Procedures

The basic training process (i.e. without reinforcing phase) mainly consists 3 phases, which are “Initialization phase”, “Calibrating phase”, and “Training (growing) phase”. After training, you can also apply the checking process, which inputs the training data again and test the nodes representation in each trained audio map. Below, I will explain in detail how to run the whole process.

TRAINING

1. Preparation work

   You should make some folders (see Fig. 1).

   - codes: stores code files.
   - gmaps: stores trained gmap files.
   - logs: stores log files.
   - matlab: stores files to be imported into matlab for visualization.
   - training_data: stores training data file (e.g. audio_coding_1.txt)

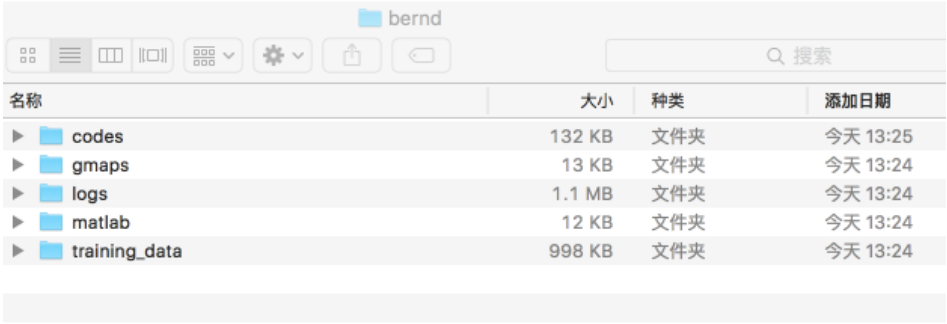

Fig 1. Folders

2. Codes

   You will see 5 python code files, and I will explain them one by one.

   - functions.py

     This is a global function file, which contains many function declarations. Other files will import this file and call functions from this file when needed.

   - gsom_initial.py

     This file declares a function *initial (vector_dimension)*, which will do the initialization work based on the dimension of training vectors. It also defines node structures, which will be used during the training process.

This file will be imported and called by “gsom_train.py”

- gsom_grow.py

  This file is also a function declaration file, and its is more focused on functions to be used during training phases.

- gsom_train.py

  This file is the main file you need to run during the training process. It will perform the “Initialization phase”, “Calibrating phase”, and “Training (growing) phase” step-by-step, and output trained gmaps.

- check.py

  This file is the main file you need to run during the checking process. It checks the performance of each trained gmap, and output files for afterwards analysis.

3. About each phase

- Initialization phase

  Initialize the network with 4 nodes. The dimension of each node is set according to vector dimension of training data. The initial weights of each node are generated randomly around 0.5. You can have a look at the *initial_weight_vector(n)* function in “functions.py”.

- Calibrating phase

  It is a phase newly added into the procedure (see the InterSpeech paper for details). The main functions of this phase are (1) to make the 4 initial nodes properly distributed, and (2) find suitable growth threshold (*GT*) for the growing phase. During this phase, neighborhood size (*NS*) is set to 1, and no growing is allowed. You can modify the variable “EPOCH_caliberating” in the “gsom_train.py” to set the rounds of calibrating phase you need.

- Training (growing) phase

  The growing phase is very similar to what we did before. There are mainly two differences: (1) GT is inherited from the calibrating phase, and (2) a dynamic neighborhood size (*NS*) based on the network size (*n*) is used, that:

$$NS = \begin{cases} 1, & n < 10 \\ \log_{10}(n), & otherwise \end{cases}$$

You can modify variable "EPOCH_growing" in the "gsom_train.py" to set the rounds of growing phase you need.

4. Notes
   - You need to manually set the "VECTOR_DIMENSION" constant in the "gsom_train.py".
   - You need to modify all print functions and make other necessary changes if you want to run the code in Python 3.
   - The "gsom_train.py" will generate a large number of plain text prints, which are used for log recording. You should run the code with command like "Python gsom_train.py > ../logs/training1.txt". The symbol ">" is the "re-directional mark", which will redirect the print output to the file you assigned (here is "../logs/training1.txt").

CHECKING

1. Preparation work

   You should make two more folders: you should make a folder called "check" within the folders "gmaps" and "matlab" respectively.
2. Run the code of "check.py". The code will check the performance of each gmap and output results.
3. The code will calculate the mean squared error (MSE) of each gmap and output the result into the file MSE_A in the "check" folder of "gmaps" folder. Open the MSE_A file, each line represents the MSE of a gmap ordering by the training steps. You should see a generally declining trend (see Fig. 2).

Fig 2. MSE results

4. The code will also generate lots of files in the “check” folder of “matlab” folder. I will further explain how to use matlab to visualize checked results in the next step.