Speaker and Speech Recognition from raw waveform with SincNet

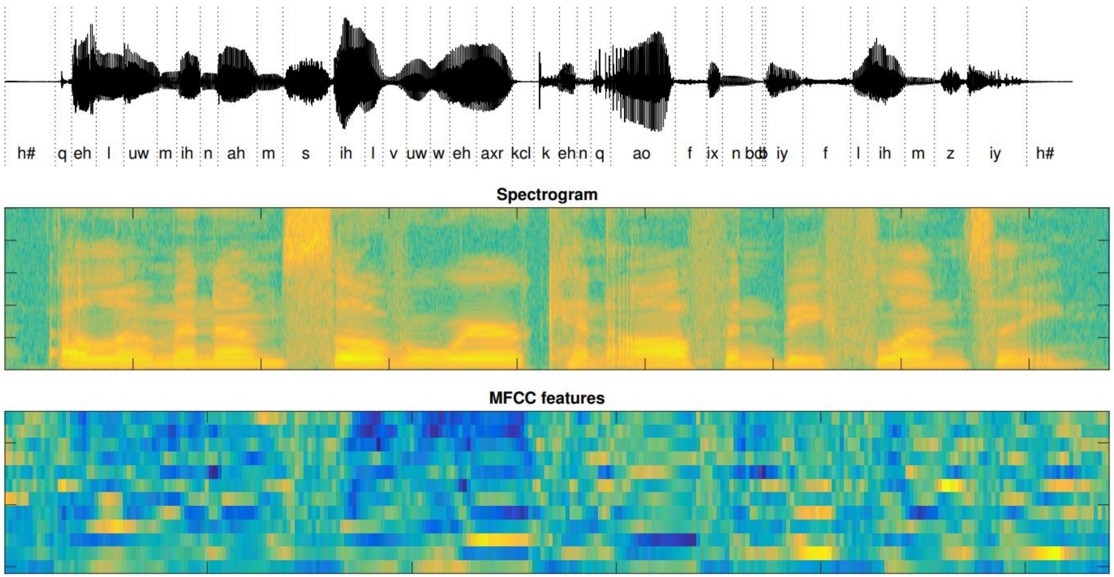
Mirco Ravanelli

Mila

On Processing Waveforms…

Problem:

- Speech/Audio sequences are very high-dimensional.



Raw Waveform:
1 second = 16000 features

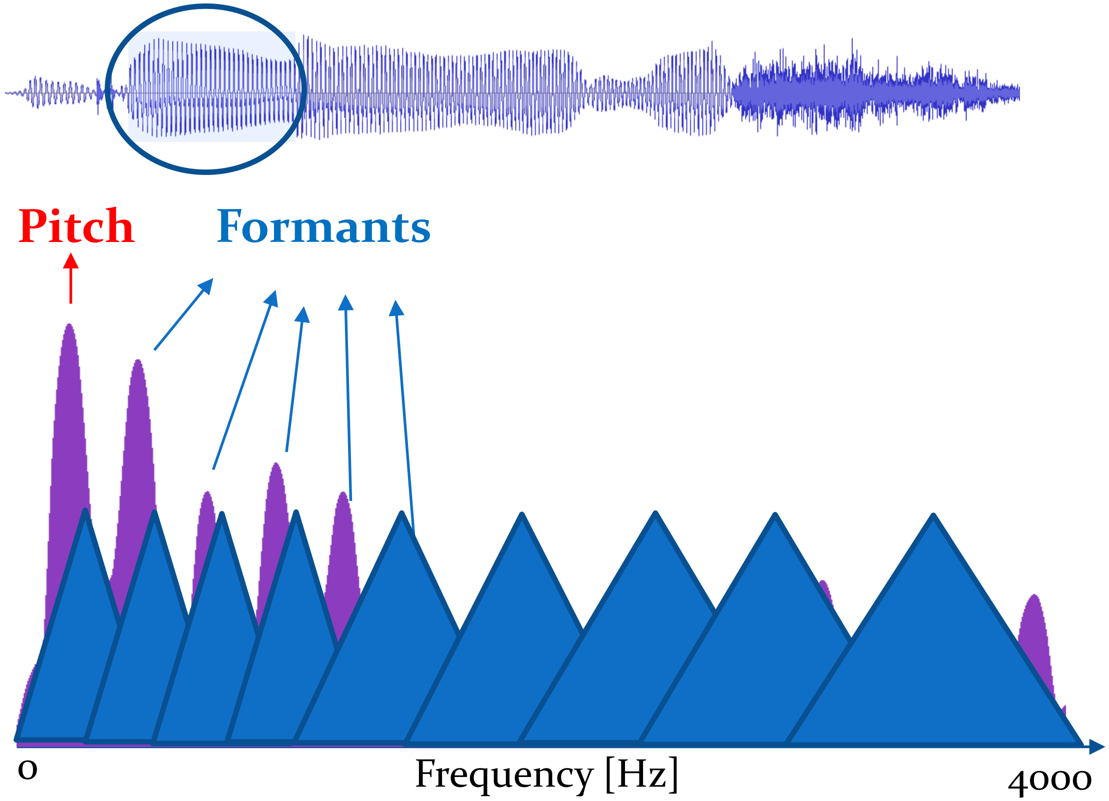
FBANKs/MFCCs:
1 second ≈ 4000 features

- *Hand-crafted* features (e.g. MFCCs, PLPs, or FBANKs) are still employed to achieve a more **compact representation**.

On Processing Waveforms…

Problem:

- *Hand-crafted* features are designed from **perceptual evidence.**



Not necessarily optimal for all audio/speech tasks!

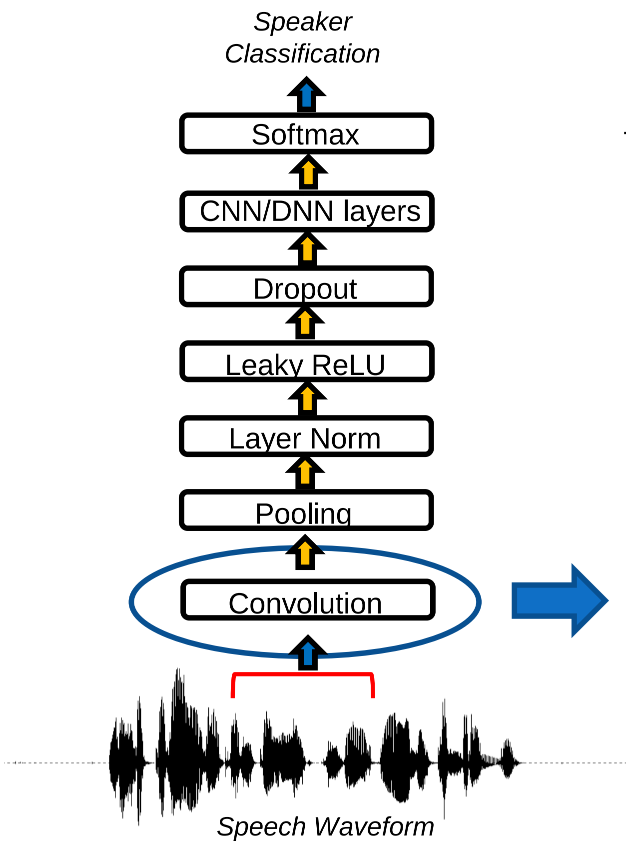
(e.g., Speaker recognition)

We **smooth the spectrum**, possibly hindering the extraction of pitch and formants.

Standard Approach

Problem:

- Recent works have proposed directly feeding CNNs with raw waveforms.



Convolution:

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

Critical Part:
- High Dimensionality
- Vanishing Gradient

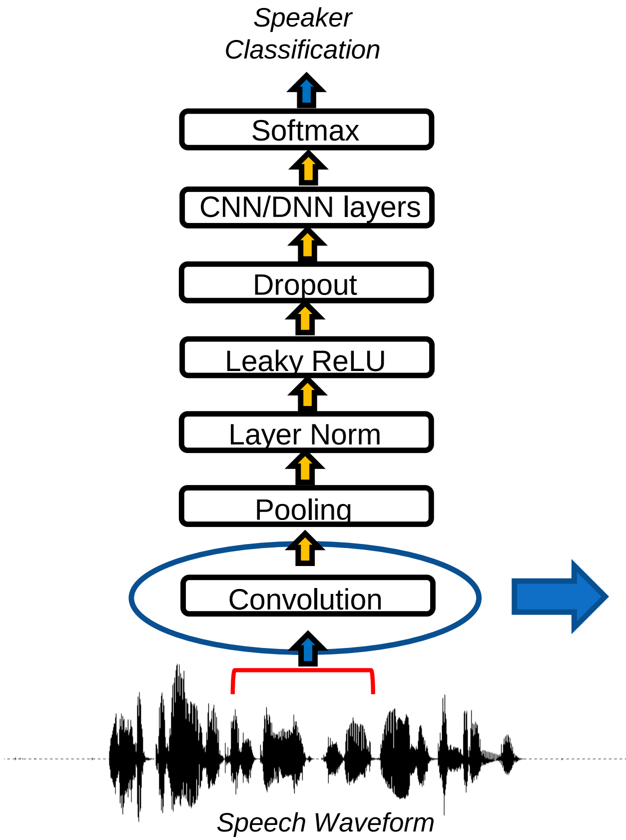
GitHub https://github.com/mravanelli/SincNet

arXiv.org M. Ravanelli, Y. Bengio, "*Speaker Recognition from raw waveform with SincNet*", in Proc. of SLT 2018.

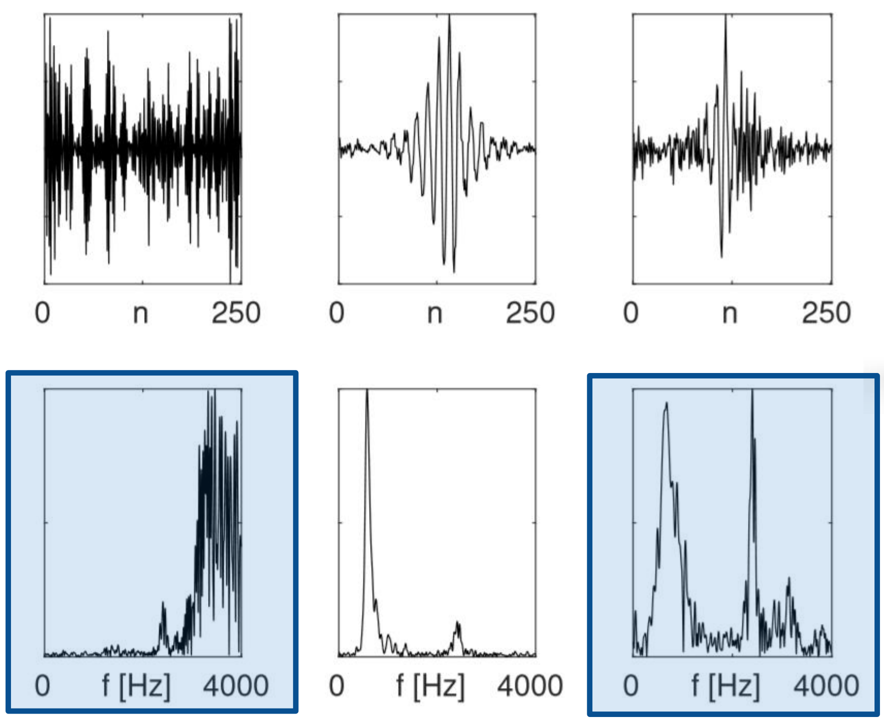
Standard Approach

Problem:

- Recent works have proposed directly feeding CNNs with raw waveforms.



CNN filters:



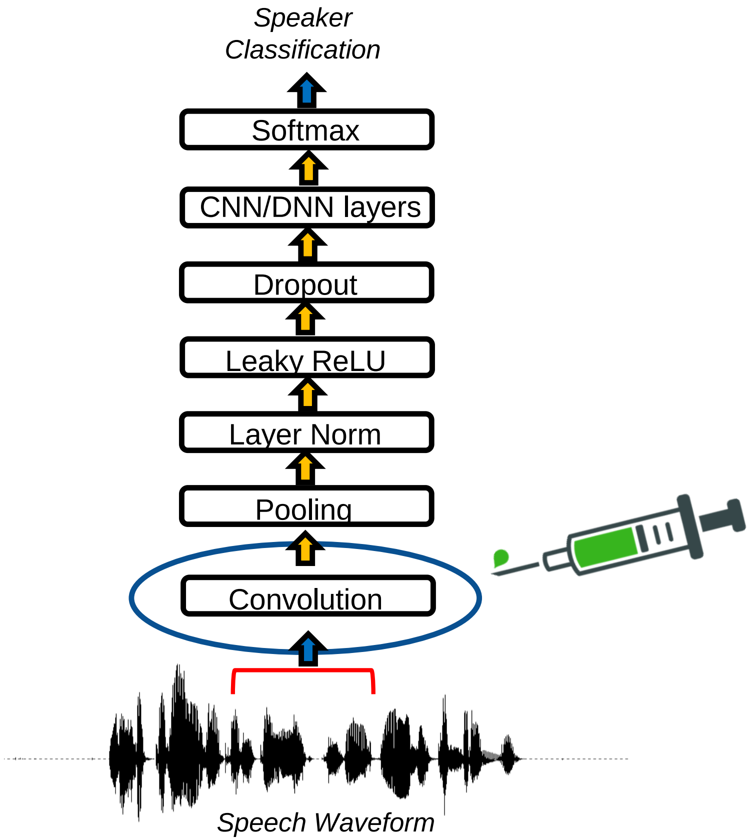
GitHub https://github.com/mravanelli/SincNet

arXiv.org M. Ravanelli, Y. Bengio, "*Speaker Recognition from raw waveform with SincNet*", in Proc. of SLT 2018.

SincNet

Problem:

- Recent works have proposed directly feeding CNNs with raw waveforms.



Can we help the CNN discover more <u>meaningful</u> filters?

<u>A simple idea</u>:
Inject prior knowledge on the filter shape

GitHub https://github.com/mravanelli/SincNet

arXiv.org M. Ravanelli, Y. Bengio, "*Speaker Recognition from raw waveform with SincNet*", in Proc. of SLT 2018.

SincNet

Standard CNN:

$$y[n] = x[n] * h[n]$$

We learn all the elements of each filter

SincNet:

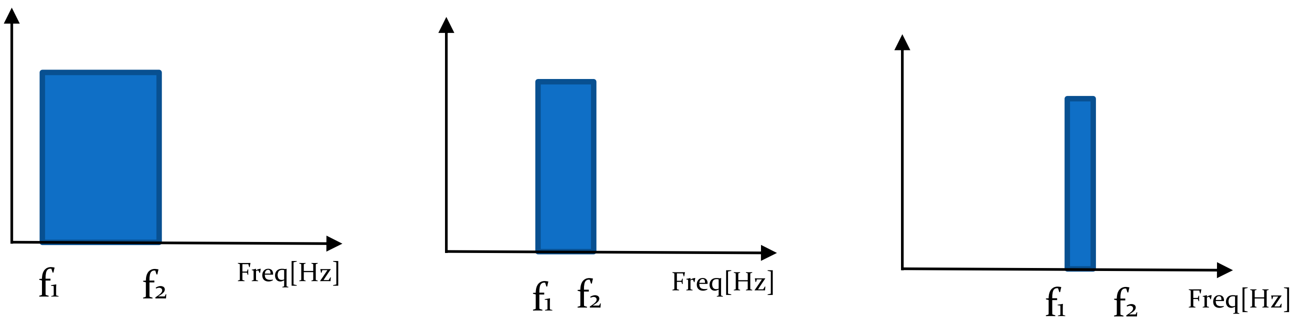
$$y[n] = x[n] * g[n, \theta]$$

We only learn the θ *parameters of the predefined kernel*

What could be a good choice for g(·)?

SincNet

- We can choose g(·) to implement a bank of **band-pass filters** where low and high **cutoff frequencies** are the only parameters **learned.**



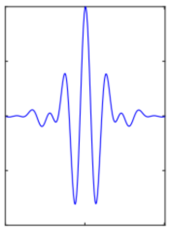
Frequency Domain:

$$G[f, f_1, f_2] = rect\left(\frac{f}{2f_2}\right) - rect\left(\frac{f}{2f_1}\right)$$

For each filter we only learn f1 and f2

Time Domain:

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

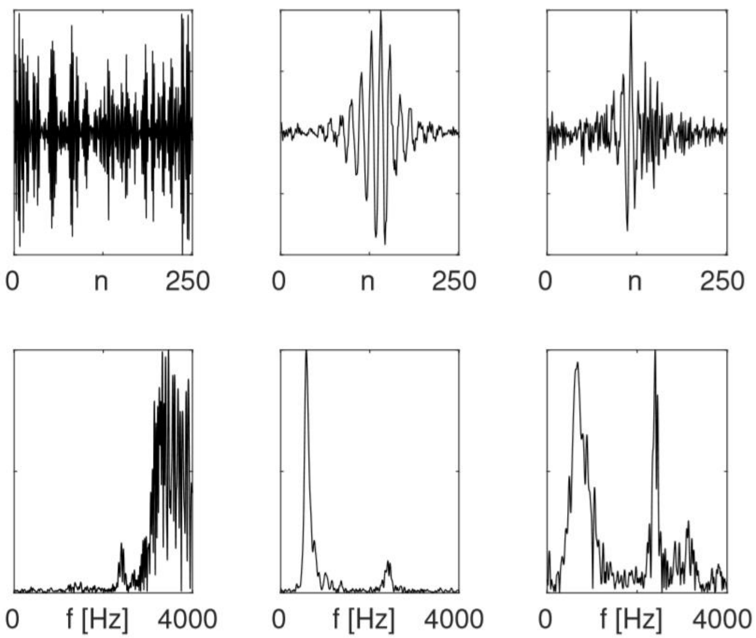


GitHub https://github.com/mravanelli/SincNet

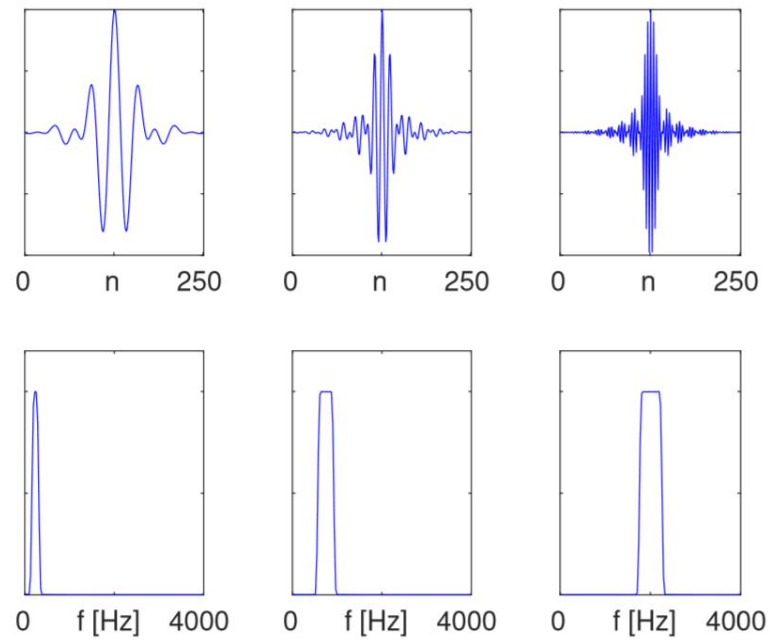
arXiv.org M. Ravanelli, Y. Bengio, "*Speaker Recognition from raw waveform with SincNet*", in Proc. of SLT 2018.

SincNet

CNN Filters:



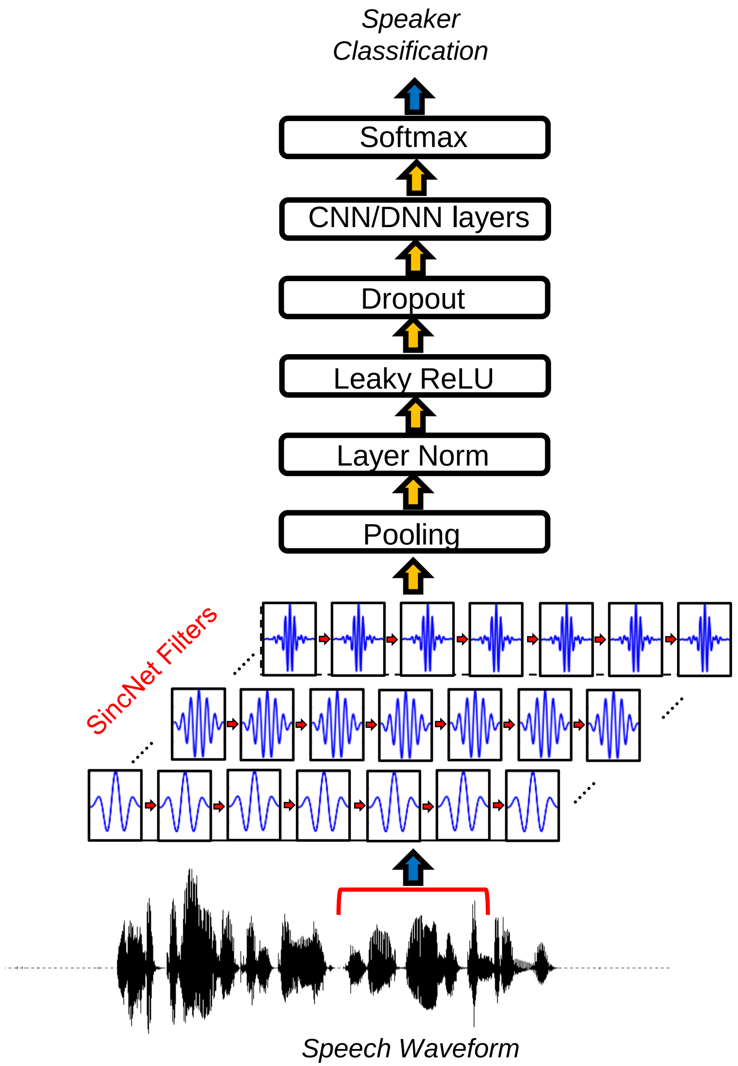
SincNet Filters:



GitHub https://github.com/mravanelli/SincNet

arXiv.org M. Ravanelli, Y. Bengio, "*Speaker Recognition from raw waveform with SincNet*", in Proc. of SLT 2018.

SincNet



Model Properties

- **Few Parameters**:

F= Number of filters (e.g. 80)
L= Length of each filter (e.g. 100)

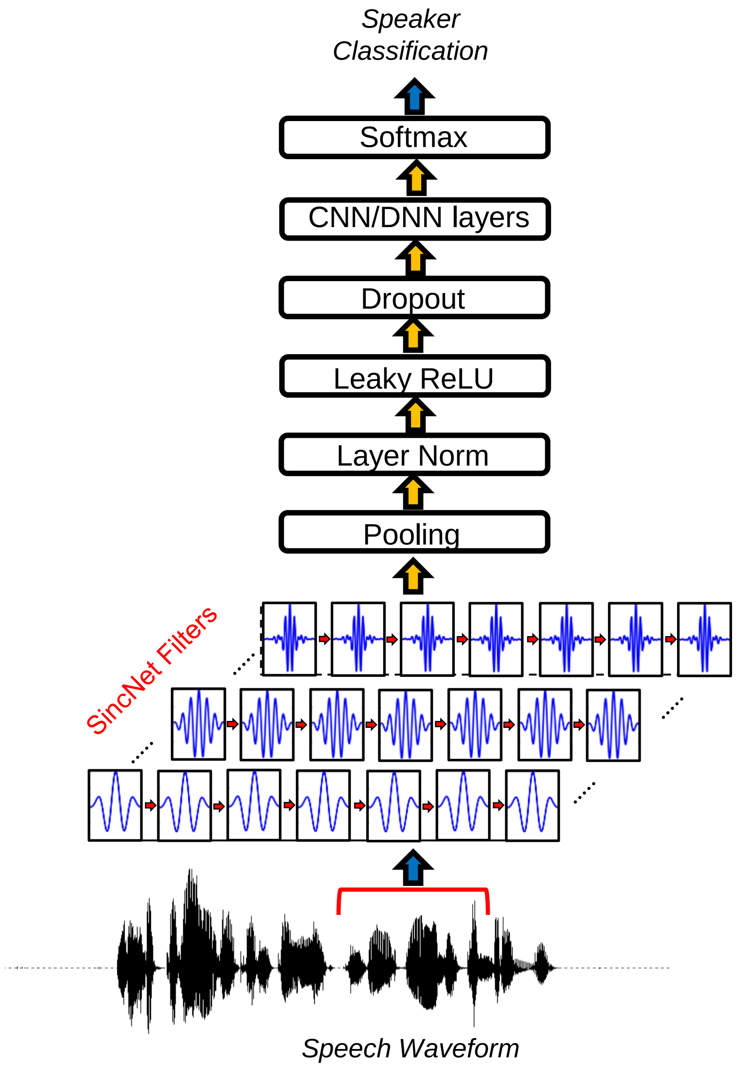
Standard CNN
F · L parameters (8k)

SincNet
2F parameters (160)

The number of parameters doesn't depend on L.

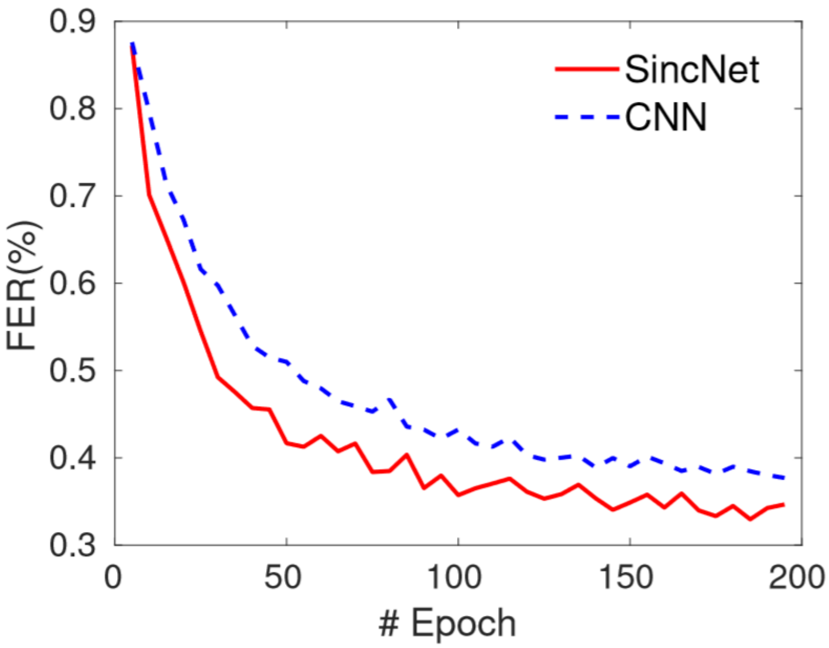
We can achieve **high frequency selectivity** without wasting parameters!

SincNet

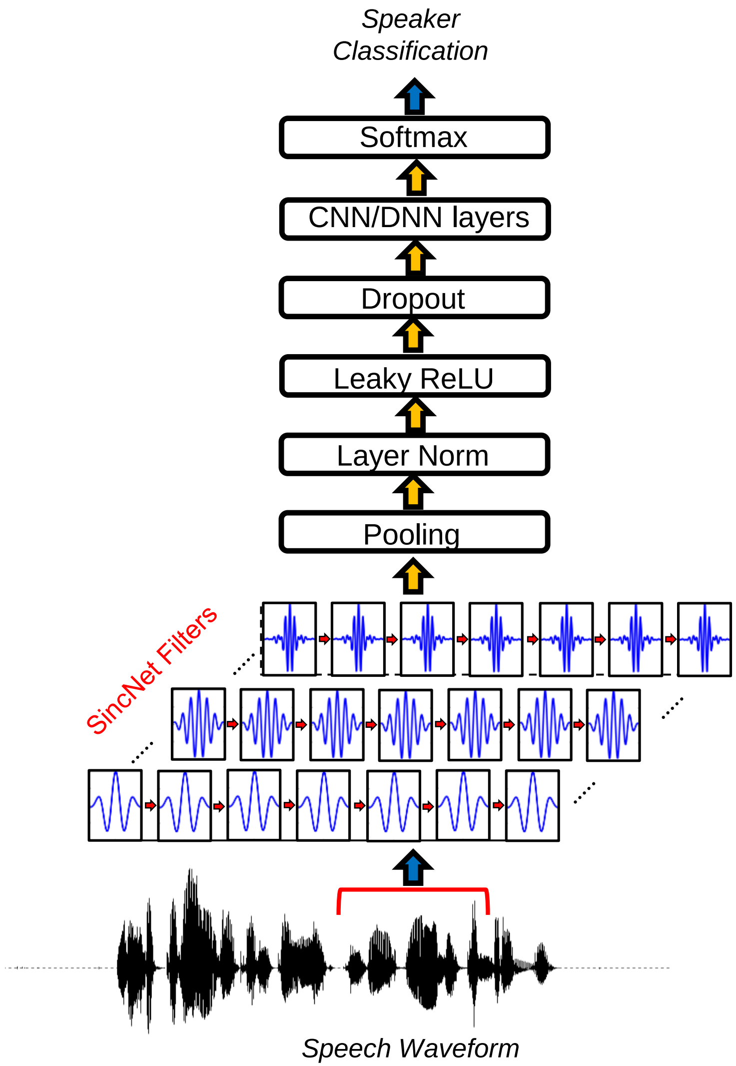


Model Properties

- **Few Parameters**
- **Fast Convergence**



SincNet



Model Properties

- **Few Parameters**
- **Fast Convergence**
- **Interpretability**

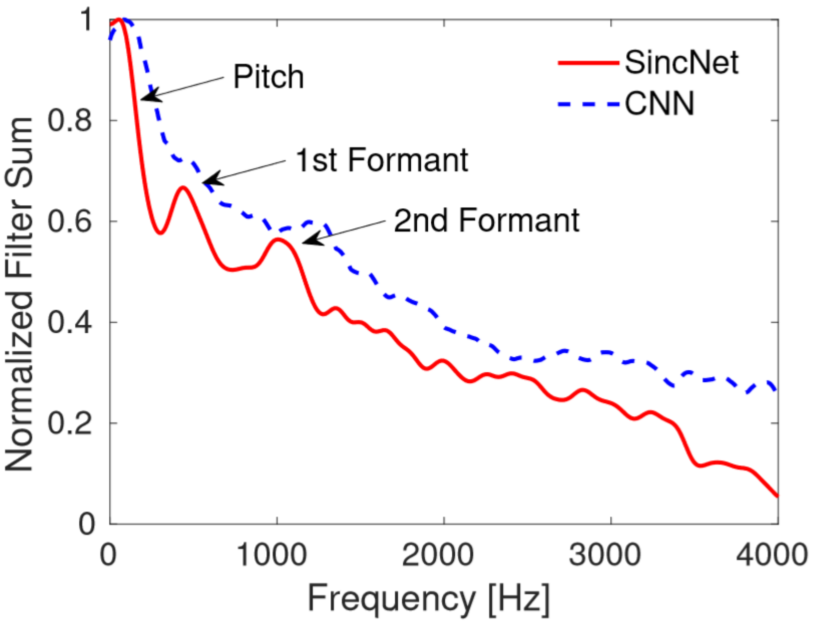
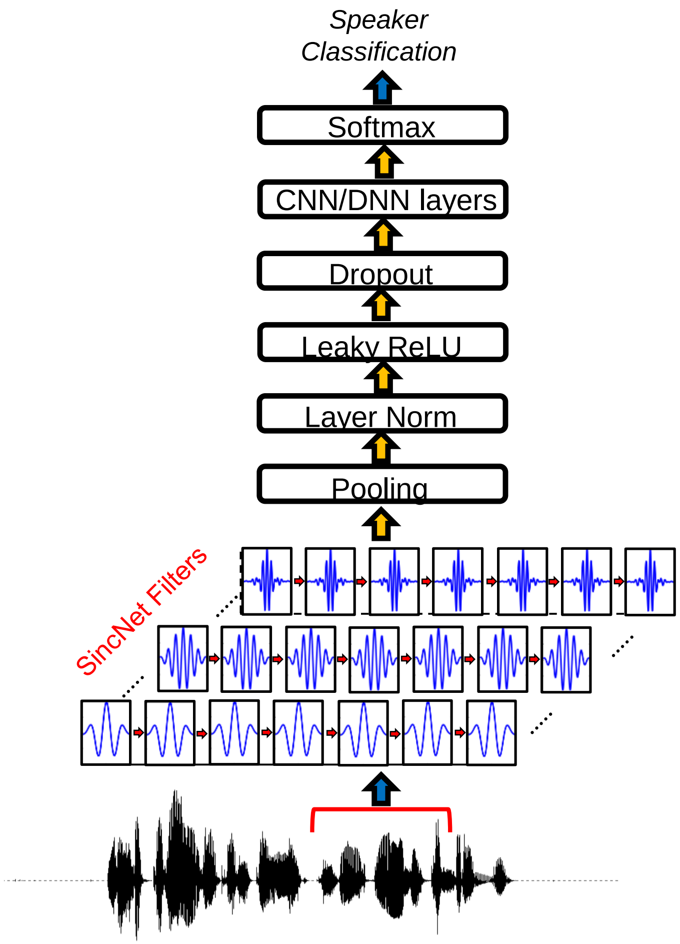
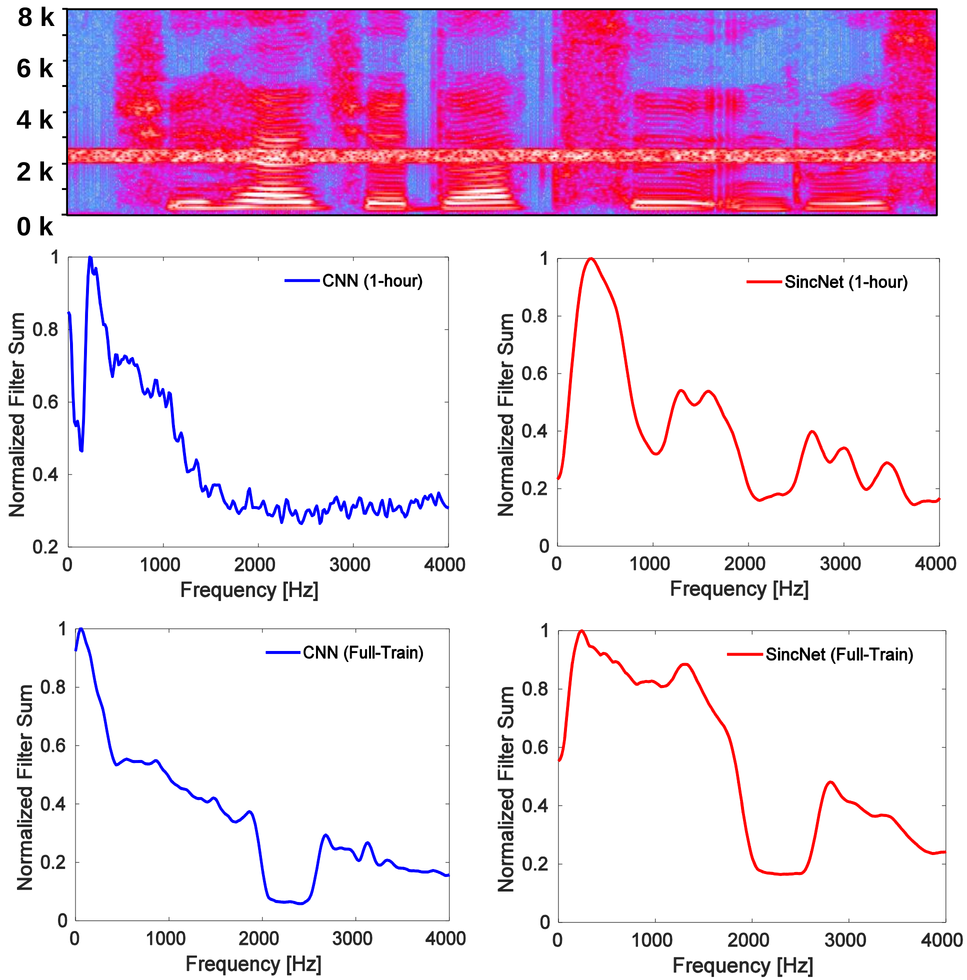


Fig. 3: Cumulative frequency response of the SincNet filters.

SinCNet



- **Interpretability**



GitHub https://github.com/mravanelli/pytorch-kaldi

arXiv.org M. Ravanelli, Y. Bengio, "Interpretable Convolutional Filters with SincNet", in Proc. of NIPS@IRASL 2018.

Speaker Recognition Results

Training: 12-15 seconds for each speaker
Test: short sentences (from 2 to 6 seconds)

Speaker Identification Performance:

	TIMIT	LibriSpeech
DNN-MFCC	0.99	2.02
CNN-FBANK	0.86	1.55
CNN-Raw	1.65	1.00
SINCNET	**0.85**	**0.96**

Tab 1: Classification Error Rate (CER%) on TIMIT (462 spks) and Librispeech (2484 spks).

Speaker Verification Performance:

	d-vector
DNN-MFCC	0.88
CNN-FBANK	0.60
CNN-Raw	0.58
SINCNET	**0.51**

Tab 2: Equal Error Rate (SER%) on Librispeech with the d-vector approach

I-Vector EER = 1.1 %

GitHub https://github.com/mravanelli/SincNet

arXiv.org M. Ravanelli, Y. Bengio, "*Speaker Recognition from raw waveform with SincNet*", in Proc. of SLT 2018.

Speech Recognition Results

	TIMIT	DIRHA
CNN-FBANK	18.3	40.1
CNN-Raw waveform	18.3	40.5
SincNet-Raw waveform	**18.0**	**38.2**

Tab 3: Speech Recognition error rates (%) obtained for TIMIT and for the DIRHA dataset.

- SincNet works for ASR as well!
- SincNet works in noisy and reverberant conditions

GitHub https://github.com/mravanelli/pytorch-kaldi

arXiv.org M. Ravanelli, Y. Bengio, "Interpretable Convolutional Filters with SincNet", in Proc. of NIPS@IRASL 2018.

Conclusion

Summary:

- SincNet has shown **promising** on speaker and speech recognition task.
- Analysis of the SincNet filters reveals that the learned filter-bank is tuned to address the specific task.

Towards Unsupervised Learning of Speech Representations

Mirco Ravanelli

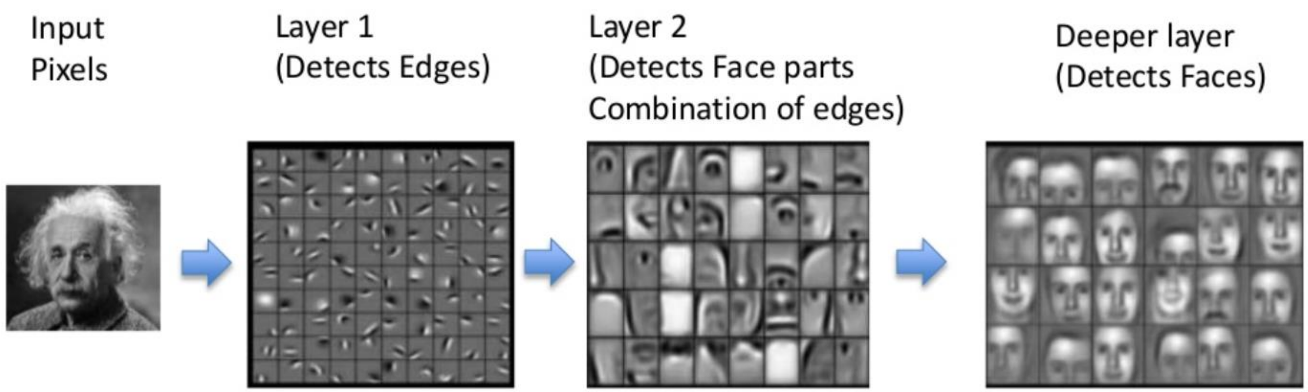
Mila

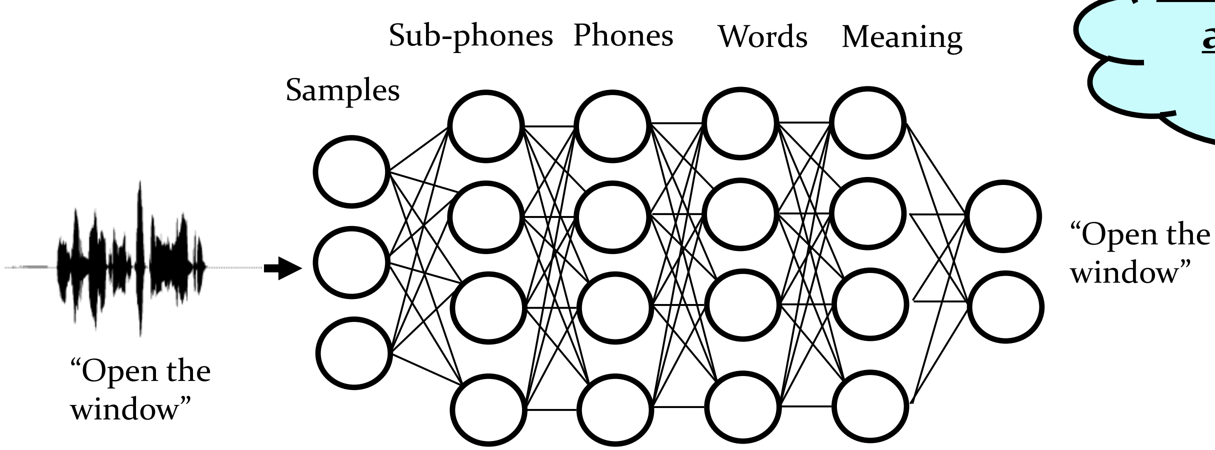
Outline

- *Why unsupervised learning?*
- *Self-Supervised Learning*
- *Local Info Max (LIM)*
- *Problem-Agnostic Speech Encoder (PASE)*
- *Conclusion*

Why Unsupervised Learning?

- **Deep learning** = Learning hierarchical representations.

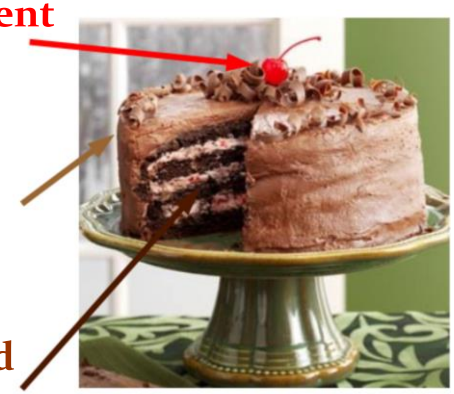




Can we learn them in an unsupervised way?

Why Unsupervised Learning?

Unsupervised Learning
=
Learning Without a Teacher

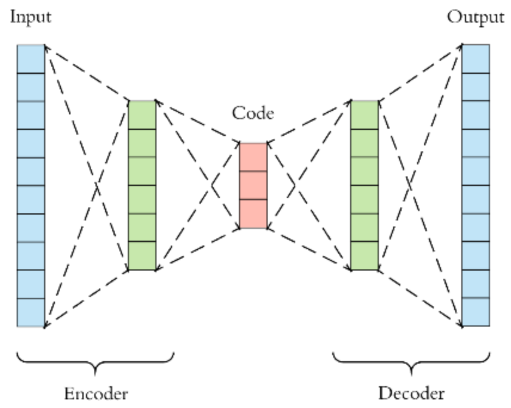


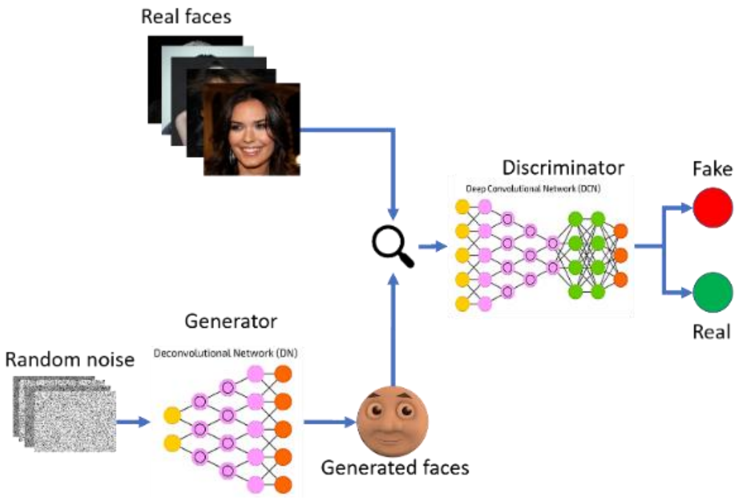
- Unsupervised learning is actually how humans/animals learn.
- Rapid generalization to a new task.
- Targets/rewards can be difficult/expensive to obtain or define.

Why Unsupervised Learning?

Some popular approaches:

- *Deep Belief Nets*
- *Autoencoders*
- *Variational Autoencoders*
- *Generative Adversarial Networks*





Self-Supervised Learning

A field that is gaining popularity in computer vision is **self-supervised learning.**

Self-supervised Learning = the supervision is extracted from the signal itself.

- In general, this is performed by applying **known transforms** to the input data and using the resulting outcomes as targets.

Relative Positioning

(Doersch et. al. , ICCV 2015)

Colourization

(Zhang et al., ECCV 2016)

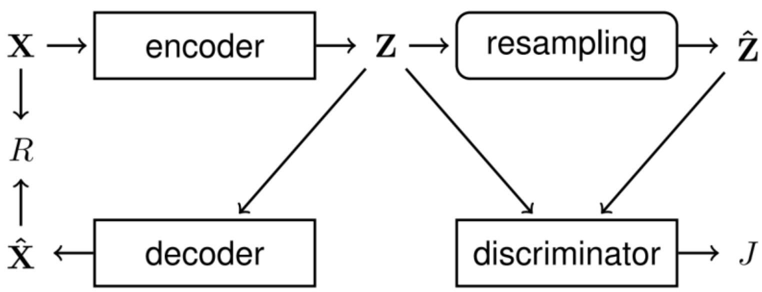
Correct Rotation

90° rotation 270° rotation

(Gidaris et al., ICLR 2018)

Self-Supervised Learning

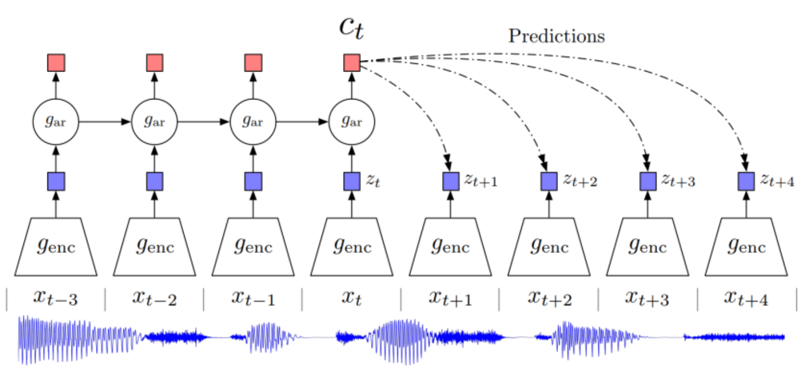
Some recent works have used self-supervised learning to learn speech representations:



We learn independent features for speech separation.

P. Brakel, Y. Bengio, "Learning independent features with adversarial nets for non-linear ICA", 2017

<u>Constructive Predicting Coding (CPC)</u>



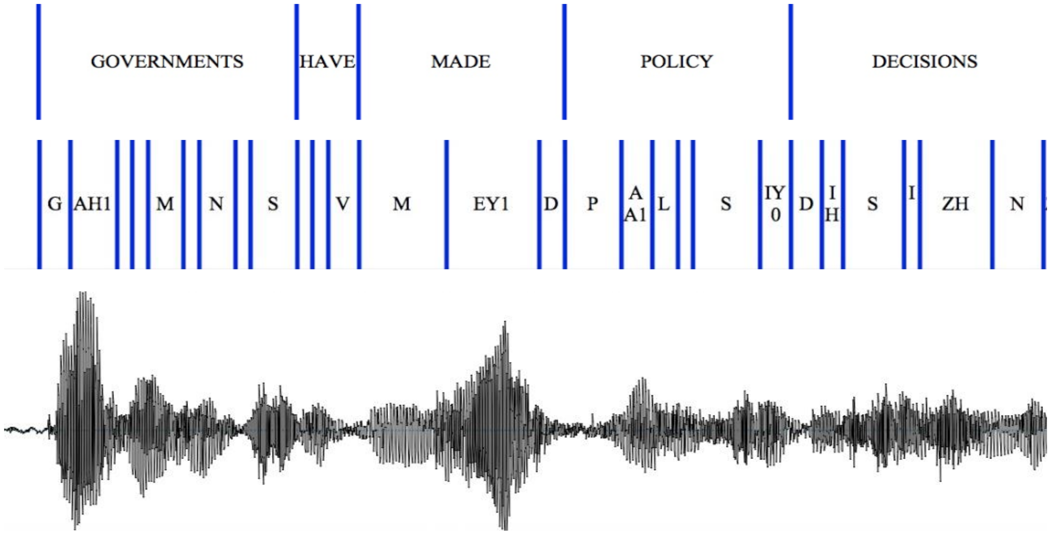
We learn features that are "predictable" about the future.

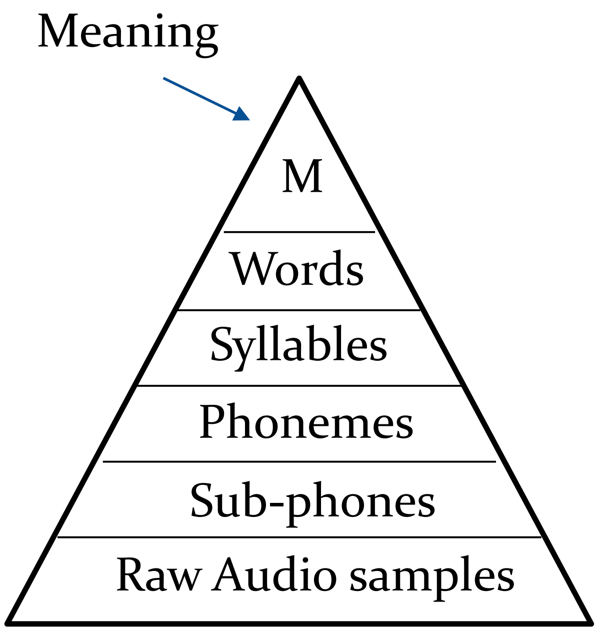
A. van den Oord, Y. Li, O. Vinyals, "Representation Learning with Contrastive Predictive Coding" ,2018

Self-Supervised Learning

<u>Self-supervised learning on speech: why is challenging?</u>

- High-dimensionality
- Long sequences
- Variable-length
- Complex hierarchical structure





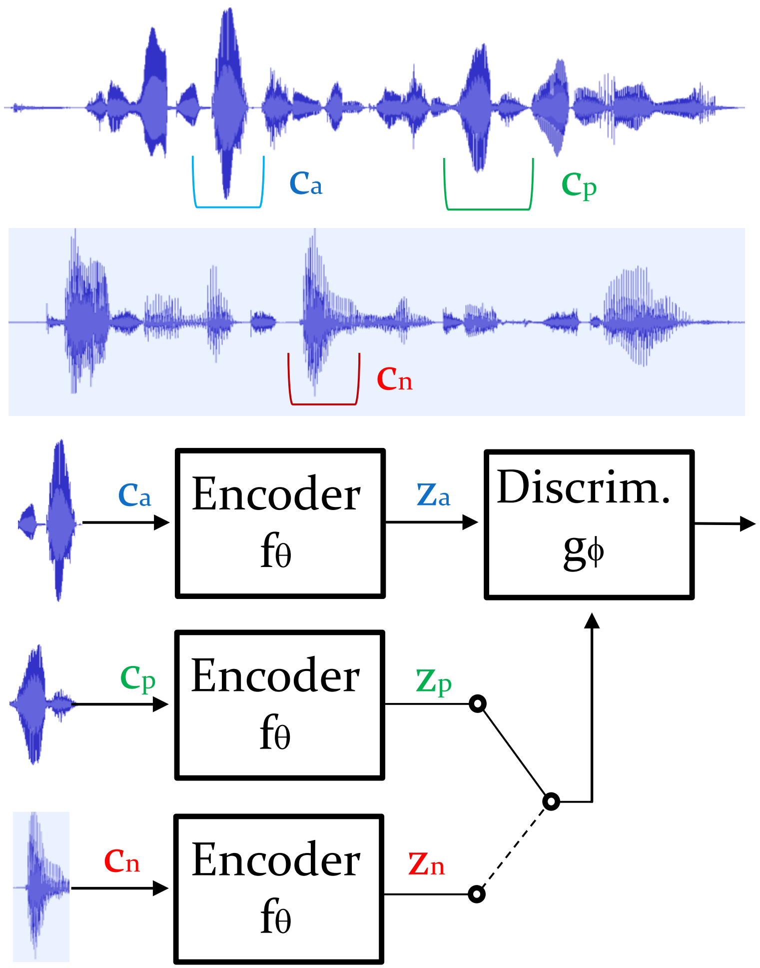
Local Info Max (LIM)

Goal: Learn good speaker representations with Mutual Information.

$$MI(z_1, z_2) = \int_{z_1}\int_{z_2} p(z_1, z_2) log\left(\frac{p(z_1, z_2)}{p(z_1)p(z_2)}\right) dz_1 dz_2$$
$$= D_{KL}\big(p(z_1, z_2)||p(z_1)p(z_2)\big)$$

- MI can capture complex **non-linear relationships** between random variables.
- MI is **difficult to compute** in high-dimensional spaces.
- MINE (*Belghazi, 2018*) found that it is possible to maximize or minimize the MI within a framework that **closely resembles** that of **GANs**.

Local Info Max (LIM)



<u>Sampling strategy:</u>

1. Choose a random chunk from a random sentence C_a (anchor).
2. Choose another random chunk from <u>the same sentence</u> C_p (positive).
3. Choose a random chunk from another random sentence C_n (negative).

<u>The game we play:</u>

1. Process C_a, C_p, C_n with an encoder.

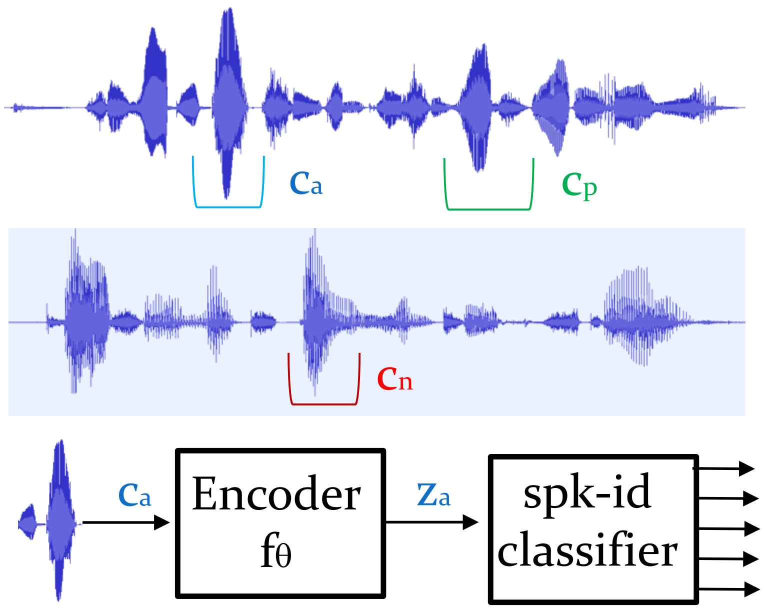
(Z_a, Z_p): sample from <u>the joint distribution</u> (positive sample).

(Z_a, Z_n): sample from the product of <u>marginal distribution</u> (negative sample).

2. We feed the discriminator with positive or negative samples.

3. The discriminator should figure out if their two inputs come from the **same** or **different** sentences.

Local Info Max (LIM)



- The discriminator loss is set to maximize the mutual information MI.
- Different choices are possible (MINE, Info-NCE, BCE).
- Encoder and discriminator are **jointly trained**.
- **Cooperative** game, **not adversarial**!
- The representations discovered by the encoder can be later used for the **supervised speaker recognition task.**

Local Info Max (LIM)

- ***Loss Comparison***

	Librispeech
Triplet Loss	1.33%
MINE	0.94%
Info-NCE	0.82%
Binary Cross Entropy (BCE)	**0.75%**

Tab. 1 Classification Error Rate obtained the speaker-id tasks (2484 spks) using LIM with various losses (the lower the better).

Insights:

- Mutual information losses (MINE, Info-NCE, BCE) **outperform** the **triplet loss**.
- Better embeddings can thus be derived with a divergence measure more meaningful than the simple **cosine distance** used in triplet loss.
- The best performance is achieved with the standard binary **cross-entropy**.

- Similar to (*D. Hjelm et. al, 2018*), we have observed that this **bounded** metric is more **stable** and **easier** to optimize.

Local Info Max (LIM)

- ***Speaker Identification on Librispeech (2484 spks)***

	Clean	**Rev**
Supervised	0.80	17.1
LIM (Frozen)	0.75	15.2
LIM (FineTuned)	0.56	9.6
LIM (joint Training)	**0.52**	**9.3**

Tab. 2 Classification Error Rate (CER%) obtained on speaker-id in clean and reverberant conditions (the lower the better).

Insights:

- LIM outperforms a **fully-supervised classifier**.
- The gap becomes more evident when **pre-training** the encoder with LIM and fine-tune it with the classifier (*LIM-FineTuned*).

- **Jointly training** from scratch **encoder**, **discriminator**, and **classifier** (*LIM-joint Training*) yields the best performance.

Local Info Max (LIM)

Strengths

LIM highlights high-quality speaker representations.

LIM is simple and efficient (local information only).

Issue

The LIM representations are very **task-specific**.

All previous approaches were based **on single self-supervised tasks** only.

But, Is it really possible to capture the complex structure of speech with a single-tasks only?

The risk is to focus on "specific" aspects of the speech signal only.

arXiv.org M. Ravanelli, Y. Bengio, "Learning Speaker Representations with Mutual Information", 2018

Problem-agnostic Speech Encoder (PASE)

Idea: jointly tackle **multiple self-supervised tasks**

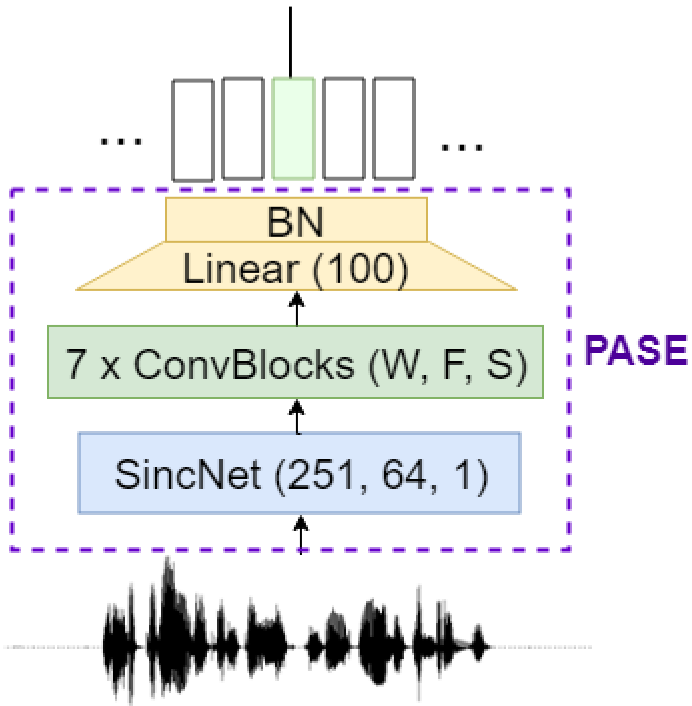
Where an ensemble of neural networks must **cooperate** to discover **good speech representations**.

Intuition:

- Each self-supervised task brings **a different "view"** on the speech signal.
- A **consensus** across these different "views" is needed, imposing several "*soft constraints*" to the representation.
- This way, our approach is more likely to learn **general**, **robust**, and **transferable features**.

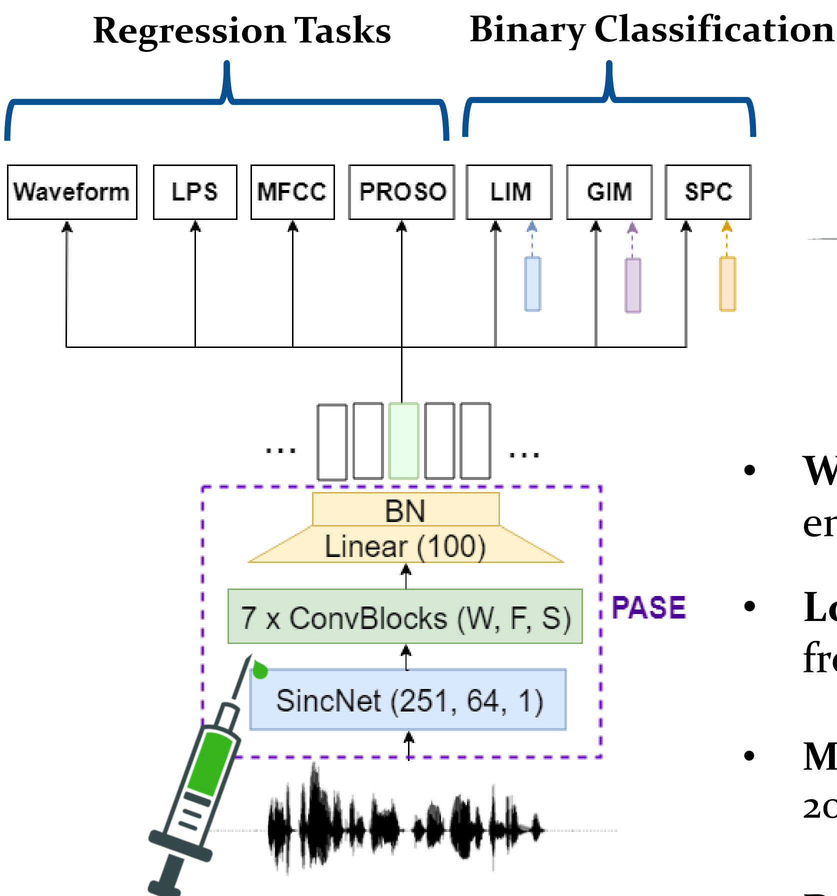
arXiv.org *S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, Y. Bengio " Learning Problem-agnostic Speech Representations from Multiple Self-supervised Tasks", 2019.*

Problem-agnostic Speech Encoder (PASE)



https://github.com/santi-pdp/pase

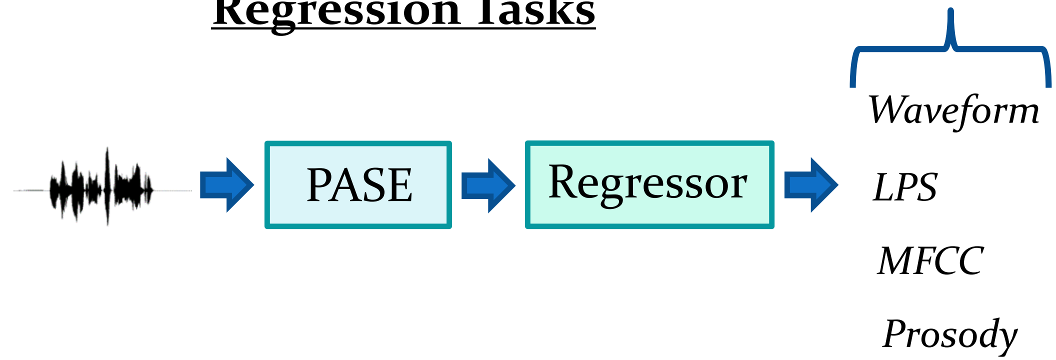
Problem-agnostic Speech Encoder (PASE)



We inject prior knowledge into the encoder!

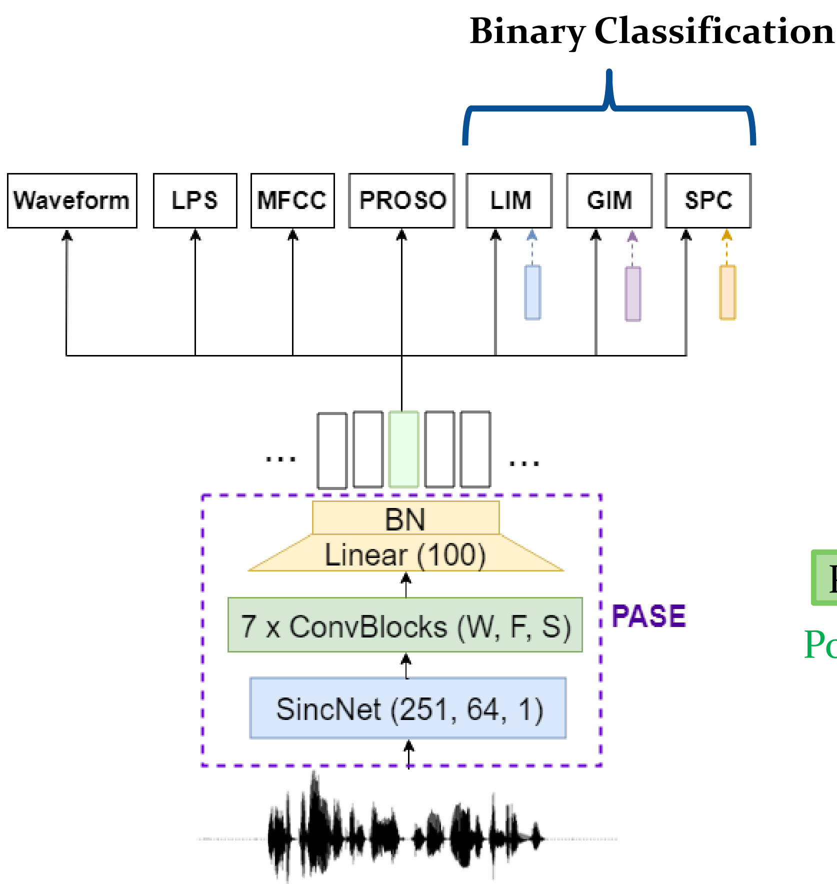
Known speech representations

Regression Tasks

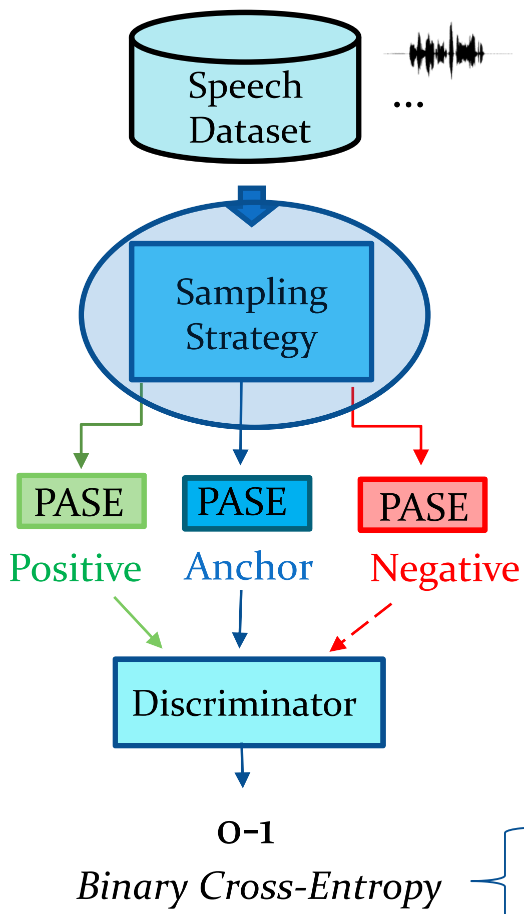


- **Waveform**: we predict the input waveform in an auto-encoder fashion.
- **Log power spectrum (LPS):** we compute it using 1024 frequency bins.
- **Mel-frequency cepstral coefficients (MFCC):** we extract 20 coefficients from 40 mel filter banks.
- **Prosody:** we predict fundamental freq., voiced/unvoiced probability, zero-crossing rate, and energy.

Problem-agnostic Speech Encoder (PASE)



Binary Classification Tasks



1. *We **sample** three speech chunks (i.e., **anchor**, **positive, negative chunks**). according to a predefined strategy.*

2. *We process all the chunks with PASE.*

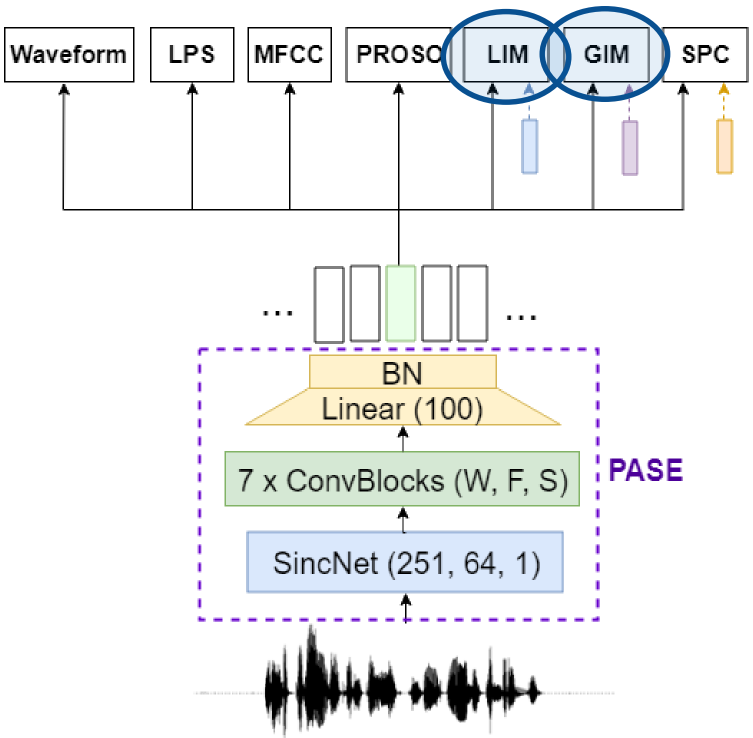
3. *Given the anchor, the **discriminator** should figure out if the other input is the positive or the negative one.*

<u>Intuition</u>:

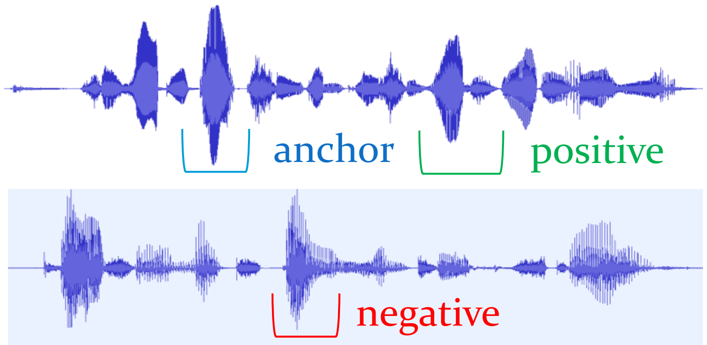
Positive + Anchor => “Close”

Negative + Anchor => “Distant”

Problem-agnostic Speech Encoder (PASE)

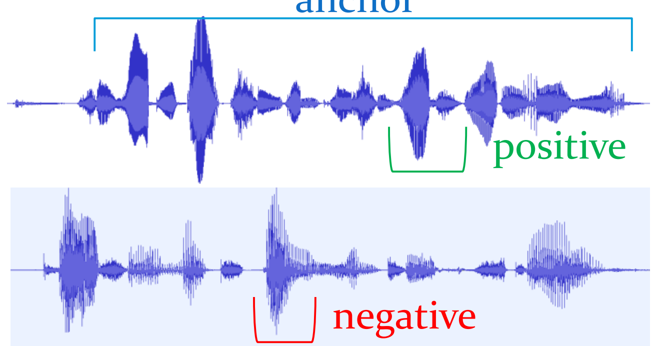


<u>Local Info Max (LIM)</u>



This way we highlight <u>speaker identities.</u>

<u>Global Info Max (GIM)</u> *(Hjelm,2018)*

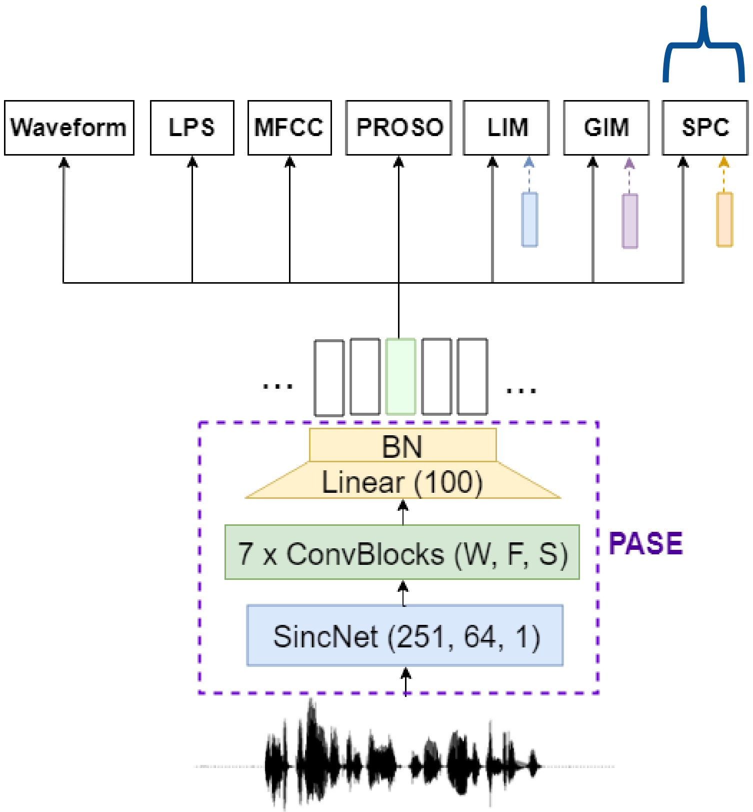


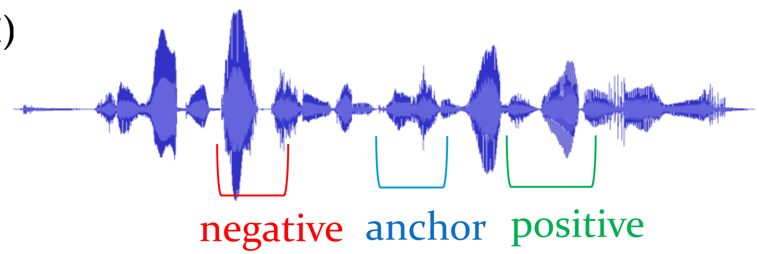
This way we highlight “global” information.

arXiv.org *S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, Y. Bengio " Learning Problem-agnostic Speech Representations from Multiple Self-supervised Tasks“, 2019.*

Problem-agnostic Speech Encoder (PASE)

Sequence Predictive Coding (SPC)





<u>Sampling strategy:</u>

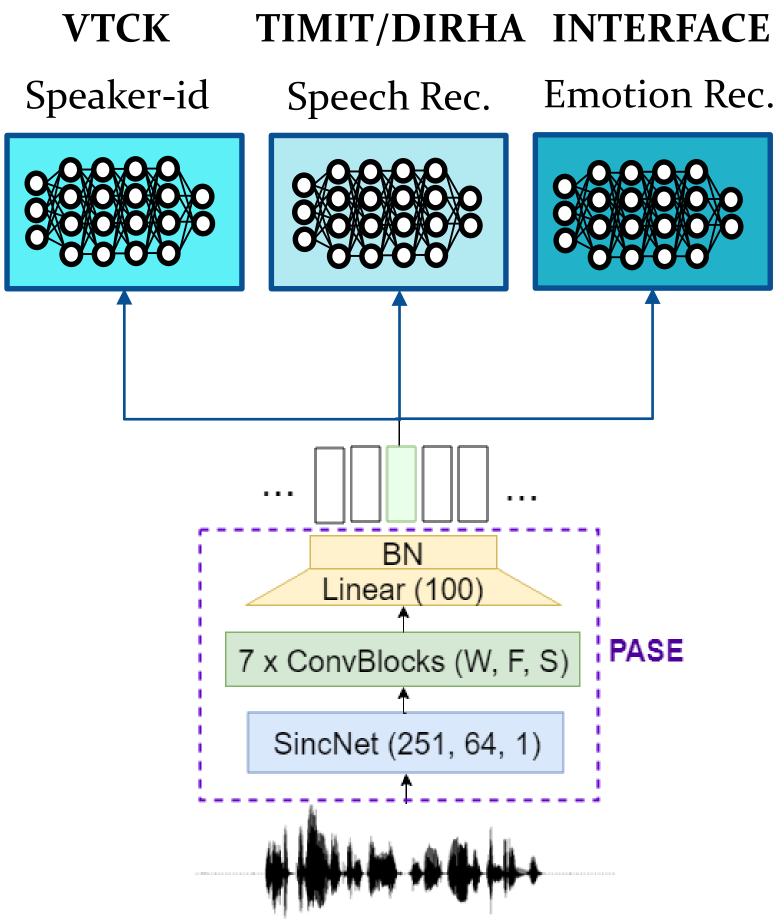
1. Choose a random chunk from a random sentence (anchor).
2. Choose another random chunk from <u>the future</u> of the same sentence (positive).
3. Choose another random chunk from <u>the past</u> of the same sentence (negative).

This way we want to capture **longer contextual information**.

Similar to CPC (van den Oord, 2018), but samples here are all from the same sentence.

A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding" 2018

Problem-agnostic Speech Encoder (PASE)



- The total loss that is computed as the **average of each worker cost.**
- Encoder and workers are **jointly trained** (using Librispeech).
- The encoder parameters will be updated pointing to a direction that is a **compromise among all the worker losses**.
- The representations discovered by the encoder can be later used for **supervised classification:**

PASE-Frozen: *we keep the encoder frozen during supervised training.*

PASE-FineTuned: *fine-Tuning the encoder during supervised training.*

Problem-agnostic Speech Encoder (PASE)

- ***Which self-supervised taks are needed?***

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

Insights:

- **No worker is dispensable** (the best results are achieved with all workers).
- Some workers are **helpful for all the speech tasks** (e.g., Waveform, LPS, and MFCC).
- Others turn out to be more **application-dependent** (e.g., Prosody, LIM, GIM, SPC).

arXiv.org *S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, Y. Bengio " Learning Problem-agnostic Speech Representations from Multiple Self-supervised Tasks", 2019.*

Problem-agnostic Speech Encoder (PASE)

- **Comparison with Standard Features**

Table 2: *Accuracy comparison on the considered classification tasks using MLPs and RNNs as classifiers.*

Model	Classification accuracy [%]					
	Speaker-ID *(VCTK)*		Emotion *(INTERFACE)*		ASR *(TIMIT)*	
	MLP	RNN	MLP	RNN	MLP	RNN
MFCC	96.9	72.3	90.8	91.1	81.1	84.8
FBANK	98.4	75.1	94.1	92.8	80.9	85.1
PASE-Supervised	97.0	80.5	93.8	92.8	82.1	84.7
PASE-Frozen	97.3	82.5	91.5	92.8	81.4	84.7
PASE-FineTuned	**99.3**	**97.2**	**97.7**	**97.0**	**82.9**	**85.3**

Insights:

- PASE features are **often better** than MFCCs and FBANKs, even when freezing the encoder (PASE-Frozen).
- The improvement is more evident when **pre-training the encoder and fine-tuning it** with the supervised task of interest (PASE-FineTuned).
- This approach consistently provides the **best performance over all the tasks and classifiers** considered here.

arXiv.org *S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, Y. Bengio " Learning Problem-agnostic Speech Representations from Multiple Self-supervised Tasks", 2019.*

Problem-agnostic Speech Encoder (PASE)

- ***Transferability***

Table 3: *Word error rate (WER) obtained on the DIRHA corpus.*

	WER [%]
MFCC	35.8
FBANK	34.0
PASE-Supervised	33.5
PASE-Frozen	32.5
PASE-FineTuned	**29.8**

Insights:

- Finally, we study the **exportability** of PASE to **acoustic conditions** that are **very different** from the clean one used to train it.

- Interestingly, PASE clearly outperforms the other systems even if it is not specifically designed to address **noise** and **reverberation**.

arXiv.org *S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, Y. Bengio " Learning Problem-agnostic Speech Representations from Multiple Self-supervised Tasks", 2019.*

Conclusion

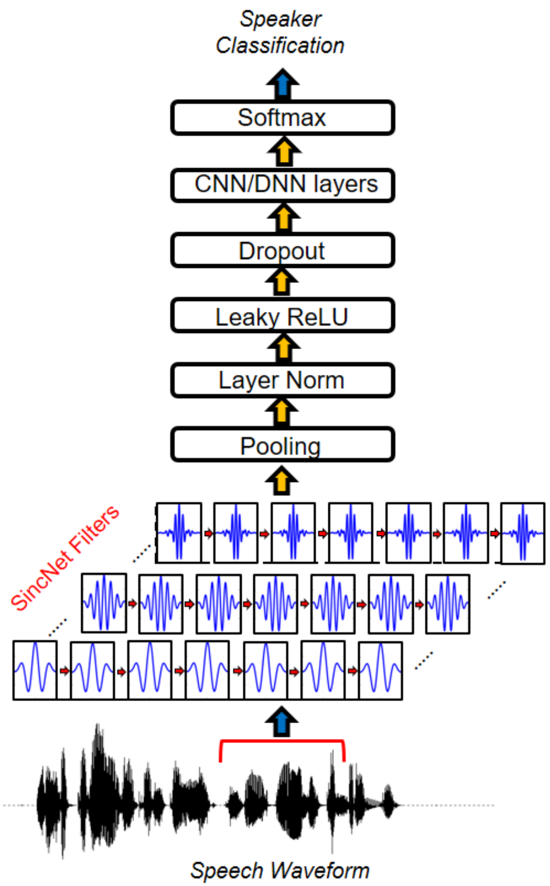
- PASE is **neural speech encoder** trained using multiple **self-supervised** tasks.
- The discovered embeddings turn out to carry important information related to, at least, **speaker identity**, **phonemes**, and **emotional cues.**
- It is designed to be **efficien**t and **fully parallelizable**.
- PASE can be used a standard **feature extractor** or as a **pre-trained model** (as commonly done in computer vision).
- It can be seen as a first step towards a **universal speech feature extractor**.

https://github.com/santi-pdp/pase

arXiv.org *S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, Y. Bengio " Learning Problem-agnostic Speech Representations from Multiple Self-supervised Tasks", 2019.*

Other Research Directions: SincNet

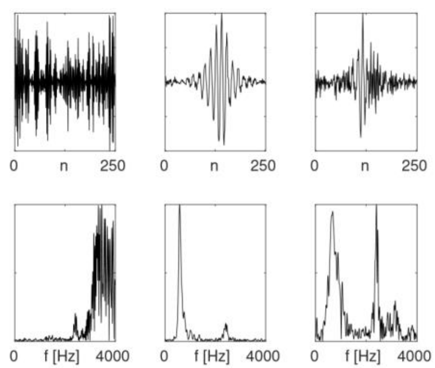
- SincNet is a convolutional architecture for efficiently processing **raw audio samples**.



Standard CNN:

$$y[n] = x[n] * h[n]$$

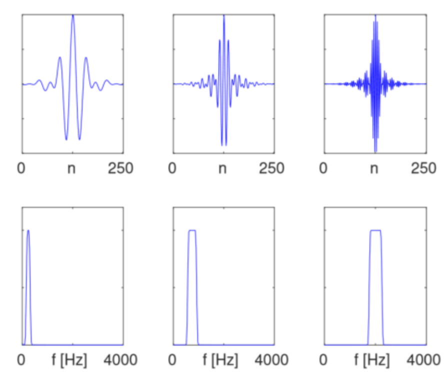
- We perform the convolution with a set of FIR filters.
- We learn **all the taps** of each filter.



SincNet:

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

- We perform the convolution with **sinc-based kernels** that implement band-pass filters.
- We learn only **low and high cut-off frequencies** of each filter.

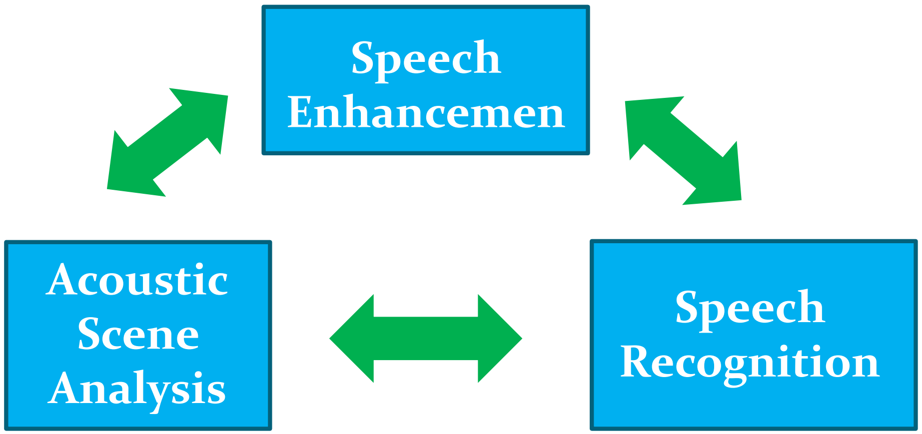


GitHub https://github.com/mravanelli/SincNet

arXiv.org M. Ravanelli, Y. Bengio, "*Speaker Recognition from raw waveform with SincNet*", in Proc. of SLT 2018.

Other Research Directions

Cooperative Neural Networks



M. Ravanelli, P. Brakel, M. Omologo, Y. Bengio, "*A network of deep neural networks for distant speech recognition*", in Proceedings of ICASSP 2017 (*best IBM student paper award*).

The PyTorch-Kaldi Project

PyTorch-Kaldi

https://github.com/mravanelli/pytorch-kaldi

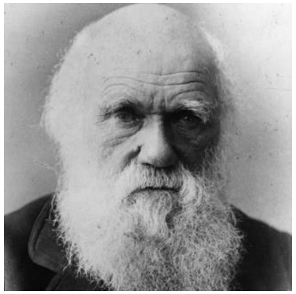
M. Ravanelli, T. Parcollet, Y. Bengio, "The PyTorch-Kaldi Speech Recognition Toolkit", in Proc. of ICASSP 2019.

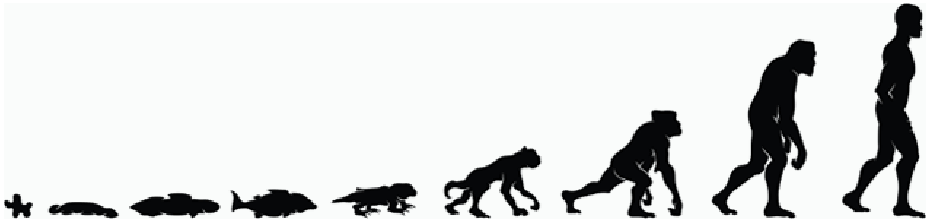
Montreal: the silicon valley of AI

Cooperative Networks of Deep Neural Networks

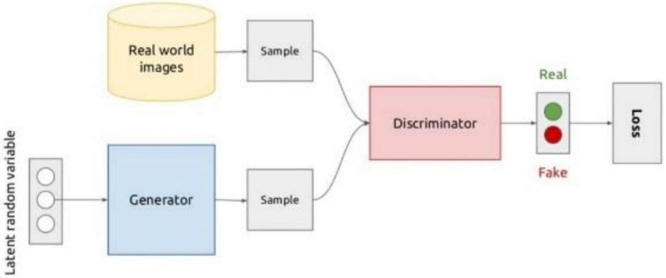
M. Ravanelli, P. Brakel, M. Omologo, Y. Bengio, "*A network of deep neural networks for distant speech recognition*", in Proceedings of ICASSP 2017 (*best IBM student paper award*).

Cooperative Networks of DNNs





DNN Competition:



AlphaGo

- **Competition** played a crucial role for the evolution of living forms.

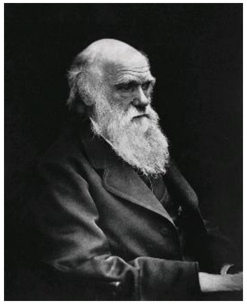
- Several deep learning systems are inspired by the "principle of competition"

M. Ravanelli, P. Brakel, M. Omologo, Y. Bengio, "A network of deep neural networks for distant speech recognition", in Proceedings of ICASSP 2017 (best IBM student paper award)

Cooperative Networks of DNNs

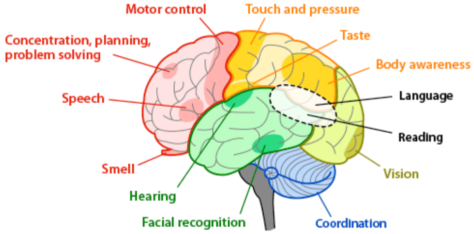
What about cooperation?

- **Cooperation** played a crucial role for the evolution of living forms as well.



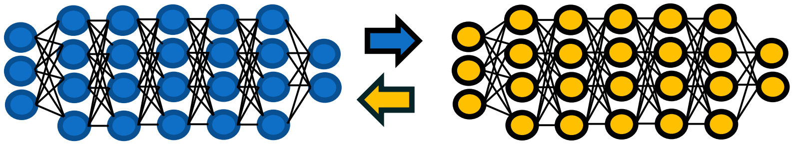
- Can we train **multiple DNNs** that learn how to cooperate?

- Cooperation can be helpful to **counteract uncertainty.**



- This paradigm can be exploited to solve **challenging problems**.

- **Distant speech recognition** represents the natural application field for this approach!



M. Ravanelli, P. Brakel, M. Omologo, Y. Bengio, "*A network of deep neural networks for distant speech recognition*", in Proceedings of ICASSP 2017 (*best IBM student paper award*).

Distant Speech Recognition (DSR)

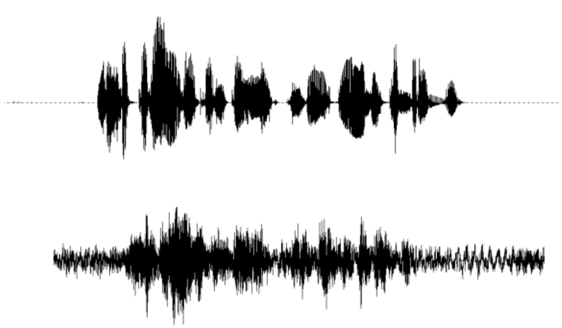
DIRHA system

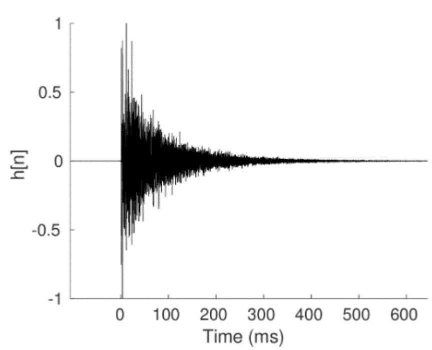
Amazon Echo

Google Home

<u>Applications:</u>

- Home Automation
- Smart TV
- Meeting Transcriptions
- Healthcare
- Robotics





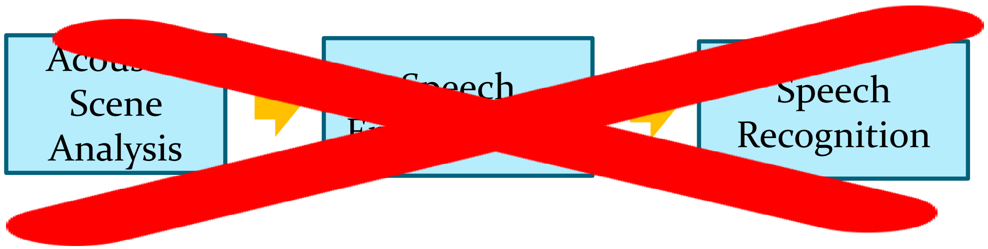
DSR is very challenging due to ***noise*** *$n(t)$ and* ***reverberation*** *$h(t)$:*

$$y(t) = x(t) * h(t) + n(t)$$

M. Ravanelli, "*Deep Learning for Distant Speech Recognition*", PhD Thesis, Dec. 2017 arXiv.org

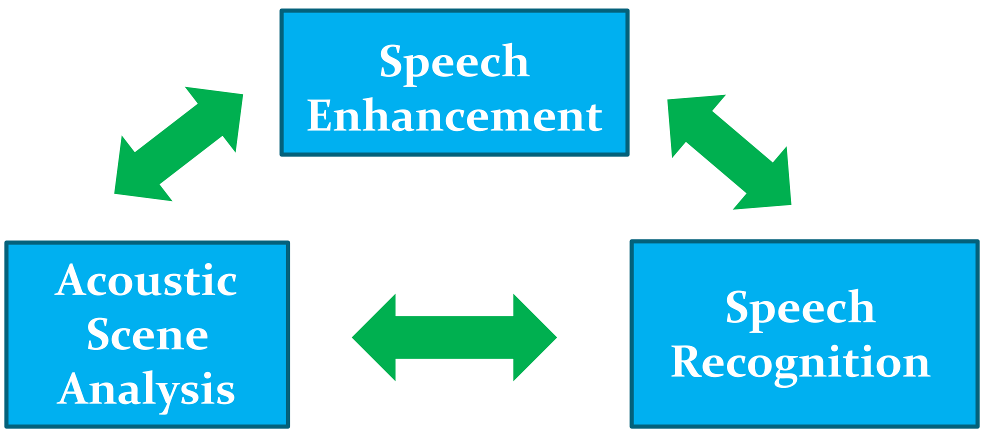
Cooperative Networks of DNNs

Breaking the pipeline



Lack of matching

Lack of communication



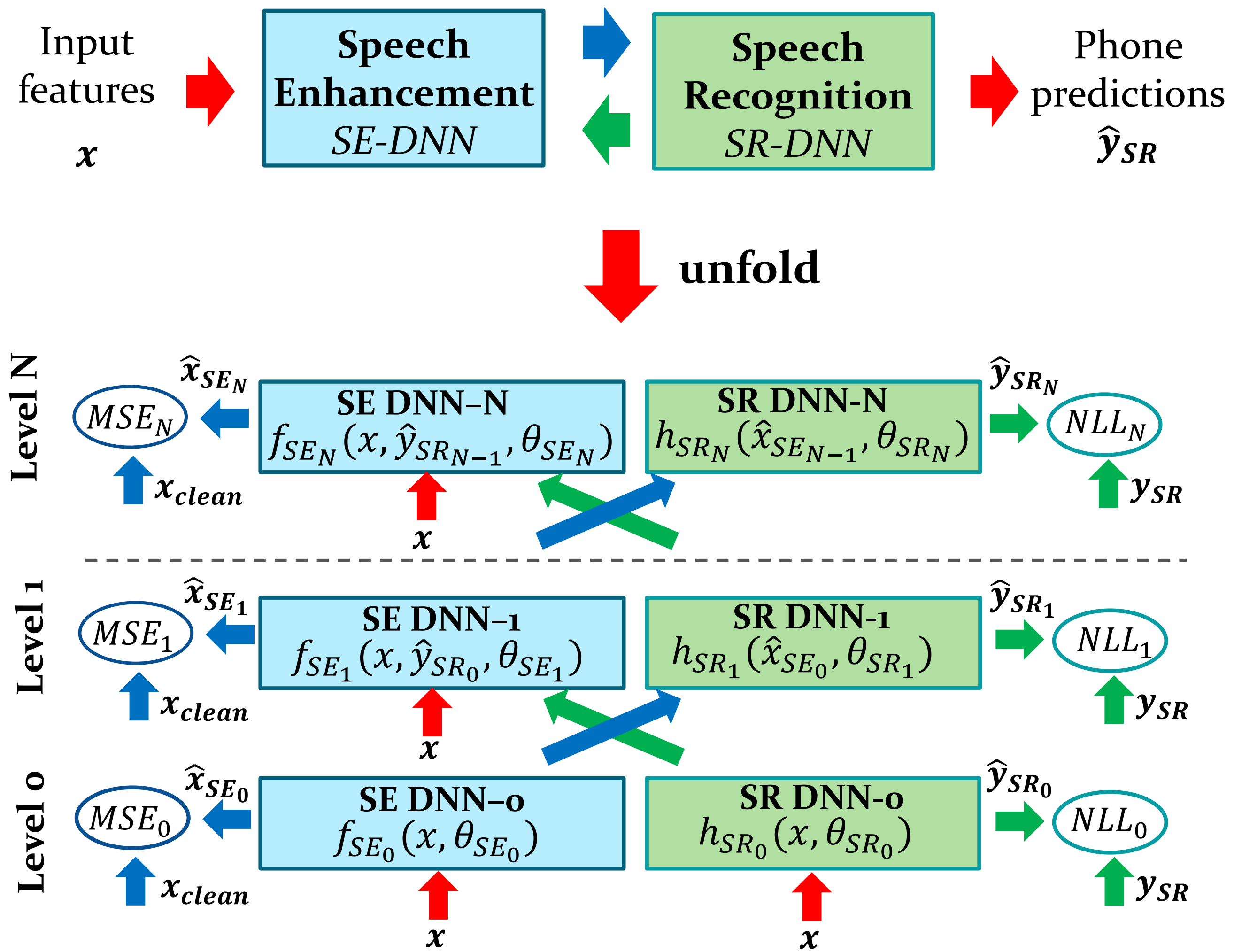
Improved matching

Full-communication

M. Ravanelli, P. Brakel, M. Omologo, Y. Bengio, "*A network of deep neural networks for distant speech recognition*", in Proceedings of ICASSP 2017 (*best IBM student paper award*).

Cooperative Networks of DNNs

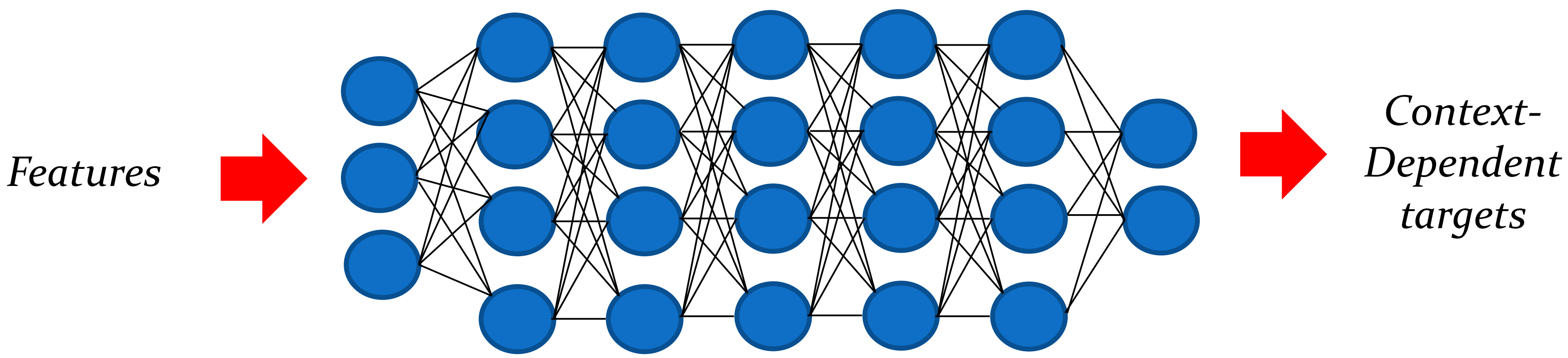
How we train it?



Network of DNNs

ASR results

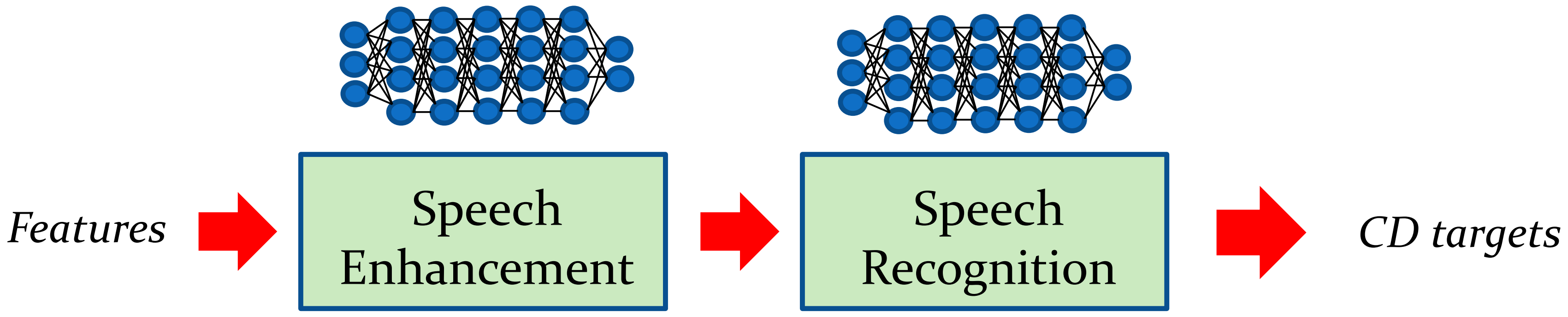
Systems	TIMIT Rev	DIRHA WSJ Rev	DIRHA WSJ Rev+Noise
Single DNN	**31.9**	**8.1**	**14.3**
Joint SE-SR training			
Network of DNNs			



Network of DNNs

ASR results

Systems	TIMIT Rev	DIRHA WSJ Rev	DIRHA WSJ Rev+Noise
Single DNN	31.9	8.1	14.3
Joint SE-SR training	**29.1**	**7.8**	**12.7**
Network of DNN			

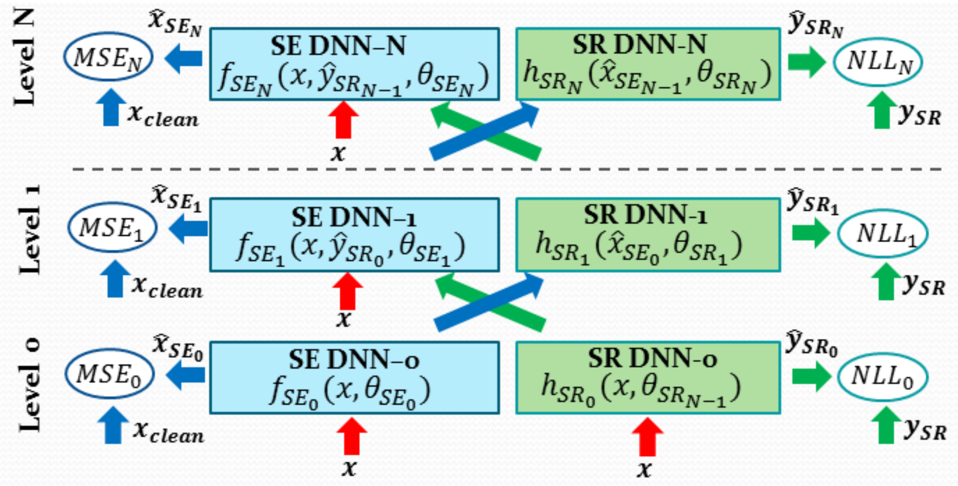


M. Ravanelli, P. Brakel, M. Omologo, Y. Bengio, "*Batch-normalized joint training for DNN-based distant speech recognition*", in Proceedings of STL 2016.

Network of DNNs

ASR results

Systems	TIMIT Rev	DIRHA WSJ Rev	DIRHA WSJ Rev+Noise
Single DNN	31.9	8.1	14.3
Joint SE-SR training	29.1	7.8	12.7
Network of DNNs	**28.7**	**7.6**	**12.3**



Conclusion

- Networks of DNNs **can counteract noise with cooperation**.
- To better exploit this paradigm further studies are needed in the future.
- We can apply this paradigm to many other fields!

M. Ravanelli, P. Brakel, M. Omologo, Y. Bengio, "*A network of deep neural networks for distant speech recognition*", in Proceedings of ICASSP 2017 (*best IBM student paper award*).

The PyTorch-Kaldi Speech Recognition Toolkit

Mirco Ravanelli

What is PyTorch-Kaldi?

- PyTorch-Kaldi is an **open-source** toolkit for developing state-of-the-art DNN/HMM speech recognition systems.

- The PyTorch-Kaldi project aims to **bridge the gap** between the Kaldi and the PyTorch toolkits.

- It inherits the **efficiency** of Kaldi and the **flexibility** of PyTorch.

- The toolkit is released under a **Creative Commons Attribution 4.0 International license**

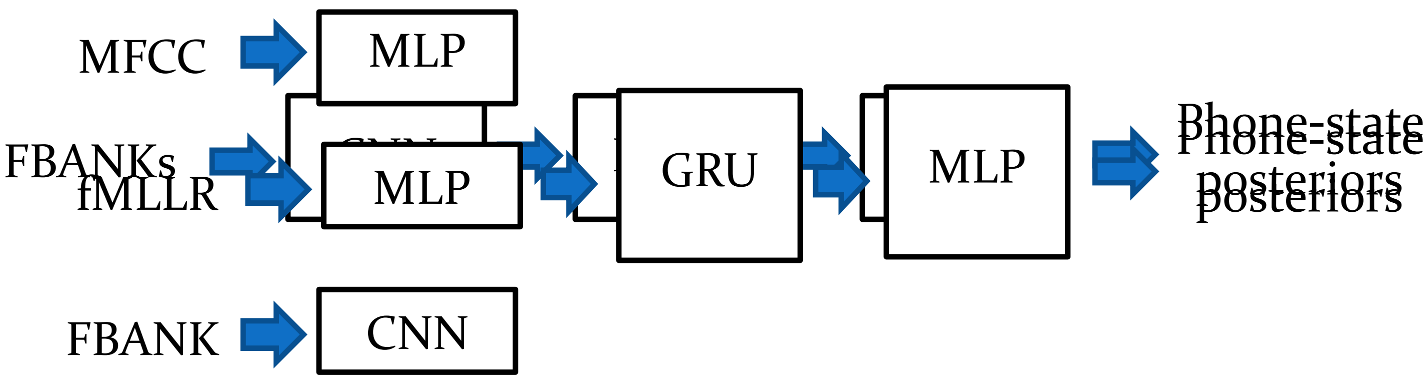
GitHub https://github.com/mravanelli/pytorch-kaldi

arXiv.org M. Ravanelli, T. Parcollet, Y. Bengio, "The PyTorch Kaldi Speech Recognition Toolkit", 2018

It’s more than a simple interface…

PyTorch-Kaldi is not only a simple interface between these toolkits, but it embeds several useful features and utilities for developing modern speech recognizers:

- Several **pre-implemented models** (MLP, CNN, LSTM, GRU, Li-GRU, SincNet).
- Easy and **flexible configuration files**.
- Natural implementation of **complex models** based on multiple features, labels, and neural architectures.



GitHub https://github.com/mravanelli/pytorch-kaldi

arXiv.org M. Ravanelli, T. Parcollet, Y. Bengio, “The PyTorch Kaldi Speech Recognition Toolkit", 2018

It's more than a simple interface…

- Easy plug-in of **user-defined models**.

```
class my_NN(nn.Module):
    def __init__(self, options):
        super(my_NN, self).__init__()
        # Definition of Model Parameters
        # Parameter Initialization

    def forward(self, minibatch):
        # Definition of Model Computations
        return [output_prob]
```

PyTorch-Kaldi

GitHub https://github.com/mravanelli/pytorch-kaldi

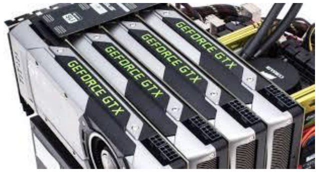
arXiv.org M. Ravanelli, T. Parcollet, Y. Bengio, "The PyTorch Kaldi Speech Recognition Toolkit", 2018

It's more than a simple interface...

- Designed to work locally or on **HPC clusters**.

- **Automatic recovery** from the last processed chunk.

- **Multi-GPU** training.



- Easy **hyperparameter tuning**.

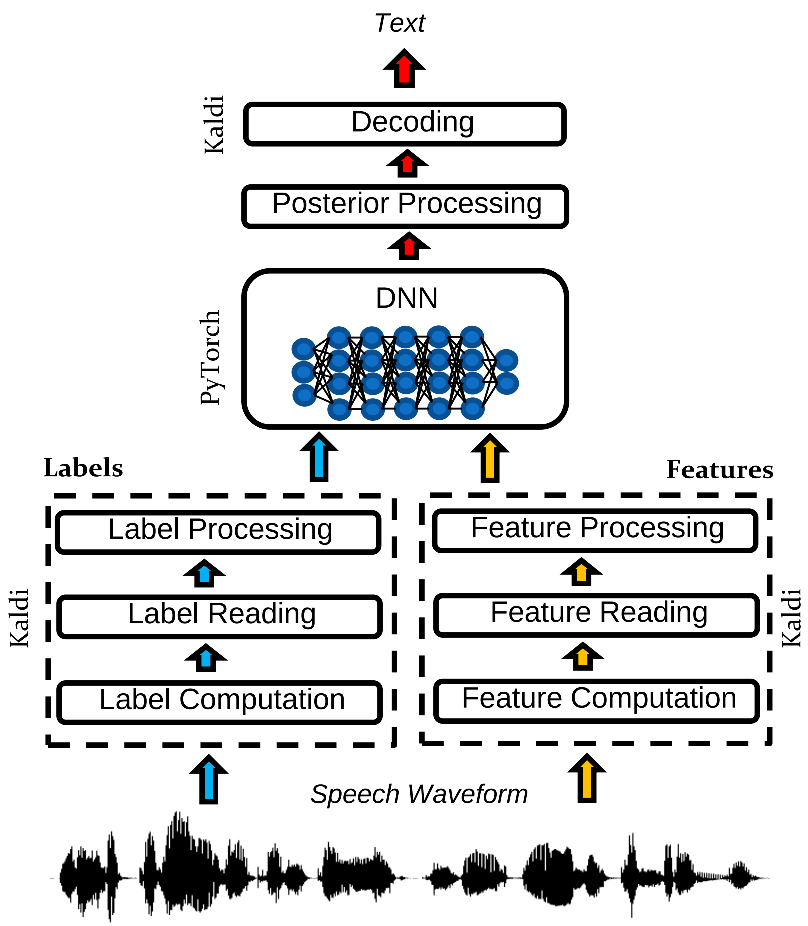
- **Rich Documentation** with tutorials

PyTorch-Kaldi

GitHub https://github.com/mravanelli/pytorch-kaldi

arXiv.org M. Ravanelli, T. Parcollet, Y. Bengio, "The PyTorch Kaldi Speech Recognition Toolkit", 2018

PyTorch-Kaldi Architecture



GitHub https://github.com/mravanelli/pytorch-kaldi

arXiv.org M. Ravanelli, T. Parcollet, Y. Bengio, “The PyTorch Kaldi Speech Recognition Toolkit", 2018

Baselines

Table 1: PER(%) obtained for the test set of TIMIT with various neural architectures.

	MFCC	FBANK	fMLLR
MLP	18.2	18.7	16.7
RNN	17.7	17.2	15.9
LSTM	15.1	14.3	14.5
GRU	16.0	15.2	14.9
Li-GRU	15.3	14.6	**14.2**

Table 2: PER(%) obtained on TIMIT when progressively applying some techniques implemented within PyTorch-Kaldi.

	RNN	LSTM	GRU	Li-GRU
Baseline	16.5	16.0	16.6	16.3
+ Incr. Seq. length	16.6	15.3	16.1	15.4
+ Recurrent Dropout	16.4	15.1	15.4	14.5
+ Batch Normalization	16.0	14.8	15.3	14.4
+ Monophone Reg.	15.9	14.5	14.9	**14.2**

M. Ravanelli, P. Brakel, M. Omologo, Y. Bengio, "Light Gated Recurrent Units for Speech Recognition", in IEEE Transactions on Emerging Topics in Computational Intelligence, 2018

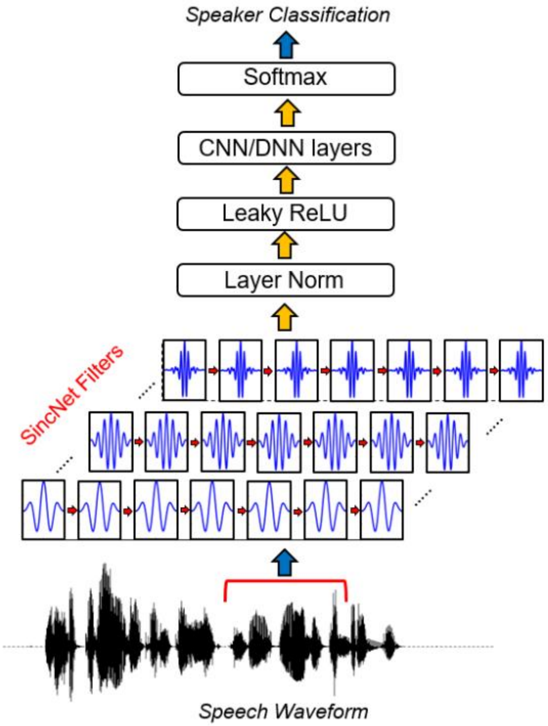
Baselines

Table 3: PER(%) obtained by combining multiple neural networks and acoustic features.

Architecture	Features	PER (%)
Li-GRU	fMLLR	14.2
MLP+Li-GRU+MLP	MFCC+FBANK+fMLLR	**13.8**

Table 4: PER(%) obtained with standard convolutional and with the SincNet architectures.

Model	Features	PER (%)
CNN	FBANK	18.3
CNN	Raw waveform	18.3
SincNet	Raw waveform	**18.1**



M. Ravanelli, Y. Bengio, “Interpretable convolutional filters with SincNet", in Proc. of NIPS@IRASL 2018

M. Ravanelli, Y. Bengio, "Speaker recognition from raw waveform with SincNet", in Proc. of SLT 2018

Baselines

Table 5: WER(%) obtained for the DIRHA, CHiME, and LibriSpeech (100h) datasets with various neural architectures.

	DIRHA	CHiME	LibriSpeech
MLP	26.1	18.7	6.5
LSTM	24.8	15.5	6.4
GRU	24.8	15.2	6.3
Li-GRU	**23.9**	**14.6**	**6.2**

Conclusion and Future Work

- PyTorch-Kaldi is a novel toolkit to design **state-of-the-art ASR** systems.
- The project is still in its initial phase and we invite all potential contributors to **participate in it**.
- We hope to build a community of developers larger enough to progressively **maintain**, **improve**, and **expand** the functionalities of our current toolkit.