ADVERSARIAL SIGNAL DENOISING WITH ENCODER-DECODER NETWORKS

Leslie Casas, Nassir Navab

CAMP, Technische Universität München
Garching bei München,Germany

Vasileios Belagiannis

MRM, Universität Ulm
Ulm, Germany

arXiv:1812.08555v1 [cs.LG] 20 Dec 2018

ABSTRACT

In this work, we treat the task of signal denoising as distribution alignment between the clean and noisy signal. An adversarial encoder-decoder network is proposed for denoising signals, represented by a sequence of measurements. We rely on the signal's latent representation, given by the encoder, to detect clean and noisy samples. Aligning the two signal distributions results in removing the noise. Unlike the standard GAN training, we propose a new formulation that suits to one-dimensional signal denoising. In the evaluation, we show better performance than the related approaches, such as autoencoders, wavenet denoiser and recurrent neural networks, demonstrating the benefits of our approach in different signal and noise types.

Index Terms— signal denoising, adversarial learning

1. INTRODUCTION

In signal processing, the presence of noise is a common problem independent from the signal type. One way to recover the signal is to use neural networks for denoising. This approach has been particularly popular in the image domain, where learning-based models, such as denoising autoencoders [1], have advanced the field. Similarly in audio and speech processing the recent advances of deep neural networks (DNNs) have resulted in promising results [2, 3, 4]. On the other hand, the influence of learning-based methods on one-dimensional signals, such as motion signals, is rather limited.

In this work, we introduce the idea of adversarial learning for one-dimensional signal denoising. We present an adversarial encoder-decoder network architecture to denoise signals that are represented by a sequence of measurements. In our approach, a discriminator network classifies the signal into noisy or clean given the signal latent representation input. Aligning the noisy and clean signal distributions is equivalent to removing the noise. Unlike the standard GAN training [5] and adversarial autoencoders [6], we propose a new formulation that suits to one-dimensional signal denoising.

Our network architecture builds on the advances from the image domain. First, we adopt the structure of a fully convolutional network (FCN) [7] for one-dimensional data. We design the encoder to denoise the input and transform it to the latent representation. On the other hand, the decoder reconstructs the clean signal from the latent representation. To facilitate the reconstruction, we introduce residual learning with shortcut connections from the encoder to the decoder. Moreover, we use dilated convolutions for the encoder and dilated deconvolutions for the decoder. The reason is to increase the effective receptive fields of the network, compared to standard convolutions. While these operations are well-established in the image domain, they have not been yet sufficiently explored for one-dimensional signal processing.

In the evaluation, we demonstrate that adversarial learning improves the encoder-decoder architectures on different types of one-dimensional signal. We perform denoising of motion and electrocardiogram (ECG) signals. These particular signals are chosen due to the complex types of noise. We compare our results with standard signal processing, such as wavelets, and learning-based algorithms. We evaluate neural network approaches, including autoencoders, wavenet denoiser and recurrent neural networks. Notably, our adversarial encoder-decoder outperforms the related approaches in all cases.

In summary, our work makes the following contributions: (i) an adversarial encoder-decoder network for one-dimensional signal denoising , (ii) an architecture that generalizes to different signal and noise types and (iii) better performance than related approaches.

2. RELATED WORK

Learning-based signal denoising has been established with the autoencoders [1], mainly applied on image data. Autoencoders have been the motivation to explore DNNs with complex architectures [8, 9]. On the same direction, image denoising has been addressed with multi-layer perceptron [10] and ConvNets [11]. Although our approach is related to autoencoders, it is closer to the encoder - decoder architectures for image recognition [7]. An encoder-encoder network with skip [12] connections has been successfully used for image denoising too. Unlike, our focus is on signals that are represented by a sequence of measurements. Here, there is not spatial domain to shrink and then expand. Instead, we perform the same operation on the temporal domain.

Sequential modelling has been traditionally assessed with recurrent neural networks (RNNs). Recently, though, sequential tasks, such as machine translation [13] and language modeling [14], have been modelled with feed-forward networks. In WaveNet, dilated convolutions have been introduced to model the temporal domain [3]. A recent comparison between RNNs and feed-forward networks has shown that ConvNets with dilated convolutions perform as well as RNNs [15]. Our architecture's design is motivated by these findings.

Finally, generative adversarial networks [5] have revolutionized image synthesis. In denoising, adversarial learning has recently been explored [16, 6] in the image domain. In our work, it contributes to align the distribution of the noisy with clean signal, performing effectively signal denoising. In addition, we work on one-dimensional data where the challenges are different.

3. ADVERSARIAL ENCODER-DECODER

Let $\mathbf{x} \in \mathbb{R}^D$ be the corrupted version of the one-dimensional signal $\mathbf{y} \in \mathbb{R}^D$, where D is the signal length. Our goal is to estimate the clean signal $\mathbf{y}$ with the function $f : \mathbb{R}^D \rightarrow \mathbb{R}^D$ that is the composition of an encoder and decoder network, given by $f(\mathbf{x}) = \psi(\phi(\mathbf{x}))$. The encoder $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^K$ provides a denoised latent K-dimensional representation of the input. The decoder $\psi : \mathbb{R}^K \rightarrow \mathbb{R}^D$ reconstructs the denoised signal from the latent space. Training the network is done with the encoder-decoder objective, as well as, adversarial learning. For the adversarial learning, we propose to use a discriminator that detects clean and noisy signals. The network is parametrized by $\boldsymbol{\theta}$, where the parameters are learned with back-propagation and stochastic gradient descent. Next, we define the network operations, architecture and the adversarial learning.

3.1. Encoder-Decoder Network

Given a set of training data, the encoder-decoder objective that we aim to minimize is the following:

$$\mathcal{L}_{ED}(f) = \mathbb{E}_{\mathbf{x},\mathbf{y}}[f_{\boldsymbol{\theta}}(\mathbf{x}) - \mathbf{y}]^2. \quad (1)$$

The encoder and decoder parameters correspond to convolution and deconvolution operations. We model the encoder with dilated convolutions and the decoder with dilated deconvolutions (i.e. transposed convolutions). We carefully design both parts so that they are symmetric. Then, the shortcuts connect the encoder with the decoder, as illustrated by Fig. 1.

Dilated Convolutions. The dilated convolutions increase the effective receptive field of the network without increasing the number of parameters. In our model, we introduce dilated convolutions in the encoder network. Given the 1D input signal $\mathbf{x} \in \mathbb{R}^M$ and a 1D filter $\mathbf{w} \in R^r$, the dilated convolution at the position t is defined as: $\mathbf{y}[t] = \sum_{j=0}^{r-1} \mathbf{x}[t + d \cdot (j-1)] \cdot \mathbf{w}[j]$, where d is the dilation factor that we linearly increase in the encoder, while we fix the size of the filter. Furthermore, the 1D convolution kernel is centered at the t location.

Dilated Deconvolutions. Our objective is to build a symmetric decoder to the encoder, which is defined by transposed operations. We propose the dilated deconvolution to upsample the latent and feature representations. Assuming the 1D input $\mathbf{x}$ and a 1D filter $\mathbf{w}$ again, the dilated deconvolution at the location t is now defined as: $\mathbf{y}[t] = \sum_{j=0}^{r-1} \mathbf{x}[t - d \cdot j] \cdot \mathbf{w}[j]$, where $j \leq \frac{t}{d}$ and $j \geq \frac{t-|\mathbf{x}|}{d}$ in order to avoid indices out of $\mathbf{x}$ range. For a 1D input, the operation is similar to inverting the filter and applying it to the signal. The dilation factor d is also symmetric. This means that the decoder starts with larger factors that linearly decreases at every upsampling.

Residual Blocks. Residual learning [17] has been introduced for building very deep neural networks, without vanishing gradient problems. A residual block can be represented as: $\mathbf{x}_{l+1} = F(\mathbf{x}_l) + \mathbf{x}_l$, where $\mathbf{x}_l$ is the input of the l-th layer and F is the residual mapping. The mapping is usually a set of operations such as convolution, activation and batch normalization. These operations are followed by addition with a skip(shortcut) connection.

Here, we propose residual blocks to connect the encoder with the decoder. Each block includes operations from the previous one as it is illustrated in Fig. 1. We introduce a combination of convolutions and deconvolutions inside the block. The residual learning contributes to reconstructing the denoised signal from the latent space.

Network Architecture. The network architecture is formed by the encoder and decoder. The input $\mathbf{x}$ is passed through the encoder and then the decoder. The encoder is composed of a standard 1D convolution followed by three levels of dilated convolution with 3, 3 and 6 dilation factors. The decoder has a symmetric structure with a set of three levels of dilated deconvolutions with symmetric dilation factors that are 6, 3 and 3, followed by a standard 1D convolution that results in the clean signal reconstruction. The convolutions use padding to retain the data size. Furthermore, the first convolution adds 128 feature channels to the input. The same number of features propagates along the encoder-decoder network, while the last convolution decreases the number of channels to 1. Note that all convolutions and deconvolutions have kernel size 3 and are followed by non-linearity (ReLUs). Only the last convolution has a linear activation.

The shortcut connections take place after each deconvolution layer. The features of the lower layers are added to the symmetric features of the deconvolution layers. After each addition there is a convolutional layer that weights the contribution of the shortcut connection (see Fig. 1). At the end, the encoder-decoder network outputs the denoised signal with the same dimensions as the noisy input.

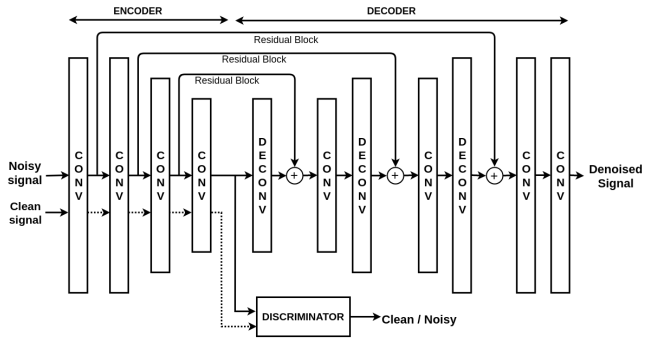


Fig. 1: Adversarial Encoder - Decoder Architecture. This is an overview of our approach. We propose the encoder with dilated convolutions and the decoder with dilated deconvolutions for sequential input. The encoder propagates features to the decoder using shortcut connections. The input to the network is noisy signal, while the output is the clean version of it. In addition, the encoder feeds the discriminator with the latent representation of the clean and noisy signals.

3.2. Adversarial Learning

Adversarial learning has been established through Generative Adversarial Networks (GANs) [5] for image generation. The idea is to build an image generation network, using a latent variable input and obtain supervision from another network, the discriminator. The discriminator's role is to distinguish generated from real images, while the overall objective is to generate images that fool the discriminator, i.e. generate images that are indistinguishable from the real ones. In this work, we modify these rules to fit in our problem.

Objective. We treat the task of denoising as distribution alignment, where the misalignment occurs because of the signal noise. The two players are the noisy and clean signal. The encoder $\phi(\cdot)$ receives the noisy sample $\mathbf{x}$ as input to produce its latent representation, therefore the role of the generator is implicitly assigned to the encoder. In addition, the encoder $\phi(\cdot)$ is used for generating the latent representation of the clean signal $\mathbf{y}$. Unlike the standard GAN problems, here the real data distribution is not given.

The discriminator $DS(\cdot)$ classifies the latent representation (i.e. noisy or clean). Finally, fooling the discriminator in adversarial learning means to align the latent representations of the two signals and that why adversarial learning performs denoising. The proposed model is shown in Fig. 1. Following the original GAN formulation from [5], we define the objective as:

$$\mathcal{L}_{GAN}(\phi, DS) = \mathbb{E}_{\mathbf{y}}\left[\log(DS(\phi(\mathbf{y})))\right] + \mathbb{E}_{\mathbf{x}}\left[\log(1 - DS(\phi(\mathbf{x}))\right]. \quad (2)$$

Note that the first term (i.e. clean data) makes use of the encoder (i.e. generator). This is the main difference from the original GAN formulation. To avoid updating the encoder with the discriminator's gradients of the clean signal, we introduce λ and re-write the objective as:

$$\mathcal{L}_{Adv}(\phi, DS, \lambda) = \lambda\mathbb{E}_{\mathbf{y}}\left[\log(DS(\phi(\mathbf{y})))\right] + \mathbb{E}_{\mathbf{x}}\left[\log(1 - DS(\phi(\mathbf{x}))\right]. \quad (3)$$

Now, the final objective that includes the adversarial and encoder-decoder terms is defined as:

$$\arg \min_{\phi, f, \lambda=0} \max_{DS, \lambda=1} \mathcal{L}_{Adv}(\phi, DS, \lambda) + \mathcal{L}_{ED}(f). \quad (4)$$

Note that the two terms could be weighted by a constant. However, we observed similar performance when balancing the two terms and thus we skip it.

Discriminator Design. The input to the discriminator is the latent representation given by the encoder. We have empirically found that a 4-layer discriminator is sufficient for our problem. This architecture is also similar to discriminators for compression [18] or domain adaptation [19]. There is first a convolution to reduce the channel dimensions to one, followed by two fully connected layers with 150 units each and ReLU activation. Finally, the signal is reduced to binary classification with the last fully connected layer and sigmoid activation.

Training. In our adversarial model, the training is joint for the encoder-decoder and the discriminator. First, the noisy and clean samples go through the encoder and discriminator. The noisy samples pass through the decoder too. During gradient update, the discriminator is updated with the gradients from the noisy and clean latent representation input. The encoder is updated with the gradients of the noisy samples, generated by labelling them as clean (as in the original GAN). Finally, the encoder and the decoder are updated from Eq. 1 gradients.

Implementation. We rely on the AdaDelta optimizer with decay rate of 5e-4. The weights of the convolutional and deconvolutional layers are initialized with Glorot uniform distribution [20] and hyper-parameters obtained by grid search. The network input for all models is raw data. Moreover, we empirically set the temporal window to 10 measurements during training, while the inference works with adaptive input.

4. EXPERIMENTS

We first rely on motion signals, where we perform denoising of the measured angular velocity. Second, we perform Electrocardiography (ECG) signal denoising. We compare with wavelets [21], a standard signal denoising method. We obtained the best parameters, after an exhaustive search, with the modified overlap wavelet transform, using Symlets 8 with 5 levels of decomposition, soft thresholding and level-dependent noise estimation. Next we focus on learning-based approaches. We implement the denoising auto-encoder (AE) [1] with three layers for encoding and another three for decoding. Note that we trained deeper AE models, but

there was not an improvement on the results. Next, an LSTM architecture [22], with two cells, is included for comparisons with recurrent neural networks. Lastly, we build a variant of a WaveNet, originally used for speech denoising [4]. The evaluation metric is the signal-to-noise ratio (SNR).

Model Components. We choose the encoder-decoder network to have a 3-layer encoder and 3-layer decoder. We tried different layer variations, but we empirically found that the 3-layer model suits well for the examined signals. We evaluate firstly the encoder-decoder network based only on Eq. 1 and without adversarial learning, similar to standard L2-loss. Secondly, we evaluate our complete model with adversarial learning based on Eq. 4. Below, we individually discuss the results for each experiment.

4.1. Motion Signal Evaluation

We select the European Robotics Challenge(EuroC) MAV dataset [23] that consists of 11 sequences of inertial measurement unit (IMU) sensors and motion capture data. Each sequence contains angular velocity and acceleration measured at 200Hz, while the 3D position and angular velocity are obtained from a motion capture system. The noise of this signal is usually Gaussian and random walk noise. We denoise the angular velocity, because it is the only one with available ground-truth. The velocity has 3 dimensions, which we treat as a sequence of measurements. We perform an 11-fold leave-one-out cross-validation. Table 1 summarizes the average results.

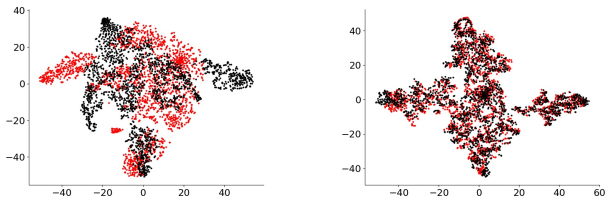
The AE performs well in increasing the SNR from 12.57dB to 23.48dB. The adversarial learning though provides the largest improvement. The performance of WaveNet is on the same level with our encoder-decoder baseline, but the LSTM is far behind. Finally, the wavelet method cannot adequately cope with unknown type of noise. We further provide a visualization of the latent space of our model, projected to two dimensions with t-SNE [24]. In detail, we used thousand clean (red) and noisy (black) samples, respectively. In Fig. 2, we show the projected samples at the beginning and at the end of training. Although, the noisy and clean data have the same range of values at the beginning, they are clearly misaligned. At the end of training, Fig. 2 shows the alignment between the two data distributions.

4.2. Electrocardiography (ECG) Evaluation

Now, we explore the generalization of our approach to denoise another type of sequential signal. We choose the Physionet ECG-ID database [25] that has 310 ECG records from 90 subjects. Each record contains the raw ECG signal and the manually filtered ground-truth version. Our sampling frequency is similar to the motion signal and thus the same network architecture is suitable for the experiment. The dataset does not have a standard evaluation protocol. For that reason, we randomly choose 10 subjects for test and use the rest for training and validation. This signal is often corrupted by power line interference, contact noise and motion artifacts. Denoising now becomes more complex because of the ECG signal is non-stationary and has overlapping spectrum with the noise. The results in Table 1 show similar behaviour to the angular velocity denoising.

Table 1: Denoising Results. We evaluate our approach on the EuroC MAV dataset [23] where ground-truth is available for the angular velocity. The Signal-to-Noise ratio (SNR) is reported. We compare with a denoising autoencoder (AE) [1], LSTM [22] and a denoising version of WaveNet [4]. We also evaluate on the Physionet ECG-ID [25]. Our baseline is the encoder-decoder network. Our adversarial encoder-decoder network has the same parameters with the baseline at inference, but more during training due to the discriminator.

	# Param.	SNR(dB)	
		Motion	ECG
Initial Noise	-	12.57	-6.72
Wavelets	-	12.79	-5.90
AE	102.855	23.48	4.67
LSTM	62.155	19.11	2.65
WaveNet Denoiser	463.747	23.33	4.45
Our Encoder-Decoder (Eq. 1)	444.161	25.21	4.24
Our Adversarial Encoder-Decoder (Eq. 4)	468.141	**32.08**	**5.30**



(a) After first epoch. (b) After last epoch.

Fig. 2: Distribution Visualization. We visualize the clean (red) and noisy (black) signal latent representations over time. We use t-SNE to project the latent representation to a two dimensional space. Although we use the same samples, the clean samples are differently distributed, because the encoder parameters keep changing over time.

5. CONCLUSION

In this work, we have proposed adversarial learning for one-dimensional signal denoising. Our adversarial encoder-decoder network generalizes to different signal and noise types. In our experiments, we have demonstrated better performance than related approaches and constant improvement using adversarial learning.

6. REFERENCES

[1] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[2] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori, "Speech enhancement based on deep denoising autoencoder.," in *Interspeech*, 2013, pp. 436–440.

[3] Aron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," in *Arxiv*, 2016.

[4] Dario Rethage, Jordi Pons, and Xavier Serra, "A wavenet for speech denoising," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5069–5073.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[6] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[7] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[8] Forest Agostinelli, Michael R Anderson, and Honglak Lee, "Adaptive multi-column deep neural networks with application to robust image denoising," in *NIPS*, 2013.

[9] Junyuan Xie, Linli Xu, and Enhong Chen, "Image denoising and inpainting with deep neural networks," in *NIPS*. 2012.

[10] Viren Jain and Sebastian Seung, "Natural image denoising with convolutional networks," in *NIPS*, 2009.

[11] Harold C Burger, Christian J Schuler, and Stefan Harmeling, "Image denoising: Can plain neural networks compete with bm3d?," in *CVPR*, 2012.

[12] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *NIPS*. 2016.

[13] Jonas Gehring, Michael Auli, David Grangier, and Yann N Dauphin, "A convolutional encoder model for neural machine translation," *arXiv preprint arXiv:1611.02344*, 2016.

[14] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," *arXiv*, 2016.

[15] S. Bai, J. Zico Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *ArXiv e-prints*, 2018.

[16] Antonia Creswell and Anil Anthony Bharath, "Denoising adversarial autoencoders," *IEEE transactions on neural networks and learning systems*, , no. 99, pp. 1–17, 2018.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *CVPR*, 2016.

[18] Vasileios Belagiannis, Azade Farshad, and Fabio Galasso, "Adversarial network compression," *arXiv preprint arXiv:1803.10750*, 2018.

[19] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, "Adversarial discriminative domain adaptation," in *CVPR*, 2017.

[20] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010.

[21] Mallat Stéphane, "Front matter," in *A Wavelet Tour of Signal Processing (Third Edition)*, pp. iii –. Academic Press, Boston, third edition edition, 2009.

[22] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[23] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, 2016.

[24] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[25] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley, "Physiobank, physiotoolkit, and physionet," *Circulation*, 2000.