ProsoDeep Documentation

Release 1.0

gerazov

May 12, 2019

Contents:

This is the documentation for the code implementation for all of the models created within the ProsoDeep framework. The code is written in Python and is available as free software under a GNU General Public License v3 on GitHub https://github.com/gerazov/prosodeep

The code includes a code base package named `prosodeep` and a main execution script `prosodeep.py`. The script controls the execution flow and carries out data loading, model initialisation, training and evaluation.

CHAPTER 1

Python ecosystem

Python was chosen as an implementation language because of the powerful scientific computing environment that is completely based on free software. The code is built upon NumPy within the SciPy ecosystem. The neural network models and their training were implemented in PyTorch, which is a powerful deep learning platform centered on Python that allows for rapid model prototyping and easy debugging. Great attention was put on code readability, which is also one of the features of good Python, augmented with detailed functions docstrings, and comments. The code is segmented in Spyder cells for rapid prototyping. Other packages used in the code include:

- scikit-learn – for machine learning utility functions,
- pandas – for data structuring and manipulation,
- matplotlib – for result plotting, and
- seaborn – for plotting histograms.

Finally, the whole code has been licensed as free software with a GNU General Public License v3. The code can be found on GitHub: https://github.com/gerazov/prosodeep

CHAPTER 2

Modules

The prosodeep package comprises the following modules:

- prosodeep_params.py – parameter setting module that includes (this list is not exhaustive):
  - model selection – which type of contour generators to use, and what training approach to use, i.e. joint backpropagation training or analysis-by-synthesis,
  - model hyperparameters – number of points to be sampled from the pitch, number of training iterations, number of hidden units, layers, type of activation, learning rate, regularisation parameters, type of optimizer, early stopping, validation and test set ratios,
  - data related parameters – type of input data, phrase level functional contours, local functional contours,
  - general control parameters – use of preprocessed data, use of trained models, plotting, saving.
- prosodeep_corpus.py – holds all the functions that are used to consolidate and work with the corpus of data that is directly fed and output from the SFC model. The corpus is a Pandas data frame object, which allows easy data access and analysis.
- prosodeep_data.py – comprises functions that read the input data files and calculate the $f_0$ and duration coefficients,
- prosodeep_learn.py – holds the training functions for backpropagation and analysis_by_synthesis,
- prosodeep_model.py – holds all of the neural network models used by the various prosody models,
- prosodeep_dsp.py – holds DSP functions for smoothing the pitch contour based on SciPy,
- prosodeep_plot.py – holds the plotting functions based on matplotlib and seaborn.

Currently, prosodeep supports the standard Praat TextGrid annotations, and calculates pitch based on Praat PointProcess pitch mark files. We plan to integrate state-of-the-art pitch extractors in the near future, e.g. the Kaldi pitch extractor.