2022 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-9620-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIP46576.2022.9897173

NON-SMOOTH ENERGY DISSIPATING NETWORKS

Hannah Dröge[⋆] *Thomas Möllenhoff*[†] *Michael Möller*[⋆]

[⋆] University of Siegen, Germany
[†] RIKEN Center for AI Project, Tokyo, Japan

ABSTRACT

Over the past decade, deep neural networks have been shown to perform extremely well on a variety of image reconstruction tasks. Such networks do, however, fail to provide guarantees about these predictions, making them difficult to use in safety-critical applications. Recent works addressed this problem by combining model- and learning-based approaches, e.g., by forcing networks to iteratively minimize a model-based cost function via the prediction of suitable descent directions. While previous approaches were limited to continuously differentiable cost functions, this paper discusses a way to remove the restriction of differentiability. We propose to use the Moreau-Yosida regularization of such costs to make the framework of energy dissipating networks applicable. We demonstrate our framework on two exemplary applications, i.e., safeguarding energy dissipating denoising networks to the expected distribution of the noise as well as enforcing binary constraints on bar-code deblurring networks to improve their respective performances.

Index Terms— energy minimization, neural networks, energy dissipation, image reconstruction, moreau envelope

1. INTRODUCTION

Many image processing problems, e.g. in medical imaging or the reconstruction of impaired corrupted images such as down-sampled, noisy, or blurred images, can be written as linear inverse problems, where a desired quantity $\hat{u}$ ought to be recovered from measured data f that relates to the true solution via

$$f = A\hat{u} + n, \tag{1}$$

for additive noise n and a linear mapping A. A classical way to approach such problems have been maximum a-priori (MAP) estimates which motivate the reconstruction of u as the argument that minimizes a suitable cost function

$$\tilde{u} \in \text{argmin}_u - \log p(u|f) - \log p(u), \tag{2}$$

where $p(u|f)$ refers to the conditional probability of u being the true image after measuring f and $p(u)$ is the (data-independent) prior probability of u. Such *energy minimization methods* therefore consist of a *data fidelity term*, $-\log(p(u|f))$ that depends on the distribution of the noise, and a regularizer $-\log(p(u))$.

Over the past decade, approaches like (2) have largely been outperformed and therefore replaced by deep learning based techniques that directly predict a suitable estimate $\tilde{u} = \mathcal{G}(f; \theta)$ for a (deep convolutional) neural network $\mathcal{G}$ with learnable parameters θ. Despite their performance, it is, however, difficult for such network to *guarantee* a certain *constraint* on its output. This can be a severe limitation particularly for safety-critical applications, where one at least needs to ensure that – if the distribution of the noise can be characterized as $p(u|f) \propto \exp(-d(Au, f))$ – the prediction $\tilde{u}$ respects the data up to the expected noise level $\delta = d(A\hat{u}, f)$, i.e.

$$d(A\tilde{u}, f) \leq \delta. \tag{3}$$

Previous works have addressed similar problems by training networks that are safeguarded by a suitable cost function in order to guarantee bounds such as (3), see [1, 2]. Unfortunately, the approach of Moeller *et al.* [1] is limited to the case where $d(Au, f)$ is *continuously differentiable* in u, e.g. the investigated classical case of Gaussian noise where $d(Au, f) = \|Au - f\|^2$. Yet, some distributions of high practical relevance such as the Laplace (or even more heavy-tailed) distributions cannot be tackled with their approach. Moreover, for cost functions that do not possess a Lipschitz-continuous gradient with reasonably small Lipschitz constant, the stated convergence can become very slow.

In this paper, we extend the method from [1] to apply to semiconvex and non-differentiable costs by descending on their *Moreau envelope*, a smooth approximation with identical minimizers. We discuss appropriate step size rules of the resulting descent scheme to ensure convergence, and showcase the importance of using non-smooth loss functions in two exemplary applications.

2. RELATED WORK

There is a large body of literature for solving ill-posed inverse problems with a known data formation process (1) via energy minimization methods (2) with different regularization terms including prominent examples such as the total variation [3], extensions thereof [4], wavelets [5], shearlets [6], or dictionary learning approaches [7].

An alternative to such classical techniques is to solve them by data-driven neural network training on given ground truth data and their degraded measurements. In recent decades, many (inverse) problems have been solved by training neural networks for widely varying applications [8, 9, 10, 11]. However, disadvantages of neural networks are the lack of theoretical understanding and the missing guarantee and control over the result of the network.

To simultaneously take advantage of the mathematical understanding of the behavior of classical model-based methods and data-based learned networks, hybrid methods have been developed. These include methods where an inverse problem is optimized using a learned regularizer, as in [7, 12, 13], methods that use classical optimization algorithms as network frameworks [14, 15, 16] or methods in which the structure of a neural network serves as regularization [17]. Beside this, safeguarding methods are developed to ensure the provable convergence, as [18], proving convergence of a learned optimizer by switching it with a generic learning algorithm. The works [19, 20, 21] propose to incorporate network-based update steps and to control and correct convergence behavior through some feedback mechanism. Most closely related to our approach, the work [1] trains a network $\mathcal{G}$ on iteratively predicting update directions that lie in a suitable convex set $C(\zeta_1, \zeta_2, \nabla E(u^k))$ of descent directions of a continuously differentiable cost function E at the current estimate u^k, where

$$C(\zeta_1, \zeta_2, g) = \{d | \langle d, g\rangle \geq \ \zeta_1 \|g\|^2, \|d\| \leq \zeta_2 \|g\|\}. \tag{4}$$

The intuition behind the projection onto the above set is to control the angular deviation between the network's predicted direction and the gradient direction g of the energy E. The work subsequently conducts a line-search algorithm for an update of the form

$$u^{k+1} = u^k - \tau^k \mathcal{G}(u^k, \nabla E(u^k), f; \theta) \tag{5}$$

for input data f and network parameters θ, to ensure a monotonic decrease of energy and, under some additional assumptions, a convergence to minimizers of the energy E as well as convergence rates (as common for first order descent methods). The work [2] additionally combined this approach with plug-and-play networks. Yet, both approaches fundamentally rely on the ability to differentiate the energy and obtain reasonable step sizes (e.g. via Lipschitz continuous gradients with reasonably small Lipschitz constants). In this work we tackle both of the aforementioned drawbacks e.g. for applications using regularizers and robust losses, involving non-smooth functions.

3. NON-SMOOTH ENERGY DISSIPATION

For the remainder of this work, let $E : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ be a proper, lower semi-continuous cost function that has a minimizer (e.g. by being coercive). The minimization of such costs is very well studied in the literature, particularly in the case where E is convex, with customized versions of the proximal point method [22, 23]

$$u^{k+1} = \text{prox}_{\tau E}(u^k), \tag{6}$$

with

$$\text{prox}_E(u) = \arg\min_v E(v) + \frac{1}{2}\|u - v\|^2 \tag{7}$$

being at the heart of several techniques such as the (accelerated) proximal gradient method [24], the primal-dual algorithm [25] or the alternating direction method of multipliers (ADMM) [26].

In the convex setting, the proximal point algorithm (6) can be interpreted as a conventional gradient descent method on a smoothed version of the original costs. It is well-known (see, e.g., [23]) that for the Moreau-Yosida regularization

$$E_\mu(u) = \inf_v \ E(v) + \frac{1}{2\mu}\|u - v\|^2, \tag{8}$$

one can write the proximal point algorithm (6) as gradient descent on E_μ with step size μ, i.e.

$$u^{k+1} = \text{prox}_{\mu E}(u^k) = u^k - \mu \nabla E_\mu(u^k). \tag{9}$$

This property makes an explicit gradient descent on the Moreau envelope of a convex non-smooth function E interesting for the framework of energy dissipating networks. To go beyond the fully convex case, let us assume that E is α-semi-convex, i.e., that there exists a constant α such that $E(u) + \frac{\alpha}{2}\|u\|^2$ is convex.

Proposition 1. *Let E be proper, lower semi-continuous, and α-semi-convex. For $\frac{1}{\mu} > \alpha$ the gradient of the Moreau envelope E_μ,*

$$\nabla E_\mu(u) = \frac{1}{\mu}(u - prox_{\mu E}(u)). \tag{10}$$

is L-Lipschitz continuous with a constant of at most $\frac{1}{\mu}(1 + \frac{1}{(1-\mu\alpha)^2})$.

Proof. If we denote $\tilde{v} = \text{prox}_{\mu E}(v)$, $\tilde{z} = \text{prox}_{\mu E}(z)$, and note that $\tilde{E}(u) = E(u) + \frac{\alpha}{2}\|u\|^2$ is convex, the optimality conditions yield that

$$\begin{aligned} &(\frac{1}{\mu} - \alpha)(\tilde{v} - \tilde{z}) = \frac{1}{\mu}(v - z) - (p_v - p_z), \\ &p_v \in \partial\tilde{E}(\tilde{v}),\ p_z \in \partial\tilde{E}(\tilde{z}). \end{aligned} \tag{11}$$

Multiplying by μ, taking the inner product with $\tilde{v} - \tilde{z}$, and using that $\langle p_v - p_z, \tilde{v} - \tilde{z}\rangle \geq 0$ yields

$$\begin{aligned} &(1 - \mu\alpha)\|\tilde{v} - \tilde{z}\|^2 \leq \langle v - z, \tilde{v} - \tilde{z}\rangle \\ &\leq \frac{1}{2}(1 - \mu\alpha)\|\tilde{v} - \tilde{z}\|^2 + \frac{1}{2(1 - \mu\alpha)}\|v - z\|^2 \end{aligned} \tag{12}$$

such that

$$\|\tilde{v} - \tilde{z}\|^2 \leq \frac{1}{(1 - \mu\alpha)^2}\|v - z\|^2. \tag{13}$$

As this shows that $\text{prox}_{\mu E}$ is $\frac{1}{(1-\mu\alpha)^2}$-Lipschitz continuous, the assertion follows by simple addition of Lipschitz constants. □

Therefore, we propose the following approach: Let E be a given semi-convex but possibly non-smooth cost function with which we'd like to control the behavior of a data driven (deep learning) approach.

We design an arbitrary (e.g. deep convolutional) neural network $\mathcal{G}$ of our choice, that gets as an input the current estimate u^k, the input data f and the gradient of the Moreau envelope at the current estimate $\nabla E_\mu(u^k)$ and predicts a descent direction $\mathcal{G}(u^k, \nabla E_\mu(u^k), f)$, s.t.

$$u^{k+1} = u^k - \tau\mathcal{G}(u^k, \nabla E_\mu(u^k), f) \tag{14}$$

converge to a minimizer of a non-smooth energy E.

To satisfy the descent constraints for a non-smooth energy E we use a surjective mapping onto $C(\zeta_1, \zeta_2, \nabla E_\mu(u^k))$ (see Eq. (4)) as the last layer of $\mathcal{G}$, which is given by

$$z \mapsto \Pi_{[\zeta_1, \zeta_2]}(\eta)g + \Pi_B(z - \eta g), \tag{15}$$

with Π_B being a projection onto $B = \{d | \|d\| \leq \sqrt{\zeta_2^2 - \eta^2}\|g\|\}$, with $\eta = \langle z, g\rangle / \|g\|^2$ and in this setting $g = \nabla E_\mu$, see [1].

In order to train the network on data that it could face during descent, prior to each training step, the data is transformed into a potential sample generated from the space of possible inputs as shown in Algorithm 1. For this purpose, starting from the input data u^0 (e.g. $u_i^0 = \frac{1}{n}\sum_j^n f_j$ in section 4.1 and $u^0 = f$ in section 4.2), an arbitrary number of descent steps (14) are performed to generate a sample that is potentially visited during descent with the current model. This potential sample $u^{\tilde{k}}$ for $\tilde{k} \in \{0, \dots, N\}$ is used as input for training the neural network. The network is trained by minimizing the sum of losses

$$\|N_\theta(u^{\tilde{k}}, \nabla E_\mu(u^{\tilde{k}}), f) - (\hat{u} - f)\|_2^2 \tag{16}$$

over all training examples for θ, where $\hat{u}$ represent the desired (ground truth) predictions. Please note the increased computational cost by computing new training samples, in comparison to other training based networks.

Proposition 2. *For a Moreau envelope with L-Lipschitz continuous gradient, the descent steps in (14) converge with constant step size $\tau^k < \frac{\zeta_1}{\zeta_2 L}$ for a model $\mathcal{G}(u^k, \nabla E_\mu(u^k), f)$ that satisfies*

$$\mathcal{G}(u^k, \nabla E_\mu(u^k), f) \in C(\zeta_1, \zeta_2, \nabla E_\mu(u^k))$$

```
Data: starting point $u^0$, network $\mathcal{G}$, gradient of the moreau
      envelope $\nabla E_\mu$, input data $f$, stepsize $\tau$, maximal
      number of iterations $N$
Result: $u^{\tilde{k}}$
$\tilde{k} \in \{0, ..., N\}$
for $k \in \{0, ..., \tilde{k}\}$ do
    $u^{k+1} \leftarrow u^k - \tau\mathcal{G}(u^k, \nabla E_\mu(u^k), f)$
end
```

Algorithm 1: Learned descent steps by an energy dissipating network satisfying the descent constraints for an energy E_μ.

Proof. According to Taylor's theorem it holds that

$$\begin{aligned} E_\mu(u^{k+1}) =& E_\mu(u^k) + \langle \nabla E_\mu(u^k), u^{k+1} - u^k \rangle \\ &+ \langle \nabla E_\mu(\xi) - \nabla E_\mu(u^k), u^{k+1} - u^k \rangle, \end{aligned} \tag{17}$$

for some ξ on the line segment between u^k and u^{k+1}. Using that $u^{k+1} = u^k - \tau^k d^k$ it holds that

$$\begin{aligned} &E_\mu(u^{k+1}) - E_\mu(u^k) \\ &\leq -\tau^k \langle \nabla E_\mu(u^k), d^k \rangle + \|\nabla E_\mu(\xi) - \nabla E_\mu(u^k)\| \|u^{k+1} - u^k\|, \\ &\leq -\tau^k \zeta_1 \|\nabla E_\mu(u^k)\|^2 + \tau^k L \|\xi - u^k\| \|d^k\|, \\ &\leq -\tau^k \zeta_1 \|\nabla E_\mu(u^k)\|^2 + (\tau^k)^2 L \|d^k\|^2, \\ &\leq -\tau^k \zeta_1 \|\nabla E_\mu(u^k)\|^2 + (\tau^k)^2 L \zeta_2 \|\nabla E_\mu(u^k)\|^2, \\ &= \tau^k \|\nabla E_\mu(u^k)\|^2 \cdot (-\zeta_1 + \tau^k L \zeta_2). \end{aligned}$$

Thus, for $\tau^k < \frac{\zeta_1}{L\zeta_2}$ we are descending in the energy. Second, coercivity of the energy ensures the existence of a convergent subsequence. Third, since $d^k \in C(\zeta_1, \zeta_2, \nabla E_\mu(u^k))$, it holds that $\|\nabla E_\mu(u^k)\| \leq \frac{1}{\zeta_1}\|d^k\| = \frac{1}{\zeta_1 \tau^k}\|u^{k+1} - u^k\|$. The convergence then follows from standard results in descent-based methods such as *Theorem 2.9 (Convergence to a critical point)* in [27]. □

Similar, but local convergence results can possibly be generalized beyond semi-convex functions E by using the notion of prox-regularity, e.g. following [28], but go beyond the scope of this paper.

4. APPLICATIONS

For proof of concept we implemented salt and pepper denoising to demonstrate energy dissipating networks on non-smooth energies and binary deblurring to show energy dissipating networks on non-smooth and semi-convex energies.

4.1. Salt and Pepper Denoising

For denoising images with salt and pepper noise using dissipation neural networks, an appropriate convex data fidelity term is the ℓ_1 norm of the distance to the noisy image f:

$$E(u) = \|u - f\|_1. \tag{18}$$

We construct the Moreau envelope of (18) as

$$E_\mu(u) = \sum_i e_\mu(u_i) \tag{19}$$

with

$$e_\mu(u_i) = \begin{cases} |u_i - f_i| - \frac{\mu}{2} & \text{if } |u_i - f_i| > \mu \\ \frac{1}{2\mu}(u_i - f_i)^2, & \text{otherwise,} \end{cases} \tag{20}$$

noise	$\|u-v\|_1$	$\|u-v\|_2^2$	median filter	TV
1%	**39.87/0.97**	39.07/0.97	30.25/0.87	34.24/0.97
5%	**34.99/0.95**	32.17/0.89	29.34/0.85	30.00/0.92
10%	**28.62/0.82**	16.93/0.45	26.61/0.80	27.83/0.86
25%	15.13/0.39	15.13/0.39	14.87/0.24	**24.75/0.72**

Table 1: Measured mean PSNR and SSIM value on the validation dataset of BSDS500 for energy dissipation network algorithm with an ℓ_1 and an ℓ_2 data fidelity term for $\zeta_1 = 0.05$ and $\zeta_2 = 30$, for running a median filter with kernel size 3, and for running TV denoising.

and train an energy dissipating network on noisy data with a surjective mapping to the gradient ∇E_μ in its last layer. Based on Proposition 2, the dissipating network minimizes the data fidelity term (18), but takes a path that tries to get as close as possible to the noise-free image in a data-driven way. Thus, there is a point during the minimization where the image is denoised best, and afterwards, due to convergence to the minimizer of (18), approaches the noisy image v.

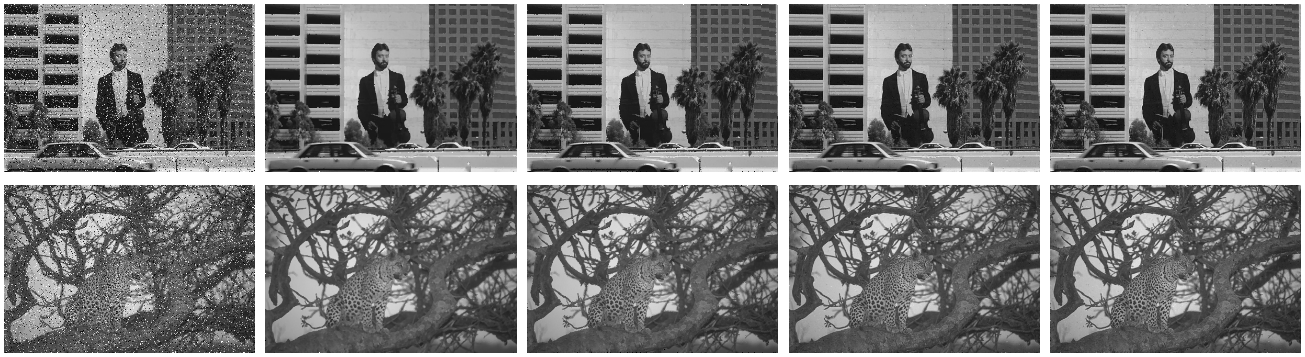
To stop minimization when denoised best, a popular a posteriori stopping rule is the discrepancy principle [29]: Similar to (3) we stop the algorithm at the minimum distance of the expected noise level $\delta = \|\hat{u} - f\|_p$ for $p = 1$ to the distance of the calculated image to the noisy image

$$\arg\min_k |\|u^k - f\|_p - \delta|. \tag{21}$$

In the following experiments, we train an energy dissipating network on salt and pepper denoising with mapping on the gradient of the Moreau envelope (19) and compare the approach with using dissipating networks on the ℓ_2 norm $E_{\ell_2}(u) = \|u - f\|^2$. For the latter, the predicted descent direction is mapped to $\nabla E_{\ell_2}(u) = 2(u^k - f)$, s.t. the update step becomes $d^k = N_\theta(u^k, \nabla E_{\ell_2}(u^k), f)$. We also compare our results to denoising using a median filter and total variation (TV) denoising by $\min_u \|Du\|_1$ s.t. $\|u - f\|_1 \leq \delta.$, with D being a finite difference matrix.

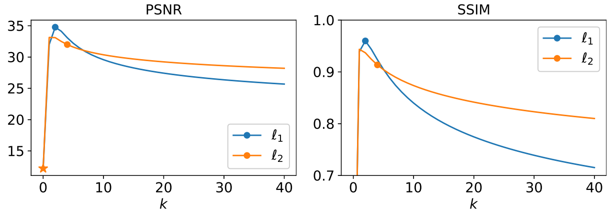
As train and validation data, we use the given images from BSDS500 [30] and apply salt and pepper noise with a probability of 5% each that a pixel takes the value 0 or 1. For training, we extract patches of dimension 52 × 52 and use them to train the network with the architecture of [8] for 30000 iterations on the loss function in (16) using Adam optimizer.

Since the step size given in Proposition 2 turns out to be too small at the beginning to efficiently minimize the energy, we choose our step size as $\tau_i = \max(\frac{1}{i+1}, \tau_{\min})$. In validation, we compare PSNR (peak signal-to-noise ratio) and SSIM (structural similarity index measure) averaged over 100 validation images for different fractions of noise, i.e., images that originated outside the potential sampling space of the training. A quantitative evaluation is given in Table 1, where the values are measured at the stopping point triggered by the criterion in (21). It shows good performance for the ℓ_1 norm as surrogate energy, better than for the ℓ_2 norm (up to 8.8% in PSNR for 5% noise), the median filter (with kernel size 3), or TV denoising, except for the highly degraded data with 25% noise, which appears to be too far outside of the range of our training examples, as in our method the network was trained on data with 5% probability of noise. Exemplary denoising results for 5% noisy data are shown in Fig. 1, with the corresponding PSNR and SSIM curve (for the upper images) over the iterations in Fig. 2, showing the peak of PSNR and SSIM at a certain point during minimization.



(a) Noisy image (b) Median filter (c) TV (d) ℓ_1 fidelity term (e) ℓ_2 fidelity term

Fig. 1: Exemplary denoising results of noisy images (a) by running median filter with kernel size 3 (b), by running TV denoising (c), by running descent on an energy dissipating network satisfying descent constraints for an ℓ_1 fidelity term (d) and an ℓ_2 fidelity term (e).



(a) PSNR over the iterations (b) SSIM over the iterations

Fig. 2: Exemplary comparison of the PSNR and SSIM for running descent on an energy dissipating network satisfying descent constraints for an ℓ_1 fidelity term and an ℓ_2 fidelity term with circles shaped markings of the iteration where the stopping criterion is triggered for $p = 1$ and a star shaped marking where the stopping criterion is triggered for $p = 2$.

4.2. Binary Deblurring

To demonstrate the concept of non-smooth and semi-convex energy dissipating networks, we consider the deblurring of binary images u with pixels that are supposed to be either zero or one $u_i \in \{0, 1\}$, e.g. having the reconstruction of bar-codes or QR-codes in mind. To ensure binary outputs, we consider the function

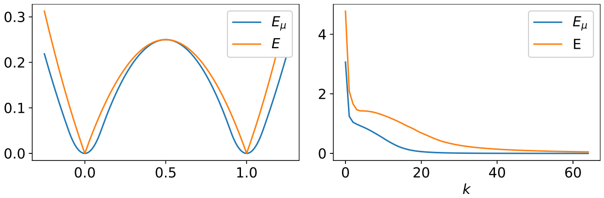
$$E(u) = \|(u - 0.5)^2 - 0.25\|_1, \tag{22}$$

and its Moreau envelope, which is illustrated in Fig. 3 (a) and train a dissipating network that satisfies the descent constraints for (22).

As data set we use generated bar-codes $b_i \in \{0, 1\}^{180}$ of type *Code 128*, by encoding randomly chosen sequences of 5 numbers and letters, and blur them using a Gaussian filter with radius 1.5. Our trainingsset consists of 40960 arrays, our testset of 1024 arrays. In our experiments we use the network architecture of [8] and decrease the network depth to 12 convolutional layers and the width to 32. We train the network using losses of the form (16) for 30000 iterations.

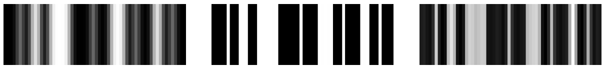
As shown in Fig. 3 (b), the network-based updates lead to a monotonically decreasing cost function, converging to zero, i.e., to binary predictions in about 60 iterations.

Fig. 4 shows an input bar code with a blur radius of 1.0 and the result of a dissipating network in comparison to an unconstrained network, with the same network architecture and trained on the same blurred data with a radius of 1.5, just without energy dissipation. Here, for 2D visualization, the arrays were repeated along the height and cropped in width. As it turns out, the unconstrained network, which has no guarantee of predicting a deblurred binary image on unknown data, fails to predict a binary image, unlike the constrained energy dissipating network.



(a) Illustration of E and E_μ (b) Energy over the iterations

Fig. 3: Illustration of Energy E (22) and its Moreau envelope E_μ for $\mu = 0.1$ (a) and their energy curve over the iterations when running descent on an energy dissipating network (b).



(a) Blurred bar-code (b) Dissipating network (c) Neural network G

Fig. 4: Deblurring results for the blurred input bar-code using a Gaussian filter with radius 1.0 (a), by running descent on an energy dissipating network (b), satisfying (22), and by applying a trained neural network (c).

5. CONCLUSION

In this paper, we discussed how to extend the framework of energy-dissipating networks to non-smooth energies by using the equivalence of the proximal point algorithm to gradient descent on the Moreau envelope. In numerical experiments, we practically applied this approach to the ℓ_1-norm for salt and pepper denoising and a non-smooth (semi-convex) function for binary deblurring, demonstrating that iterative incorporating constraints like the desire of binary outputs can improve over an unconstrained network.

6. REFERENCES

[1] M. Moeller, T. Möllenhoff, and D. Cremers, "Controlling neural networks via energy dissipation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3256–3265.

[2] H. Sommerhoff, A. Kolb, and M. Moeller, "Energy dissipation with plug-and-play priors," *NeurIPS 2019 Workshop on Solving Inverse Problems with Deep Networks*, 2019.

[3] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[4] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.

[5] S. Mallat, *A wavelet tour of signal processing*, Elsevier, 1999.

[6] G. Easley, D. Labate, and W.-Q. Lim, "Sparse directional image representations using the discrete shearlet transform," *Applied and Computational Harmonic Analysis*, vol. 25, no. 1, pp. 25–46, 2008.

[7] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[8] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[10] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[11] Y. Yang, J. Sun, H. Li, and Z. Xu, "Admm-net: A deep learning approach for compressive sensing mri," *arXiv preprint arXiv:1705.06869*, 2017.

[12] S. Roth and M. J. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 205–229, 2009.

[13] S. Hawe, M. Kleinsteuber, and K. Diepold, "Analysis operator learning and its application to image reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2138–2150, 2013.

[14] E. Kobler, T. Klatzer, K. Hammernik, and T. Pock, "Variational networks: connecting variational methods and deep learning," in *German Conference on Pattern Recognition*. Springer, 2017, pp. 281–293.

[15] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2774–2781.

[16] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1256–1272, 2016.

[17] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[18] H. Heaton, X. Chen, Z. Wang, and W. Yin, "Safeguarded learned convex optimization," *arXiv preprint arXiv:2003.01880*, 2020.

[19] R. Liu, S. Cheng, X. Liu, L. Ma, X. Fan, and Z. Luo, "A bridging framework for model optimization and deep propagation," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[20] R. Liu, Y. Zhang, S. Cheng, X. Fan, and Z. Luo, "A theoretically guaranteed deep optimization framework for robust compressive sensing mri," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 4368–4375.

[21] R. Liu, L. Ma, Y. Wang, and L. Zhang, "Learning converged propagations with deep prior ensemble for image enhancement," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1528–1543, 2018.

[22] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[23] Neal Parikh and Stephen Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[24] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[25] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[26] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[27] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods," *Mathematical Programming*, vol. 137, no. 1, pp. 91–129, 2013.

[28] P. Ochs, "Local convergence of the heavy-ball method and ipiano for non-convex optimization," *Journal of Optimization Theory and Applications*, vol. 177, no. 1, pp. 153–180, 2018.

[29] V. A. Morozov, "On the solution of functional equations by the method of regularization," in *Doklady Akademii Nauk*. Russian Academy of Sciences, 1966, vol. 167, pp. 510–512.

[30] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2010.