DSP - Digital Signal Processing
學號：R07942089
系級：電信一
姓名：劉廷緯

DSP
Assignment 2-1 HMM Training and Testing

Part 1 - Run Baseline

Following the steps in homework description, set path to:

```
set_htk_path.sh
1  PATH=$PATH:"../htk341_debian_x86_64"
```

And run the given default scripts, an accuracy of **74.34%** is obtained:

```
====================== HTK Results Analysis =======================
  Date: Tue Nov 13 11:05:25 2018
  Ref : labels/answer.mlf
  Rec : result/result.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=38.54 [H=185, S=295, N=480]
WORD: %Corr=96.61, Acc=74.34 [H=1679, D=13, S=46, I=387, N=1738]
===================================================================
```

Part 2 - Improve Accuracy

By adjusting the parameters in “lib/proto”, “lib/mix2_10.hed”, and “03_training.sh”, accuracy was improved significantly. An optimal accuracy of **97.76%** is obtained, the implementation details that achieves this performance is described in the next section.

```
====================== HTK Results Analysis =======================
  Date: Tue Nov 13 14:50:27 2018
  Ref : labels/answer.mlf
  Rec : result/result.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=92.71 [H=445, S=35, N=480]
WORD: %Corr=97.93, Acc=97.76 [H=1702, D=27, S=9, I=3, N=1738]
===================================================================
```

Part 3 - Describing Training Process and Accuracy

Parameters that achieves the highest accuracy:

- Number of states: **15**
- Gaussian mixtures: **8**
- Iterations: **16 and 32**

Implementation Details:

- lib/proto

1) Number of states is modified to:

```
<NumStates> 15
```

2) Add <state, mean, variance> until 14:

```
<State> 14
<Mean> 39
0.0 0.0 0.0 0.0 0.
0.0 0.0 0.0
<Variance> 39
1.0 1.0 1.0 1.0 1.
1.0 1.0 1.0
```

3) TransP is modified to:

```
<TransP> 15
```

4) Transition matrix is modified so that the probability of jumping to the next state and the original state are both 0.5:

```
0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

- lib/mix2_10.hed

1) Every *.state[2-9].mix is modified to: `.state[2-14].mix`

2) Number of Gaussian is modified to: `MU 8`

- 03_training.sh

1) The first two for loops are modified to: `for i in $(seq 0 15) ;`

2) The last for loop is modified to: `for i in $(seq 0 31) ;`

Training Process:

I wrote an additional shell script to help me run the whole training and testing pipeline, after the above modification, run these commands and the result shown in Part 2 can be reproduced:

run.sh

```
#!/bin/bash
bash 00_clean_all.sh
bash 01_run_HCopy.sh
bash 02_run_HCompV.sh
bash 03_training.sh
bash 04_testing.sh
echo "Complete."
```

Experiments:

Below I will demonstrate some interesting discoveries I found during my attempts to achieve a higher accuracy. We can briefly analyze how the above change of parameters affects the performance by changing the parameters one at a time, and compare its performance with the baseline model (The one in Part 1):

- **The effect of number of states:** set the values of Gaussian mixtures and iterations to the default baseline values, and increase the number of states. Number of states is tripled in this experiment (5 → 15), accuracy is increased from **74.34% to 95.91%**.

```
====================== HTK Results Analysis =======================
  Date: Tue Nov 13 14:24:44 2018
  Ref : labels/answer.mlf
  Rec : result/result.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=87.50 [H=420, S=60, N=480]
WORD: %Corr=96.26, Acc=95.91 [H=1673, D=41, S=24, I=6, N=1738]
===================================================================
```

• **The effect of Gaussian mixtures:** Set the values of number of states and iterations to the default baseline values, and increase the number of Gaussian mixtures. Number of Gaussian mixtures is doubled in this experiment (2 → 4), accuracy is increased from **74.34% to 76.47%**.

```
====================== HTK Results Analysis =======================
  Date: Tue Nov 13 14:05:24 2018
  Ref : labels/answer.mlf
  Rec : result/result.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=43.75 [H=210, S=270, N=480]
WORD: %Corr=96.72, Acc=76.47 [H=1681, D=12, S=45, I=352, N=1738]
===================================================================
```

• **The effect of Iteration:** Set the values of number of states and Gaussian mixtures to the default baseline values, and increase the number of iterations. Number of iteration is doubled in this experiment (3 → 6 and 6 → 12), accuracy is increased from **74.34% to 76.29%**.

```
====================== HTK Results Analysis =======================
  Date: Tue Nov 13 13:54:46 2018
  Ref : labels/answer.mlf
  Rec : result/result.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=42.92 [H=206, S=274, N=480]
WORD: %Corr=96.84, Acc=76.29 [H=1683, D=11, S=44, I=357, N=1738]
===================================================================
```

We can easily observe that the **increment in the number of states, contributes the most to the boost in performance.** Increasing the number of Gaussian mixtures and the number of iteration also increase the performance by a small margin. Hence I use a mixture of the above adjustments, and obtain the best result shown in Part 2.