Fast and Efficient Speech Enhancement with Variational Autoencoders

Mostafa Sadeghi, Romain Serizel

Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France

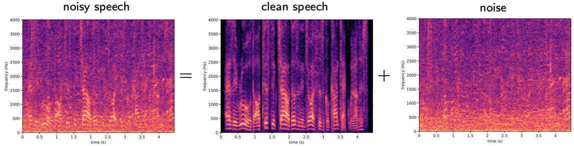
2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2023)

June 04-10, Rhodes island, Greece.

Introduction

Speech Enhancement

Short-time Fourier transform (STFT) representation:



$\mathbf{x}$ $\mathbf{s}$ $\mathbf{n}$

- $\mathbf{x} = \{\mathbf{x}_t\}_{t=1}^{T}$ (similarly for $\mathbf{s}$ and $\mathbf{n}$).

Given noisy speech observation $\mathbf{x} = \mathbf{s} + \mathbf{n}$, estimate the clean speech signal, $\mathbf{s}$.

SE approaches

▷ **Supervised (discriminative):** Model $p_\Theta(\mathbf{s}|\mathbf{x})$, and learn Θ

- Train a deep neural network (DNN) on pairs of noisy-clean data $\{\mathbf{x}_i, \mathbf{s}_i\}$.
- *Implicit* prior modeling $p_\theta(\mathbf{s})$ via inductive biases (architecture, optimizer, etc.).

▷ **Unsupervised (generative):** Speech enhancement without training on noise.

Model $p_\Theta(\mathbf{s}|\mathbf{x}) \propto \underbrace{p_\psi(\mathbf{x}|\mathbf{s})}_{\text{Inference}} \cdot \underbrace{p_\theta(\mathbf{s})}_{\text{Training}}$, and learn $\Theta = \theta \cup \psi$:

- Training - Learn speech's prior distribution $p_\theta(\mathbf{s})$
- Inference - Model $p_\psi(\mathbf{x}|\mathbf{s})$, and infer $\mathbf{s}$ using $p_\theta(\mathbf{s})$

☞ We focus on *unsupervised SE* approaches.

Speech generative modeling

How to learn speech's prior distribution?

- Latent variable generative models: Variational autoencoder (VAE),[1] Normalizing Flow (NF),[2] etc.

- Score-based generative models:[3] Learn the score function $\nabla_{\mathbf{s}} \log p_\theta(\mathbf{s})$.

We focus on VAE:

$$p_\theta(\mathbf{s}) = \int p_\theta(\mathbf{s}|\mathbf{z}) p_\theta(\mathbf{z}) d\mathbf{z}$$

- $\mathbf{z} = \{\mathbf{z}_t \in \mathbb{R}^L\}$: (real-valued, low-dimensional, $L \ll F$) latent variables

[1] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," ICLR, 2014.

[2] D. Rezende, D. Mohamed, "Variational inference with normalizing flows," ICML, 2015.

[3] Y. Song, S. Ermon, "Generative modeling by estimating gradients of the data distribution," NeurIPS, 2019

VAE-based speech modeling

A Gaussian generative model ($\forall t$):[4]

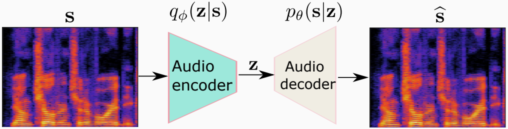
$$\begin{cases} p_\theta(\boldsymbol{s}_t|\boldsymbol{z}_t) = \mathcal{N}_c\Big(\mathbf{0}, \mathrm{diag}(\boldsymbol{\sigma}_\theta(\boldsymbol{z}_t))\Big), \\ p_\theta(\boldsymbol{z}_t) = \mathcal{N}(\mathbf{0}, \mathbf{I}) \end{cases}$$

$\boldsymbol{\sigma}_\theta(.)$: a neural network (*decoder*)

Need to maximize $\log p_\theta(\mathbf{s})$, which is intractable. However:

$$\log p_\theta(\mathbf{s}) = \log \int p_\theta(\mathbf{s}|\mathbf{z}) p_\theta(\mathbf{z}) d\mathbf{z} \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{s})}\left[\log \frac{p_\theta(\mathbf{s}|\mathbf{z}) p_\theta(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{s})}\right] \triangleq \mathcal{L}(\theta, \phi)$$

▷ Reparametrization trick + Adam optimizer[5]



[4] S. Leglaive *et al.* "A variance modeling framework based on variational autoencoders for speech enhancement," MLSP, 2018.

[5] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," ICLR, 2014.

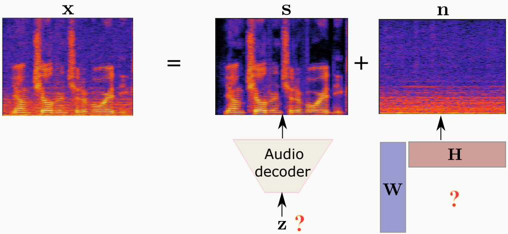
Speech Enhancement

Observation model: $\forall t: \quad \boxed{\boldsymbol{x}_t = \boldsymbol{s}_t + \boldsymbol{n}_t}$

Noise model: Non-negative Matrix Factorization (NMF)

$$\forall t: \quad \boldsymbol{n}_t \sim \mathcal{N}_c\left(\mathbf{0}, \mathrm{diag}(\mathbf{WH}[:,t])\right), \quad \mathbf{W}, \mathbf{H} \geq 0$$

Clean speech model: Trained generative (decoder) network.



☞ Need to estimate the parameters $\psi = \{\mathbf{W}, \mathbf{H}\}$.

Parameter estimation

Expectation-Maximization (EM):

$$\psi^* = \underset{\psi}{\operatorname{argmax}} \sum_{t=1}^{T} \mathbb{E}_{p_\phi(\boldsymbol{z}_t|\boldsymbol{x}_t)}\{\log p_\psi(\boldsymbol{x}_t, \boldsymbol{z}_t)\} = \underset{\psi}{\operatorname{argmax}} \sum_{t=1}^{T} \mathbb{E}_{p_\psi(\boldsymbol{z}_t|\boldsymbol{x}_t)}\{\log p_\psi(\boldsymbol{x}_t|\boldsymbol{z}_t)\}$$

⚠ Intractable expectation (E-step). Approximate solutions:

- Monte Carlo EM (MCEM)[6]: Sample from $p_\psi(\mathbf{z}|\mathbf{x}) \propto p_\psi(\mathbf{x}|\mathbf{z})p(\mathbf{z})$.
- Variational EM (VEM)[7]: Fine-tune the learned encoder, $q_\psi(\mathbf{z}|\mathbf{s})$, on $\mathbf{x}$:

$$p_\psi(\mathbf{z}|\mathbf{x}) \approx q_\psi(\mathbf{z}|\mathbf{x})$$

- Point Estimate EM (PEEM)[7]: Find only the mode of $p_\psi(\mathbf{z}|\mathbf{x})$.

Once ψ is learned, $\hat{\mathbf{s}} = \mathbb{E}_{p_{\psi^*}(\mathbf{s}|\mathbf{x})}\{\mathbf{s}\}$.

---

[6] S. Leglaive *et al.* "A variance modeling framework based on variational autoencoders for speech enhancement," MLSP, 2018.

[7] S. Leglaive *et al.* "A recurrent variational autoencoder for speech enhancement," ICASSP, 2020.

Langevin Dynamics Expectation Maximization (LDEM)

Motivation

Lack of a good trade-off between complexity & performance:

Method	Complexity	Performance
MCEM	High	High
VEM	High	High
PEEM	Low	Low

- We develop **Langevin dynamics EM (LDEM)**.
- Sampling by taking gradient steps on log posterior + noise injection.
- Total variation (TV) regularization on the latent vectors.
- LDEM makes an *effective compromise* between complexity & performance.

LDEM: Basic version

Stochastic Gradient LD:[8]

Sample from $p_\phi(\boldsymbol{z}_t|\boldsymbol{x}_t)$ using the *score function*:

$$f(\boldsymbol{z}_t) = \nabla_{\boldsymbol{z}_t} \log p_\phi(\boldsymbol{z}_t|\boldsymbol{x}_t) = \nabla_{\boldsymbol{z}_t}\Big(\log p_\phi(\boldsymbol{x}_t|\boldsymbol{z}_t) + \log p(\boldsymbol{z}_t)\Big)$$

Given an initial state $\boldsymbol{z}_t^{(0)}$, the next states (samples) are $(k \geq 0)$:

$$\boxed{\boldsymbol{z}_t^{(k)} = \boldsymbol{z}_t^{(k-1)} + \frac{\eta}{2} f(\boldsymbol{z}_t^{(k-1)}) + \sqrt{\eta}\zeta} \quad \zeta \sim \mathcal{N}(0, \boldsymbol{I}),\ \eta > 0,$$

- Noise injection for better exploration of high-density regions.
- When $k \to \infty$ and $\eta \to 0$, then $\boldsymbol{z}_t^{(k)} \sim p_\phi(\boldsymbol{z}_t|\boldsymbol{x}_t)$.

[8] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," ICML, 2011.

LDEM: Extended version

▷ Starting from $\boldsymbol{z} = \{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_T\}$, draw m different states $\boldsymbol{z}_{t,1}, \cdots, \boldsymbol{z}_{t,m}$ per $\boldsymbol{z}_t$:

$$\boldsymbol{z}_{t,i} | \boldsymbol{z}_t \sim \mathcal{N}(\boldsymbol{z}_t, \sigma^2 \boldsymbol{I}), \quad \forall t, i$$

or

$$\boldsymbol{z}_{t,i} = \boldsymbol{z}_t + \sigma \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$$

☞ A random walk as a second source of stochasticity.

▷ A *TV regularization* to incorporate time dependencies of $\{\boldsymbol{z}_1, \cdots \boldsymbol{z}_T\}$:

$$f_\lambda(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \Big(\sum_{t=1}^{T} g(\boldsymbol{z}_t) + \lambda \sum_{t=2}^{T} \|\boldsymbol{z}_t - \boldsymbol{z}_{t-1}\|_1 \Big)$$

LDEM: Extended version

Algorithm 1 LD

Require: $\bar{\mathbf{z}}^{(0)} = \left\{\mathbf{z}_{t,i}^{(0)}\right\}_{t,i}$, K (sampling steps), η (step-size).
for $k = 1, \cdots, K$ **do**
  $\boldsymbol{\zeta} = \left\{\boldsymbol{\zeta}_{t,i}\right\}_{t,i}$, with $\boldsymbol{\zeta}_{t,i} \sim \mathcal{N}(0, \mathbf{I})$
  $\bar{\mathbf{z}}^{(k)} = \bar{\mathbf{z}}^{(k-1)} + \frac{\eta}{2} f_\lambda(\bar{\mathbf{z}}^{(k-1)}) + \sqrt{\eta}\boldsymbol{\zeta}$,
end for
Output: $\bar{\mathbf{z}}^{(K)} = \left\{\mathbf{z}_{t,i}^{(K)}\right\}_{t,i}$

Algorithm 2 LDEM

Require: $\mathbf{x} = \{\mathbf{x}_t\}_{t=1}^{\tilde{T}}$, f_λ, σ, K, η, m, J (EM iterations).
Initialize: $\mathbf{z}$, $\mathbf{W}$, $\mathbf{H}$.
for $j = 1, \cdots, J$ **do**
  $\mathbf{z}_{t,i} = \mathbf{z}_t + \sigma\boldsymbol{\epsilon}_{t,i}$, with $\boldsymbol{\epsilon}_{t,i} \sim \mathcal{N}(0, \mathbf{I})$, $\forall t, i$
  $\{\mathbf{z}_{t,i}\}_{t,i} \leftarrow \mathrm{LD}(\{\mathbf{z}_{t,i}\}_{t,i})$
  $\phi \leftarrow \mathrm{argmax}_\phi \; \sum_{t,i} \log p_\phi(\mathbf{x}_t | \mathbf{z}_{t,i})$
end for
Output: $\phi = \{\mathbf{W}, \mathbf{H}\}$

Experiments

Setup

- NTCD-TIMIT dataset[9]: Extended version of TCD-TIMIT to noisy speech data.
  - **Training set** ($\sim$ 5 hours): 39 speakers $\times$ 98 sentences $\times$ 5 seconds
  - **Test set** ($\sim$ 1 hour): 9 speakers $\times$ 98 sentences $\times$ 5 seconds
  - **Noise levels**: -10 dB, -5 dB, 0 dB, and 5 dB
  - **Noise types**: *Living Room (LR)*, *White*, *Cafe*, *Car*, *Babble*, and *Street*
- STFT: 1024 sample-long (64 ms) sine window, 75% overlap, no zero-padding
- Baselines: MCEM, PEEM, ~~VEM~~ (public implementation did not work)

[9] A.-H. Abdelaziz, "NTCD-TIMIT: A new database and baseline for noise-robust audio-visual speech recognition," INTERSPEECH, 2017.

Setup

▷ VAE architecture:

- Single-layer, 128-node, encoder and decoder,
- Hyperbolic tangent activation function,
- Latent space's dimension: $L = 32$.

▷ EM parameters:

- Number of EM iterations: $J = 100$,
- Number of E-step iterations: $K = 10$,
- Learning rate for PEEM & LDEM: $\eta = 0.005$,
- Noise variance for LDEM: $\sigma^2 = 0.01$.

Results

Objective measures (the higher, the better)

- Signal-to-distortion ratio (SDR).
- Perceptual evaluation of speech quality (PESQ).
- Short-time objective intelligibility (STOI).

Metric	SI-SDR (dB)					PESQ					STOI				
Noise SNR (dB)	-10	-5	0	5	10	-10	-5	0	5	10	-10	-5	0	5	10
Input (unprocessed)	-18.08	-12.80	-7.72	-2.91	2.04	1.40	1.51	1.76	2.05	2.37	0.12	0.20	0.30	0.43	0.56
PEEM	-9.66	-4.35	0.57	5.49	10.33	1.60	1.80	2.06	2.36	2.67	0.15	0.24	0.36	0.49	0.63
MCEM	-7.67	-1.48	3.34	7.81	12.00	1.55	1.84	2.18	2.49	2.78	**0.17**	**0.27**	**0.40**	**0.54**	**0.66**
LDEM ($\lambda : 0, m : 1$)	**-7.20**	**-1.03**	**3.76**	**8.18**	**12.37**	1.54	1.85	2.18	2.50	2.78	0.16	0.25	0.38	0.52	0.65
LDEM ($\lambda : 0.5, m : 1$)	-7.17	-1.08	3.70	8.16	12.34	1.58	1.87	2.20	2.51	2.80	0.17	0.27	0.40	0.53	0.66
LDEM ($\lambda : 5, m : 1$)	-7.28	-1.41	3.42	7.93	12.13	1.70	1.96	2.25	2.56	2.83	0.17	0.27	0.40	0.53	0.66
LDEM ($\lambda : 5, m : 5$)	-7.10	-1.26	3.60	8.07	12.27	**1.73**	**2.01**	**2.30**	**2.59**	**2.85**	**0.17**	**0.27**	**0.40**	**0.54**	**0.67**

Average runtimes (in seconds) per test sample ($\sim$ 5-second long)

Method	PEEM	MCEM	LDEM ($m : 1$)	LDEM ($m : 5$)
runtime	5	32	5.4	18

Conclusions

- We addressed the computationally demanding EM step of VAE-based speech enhancement.
- Existing methods suffer from high computational complexity or low performance.
- We proposed Langevin dynamics EM (LDEM) as sampling-based method that effectively compromises the complexity and quality.
- Experimental results showed that LDEM outperforms previous sampling-based approaches.

Thank you for your attention!