Self-supervised learning applied to speaker and language recognition

Theo Lepage

EPITA Speaker and Language Recognition (ESLR)
Supervised by Reda Dehak

Seminar — July 2021

The issue with supervised learning

Question: Why are large human-labeled datasets important?

$\implies$ The number of trainable parameters of recent models is constantly increasing and thus larger training sets are mandatory to avoid overfitting.

Drawbacks of supervised learning:

1. Labeling datasets is expensive, tedious and slow.
2. Manual labeling is not scalable to the amount of data available today.
3. It could lead to biased models towards the considered problem.

The rise of self-supervised learning

Self-supervised learning

SSL relies on supervisory signals generated from the data itself. The principle is to train the model on a **pretext task** and use the learned representations for a different **downstream task** (through transfer learning for instance).

Different self-supervised strategies:

- **Contrastive tasks**: predicting hidden parts of the signal, maximizing mutual information between representations sampled from the same temporal context, ...
- **"Autoencoder" tasks**: predicting transformations that can be directly derived from the input signal

Our objective

Supervised learning is a bottleneck for building intelligent speaker and language recognition models because of the lack of large amount of labeled samples.

$\Longrightarrow$ The objective is to create an efficient **self-supervised model** designed for **speaker and language recognition** tasks.

SSL for audio: CPC [van den Oord et al., 2019]

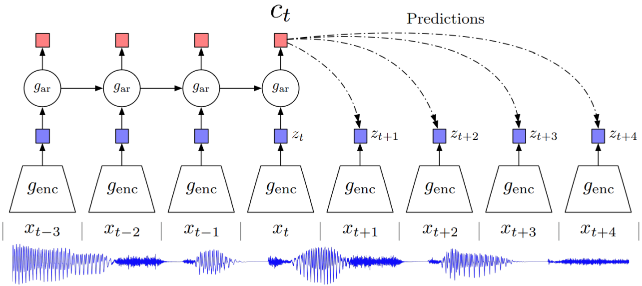

Figure: Contrastive Predictive Coding (CPC) model architecture.

$$f_k\left(x_{t+k}, c_t\right) = \exp\left(z_{t+k}^{T} W_k c_t\right) \quad (1)$$

$$\mathcal{L}_{\mathrm{NCE}} = -\underset{X}{\mathbb{E}}\left[\log \frac{f_k\left(x_{t+k}, c_t\right)}{\sum_{x_j \in X} f_k\left(x_j, c_t\right)}\right] \quad (2)$$

SSL for audio: LIM/GIM [Ravanelli and Bengio, 2019a]

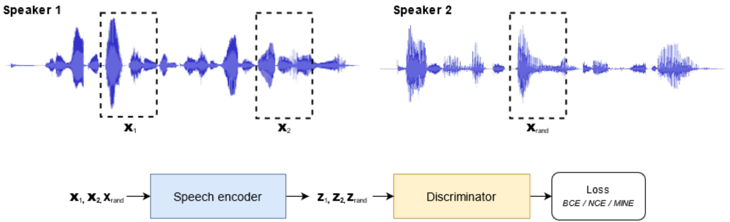

Figure: Local Info Max (LIM) model architecture.

Different objective functions to maximize mutual information:

- Binary crossentropy
- Noise-Contrastive Estimation *(NCE)* [Gutmann and arinen, 2010]
- Mutual Information Neural Estimation *(MINE)* [Belghazi et al., 2018]

SSL for audio: wav2vec 2.0 [Baevski et al., 2020b]

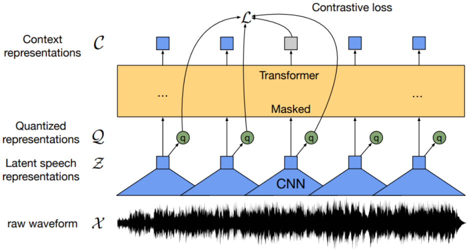

Figure: wav2vec 2.0 model architecture.

- Based on Transformers [Vaswani et al., 2017] similarly to BERT [Devlin et al., 2019]
- Specifically designed for speech recognition but extremely data-efficient
- Previous versions:
  - wav2vec [Schneider et al., 2019]
  - vq-wav2vec [Baevski et al., 2020a]

SSL for audio: PASE/PASE+ [Ravanelli et al., 2020]

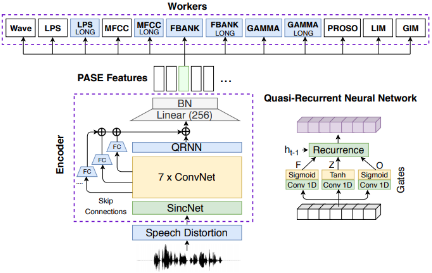

Figure: Problem Agnostic Speech Encoder (PASE+) model architecture.

- Encoder based on SincNet [Ravanelli and Bengio, 2019b]
- Produce generalist representations
- Relying on techniques introduced previously: LIM, GIM, CPC
- Use regressors modules acting as regular autoencoders

The connection with mutual information

Most SSL methods are **contrastive** and aim at **maximizing mutual information** between **representations sampled from the same temporal context**.

$$MI\left(z_1, z_2\right) = \int_{z_1} \int_{z_2} p\left(z_1, z_2\right) \log\left(\frac{p\left(z_1, z_2\right)}{p\left(z_1\right) p\left(z_2\right)}\right) dz_1 dz_2 \tag{3}$$

In the case of CPC, the optimal objective function can be written as Eq. 4.

$$\mathcal{L}_{\mathrm{NCE}}^{\mathrm{opt}} = -\mathop{\mathbb{E}}_{X} \log\left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}}\right] \geq -MI(x_{t+k}, c_t) + log(N) \tag{4}$$

Thus, we have $MI\left(x_{t+k}, c_t\right) \geq \log(N) - \mathcal{L}_{\mathrm{NCE}}^{\mathrm{opt}}$.

Objective functions to maximize mutual information

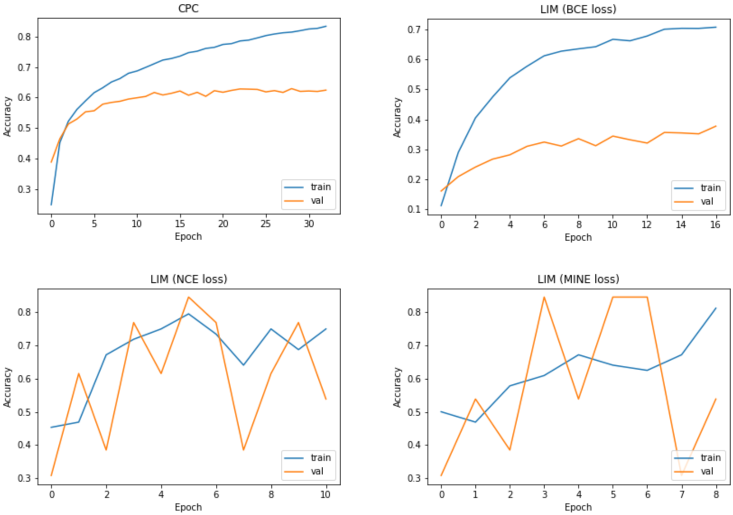

Figure: Evolution of different loss functions aiming at maximizing mutual information.

Implementation of the framework

Throughout the semester, my work was dedicated to create a Python library to train and evaluate self-supervised models for speaker and language recognition [1].

1. Implement all models presented previously with TensorFlow.
2. Handle audio datasets by caching data.
3. Evaluation on speaker recognition, speaker verification, language recognition and data-efficiency.
4. Modular configuration files.

[1] `https://github.com/theolepage/ssl-for-slr`

Vectorized implementation of CPC loss

```python
@tf.function
def cpc_loss(nb_timesteps_to_predict, predictions, X_future_encoded):
    # Shape: (batch_size, nb_timesteps_to_predict, encoded_dim)

    batch_size = tf.shape(predictions)[0]

    losses = tf.zeros((batch_size))

    for t in range(nb_timesteps_to_predict):
        dot = tf.linalg.matmul(X_future_encoded[:, t, :],
                               predictions[:, t, :],
                               transpose_b=True)

        # Determine loss
        log_softmax_dot = tf.nn.log_softmax(dot, axis=0)
        diag = tf.linalg.tensor_diag_part(log_softmax_dot)
        losses += diag

    losses /= tf.cast(nb_timesteps_to_predict, dtype=tf.float32)

    # Compute the average loss and accuracy across all batches
    loss = tf.math.reduce_mean(losses)

    return -loss
```

Listing 1: TensorFlow implementation of CPC loss

Improvements of CPC base model (1)

We introduced several improvements to CPC base model (**cpc-spk-1**):

- Sinc-based encoder [Ravanelli and Bengio, 2019b] (**cpc-spk-2**)
  Intuition: capture voice characteristics with band pass filters.
- Bidirectional GRU (**cpc-spk-3**)
  Intuition: learn to predict past frames with future frames.
- Sinc-based encoder + Data-augmentation (**cpc-spk-4**)
  Intuition: use data-augmentation techniques to learn more generalist representations.

Improvements of CPC base model (2)

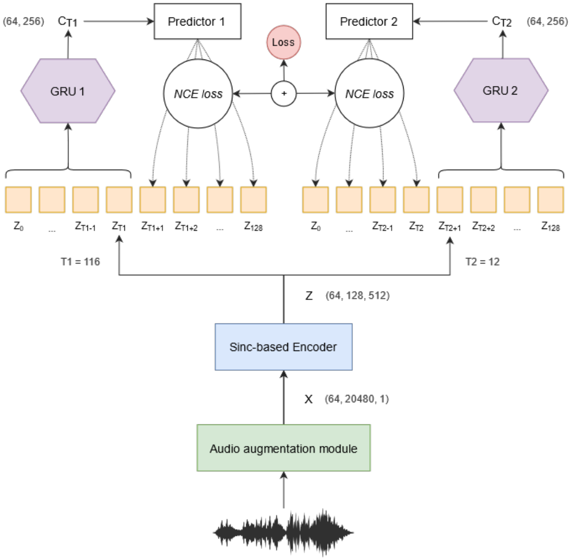

Figure: CPC model architecture with our improvements

Experimental protocol (1)

We first train models in a self-supervised way (**pretext task**) before training a classifier on top of their representations (**downstream task**).

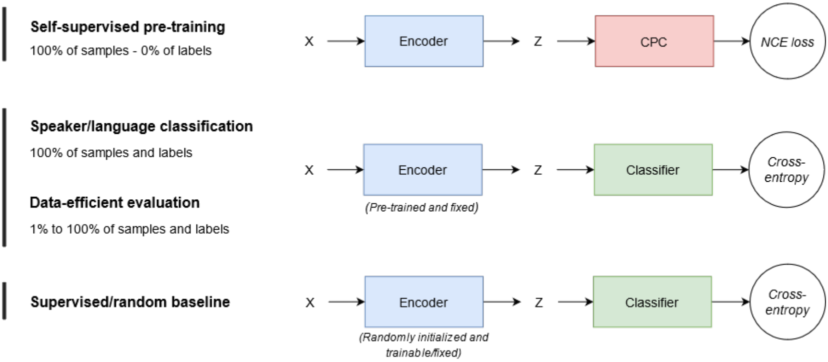

Figure: Overview of our training and evaluation framework

Experimental protocol (2)

Datasets:

- Speaker recognition: LibriSpeech [Vassil Panayotov and Khudanpur, 2015]
- Language recognition: VoxLingua107 [Valk and Alumäe, 2020]

Setup:

- We use frames of 1.28 second (20480 values at 16kHz)
- Batch size of 64
- Adam optimizer with a learning rate of 1×10^{-4}
- L2 weight normalization factor of 1×10^{-4}
- Each training stops after 5 epochs without a better validation loss
- 2x NVIDIA TITAN X GPU

Results on speaker classification

Model	Accuracy	# of params	Training time
Random initialization	1.00%	6,518M	3 hours 50 min
Supervised baseline	83.92%	6,518M	3 hours 20 min
cpc-spk-1 (Base model)	77.90%	6,518M	1 hour 30 min
cpc-spk-2 (Sinc-encoder)	72.29%	6,518M	1 hour 30 min
cpc-spk-3 (Bidirectional GRU)	86.23%	7,175M	2 hours 40 min
cpc-spk-4 (Data-augmentation)	55.10%	6,518M	1 hour 10 min

Table: Linear classification of 2338 speakers from LibriSpeech.

Model	Accuracy	# of params	Training time
Random initialization	2.00%	7,445M	2 hours 50 min
Supervised baseline	74.50%	7,445M	4 hours 30 min
cpc-spk-1 (Base model)	81.58%	7,445M	1 hour
cpc-spk-2 (Sinc-encoder)	79.51%	7,445M	1 hour
cpc-spk-3 (Bidirectional GRU)	87.78%	8,168M	1 hour 30 min
cpc-spk-4 (Data-augmentation)	61.09%	7,445M	1 hour 20 min

Table: MLP classification of 2338 speakers from LibriSpeech.

The issue faced with language recognition

(a) Speaker embeddings (b) Language embeddings

Figure: PCA on embeddings provided by the self-supervised model

$\Longrightarrow$ Our model struggles to capture language features as it is easier for the contrastive task to rely on speaker identity.

Is our approach data-efficient?

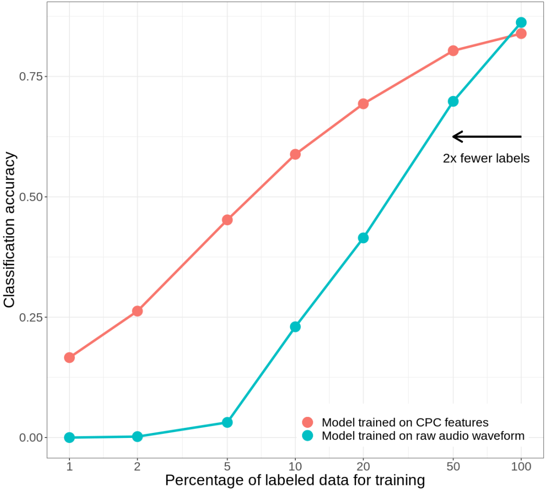

Figure: Data-efficient speaker evaluation of our best model.

Perspectives and conclusion

Future directions:

- Find a solution for language recognition.
- Train all models that have been implemented.
- Improve our experimental setup by evaluating on larger datasets (NIST SRE) and comparing our results with other self-supervised models.

$\Longrightarrow$ I am convinced that self-supervised learning is the key to build more intelligent speaker and language recognition systems.

Questions & Answers

Do you have any questions?

Reference I

Baevski, A., Schneider, S., and Auli, M. (2020a).
vq-wav2vec: Self-supervised learning of discrete speech representations.

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020b).
wav2vec 2.0: A framework for self-supervised learning of speech representations.

Belghazi, M. I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, R. D. (2018).
Mine: Mutual information neural estimation.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019).
Bert: Pre-training of deep bidirectional transformers for language understanding.

Gutmann, M. and arinen, A. H. (2010).
Noise-contrastive estimation: A new estimation principle for unnormalized statistical models.

Ravanelli, M. and Bengio, Y. (2019a).
Learning speaker representations with mutual information.

Ravanelli, M. and Bengio, Y. (2019b).
Speaker recognition from raw waveform with sincnet.

Ravanelli, M., Zhong, J., Pascual, S., Swietojanski, P., Monteiro, J., Trmal, J., and Bengio, Y. (2020).
Multi-task self-supervised learning for robust speech recognition.

Schneider, S., Baevski, A., Collobert, R., and Auli, M. (2019).
wav2vec: Unsupervised pre-training for speech recognition.

Reference II

Valk, J. and Alumäe, T. (2020).
Voxlingua107: a dataset for spoken language recognition.

van den Oord, A., Li, Y., and Vinyals, O. (2019).
Representation learning with contrastive predictive coding.

Vassil Panayotov, Guoguo Chen, D. P. and Khudanpur, S. (2015).
Librispeech: an asr corpus based on public domain audio books.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017).
Attention is all you need.