Feature Extraction with Convolutional Autoencoders

Vaggelis Lamprou

Presentation

Contents

1. Introduction

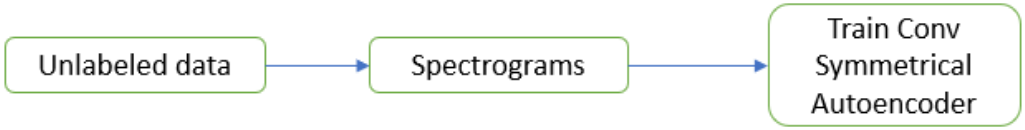
- Goal: Study the use of Convolutional Autoencoders as feature extraction tools when compared to hand-crafted features obtained by the PyAudioAnalysis Python library.
- Consider a scenario where the available training dataset consists of "a few" labelled points and "many" unlabelled points (of the same data source).
- Dataset: Valence class samples of "MSP Podcast"
  - classes: negative, neutral, positive
  - Split into speaker independent sets:

Training labelled data: 350 + 362 + 319 (1031 in total)
Training unlabelled data: 2781 + 2674 + 2622 (8077 in total)
Test data: 432 + 421 + 383 (1236 in total)

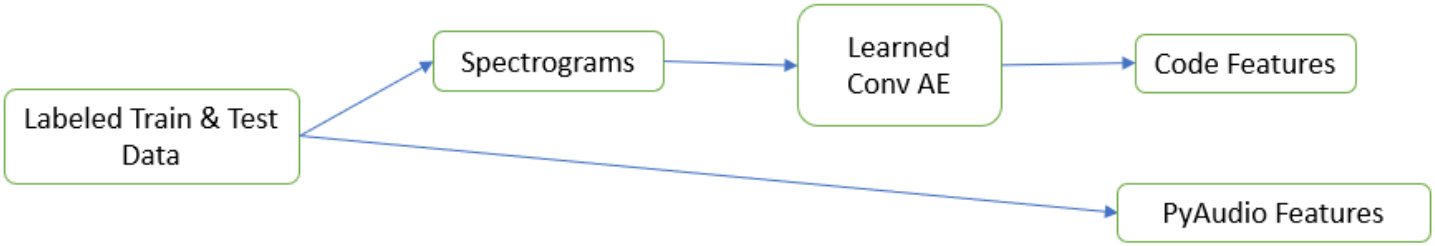
  - All data have signal size at least 30K samples

2. Graphs

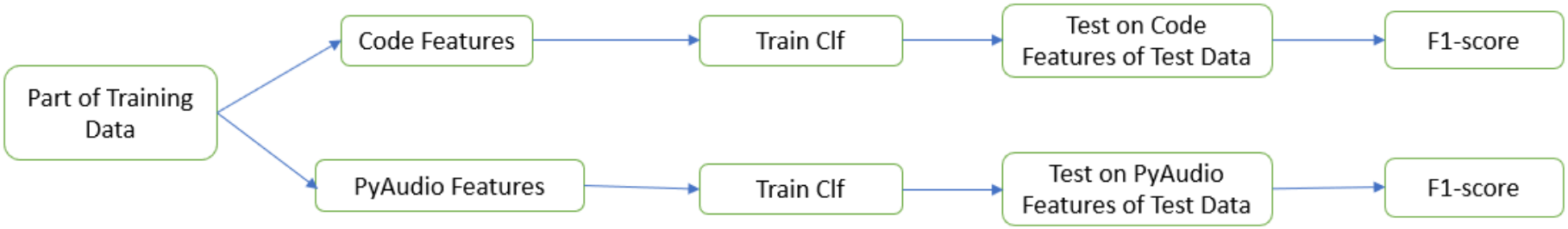
- Part I - Conv AE



- Part II - Features

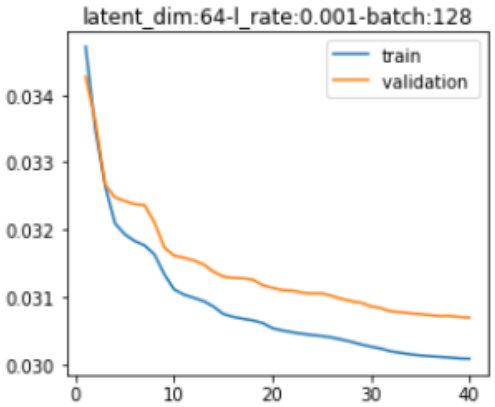


- Part III - The experiment
  Per iteration (increase size by 2%):



3. Spectograms & Conv AE

- Spectograms
  - Must be of fixed size to fit into the Conv AE.
  - Take 30K length signal sizes. Ignore tails.
- Conv AE
  - Tuned Symmetric Architectures wrt number of convolutional layers, kernels, strides, bottleneck dimension, learning rate and batch size
  - Loss - history:

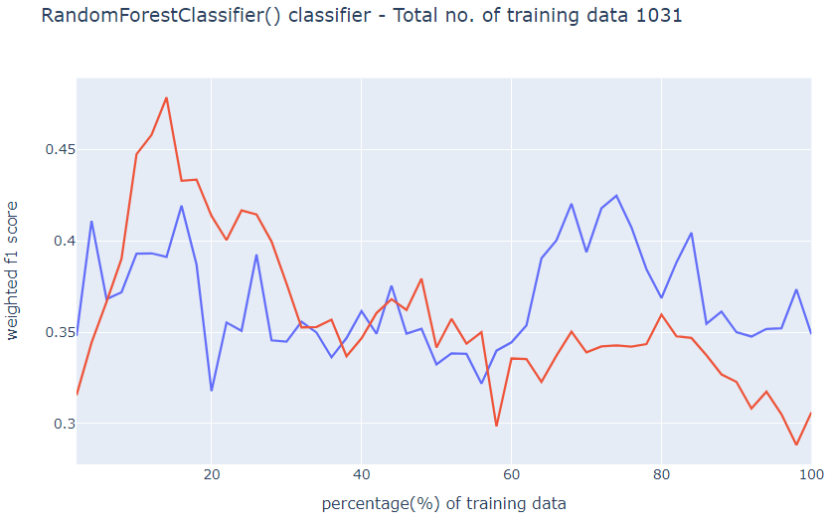


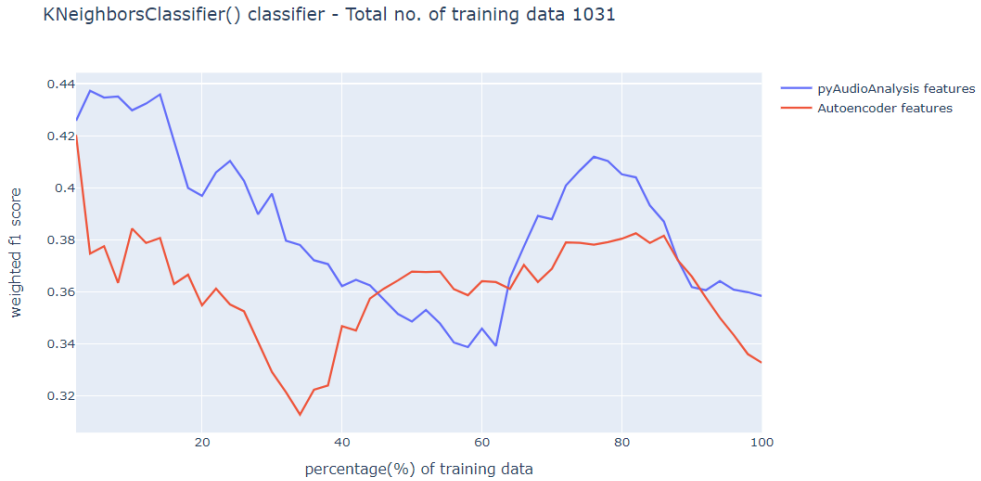
- Encoder Summary:

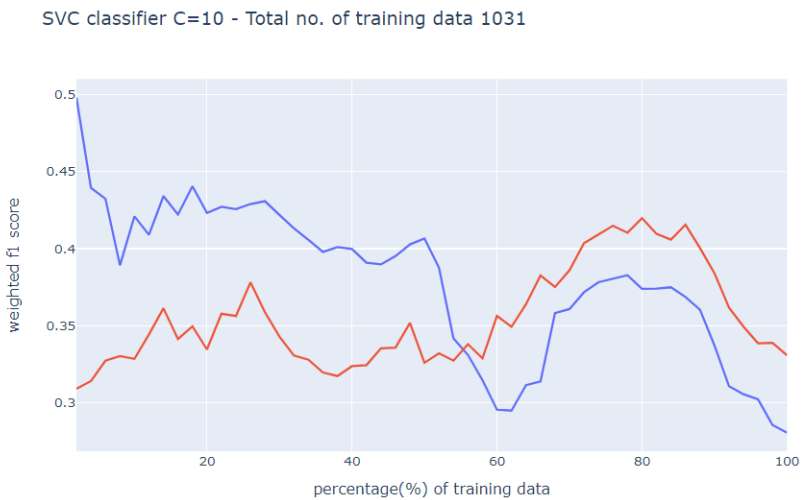
Layer (type)	Output Shape	Param #
input_5 (InputLayer)	[(None, 74, 200, 1)]	0
conv2d_16 (Conv2D)	(None, 37, 100, 16)	80
dropout_24 (Dropout)	(None, 37, 100, 16)	0
conv2d_17 (Conv2D)	(None, 18, 50, 32)	2080
dropout_25 (Dropout)	(None, 18, 50, 32)	0
conv2d_18 (Conv2D)	(None, 6, 16, 64)	18496
dropout_26 (Dropout)	(None, 6, 16, 64)	0
conv2d_19 (Conv2D)	(None, 2, 5, 128)	73856
flatten_4 (Flatten)	(None, 1280)	0
dense_16 (Dense)	(None, 256)	327936
dense_17 (Dense)	(None, 64)	16448

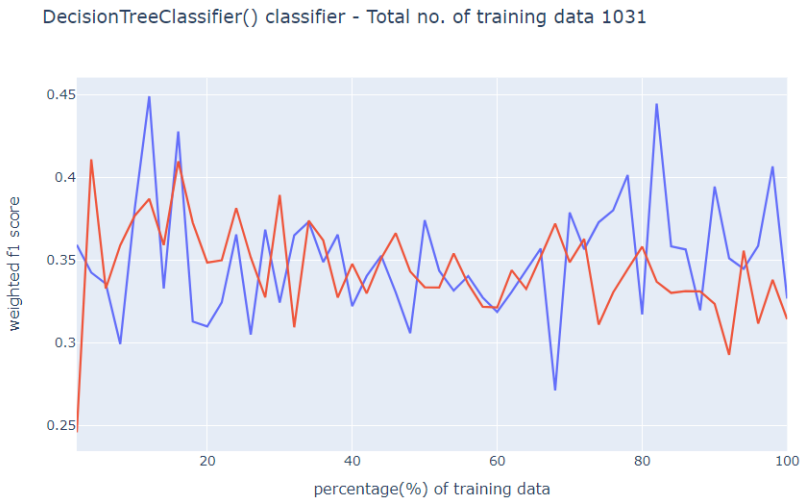
Total params: 438,896
Trainable params: 438,896
Non-trainable params: 0

4. Results


RandomForestClassifier() classifier - Total no. of training data 1031
weighted f1 score
0.45
0.4
0.35
0.3
20
40
60
80
100
percentage(%) of training data


KNeighborsClassifier() classifier - Total no. of training data 1031
weighted f1 score
0.44
0.42
0.4
0.38
0.36
0.34
0.32
20
40
60
80
100
percentage(%) of training data
pyAudioAnalysis features
Autoencoder features


SVC classifier C=10 - Total no. of training data 1031
weighted f1 score
0.5
0.45
0.4
0.35
0.3
20
40
60
80
100
percentage(%) of training data


DecisionTreeClassifier() classifier - Total no. of training data 1031
weighted f1 score
0.45
0.4
0.35
0.3
0.25
20
40
60
80
100
percentage(%) of training data

Thank You !