1. Introduction

This is the user's guide of the code developed for [1]. The idea of this work is fine-tuning and personalizing compression ratios of hearing aid compression.

For this reason, in order to decrease the number of user feedbacks and thus enable a practical implementation of the personalized compression, first a reward function is considered to model hearing preferences of a user in an asynchronous manner. This is achieved by carrying out comparison between instances of two different compressed audios. Then, an agent is trained to maximize reward. Following figure shows a block diagram of this approach:

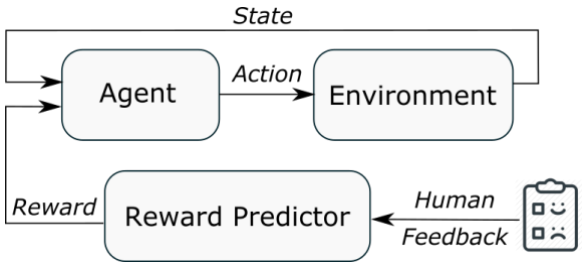
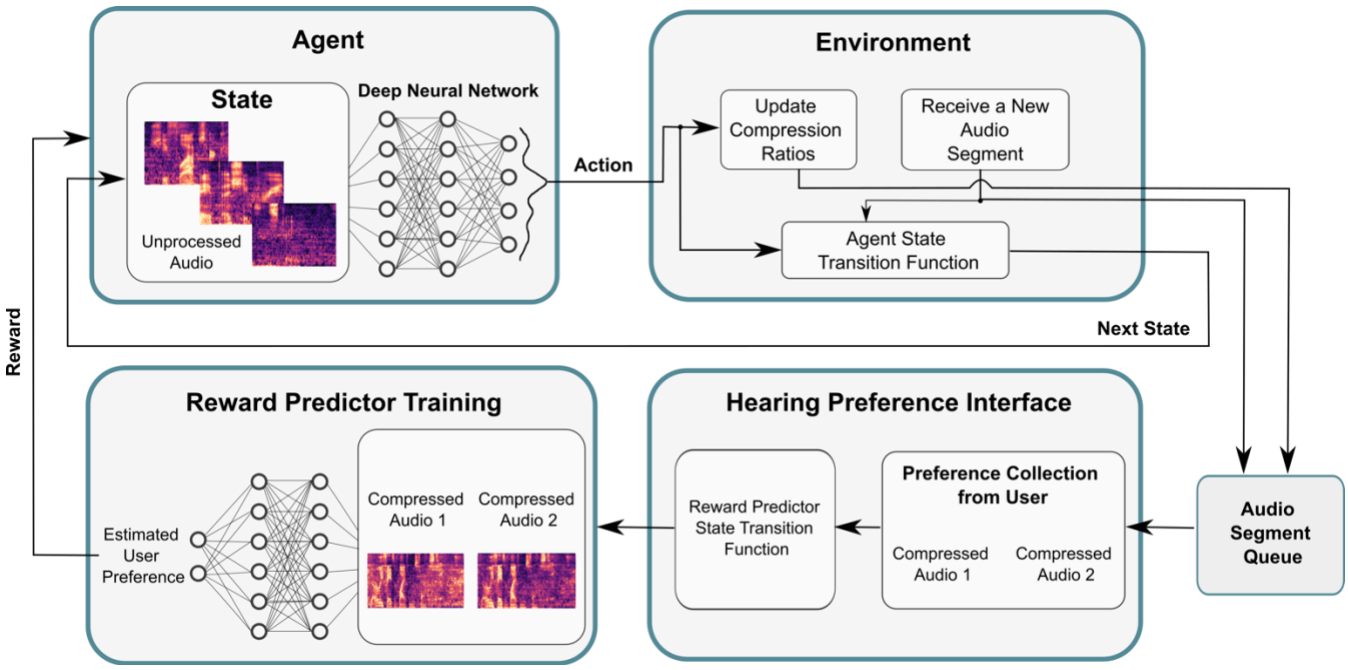
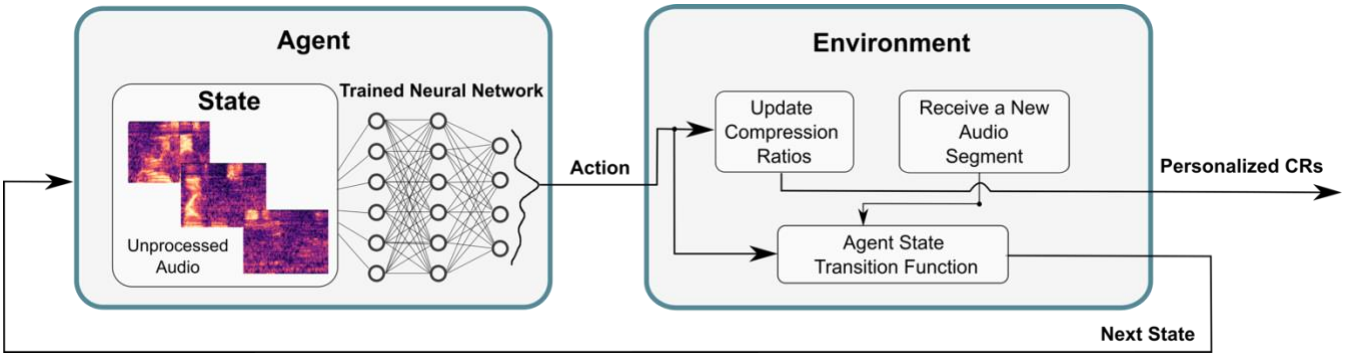


Figure 1: Block diagram of proposed approach.

Below shows how data is passes through different blocks of the personalization framework both in training and testing modes. Please refer to [1] for more details.



(a) Training mode



(b) Operation mode

Figure 2 Developed personalized compression DRL framework for (a) training mode and (b) operation mode.

2. Requirements:

Majority of codes implemented in python in the format of xxx.py. There are few python codes that are written for jupyter lab/notebook in the format of xxx.ipynb.
Requirements: python 3, Jupyter lab/notebook (it can be installed using anaconda) Anaconda can be installed in [2].

It is recommended to create an environment for the project and install all python libraries over there. Instruction can be found here:
https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

Function written for dynamic range compression is written in MATLAB. That is why in addition to python libraries, MATLAB engine needs to be installed though conda. Example of the instruction can be found in [3]:

Open Anaconda Prompt (make sure you have the correct python version):

```
cd matlabroot\extern\engines\python
python setup.py install
```

'matlabroot' is your MATLAB root directory. In my case it is 'C:\Program Files\MATLAB\R2018a'

3. Folder organization

The code has three main tasks, and each subfolder is responsible to doing one task:

Phase1: Collecting user's hearing preferences
Phase2: Data Augmentation, Training reward predictor, and then training RL agent
Phase3: Subject testing personalized compression settings versus baseline settings (assessment of effectiveness of personalization)

2.1 Phase 1 details

Main codes in subfolder Phase1 are as follows:

- gain2CR.ipynb -> to compute "Initial_CRs", and "Initialize_softG"
- run_HumanPrefCollect_real-time.ipynb

Adjustment for codes in Phase1:

- Define path of noisy speech wave files (in line 223 of *fittingEnv.py* and line 91 of *ask_human_realtime.py*)
- Define name of .txt file (in lines 129 and 224 of *askHuman_realtime.py*)
- Define "Initial_CRs", and "Initialize_softG" (in lines 24 and 25 of fittingEnv.py)

Running order: "gain2CR.ipynb", then "run_HumanPrefCollect_real-time.ipynb"

2.2 Phase 2 details

Main codes in Phase2 subfolder are as follows:

- rp_augment_main.py -> to perform augmentation
- rp_main.py -> for reward predictor training)
- rl_main.py ->for RL Agent training)

Running order*:* 1) "rp_augment_main.py" to receive augmented data. 2) "rp_main.py" to train the reward predictor model. 3) "rl_main.py" to train RL agent and generate "rl_model.h5".

2.3 Phase 3 details

Main code in subfolder Phase3 are as follows:

- *test_rlAgent.ipynb -> to generate "test_results.txt" and "RL_env.txt"*

Note: please make sure "rl_model.h5" from outcome of phase 2 is placed in phase 3 folder.

Adjustment in "fittingEnv.py":

- Define path of noisy speech wave files in line 223
- Uncomment lines [198-201]
- Define "Initial_CRs", and "Initialize_softG"

Adjustment in "test_rlAgent.ipynb":

- Define "Initial_CRs", and "Initialize_softG"
- Define path of noisy speech wave files

4. Running the code

Steps indicated in above sections to run the whole personalization framework for both training and testing can be summarized as follows:

a. *"gain2CR.ipynb "*: to convert gains in 9 frequency bands to 5 frequency bands, then computing initial compression ratios and soft gains in five frequency bands
b. *"run_HumanPrefCollect_real-time.ipynb"* : to play collection of pair compressed audios to the user and collect its hearing preferences.
c. Placing all the collected preferences *in "Human_Feedback_Data"* located in the subfolder phase 2.
d. *rp_augment_main.py* : to augment the data collected from user.
e. *rp_main.py* : to train the reward predictor model.
f. *rl_main.py* : to train RL agent and generate "rl_model.h5".
g. Placing "*rl_model.h5*" inside subfolder phase 3
h. "test_rlAgent.ipynb" -> to assess performance of personalization by asking user to compare audio compressed with personalized setting versus baseline DSL-v5 settings.
i. The results will be saved in "test_results.txt".

References

[1] N.Alamdari, E.Lobarinas, N.Kehtarnavaz, "*Personalization of Hearing Aid Compression by Human-in-the-Loop Deep Reinforcement Learning*", IEEE Access, vol. 8, pp. 203503-203515, 2020.

[2] Anaconda, https://www.anaconda.com/products/individual

[3] MATLAB Central, https://www.mathworks.com/matlabcentral/answers/346068-how-do-i-properly-install-matlab-engine-using-the-anaconda-package-manager-for-python