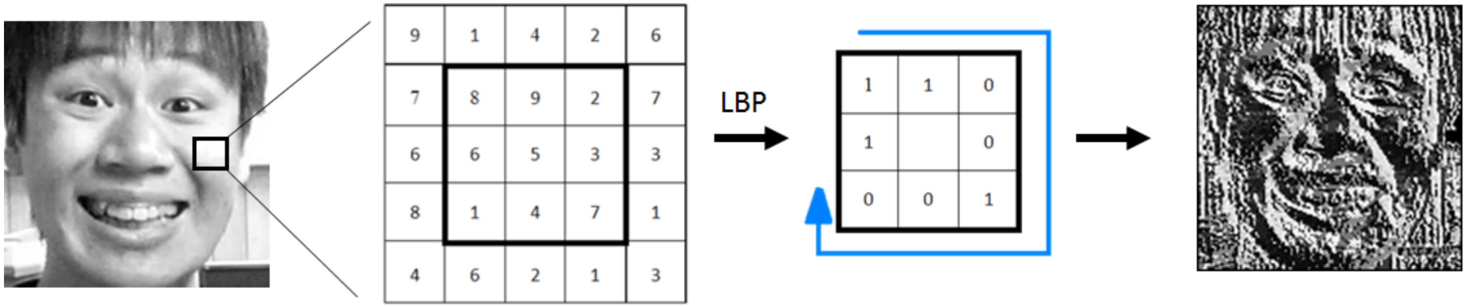
If P = 0.1, n = 20
Reject rate = 0.8784
Cascade of a series of weak classifiers can be a strong classifier

In our solution

➢ Number of stage: around 22 stages varies from model to model

Face Detection

- A little bit more detail
  - ➢ Haar-like feature vs. Local Binary Pattern feature



9	1	4	2	6
7	8	9	2	7
6	6	5	3	3
8	1	4	7	1
4	6	2	1	3

1	1	0
1		0
0	0	1

In our solution

➢ LBP window size: 24*24 bits, fixed
➢ LBP feature length: 3 pixels radius

➢ Haar feature model 22 stages, 661 KB
➢ LBP feature model 20 stages, 51 KB

Face Detection

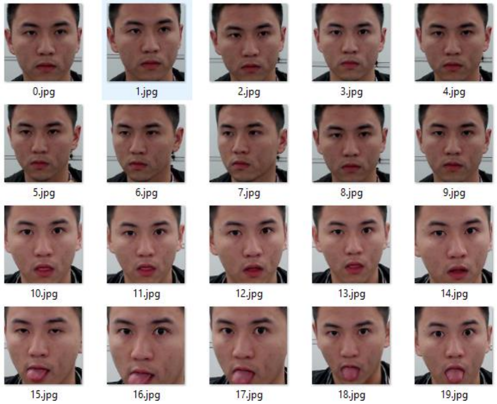
Image1: October 1927 Fifth Solvay International Conference
Source:http://zhblog.engic.org/20140308-212830/

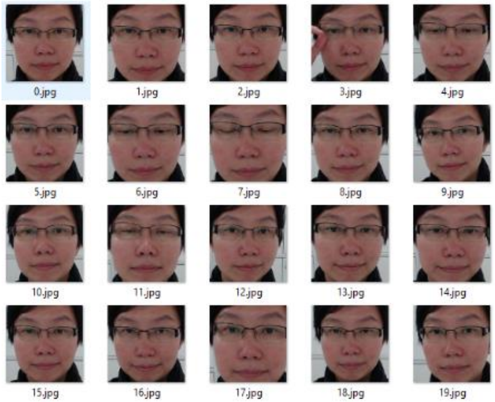
Face Recognition

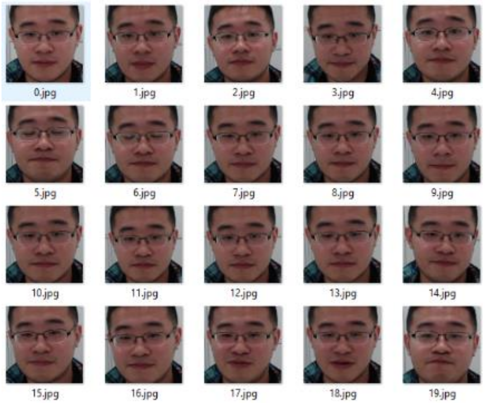
- How to recognize
  - ➢ Methodology key word: template
  - ➢ Extract most representative feature
  - ➢ Pre-process

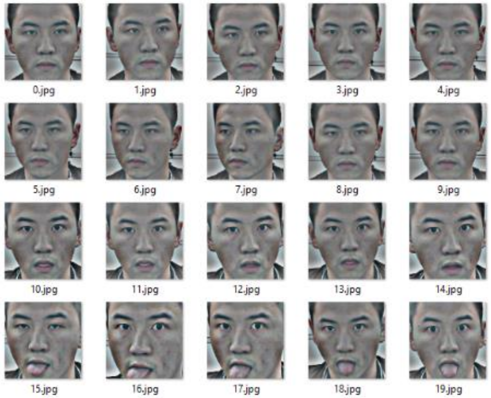
Face Recognition

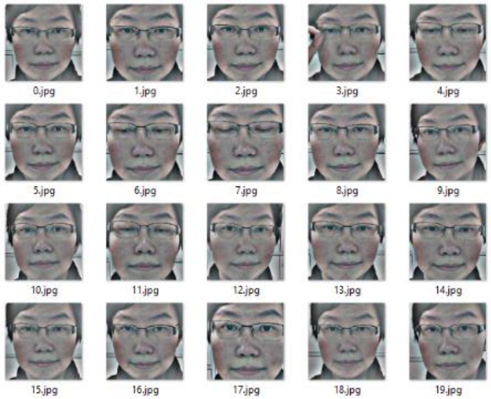
- How to recognize
  - ➢ Pre-process for eye detection and face alignment

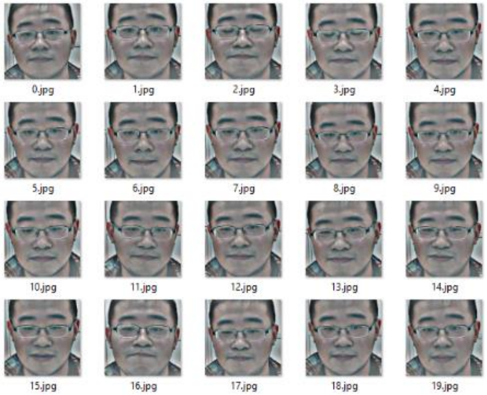






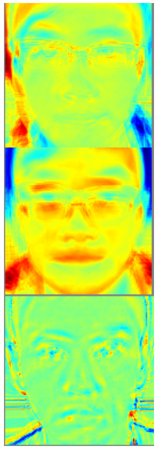




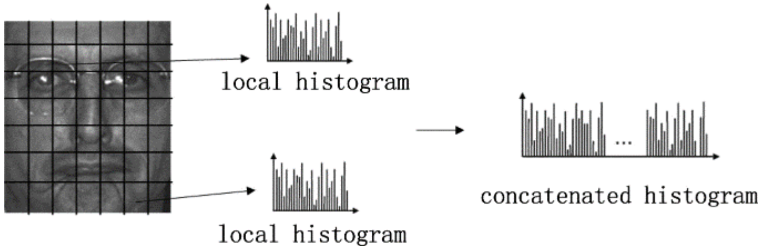


Face Recognition

- How to extract feature
  - Principal Component Analysis
    - Use a set of mathematical completeness orthogonal unified basis to represent the image
    - Calculate the covariance matrix
    - Calculate the eigenvalue and eigenvector of the covariance matrix
    - The vector of eigenvalues represent the subjects and the eigenvectors are basis

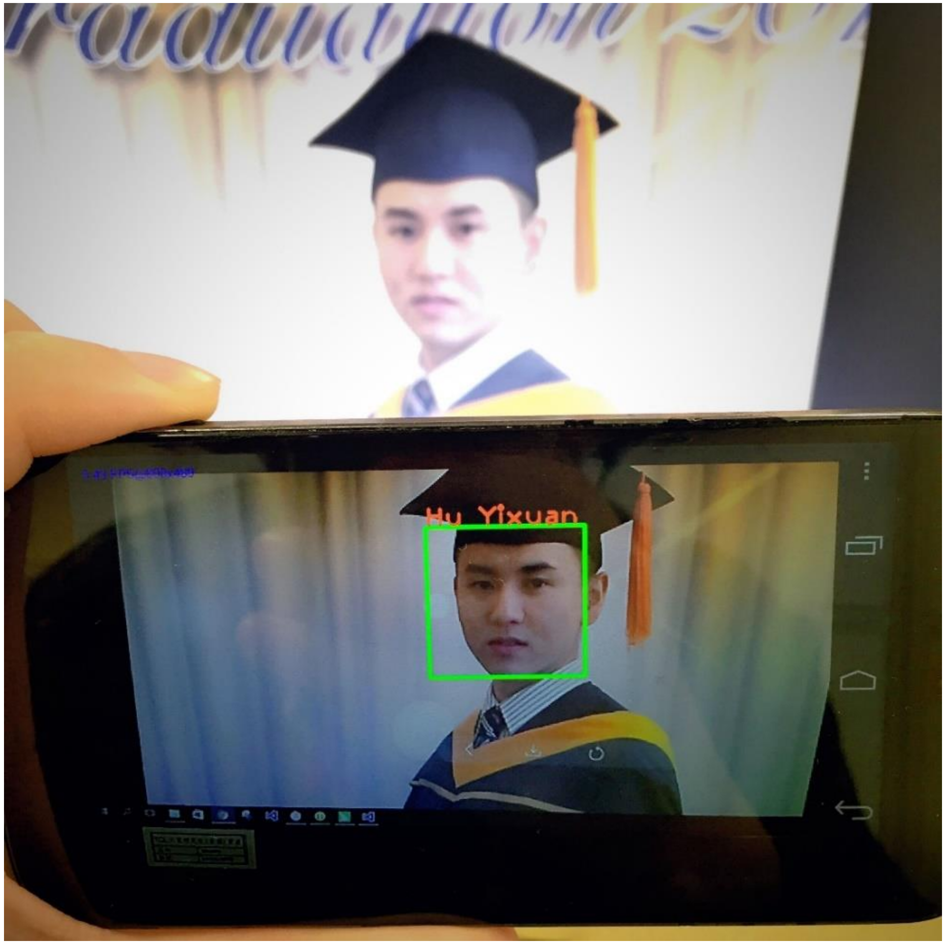


  - Linear Discrimination Analysis
    - Follow a same methodology with the PCA, but the basis of the space is not orthogonal
    - Performance is relatively better than PCA
  - Local Binary Pattern Histogram
    - Calculate the LBP feature in different region of the image
    - Compare the statistics feature with the pre-trained template



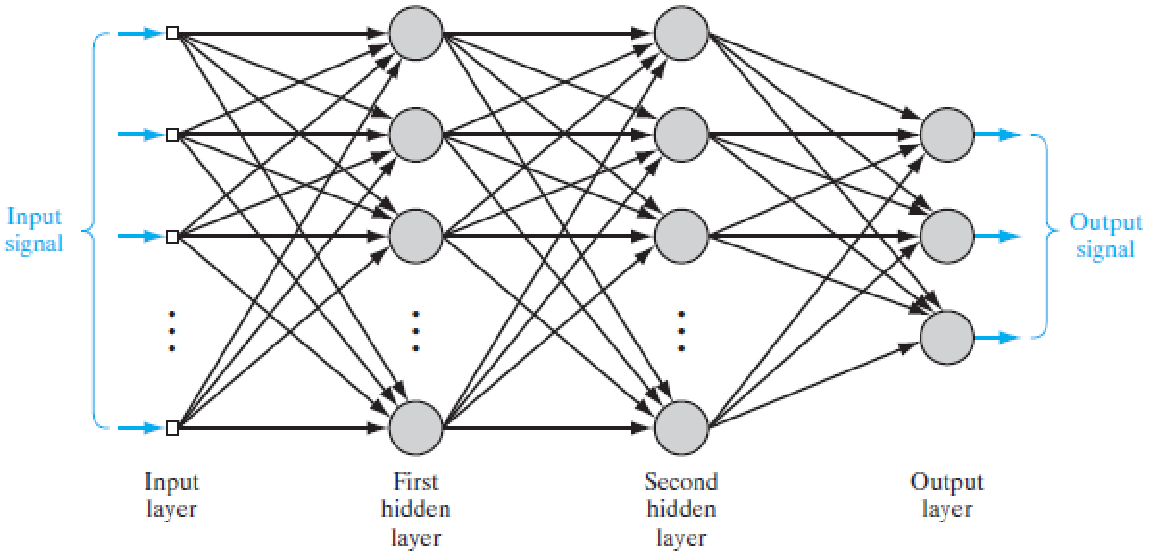
Face Recognition

Performance	LDA	LBPH
Model size	600KB	6.3MB
Frame per second	15-30	6-15
Accuracy	plain	better



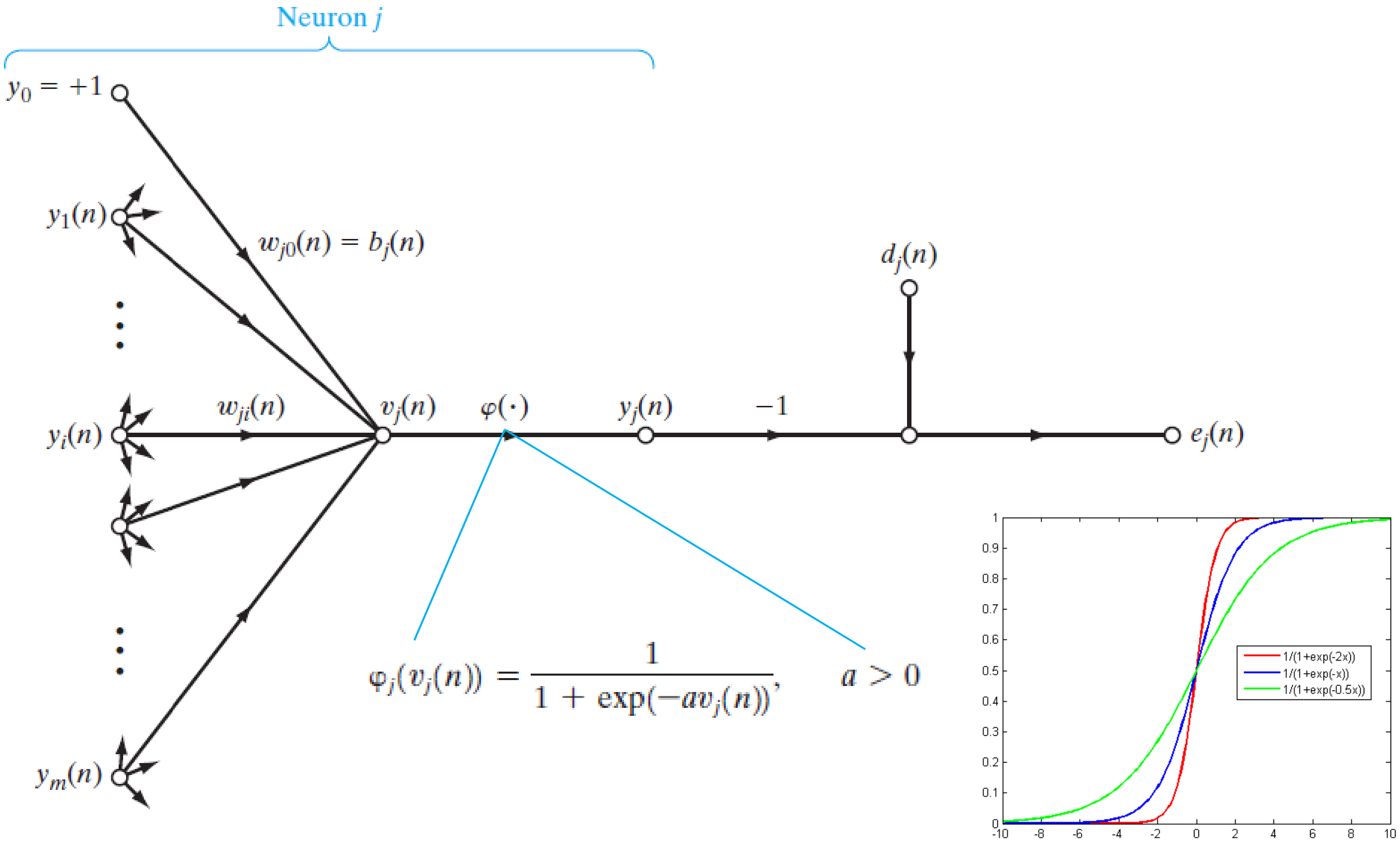
Neutral network

- Still, template, the difference is that the “template” of the neutral network is “learned” from the data through the “network”.



Back Propagation Neutral Network

- A single neuron



$$\varphi_j(v_j(n)) = \frac{1}{1 + \exp(-av_j(n))}, \quad a > 0$$

Back Propagation Neutral Network

- Weight correction

$$\begin{pmatrix} \textit{Weight} \\ \textit{correction} \\ \Delta w_{ji}(n) \end{pmatrix} = \begin{pmatrix} \textit{learning-} \\ \textit{rate parameter} \\ \eta \end{pmatrix} \times \begin{pmatrix} \textit{local} \\ \textit{gradient} \\ \delta_j(n) \end{pmatrix} \times \begin{pmatrix} \textit{input signal} \\ \textit{of neuron } j, \\ y_i(n) \end{pmatrix}$$

$$\begin{aligned} \delta_j(n) &= \frac{\partial \mathscr{E}(n)}{\partial v_j(n)} \\ &= \frac{\partial \mathscr{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \\ &= e_j(n)\varphi_j'(v_j(n)) \end{aligned}$$

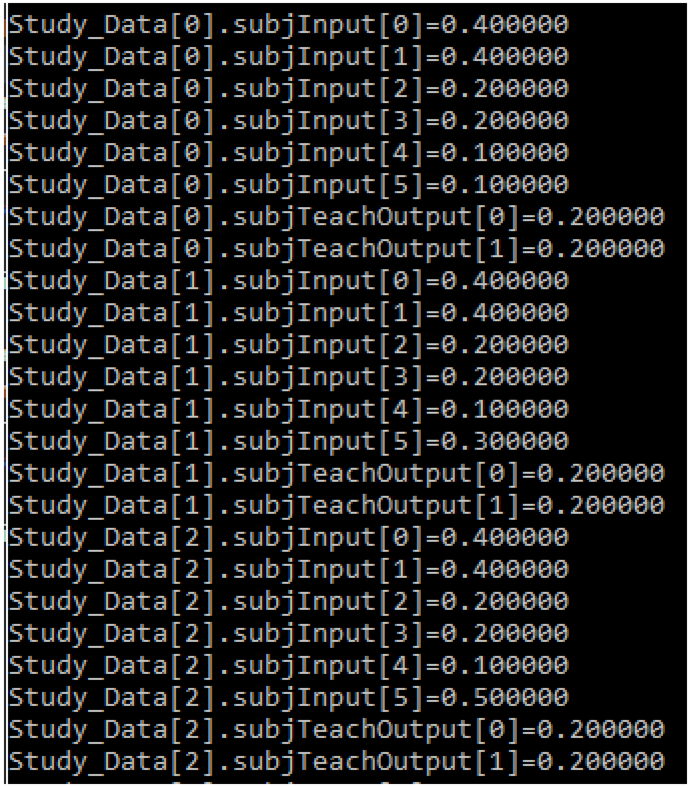
Back Propagation Neutral Network

- About the data

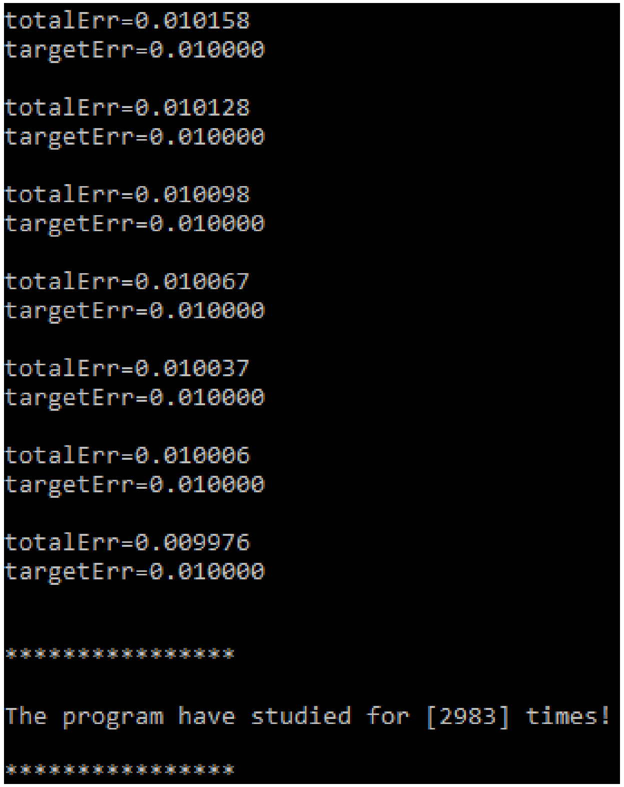
	Unacceptable	Accpetable	Good	Very good
CLASS	(F, F)	(F, T)	(T, F)	(T, T)
Price	4	4	2	2
Maint	4	2	1	1
Doors	2	2	3	4
Persons	2	4	4	4
Luggage	1	3	5	5
Safety	1	5	3	5

Data and output

```
0.4 0.4 0.2 0.2 0.1 0.1
0.4 0.4 0.2 0.2 0.1 0.3
0.4 0.4 0.2 0.2 0.1 0.5
0.4 0.2 0.2 0.4 0.3 0.5
0.4 0.2 0.2 0.4 0.5 0.5
0.4 0.2 0.2 0.5 0.5 0.5
0.2 0.1 0.3 0.4 0.1 0.5
0.2 0.1 0.3 0.4 0.3 0.5
0.2 0.1 0.3 0.4 0.5 0.3
0.2 0.1 0.3 0.5 0.3 0.5
0.2 0.1 0.4 0.4 0.3 0.5
0.2 0.1 0.4 0.4 0.5 0.5
0.2 0.2
0.2 0.2
0.2 0.2
0.2 0.8
0.2 0.8
0.2 0.8
0.8 0.2
0.8 0.2
0.8 0.2
0.8 0.8
0.8 0.8
0.8 0.8
```

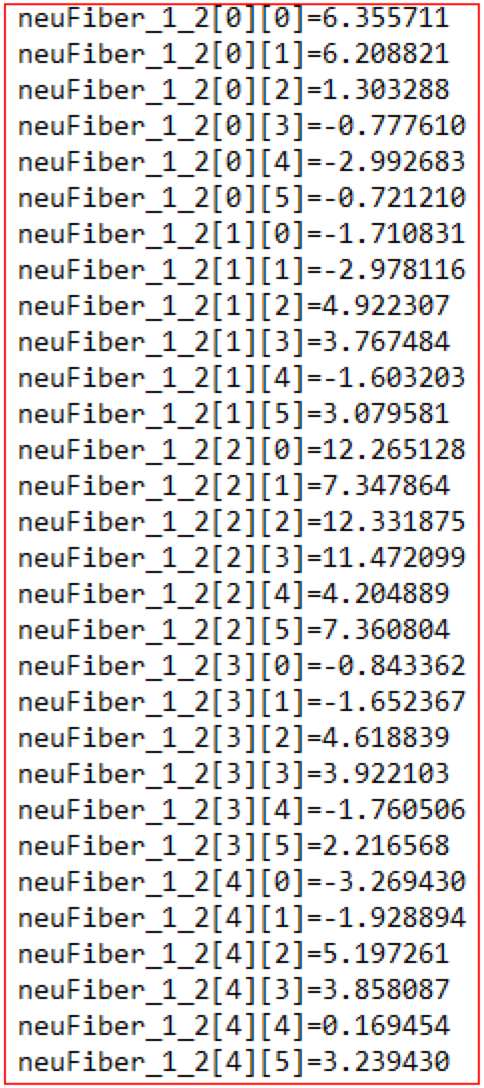


```
Study_Data[0].subjInput[0]=0.400000
Study_Data[0].subjInput[1]=0.400000
Study_Data[0].subjInput[2]=0.200000
Study_Data[0].subjInput[3]=0.200000
Study_Data[0].subjInput[4]=0.100000
Study_Data[0].subjInput[5]=0.100000
Study_Data[0].subjTeachOutput[0]=0.200000
Study_Data[0].subjTeachOutput[1]=0.200000
Study_Data[1].subjInput[0]=0.400000
Study_Data[1].subjInput[1]=0.400000
Study_Data[1].subjInput[2]=0.200000
Study_Data[1].subjInput[3]=0.200000
Study_Data[1].subjInput[4]=0.100000
Study_Data[1].subjInput[5]=0.300000
Study_Data[1].subjTeachOutput[0]=0.200000
Study_Data[1].subjTeachOutput[1]=0.200000
Study_Data[2].subjInput[0]=0.400000
Study_Data[2].subjInput[1]=0.400000
Study_Data[2].subjInput[2]=0.200000
Study_Data[2].subjInput[3]=0.200000
Study_Data[2].subjInput[4]=0.100000
Study_Data[2].subjInput[5]=0.500000
Study_Data[2].subjTeachOutput[0]=0.200000
Study_Data[2].subjTeachOutput[1]=0.200000
```

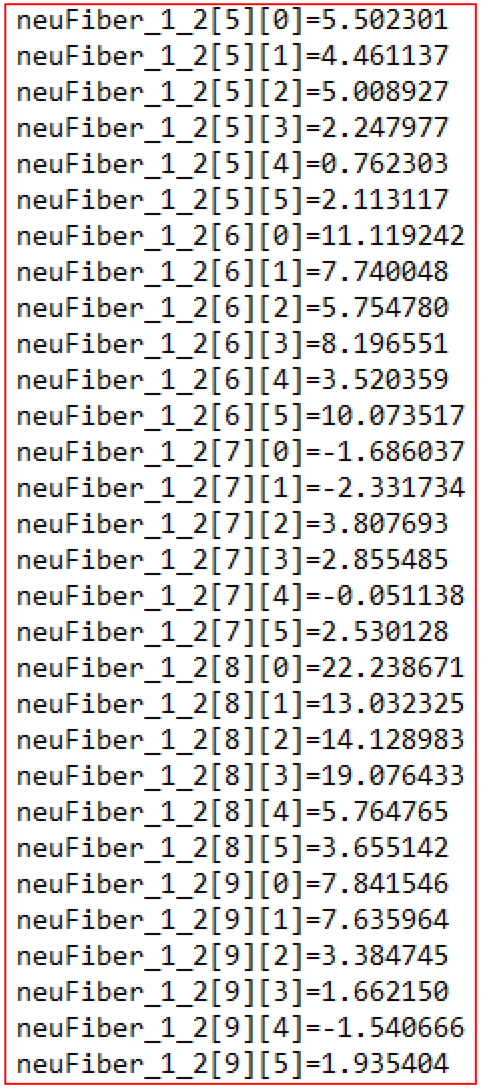


```
totalErr=0.010158
targetErr=0.010000

totalErr=0.010128
targetErr=0.010000

totalErr=0.010098
targetErr=0.010000

totalErr=0.010067
targetErr=0.010000

totalErr=0.010037
targetErr=0.010000

totalErr=0.010006
targetErr=0.010000

totalErr=0.009976
targetErr=0.010000


****************

The program have studied for [2983] times!

****************
```

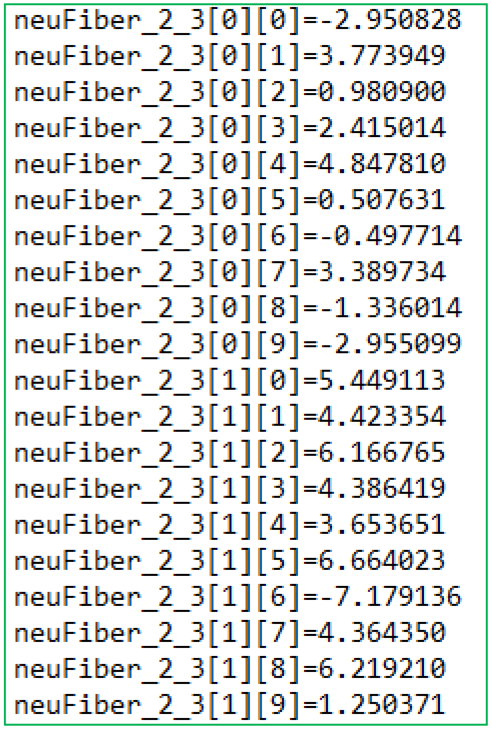
Weights



```
neuFiber_1_2[0][0]=6.355711
neuFiber_1_2[0][1]=6.208821
neuFiber_1_2[0][2]=1.303288
neuFiber_1_2[0][3]=-0.777610
neuFiber_1_2[0][4]=-2.992683
neuFiber_1_2[0][5]=-0.721210
neuFiber_1_2[1][0]=-1.710831
neuFiber_1_2[1][1]=-2.978116
neuFiber_1_2[1][2]=4.922307
neuFiber_1_2[1][3]=3.767484
neuFiber_1_2[1][4]=-1.603203
neuFiber_1_2[1][5]=3.079581
neuFiber_1_2[2][0]=12.265128
neuFiber_1_2[2][1]=7.347864
neuFiber_1_2[2][2]=12.331875
neuFiber_1_2[2][3]=11.472099
neuFiber_1_2[2][4]=4.204889
neuFiber_1_2[2][5]=7.360804
neuFiber_1_2[3][0]=-0.843362
neuFiber_1_2[3][1]=-1.652367
neuFiber_1_2[3][2]=4.618839
neuFiber_1_2[3][3]=3.922103
neuFiber_1_2[3][4]=-1.760506
neuFiber_1_2[3][5]=2.216568
neuFiber_1_2[4][0]=-3.269430
neuFiber_1_2[4][1]=-1.928894
neuFiber_1_2[4][2]=5.197261
neuFiber_1_2[4][3]=3.858087
neuFiber_1_2[4][4]=0.169454
neuFiber_1_2[4][5]=3.239430
```



```
neuFiber_1_2[5][0]=5.502301
neuFiber_1_2[5][1]=4.461137
neuFiber_1_2[5][2]=5.008927
neuFiber_1_2[5][3]=2.247977
neuFiber_1_2[5][4]=0.762303
neuFiber_1_2[5][5]=2.113117
neuFiber_1_2[6][0]=11.119242
neuFiber_1_2[6][1]=7.740048
neuFiber_1_2[6][2]=5.754780
neuFiber_1_2[6][3]=8.196551
neuFiber_1_2[6][4]=3.520359
neuFiber_1_2[6][5]=10.073517
neuFiber_1_2[7][0]=-1.686037
neuFiber_1_2[7][1]=-2.331734
neuFiber_1_2[7][2]=3.807693
neuFiber_1_2[7][3]=2.855485
neuFiber_1_2[7][4]=-0.051138
neuFiber_1_2[7][5]=2.530128
neuFiber_1_2[8][0]=22.238671
neuFiber_1_2[8][1]=13.032325
neuFiber_1_2[8][2]=14.128983
neuFiber_1_2[8][3]=19.076433
neuFiber_1_2[8][4]=5.764765
neuFiber_1_2[8][5]=3.655142
neuFiber_1_2[9][0]=7.841546
neuFiber_1_2[9][1]=7.635964
neuFiber_1_2[9][2]=3.384745
neuFiber_1_2[9][3]=1.662150
neuFiber_1_2[9][4]=-1.540666
neuFiber_1_2[9][5]=1.935404
```



```
neuFiber_2_3[0][0]=-2.950828
neuFiber_2_3[0][1]=3.773949
neuFiber_2_3[0][2]=0.980900
neuFiber_2_3[0][3]=2.415014
neuFiber_2_3[0][4]=4.847810
neuFiber_2_3[0][5]=0.507631
neuFiber_2_3[0][6]=-0.497714
neuFiber_2_3[0][7]=3.389734
neuFiber_2_3[0][8]=-1.336014
neuFiber_2_3[0][9]=-2.955099
neuFiber_2_3[1][0]=5.449113
neuFiber_2_3[1][1]=4.423354
neuFiber_2_3[1][2]=6.166765
neuFiber_2_3[1][3]=4.386419
neuFiber_2_3[1][4]=3.653651
neuFiber_2_3[1][5]=6.664023
neuFiber_2_3[1][6]=-7.179136
neuFiber_2_3[1][7]=4.364350
neuFiber_2_3[1][8]=6.219210
neuFiber_2_3[1][9]=1.250371
```

Q&A

Interesting Video Sharing

- Google soli (2015):
  https://www.google.com/atap/project-soli/
- University of Minnesota-mind control (2013):
  https://www.youtube.com/watch?v=baEYCberLUA
- MIT-eavesdrop through vibration (2014)：
  https://www.youtube.com/watch?v=FKXOucXB4a8
- Auviz-AuvizDNN (2015)：
  https://www.youtube.com/watch?v=hQVbqmuMZoM
- Facebook-“Aquila” (2015) :
  http://www.stuff.co.nz/technology/social-networking/70710497/facebooks-airforce-giant-solarpowered-aquila-drone-to-bring-internet-to-rural-areas
- Light field camera:
  https://illum.lytro.com/illum

Thanks！

TCL Corporate Research Co. Ltd (Hong Kong)
www.tcl.com
@The Creative Life