

SpeechBrain *Tutorial*



Table of Contents

1. What is SpeechBrain ?
 - a. [Introduction](#)
 - b. [Tasks Available](#)
2. Installation
 - a. [Installation](#)
3. Using Speechbrain
 - a. [Data Loader](#)
 - b. [HyperYaml](#)
 - c. [Brain Class](#)
 - d. [Using Pre-trained Models](#)
 - e. [Fine-Tuning a Model](#)
4. Conclusion
 - a. [Other Useful Links](#)
 - b. [Running on NSCC](#)
 - c. [Some NSCC Tips](#)
 - d. [Summary](#)
 - e. [References](#)
 - f. [Contribution](#)

1. What is SpeechBrain

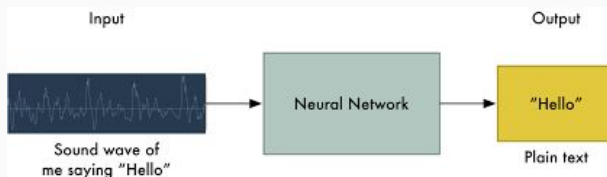
a. Introduction

- Easy to use all-in-one speech processing toolkit
- Based on **Pytorch**
- Supports multiple speech-processing tasks
- Contains pre-trained models, datasets
- Easy to customise
- Well-documented, supports multiple functionalities

b. Tasks Available

- It can be used for a wide-variety of speech tasks such as :

- Automatic Speech Recognition(ASR) :

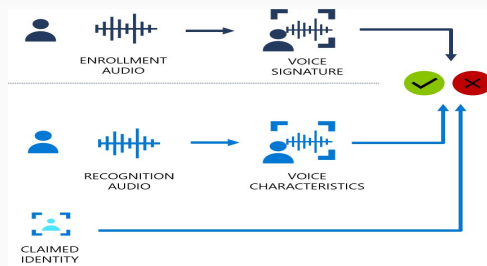


- Speech Enhancement :

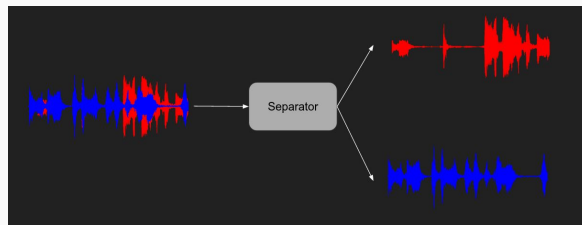


b. Tasks Available

- Speaker Verification :



- Speech Separation :



- Speaker Classification

b. Tasks Available

- Find more available tasks [here](#)
- Load available [pre-trained models](#) and perform inference for above tasks
- [Fine-Tune models](#) available on hugging face/speechbrain on custom dataset.

2. Installation

a. Installation

It can be done by :

- a) PyPI : Use when you want to directly use any functionality of speechbrain. Simply install and import using

```
pip install speechbrain  
import speechbrain as sb
```

- b) Local : To train your own speech processing system from scratch/make modifications to Speechbrain, install it locally by using

```
git clone https://github.com/speechbrain/speechbrain.git  
cd speechbrain  
pip install -r requirements.txt  
pip install --editable .
```

a. Installation

Note :

- Suggested to use different environment for different toolkits
- Better to use anaconda - create a new env *speechbrain* and install locally
- It can be done using :

```
conda create --name speechbrain python=3.9  
conda activate speechbrain
```


3. Using SpeechBrain

a. Data Loader

- Follows Pytorch Data Loading Pipeline.
- Consists of the following argument :
 - Dataset
 - Collation Function
 - Sampler
 - Data Loader
- You can also directly load the data to the `brain.fit ()` function
- The [link](#) contains detailed description

b. HyperPyYaml

- Extension of Yaml file
- Used to specify all hyperparameters (like batch_size,optimizer,learning rate etc)
- Creates readable format - separating hyperparameters from model architecture
- Saved as *file_name.yaml* file.
- [Example](#) and [demonstration](#) of HyperPyYaml

c. Brain Class

- Most important part of SpeechBrain.
- Used to perform the training loop using the **fit()** method.
- To use **fit()** method, the following two methods *need* to be defined :
 - *def compute_forward (self, batch, stage)*
 - *def compute_objectives (self, predictions, batch, stage)*

Note: stage is used to track the stages of the experiment. Defined by

TRAIN= 1 , VALID= 2, TEST= 3

Brain Class

- To define the Brain Class, we require five arguments :
 - modules
 - opt_class
 - hparams
 - run_opts
 - checkpointer

Example of brain class : `brain = SimpleBrain ({"model": model}, hparams[opt_class], hparams, run_opts={'device':device},)`

Brain Class

- The `fit()` method performs the training loop
- [Link](#) contains
 - detailed description of brian class
 - Defining basic nn (single layer)
- `evaluate()` method used to iterate over testing dataset.
- To **run an experiment** run this

```
python train.py hyp.yaml
```

d. Using pre-trained models

- SpeechBrain contains pre-trained models - imported from *speechbrain.pretrained*
- Can be used directly for inference.
- You can find the list of pre-trained models [here](#).
- The [link](#) contains code to use pre-trained models for ASR task.

e. Fine-Tuning a model

This Link contains the code for speechbrain diarization

4. Conclusions

a. Other Useful Links

- 1) [HyperPyYaml Tutorial](#)
- 2) [Checkpointing in SpeechBrain](#)
- 3) [Multi-GPUs](#)
- 4) [Pre-trained Models and Fine-Tuning](#)
- 5) [HyperParameter Optimization](#)

b. Running on NSCC Tips

- Once you get nscc login id and password. Login using the ssh command
- NSCC has an interactive command, so run the following command
 - ***qsub -l -q dgx-dev -l walltime=12:00:00 -P \$PROJECT_ID***
 - ***Replace \$PROJECT_ID with 12001458 (our project ID)***
- Since we don't have root privileges, we can use the following docker image
 - ***nscc-docker run -t nvcr.io/nvidia/pytorch:20.12-py3***
- You can then install SpeechBrain by creating a new conda env
- Will now be able to use GPU for your experiments

c. Some NSCC Tips

- While using interactive environment on NSCC, sometimes you may get “CUDA: Out of Memory error”, and following shows up if you type *nvidia-smi* command

```
(pyannote) n2202857@b4479d9b6593:~/scratch/projects/baseline$ nvidia-smi
Wed Oct 12 14:39:31 2022
```

NVIDIA-SMI 418.67		Driver Version: 418.67		CUDA Version: 11.1	
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util Compute M.
0	Tesla V100-SXM2...	On	00000000:86:00:0	Off	0
N/A	39C	P0	57W / 300W	14386MiB / 16130MiB	0%
					Default

Processes:				GPU Memory
GPU	PID	Type	Process name	Usage

- In such a case, just open another terminal and log in to nscc again.
- Then request for another job on the interactive environment while the previous one is still running.
- You can run your code here after the new job is scheduled without running into the Out of Memory Error

c. Summary

- Easy to use, flexible and well-documented toolkit
- Used for wide-variety of speech tasks
- This tutorial consists most important steps to infer from/train models for speech tasks
- Still in beta version (in progress) project with many upcoming features

d. References

- 1) <https://speechbrain.github.io/>
- 2) **Important Tutorials** - https://speechbrain.github.io/tutorial_basics.html
- 3) <https://arxiv.org/pdf/2106.04624.pdf>
- 4) <https://github.com/speechbrain/speechbrain>
- 5) <https://speechbrain.readthedocs.io/en/latest/index.html>

e. Contribution

- [Arabelly Abhinav](#)
- Thanks to Prof Chng Eng Siong, Lim Zhi Hao and Liu Chenyu for useful suggestions

THANK YOU

Finish

Demonstration of Hyperyaml file

Fine tune a Model slide