MULTI-TASK SELF-SUPERVISED LEARNING FOR ROBUST SPEECH RECOGNITION

Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski,
Joao Monteiro, Jan Trmal, Yoshua Bengio

Keypoints/ takeaways

Discover robust representations from the raw speech waveform

Outperformed more traditional hand-crafted features in different speech classification tasks such as speaker identification, emotion classification, and automatic speech recognition

Work wells also when speech that is corrupted by a considerable amount of noise and reverberation

SincNet architecture

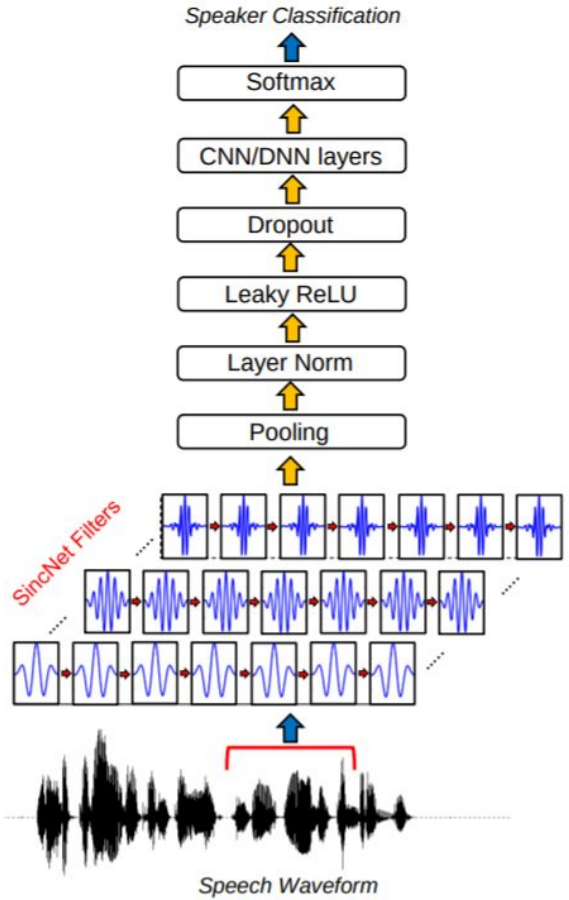

Fig. 1: Architecture of SincNet.

Ref: SincNet

SincNet Architecture

First layer of a standard CNN performs a set of **time domain convolutions** between the input waveform and some Finite Impulse Response (FIR) filters(h)

$$y[n] = x[n] * h[n] = \sum_{l=0}^{L-1} x[l] \cdot h[n-l] \qquad (1)$$

$$y[n] = x[n] * g[n, \theta]$$

x[n] is the input waveform and h[n] is the filter of length L.

Y[n] is the filtered output.

Instead of learning L elements, we use function g and we try to learn Θ

SincNet Architecture

Filter-bank composed of rectangular bandpass filters is employed.

$$g[n, f_1, f_2] = 2f_2 sinc(2\pi f_2 n) - 2f_1 sinc(2\pi f_1 n), \quad (4)$$

where the sinc function is defined as $sinc(x) = sin(x)/x$.

f1, f2 are the cutoff frequencies of rectangular bandpass.

These are the only parameters to be learnt.

They can be sampled initially from [0,fs/2] where fs is sampling frequency.

SincNet Architecture

They have used a windowing technique to perform convolution.

Using a rectangular window creates discontinuity at the end of window. So hamming window is used.

$$g_w[n, f_1, f_2] = g[n, f_1, f_2] \cdot w[n]. \tag{7}$$

This paper uses the popular Hamming window [36], defined as follows:

$$w[n] = 0.54 - 0.46 \cdot cos\left(\frac{2\pi n}{L}\right). \tag{8}$$

PASE architecture

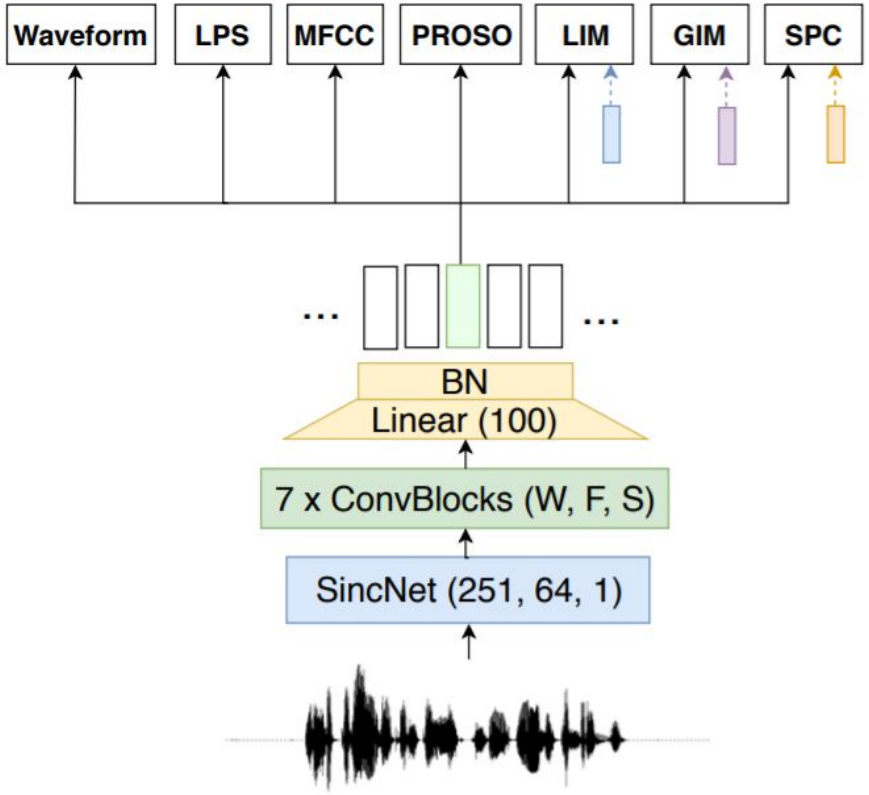

Figure 1: *The PASE architecture, with the considered workers.*

Ref: PASE

Encoder

- Based on the SincNet model.
- The subsequent layers are composed of a stack of 7 convolutional blocks
- Emulate an overlapping sliding window using a set of convolutions.
- The convolution extracts localized patterns at different time shifts.
- Input signal is decimated in time by a factor of 160.

Pre-training

- Workers are fed by the encoded representation
- Solve seven self-supervised tasks, defined as regression or binary discrimination tasks.
- In all cases, workers are based on very small feed-forward networks, composed of a single hidden layer of 256 units with PReLU activation.
- These workers are trained to minimize the mean squared error (MSE) between the target features and the network predictions

We first consider the use of regression workers, which break down the signal components at many levels in an increasing order of abstraction. These workers are trained to minimize the mean squared error (MSE) between the target features and the network predictions (again the waveform worker is an exception, see below). Features are extracted with librosa [25] and pysptk [26] using default parameters, if not stated otherwise.

we predict the input waveform in an auto-encoder fashion. The waveform decoder employs three deconvolutional blocks with strides 4, 4, and 10 that upsample the encoder representation by a factor of 160. After that, an MLP of 256 PReLU units is used with a single output unit per time-step. This worker learns to reconstruct waveforms by means of mean absolute error (L1) minimization. The choice of L1 is driven by robustness, as the speech distribution is very peaky and zero-centered with prominent outliers [27].

Regression pre-training

- **Waveform**
  - This worker learns to reconstruct waveforms by means of mean absolute error (L1) minimization.
  - The choice of L1 is driven by robustness, as the speech distribution is very peaky and zero-centered with prominent outliers.
- Log power spectrum (LPS)(wiki)

Regression pre-training

- Mel-frequency cepstral coefficients (MFCC): Extract 20 coefficients from 40 mel filter banks (FBANKs) (wiki)

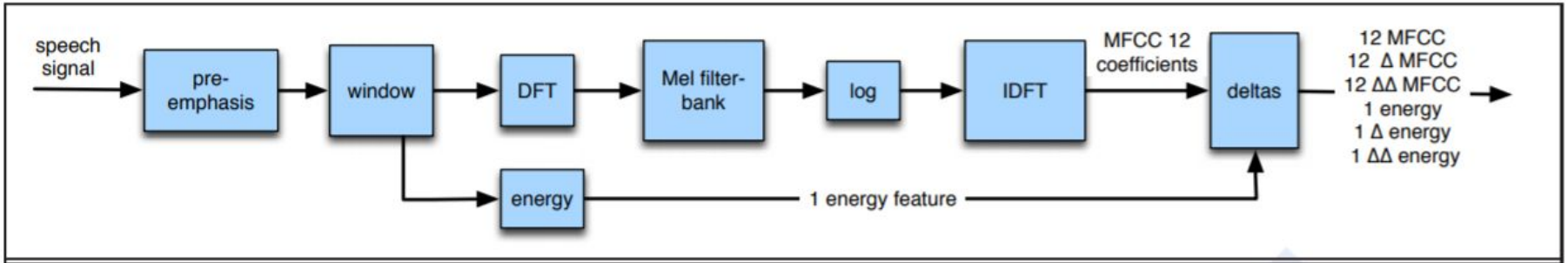

Figure 9.8 Extracting a sequence of 39-dimensional MFCC feature vectors from a quantized digitized waveform

Regression pre-training

- Prosody: Predict four basic features per frame, namely the interpolated logarithm of the fundamental frequency, voiced/unvoiced probability, zero-crossing rate, and energy. (Captures emotion)
    - The fundamental frequency, often referred to simply as the fundamental, is defined as the lowest frequency of a periodic waveform
    - **voiced** and **unvoiced**. Many consonant sounds come in **pairs**. For example, P and B are produced in the same place in the mouth with the tongue in the same position. The only difference is that P is an **unvoiced** sound (no vibration of the vocal cords) while B is a **voiced** sound (vocal cords vibrate).

    - **Zero crossing rate** of any signal frame is the **rate** at which a signal changes its sign during the frame. It denotes the number of times the signal changes value, from positive to negative and vice versa, divided by the total length of the frame.

Binary discrimination tasks

Anchor x_a, a positive x_p, and a negative x_n sample

An MLP then minimizes the binary cross-entropy

$$L = \mathbb{E}_{X_p}[\log(g(x_a, x_p))] + \mathbb{E}_{X_n}[\log(1 - g(x_a, x_n))]$$

How to sample positive and negative samples?

Local info max (LIM): Positive sample from the same sentence of the anchor and a negative sample from another random sentence that likely belongs to a different speaker

Global info max (GIM): The anchor representation is obtained by averaging all the PASE encoded frames of a random utterance. The positive sample is similarly derived from another random chunk within the same sentence, while the negative one is obtained from another sentence.

Sequence predicting coding (SPC): Anchor is a single frame, while positive and negative samples are randomly extracted from its future and past elements. In particular, x_p contains 5 consecutive future frames, while x_n gathers 5 consecutive past ones.

PASE+

PASE+ improves our previous encoder architecture as follows:

Skip connections: Skip connections introduce shortcuts in the encoder architecture, which shuttle different levels of abstractions to the final representation as well as improving gradient flows. (1)

Quasi-RNN: PASE+ can learn long-term dependencies efficiently with a QRNN placed on the top of the convolutional layers. The QRNN gates do not rely on previous computations and can be computed in parallel for all the time steps

PASE+ architecture

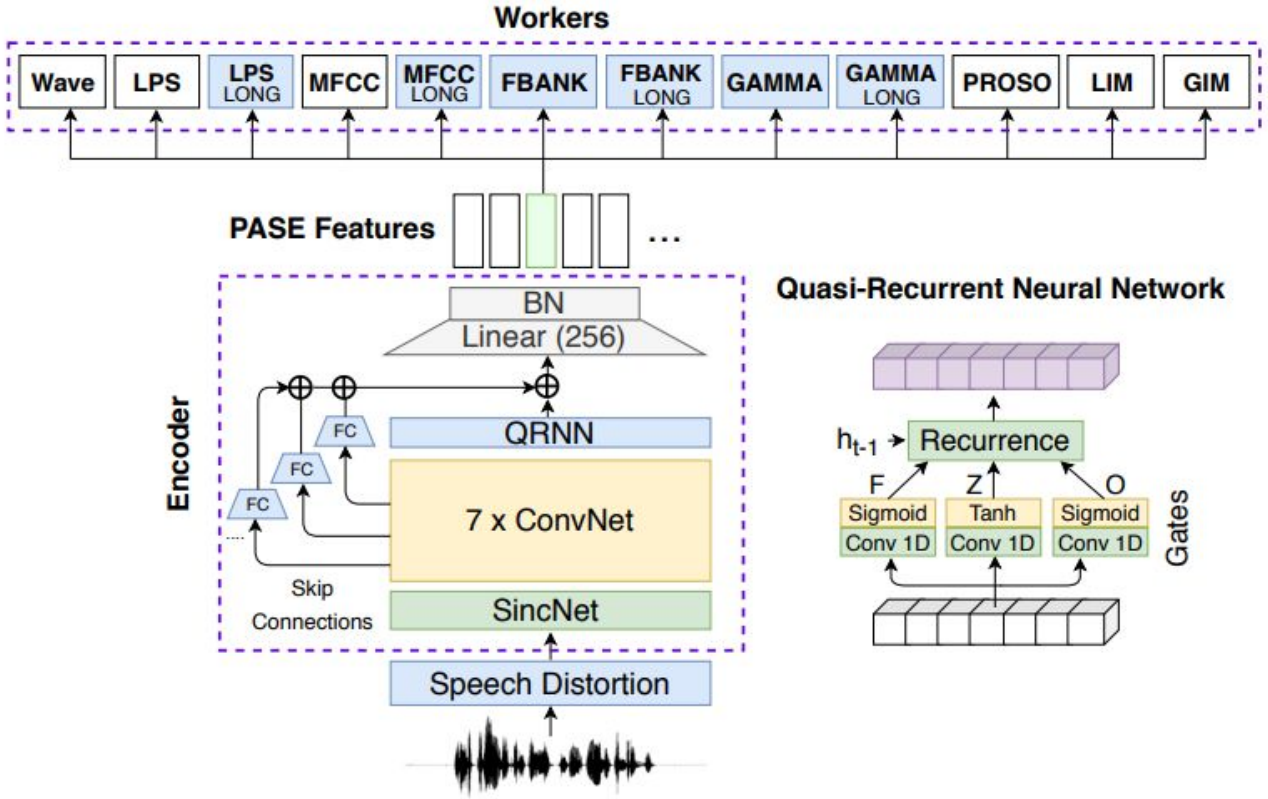

Fig. 1. The proposed PASE+ architecture for self-supervised learning. In blue are the main differences with the previous version of PASE.

Extended pretraining

Regression tasks

Adding more features: Added new workers that estimate 40 FBANKS and 40 Gammatone features

Estimating longer context: Instead of estimating the current feature only, jointly estimate multiple features within a context window of seven neighbouring frames.

Estimating features on longer windows: Estimated the aforementioned features computed over longer analysis windows of 200 ms rather than the usual 25 ms used by the other regressors

PASE results

Table 1: *Accuracies using PASE and an MLP as classifier. Rows below the "all workers" model report absolute accuracy loss when discarding each worker for self-supervised training.*

Model	Classification accuracy [%]		
	Speaker-ID *(VCTK)*	Emotion *(INTERFACE)*	ASR *(TIMIT)*
PASE (All workers)	97.5	88.3	81.1
– Waveform	−1.3	−3.9	−0.3
– LPS	−1.5	−5.3	−0.5
– MFCC	−2.4	−3.2	−0.7
– Prosody	−0.5	−5.3	−0.1
– LIM	−0.8	−1.3	−0.0
– GIM	−0.6	−0.5	−0.3
– SPC	−0.4	−1.6	−0.0

PASE+ results

	TIMIT Clean	TIMIT Rev+Noise
PASE (10h) [15]	18.6	41.1
+ 50 hours	18.3	39.9
+ Speech distortions	18.1	37.6
+ QRNN	18.1	37.0
+ Skip connection	18.0	36.2
+ Embedding 256	17.8	34.8
+ New workers	**17.2**	**33.8**

Table 2. Phone error rate (PER) obtained on the TIMIT corpus (clean and noisy) with different versions of PASE.

PASE+ results

	TIMIT rev+noise	DIRHA rev+noise
MFCC	37.1	35.8
FBANK	37.8	34.0
GAMMATONE	38.4	35.6
MFCC+FBANK+GAMM.	37.1	32.0
PASE+ (Supervised)	35.6	31.5
PASE+ (Frozen)	33.8	28.3
PASE+ (FineTuned)	**32.7**	**27.4**

Table 3. Phone error rate (PER) obtained on the TIMIT and DIRHA corpora (noise+reverb versions) with different input features.

	dev	eval
MFCC	75.9	69.5
MFCC + ivectors	74.1	65.7
PASE (Frozen, 10h)	77.9	72.0
PASE+ (Frozen)	74.1	67.5
+MFCC	73.6	66.7
+ivectors	**73.3**	**65.0**

Table 4. CHiME-5 WERs(%) on distant beamformed microphones.