The Neural Tangent Link Between CNN Denoisers and Non-Local Filters

Julián Tachella, Junqi Tang and Mike Davies
School of Engineering
University of Edinburgh
{julian.tachella, j.tang, mike.davies}@ed.ac.uk

Abstract

Convolutional Neural Networks (CNNs) are now a well-established tool for solving computational imaging problems. Modern CNN-based algorithms obtain state-of-the-art performance in diverse image restoration problems. Furthermore, it has been recently shown that, despite being highly overparameterized, networks trained with a single corrupted image can still perform as well as fully trained networks. We introduce a formal link between such networks through their neural tangent kernel (NTK), and well-known non-local filtering techniques, such as non-local means or BM3D. The filtering function associated with a given network architecture can be obtained in closed form without need to train the network, being fully characterized by the random initialization of the network weights. While the NTK theory accurately predicts the filter associated with networks trained using standard gradient descent, our analysis shows that it falls short to explain the behaviour of networks trained using the popular Adam optimizer. The latter achieves a larger change of weights in hidden layers, adapting the non-local filtering function during training. We evaluate our findings via extensive image denoising experiments[1].

1. Introduction

Convolutional neural networks are now ubiquitous in deep learning solutions for computational imaging and computer vision, ranging from image restoration tasks such as denoising, deblurring, inpainting and super-resolution, to image reconstruction tasks such as computed tomography [19] and magnetic resonance imaging [18]. However, the empirical success of CNNs is in stark contrast with our theoretical understanding. Contrary to traditional sparse models [8], there is little understanding of the implicit assumptions on the set of plausible signals imposed by CNNs.

Perhaps surprisingly, Ulyanov et al. [29] discovered that training a CNN only with a single corrupted image (the one being restored) could still achieve competitive reconstructions in comparison to fully trained networks, naming this phenomenon the deep image prior (DIP). This discovery challenges traditional wisdom that networks should be trained with large amounts of data and illustrates the powerful bias of CNN architectures towards natural images. Similar ideas have also been explored in Noise2Self [4] and other variants [14]. In this setting, the number of weights (e.g., 2,000,000 for a U-Net CNN [29, 26]) is much larger than the number of pixels in the training image (e.g., 50,000 pixels of a standard 128×128 color image). The clean version of the corrupted image is obtained by early-stopping the optimization process before the network fully matches the noisy image or by considering a loss that does not allow the network to learn the corrupted image exactly [4]. These surprising results raise the following questions: how, amongst all possible optimization trajectories towards the multiple global minima of the training loss, the procedure consistently provides close to state-of-the-art reconstructions? What is the role of the optimization algorithm on the trajectory towards the global minima, and how does it affect the bias towards clean images?

Despite their surprisingly good performance, these methods provide comparable or slightly worse denoising results than classical patch-based non-local filtering techniques, such as non-local means (NLM) [5] or BM3D [7], which also only have access to the corrupted image. Moreover, training a large neural network is more computationally intensive. Subsequent questions then arise: is the neural network performing a similar filtering process? Can we avoid the slow training, and apply this filter in a more direct way? These insights are important to build a better framework in which we can optimize and design new denoisers and other low-level computer vision algorithms.

Denoising is generally considered as the fundamental building block of any image restoration problem. In many applications, CNNs are used to perform denoising steps, either in unrolled schemes [19] or in the context of plug-and-play methods [25, 31]. Hence, understanding better the bias

[1] Code available at https://gitlab.com/Tachella/neural_tangent_denoiser

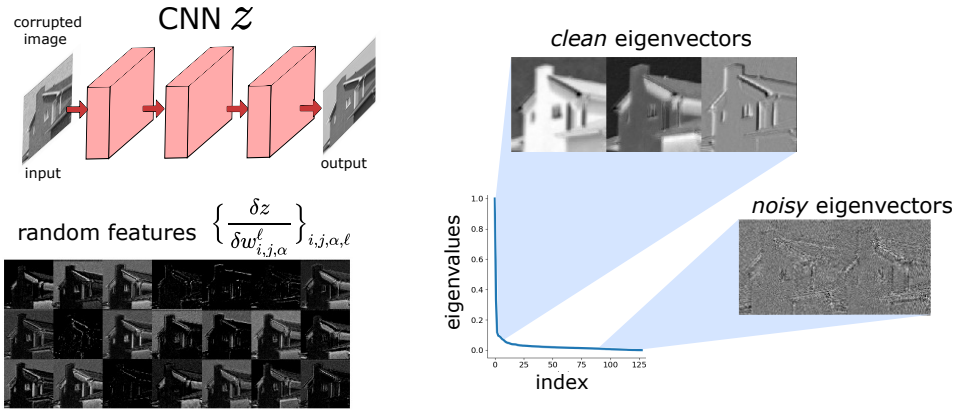

Figure 1: A convolutional neural network z trained with gradient descent on a single corrupted image can achieve powerful denoising. The left singular vectors of the Jacobian offer a representation based on patch similarities which is robust to noise.

of CNNs towards clean images is the first step towards more general imaging problems.

On another line of work, researchers have also observed that increasing the amount of overparameterization does not necessarily harm the generalization of the network [38] in the context of classification. Recently, Jacot et al. showed that overparameterized neural networks trained with (stochastic) gradient descent (GD) converge to a Gaussian process as the number of weights tends to infinity, with a kernel that depends only on the architecture and variance of the random initialization, named the neural tangent kernel (NTK) [12]. While the properties and accuracy of the kernel were analyzed for image classification [1], to the best of our knowledge, little is known in the context of high-dimensional image restoration with no clean data. Can this theory explain the good denoising performance of networks trained with a single corrupted image?

In this paper, we study overparameterized convolutional networks and their associated neural tangent kernel in the context of the image denoising, formalizing strong links with classical non-local filtering techniques, but also analyzing the short-comings of this theory to fully explain the results obtained by the DIP. The main contributions of this paper are as follows:

1. We show that GD trained CNN denoisers with a single corrupted image (placed both at the input and as a target) in the overparameterized regime equate to performing an existing iterative non-local filtering technique known as *twicing* [20], where the non-local filter is characterized by the architectural properties of the network. Moreover, these filters impose a form of low-dimensionality due to their fast eigenvalue decay, and efficient filtering can be performed directly without the CNN, using the Nyström approximation [33].

2. Departing from previous explanations [6, 11], we show that the DIP cannot be solely understood as a prior promoting low-pass images. We link this short-coming to the choice of the optimization algorithm. When trained with GD, the DIP has poor performance as predicted by the NTK theory, and maintains a fixed low-pass filter throughout training. However, training with the popular Adam optimizer as in the original DIP is able to adapt the filter with non-local information from the target image.

3. We evaluate our findings with a series of denoising experiments, showing that the fixed non-local filter associated with gradient descent performs significantly better when the corrupted image is placed at the input, whereas the Adam optimizer adapts the filter during training, providing good results for both scenarios.

2. Related Work

Neural networks as Gaussian processes: Neal [22] showed that a randomly initialized fully-connected networks converge to Gaussian process. This result was recently extended to the convolutional case [23]. Jacot et al. [12] showed that the network remains a Gaussian process throughout GD training, but with a different kernel, the NTK. Arora et al. [1] studied the kernel of a convolutional architecture for image classification, while Yang [37] extended these results to a wider set of architectures. All these works focus on classification with a set of training pairs of images and labels, whereas we study high-dimensional regression (denoising) with no clean training data.

Non-local (global) filtering: A powerful class of denoisers in image processing use patch-based filtering, e.g., NLM [5] and BM3D [7]. Milanfar studied these from a kernel function perspective [20], identifying the associated affinity (kernel) matrices, along with different iterative denoising techniques.

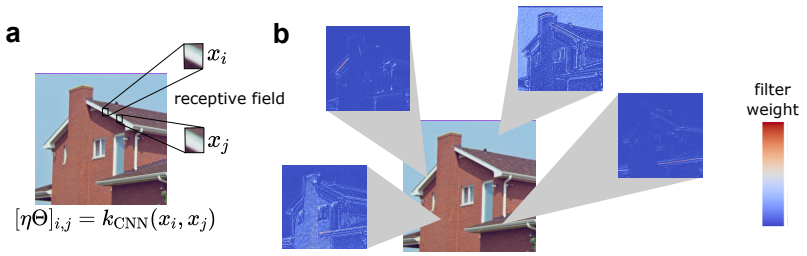

Figure 2: Non-local filter associated to the tangent kernel of a CNN with a single hidden layer. a) The filter can be obtained in closed form as the number of channels tends to infinity, where each (i, j)th entry corresponds to the similarity between the patches centered at pixels i and j. b) Filter weights for different pixels in the house image, where red/white indicates a higher weight, and blue indicates a zero weight.

CNNs as non-local filters: Recently, Mohan et al. [21] showed that a fully-trained denoising CNN without biases can be interpreted as a non-local filter by examining the input-output Jacobian of the network. They perform a local analysis of trained networks, whereas we study the global convergence during training, providing analytical expressions for the filters.

Self-supervised image denoising: In Noise2Noise [16], the authors show that training a denoising network with noisy targets can achieve similar performance to a network trained with clean targets. Noise2Void [14] and Noise2Self [4] present a self-supervised training procedure that achieves good performance, even with a single noisy image.

Deep image prior interpretations: Cheng et al. [6], analyzed the spatial (low-pass) filter associated to a U-Net CNN at initialization, following the Gaussian process interpretation of [22]. Similarly, Heckel and Soltanolkotabi [11] show that CNN decoders generate low-pass filters under GD learning, and attribute the DIP's success to this. Our work differs significantly from theirs, as we study the non-local filter behaviour of the learning process, showing that the low-pass filter behaviour does not explain DIP's state-of-the-art performance. In contrast to the spatial filters in [6, 11], the induced filters studied here can be made dependent on non-local information of the corrupted image, hence providing competitive performance to other patch-based methods.

3. Preliminaries

3.1. Convolutional neural networks

An L-layer vanilla[2] convolutional neural network with c channels at each hidden layer is defined as

$$a_i^1(x) = W_{i,1}^1 x \tag{1}$$

$$a_i^\ell(x) = \sum_{j=1}^{c} W_{i,j}^\ell \phi\left(a_j^{\ell-1}(x)\right) \tag{2}$$

$$z(x) = \sum_{j=1}^{c} W_{1,j}^L \phi\left(a_j^{L-1}(x)\right) \tag{3}$$

where $\phi : \mathbb{R}^d \mapsto \mathbb{R}^d$ is an element-wise activation function, $a_i^\ell \in \mathbb{R}^d$ are the ith channel preactivations at layer ℓ, $W_{i,j}^\ell \in \mathbb{R}^{d\times d}$ are circulant matrices associated with convolution kernels of size $r \times r$ with trainable parameters $\{w_{i,j,\alpha}^\ell : \alpha = 1, \dots, r^2\}$, the input and output are vectorized images of d pixels, denoted as $x \in \mathbb{R}^d$ and $z \in \mathbb{R}^d$ respectively. We focus on restoration with no clean training data, where only the corrupted image y is available as a training target. For the input there are 2 options:

1. Corrupted image: we place the noisy target as the input, i.e., $x = y$, i.e., Noise2Self and variants [14, 4].
2. Noise: the input is assigned with iid noise, i.e., $x \sim \mathcal{N}(0, I)$, i.e., the DIP setting [29].

As there is a single input to the network, we will drop the dependence of z with respect to x for the sake of clarity, only focusing on the dependence with respect to the weights, denoted as $z(w)$, where the high-dimensional vector w contains all individual weights $w_{i,j,\alpha}^\ell$. We assume that the weights of the network are drawn iid using the standard *He initialization*[3] [10], $w_{i,1,\alpha}^1 \sim \mathcal{N}(0, \frac{\sigma_w^2}{r^2})$ and $w_{i,j,\alpha}^\ell \sim \mathcal{N}(0, \frac{\sigma_w^2}{r^2 c})$ for $\ell = 2, \dots, L$, which avoids vanishing or divergent outputs in deep networks, where σ_w^2 is chosen depending on the non-linearity [36], e.g., $\sigma_w^2 = 2$ for relu. As in most image restoration problems, we assume training is performed on the squared loss, defined as $\mathcal{L}(w) = \frac{1}{2}\|z(w) - y\|_2^2$.

3.2. Non-local denoisers

Multiple existing non-local collaborative filtering techniques [20], such as the well-known NLM, BM3D or

[2]While our derivations focus on a simple CNN structure for the sake of clarity of the presentation, the analysis can be extended to account for multiple channels at the input and output (e.g., RGB images), biases, skip connections, downsampling and upsampling operations, see Supplementary Materials (SM) A, E and F.

[3]The analysis also covers other popular choices such as *LeCun* and *Xavier*, both with uniform or Gaussian distributions, as long as they verify that the size of intermediate layers' iid weights scale as $\mathcal{O}(c^{-1/2})$.

LARK [27], consist in computing a filtering matrix W with the (i,j)th entry related to the affinity between a (noisy) image patch y_i centered at pixel i and another (noisy) image patch y_j centered at pixel j. For example, the NLM affinity function[4] with patch size of $r \times r$ and parameter σ^2 is

$$[W]_{i,j} = k_{\text{NLM}}(y_i, y_j) = e^{-\frac{\|y_i - y_j\|_2^2}{\sigma^2}} \tag{4}$$

The most basic denoising procedure[5] consists of applying W directly to the noisy image $z = Wy$. However, the performance can be improved using an iterative procedure named *twicing* [20]

$$z^{t+1} = z^t + W(y - z^t) \tag{5}$$

Given a fixed positive semidefinite filter matrix with eigendecomposition $W = V\text{diag}\,(\lambda_1, \ldots, \lambda_d)\, V^\top$, we can express the output in the orthogonal basis V, i.e.,

$$z^t = \sum_{i=1}^{d} (1 - (1 - \lambda_i)^t)(v_i^\top y) v_i \tag{6}$$

where v_i is the ith column of V and $0 \leq \lambda_i < 2$. Assuming that W is approximately independent of the noise [20], the mean squared error (MSE) can be easily estimated as

$$\text{MSE} \approx \sum_{i=1}^{d} (1 - \lambda_i)^{2t} (v_i^\top \hat{x})^2 + (1 - (1 - \lambda_i)^t)^2 \sigma^2 \tag{7}$$

where $\hat{x}$ denotes the noiseless image, and the first and second terms represent the (squared) bias and variance respectively. As it can be seen in eq. (7), the twicing strategy trades bias for variance, starting with a blurry estimate and converging towards the noisy target y as $t \to \infty$. As with the early-stopped neural networks, the procedure is stopped before overfitting the noise.

For a fixed signal-to-noise ratio, the denoising performance will depend on how concentrated is the energy of the signal x on the leading eigenvectors of V (controlling the bias term) and how fast is the decay of the eigenvalues of the filter (controlling the variance term). It will also depend on how close are the computed non-local similarities using the noisy image from the oracle ones (computed with the clean image). For example, BM3D also adapts the filtering matrix, by using a prefiltered version of the noisy image to calculate the affinity between pixels [7].

4. Neural tangent kernel analysis

The seminal work in [12], and subsequent works [15, 37, 1], pointed out that as the number of parameters goes to infinity, which equates to taking $c \to \infty$, a network trained with GD and learning rate η of order[6] $\mathcal{O}(c^{-1})$, leads to a vanishingly small change of each individual weight [15, 1]

$$\max_t |(w_{i,j,\alpha}^\ell)^t - (w_{i,j,\alpha}^\ell)^0| = \begin{cases} \mathcal{O}(c^{-1}) & \text{if } \ell = L \\ \mathcal{O}(c^{-3/2}) & \text{otherwise} \end{cases} \tag{8}$$

where t denotes the gradient descent iteration, such that the overall change of the parameter vector $\|w - w^0\|_2$ is of order $\mathcal{O}(c^{-1/2})$. Hence, the evolution of the network can be well described by a first order expansion around the random initialization, $z(w^t) \approx z(w^0) + \frac{\delta z}{\delta w}(w^t - w^0)$, where $\frac{\delta z}{\delta w}$ is the Jacobian of the network at initialization, whose columns are shown in Figure 1. In this regime, the training dynamics reduce to

$$z^{t+1} = z^t + \eta \Theta_L^0 (y - z^t) \tag{9}$$

with $z^0 = z(w^0)$ and the positive semidefinite NTK Gram matrix (1 training sample and d outputs) given by

$$\Theta_L^0 = \frac{\delta z}{\delta w} (\frac{\delta z}{\delta w})^\top \Big|_{w=w^0} \tag{10}$$

which stays constant throughout training as $c \to \infty$.

The denoising process in eq. (9) is identical to the twicing procedure in eq. (5), where the filter W is given by the non-local affinity matrix $\eta\Theta_L^0$. The resulting pixel affinity function depends on the architecture, such as depth, convolution kernel size and choice of non-linearity. The size of each patch is given by the network's receptive field, as illustrated in Figure 2. As with non-local filters, the denoising performance depends on the alignment between the noiseless image and the leading eigenvectors of $\eta\Theta_L^0$. As shown in Figure 1, the filter associated with a CNN exhibits a fast decay of its eigenvalues and the image contains most of its energy within the leading eigenvectors.

The filter $\eta\Theta^0$ can be computed in closed form via the following recursion[7] [1]

$$\begin{cases} \Sigma_{a^\ell}^0 = \mathcal{A}\left(V\left(\Sigma_{a^{\ell-1}}^0\right)\right) \\ \eta\Theta_\ell^0 = \Sigma_{a^\ell}^0 + \mathcal{A}\left(V'\left(\Sigma_{a^{\ell-1}}^0\right) \circ \eta\Theta_{\ell-1}^0\right) \end{cases} \tag{11}$$

with base case (one hidden layer)

$$\Sigma_{a^2}^0 = \eta\Theta_2^0 = V\left(\mathcal{A}\left(xx^\top\right)\right) \tag{12}$$

where $\circ$ denotes element-wise matrix multiplication, and $\Sigma_{a^\ell}^0$ denotes the covariance of the preactivations of the network a_i^ℓ for all $i = 1, \ldots, c$. The convolution operator related to

[4] There is a subtle, but important point: the NLM filter matrix is normalized [20] as $W' = \text{diag}(1/1^\top W)W$ or using Sinkhorn's positive semidefinite approximation.

[5] Although this procedure seems to be linear it is in fact nonlinear due to the dependence of W on y.

[6] The learning rate cannot be larger than $\mathcal{O}(c^{-1})$ in order to converge to a global minimum [13]. We have also observed in our experiments using the larger learning rates leads to a divergent output.

[7] A detailed derivation is provided in SM D.

a filter size of $r \times r$ pixels is a mapping between positive semidefinite matrices $\mathcal{A} : \text{PSD}_d \mapsto \text{PSD}_d$ defined as [36]

$$[\mathcal{A}(\Sigma)]_{i,j} = \frac{1}{r^2} \sum_{i',j'} [\Sigma]_{i',j'} \tag{13}$$

where i' and j' indicate the pixels within patches of size $r \times r$ centered at pixels i and j respectively. The maps $V : \text{PSD}_d \mapsto \text{PSD}_d$ and $V' : \text{PSD}_d \mapsto \text{PSD}_d$ are defined by the choice of non-linearity and its derivative as

$$V(\Sigma) = \sigma_w^2 \mathbb{E}_{h \sim \mathcal{N}(0,\Sigma)} \{\phi(h)\, \phi(h^\top)\} \tag{14}$$

$$V'(\Sigma) = \sigma_w^2 \mathbb{E}_{h \sim \mathcal{N}(0,\Sigma)} \{\phi'(h)\, \phi'(h^\top)\} \tag{15}$$

which are available in closed form for many popular non-linearities including relu (see SM B).

For example, a relu CNN with a single hidden layer and a convolution kernel size of $r \times r$ pixels has an associated affinity function

$$k_{\text{CNN}}(x_i, x_j) = \frac{\|x_i\|_2 \|x_j\|_2}{\pi} (\sin(\varphi) + (\pi - \varphi)\cos(\varphi)) \tag{16}$$

with $\varphi = \arccos \frac{x_i^\top x_j}{\|x_i\|_2 \|x_j\|_2}$. Figure 2 illustrates the filter computed using the closed-form kernel in eq. (16), which weights similar patches more strongly.

4.1. Computing the analytic filter

Instead of training a neural network as in the DIP, we can explicitly compute the filtering matrix $\eta\Theta^0$, and use eq. (5) to perform the denoising. As the size of the filter matrix ($d \times d$) is prohibitively big to compute and store for large images, we instead only compute a random selection of $m \ll d$ columns of $\eta\Theta^0$, and approximate the matrix with its leading eigenvalues and eigenimages using the Nyström method [33]. The columns are chosen by selecting random pixels uniformly distributed in space, as in the global image denoising algorithm [28]. A detailed description of the algorithm can be found in SM G.

5. Adaptive filtering and the deep image prior

5.1. The DIP is not a low-pass filter

In the DIP paper, the input is chosen to be random iid noise. In this case, the resulting filter $\eta\Theta^0$ does not depend in any way on the target image y, and the non-local similarities are computed using the input noise. How bad can this filter be? Applying eq. (11) with noise at the input [8] we get in expectation

$$[\eta\Theta^0]_{i,j} = \frac{1}{d} \begin{cases} 1 & \text{if } i = j \\ \kappa_L & \text{otherwise} \end{cases} \tag{17}$$

with $\kappa_L \approx 0.25$ for large L, which has a very large first eigenvalue $\lambda_1 = (1 - \kappa_L)/d + \kappa_L \approx 0.25$ associated with a constant image $v_1 = [1, \ldots, 1]^\top / \sqrt{d}$ and the rest of the eigenvalues of small size $\lambda_i = 0.75/d$ for $i = 2, \ldots, d$. Hence, this (linear) filter would just be useful for constant images. In the case of an autoencoder (AE) architecture with downsampling and upsampling layers, the resulting filter is a crude low-pass filter, but still does not depend on the target image. Previous works [6, 11] hypothesized that this filter can explain the bias towards clean images. However, it is well-known that low-pass filters don't provide good denoising results, as they tend to oversmooth the edges and fine details of the image. We show that this gap between theory and practice is because the DIP in [29] is not trained with GD but Adam.

5.2. Adaptive filtering with Adam

The Adam optimizer updates the weights according to

$$w^{t+1} = w^t - \tilde{\eta} H^t \frac{\delta \mathcal{L}}{\delta w}(w^t) + \beta_1 (w^t - w^{t-1}) \tag{18}$$

where β_1 is a hyperparameter controlling the momentum and learning rate, $\tilde{\eta} = \eta(1 - \beta_1)$ and H^t is a diagonal matrix containing the inverse of a running average of the squared value of the gradients, computed using the hyperparameter β_2. The resulting filter is adapted at each iteration,

$$\tilde{\Theta}_L^t = \frac{\delta z}{\delta w} H^t \left(\frac{\delta z}{\delta w}\right)^\top \Big|_{w=w^t} \tag{19}$$

and the denoising process can be written as

$$z^{t+1} = z^t + \tilde{\eta} \tilde{\Theta}_L^t (y - z^t) + \beta_1 (z^t - z^{t-1}) \tag{20}$$

The matrix H^k imposes a metric in the weight space which differs from the standard Euclidean metric of GD. Unfortunately, as shown by Gunasekar et al. [9], this metric depends on the choice of the learning rate, rendering a general analysis of the adaptation intractable[9]. Moreover, as shown in our experiments, all weights, including intermediate layers, undergo a larger change than in gradient descent, that is

$$\max_t |(w_{i,j,\alpha}^\ell)^t - (w_{i,j,\alpha}^\ell)^0| = \mathcal{O}(c^{-1}) \quad \forall \ell = 1, \ldots, L \tag{21}$$

such that the overall change of the parameter vector $\|w^t - w^0\|_2$ is $\mathcal{O}(1)$, and a Taylor expansion around the initialization does not model accurately the training dynamics[10]. Nonetheless, here we provide insight into how the resulting

[8] Here we assume that the number of noise input channels is of order $\mathcal{O}(c)$, as in the DIP [29].

[9] Removing the adaptation ($\beta_1, \beta_2 \to 0$) the algorithm reduces to sign gradient descent, i.e., steepest descent with respect to the ℓ_∞ norm [3], but it is still sensitive to the choice of learning rate [9]

[10] Note that a higher order expansion [2] cannot explain the good performance of the DIP, as higher order derivatives are still independent of the target. Moreover, the Hessian would not describe $\mathcal{O}(1)$ perturbations.

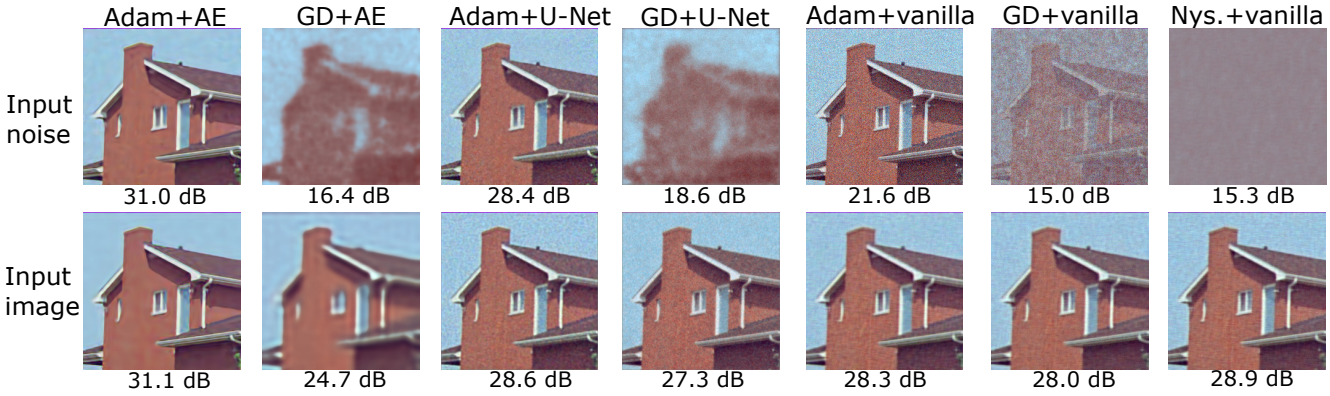

Figure 3: Results for the 'house' image. PSNR values are reported below each restored image. The best results are obtained by the autoencoder trained with Adam, which is able to provide smoother estimates while preserving sharp edges. However, it provides worse estimates of images with noise-like textures, such as the 'baboon' image (see SM I).

	Vanilla CNN		U-Net		Autoencoder	
	Noise	Image	Noise	Image	Noise	Image
Adam	19.6	27.4	28.3	28.1	29.2	**29.3**
Gradient descent	15.2	27.5	16.5	27.1	15.0	26.8
Nyström	15.2	28.3				

Table 1: Average peak-signal-to-noise ratio (PSNR) [dB] achieved by different combinations of network architecture, input and optimizer on the dataset of 9 color images [7].

filtering kernel can still absorb non-local properties from the *target output*. Similarly to the output dynamics, the evolution of the preactivations can be well described by its (time-varying) first order expansion[11]:

$$(a_i^\ell)^{t+1} \approx (a_i^\ell)^t - \eta \frac{\delta a_i^\ell}{\delta w} H^k \left(\frac{\delta a_i^\ell}{\delta w}\right)^\top \frac{\delta \mathcal{L}}{\delta a_i^\ell} \tag{22}$$

$$\approx (a_i^\ell)^t - \eta \tilde{\Theta}_\ell^t (\delta_i^\ell)^t \tag{23}$$

where the error gradient at layer ℓ and channel i is defined as $\delta_i^\ell \overset{\text{def}}{=} \frac{\delta \mathcal{L}}{\delta a_i^\ell} \in \mathbb{R}^d$. At initialization, this vector carries non-local information (via operator $\mathcal{A}$) about the target y, with covariance given by the recursion [36, 37]

$$\Sigma^0_{\delta^\ell} = \mathcal{A}\left(\Sigma^0_{\delta^{\ell+1}}\right) \circ V'\left(\Sigma^0_{a^\ell}\right) \tag{24}$$

starting with

$$\Sigma^0_{\delta^{L-1}} = c^{-1} \mathcal{A}\left((y - z^0)(y - z^0)^\top\right) \circ V'\left(\Sigma^0_{a^{L-1}}\right) \tag{25}$$

which depends on the target image via the residual $(y - z^0)$. A full derivation of eq. (24) is provided in SM C. In the case of GD training, the change in the preactivations from initialization is negligible as the error terms δ_i^ℓ are of order $\mathcal{O}(c^{-1/2})$ due to the c^{-1} scaling in eq. (25). However, the larger change in intermediate layers when using Adam yields a non-negligible change in the preactivations, while the exact adaptation depends on the choice of hyperparameters η, β_1 and β_2. This larger change lies at the heart of the improved performance of the DIP in comparison with GD training.

[11] Here we assume $\beta_1 = 0$ for simplicity.

6. Experiments

We analyze the performance of single-image denoising neural networks both with the corrupted image or iid noise at the input of the network on a standard dataset of 9 color images [7] corrupted with Gaussian noise of $\sigma = 25$. We evaluate 3 different architectures, a simple vanilla CNN with a single hidden layer and a kernel size of 11×11 pixels, a U-Net with 3 downsampling and upsampling stages and a kernel size of 3×3 pixels and an autoencoder with the same architecture as the U-Net but no skip-connections. All architectures use relu non-linearities. A detailed description of the chosen architectures can be found in SM H. For each combination of input and architecture, we optimize the network using Adam with standard hyperparameters ($\beta_1 = 0.9$ and $\beta_2 = 0.99$) and vanilla GD (no momentum). We also include results achieved by taking the infinite channel limit of the vanilla CNN, and computing the associated NTD filter. In this case, we use the Nyström approximation to reduce the memory requirements of storing the full matrix $\eta\Theta$. We found that computing only 2% of its columns gives a negligible reduction of performance with respect to computing

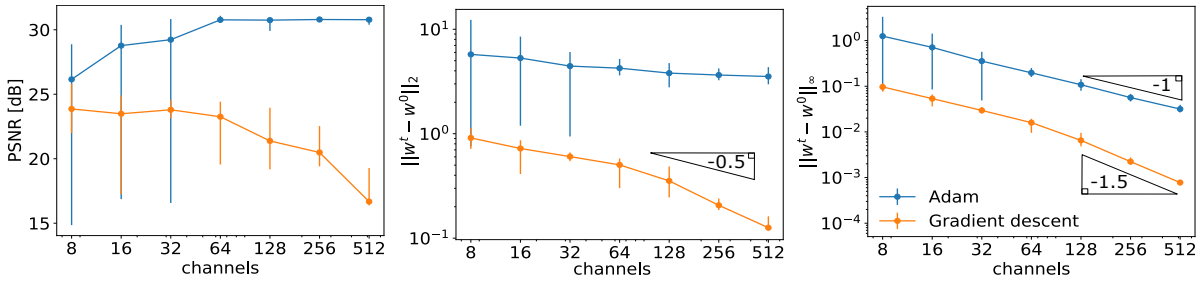

Figure 4: Comparison of Adam and GD training of an autoencoder with noise at the input as a function of the number of channels. The PSNR for the 'house' image is shown on the left plot, whereas the average ℓ_2 and ℓ_∞ change of weights in hidden layers is shown on the center and right plots respectively. The error bars denote the maximum and minimum values obtained in 10 Monte Carlo repetitions.

the full matrix. In the experiments with the image at the input, we remove the random initial output by redefining the network function as $\tilde{z} = z - z^0$ with a fixed translation z^0, such that $\tilde{z}^0 = 0$ (as with standard twicing). We run the optimization until there is no further improvement of the peak signal-to-noise ratio (PSNR)[12] or a maximum of 10^6 iterations is reached, and keep the best performing output.

6.1. Denoising performance

The average PSNR obtained by all possible configurations is shown in Table 1. The results for one of the images in the dataset are shown in fig. 3.

The best performances are achieved by the autoencoder architecture optimized with Adam, followed by the induced filter of the vanilla CNN, computed with the Nyström approximation. It is worth noting that while the autoencoder in the DIP uses batch normalization, biases, leaky relus instead of relus and a Swish activation function at the output, it does not perform significantly better without them (same average PSNR as the results reported in [29] and 0.1 dB improvement when placing the corrupted image at the input). Furthermore, the best results are obtained when placing the corrupted image at the input, without requiring the carefully-designed loss functions of Noise2Void and Noise2Self.

As predicted in Section 5.1, GD provides very poor reconstructions when inputting noise, but improves considerably with the corrupted image as the input, as only the latter has access to the non-local structure. While the vanilla CNN trained with GD and its Nyström approximation should in theory perform the same, the difference can be attributed to Nyström's lower rank approximation. Even though Adam plays a big role in adapting the autoencoder filter (8 hidden layers), it does not modify significantly the filter associated with a single hidden layer vanilla CNN. Denoising using the Nyström approximation of the analytic filter takes an average of 3 seconds per image, while training the autoencoder with Adam required 806 seconds[13]. This significant difference illustrates the potential speed up that can be obtained by having a better theoretical understanding of the denoising network.

Figure 6 shows the performance of different vanilla architectures trained via GD and their associated Nyström approximations. The evaluated networks have the same receptive fields but different depths. In this setting (NTK regime), shallower networks achieve better performance than deeper counterparts.

Interestingly, the fixed CNN filter (via GD) induced by the vanilla architecture performs better than its autoencoder counterpart. Despite having a larger receptive field (i.e., comparing larger patches), we observed that the autoencoder's eigenimages are more blurry than the vanilla CNN.

Despite being able to denoise with a single noisy image as training data, we note that the evaluated methods are below the performance of color NLM and CBM3D which obtain a PSNR of 30.26 and 31.42 dB respectively. However, we emphasize that the goal of this paper is to understand the implicit bias of CNNs rather than provide new state-of-the-art denoisers.

6.2. Change of weights during training

Figure 4 shows the PSNR obtained on the house image, and the change of weights in intermediate layers, $\|w^t - w^0\|_2$ and $\sum_{1<\ell<L} \frac{1}{L-2} \max_{i,j,\alpha} |(w^\ell_{i,j,\alpha})^t - (w^\ell_{i,j,\alpha})^0|$ as a function of the number of channels when applying GD and Adam with an autoencoder architecture. For Adam, the denoising performance only improves as the number of channels increases, suggesting that the adaptive kernel property is not due to a finite network effect. As explained in Section 5, when using Adam the weights in intermediate layers suffer a change of $\mathcal{O}(1)$ with respect to the ℓ_2 norm, hence adapting the filter at initialization, whereas GD attains

[12] While we use the oracle image for a fair comparison of all methods, a SURE estimator of the mean squared error [24] could be used in practical applications.

[13] All the experiments were run with a GPU NVIDIA GTX 1080 Ti using the PyTorch library.

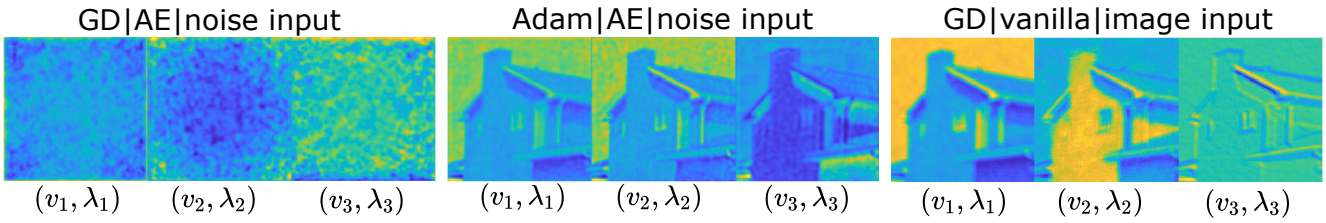

Figure 5: First 3 leading eigenvectors of the covariance of the last preactivations, $\Sigma_{a^{L-1}}$, after 500 iterations of training with Adam or gradient descent with different inputs (noise or image).

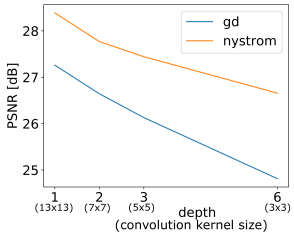

Figure 6: Denoising performance of the vanilla CNN on the dataset of 9 images for varying depth and a fixed total receptive field of 13×13 pixels.

a change of the order $\mathcal{O}(c^{-1/2})$, which corresponds to a fixed filter as $c \to \infty$. Furthermore, all individual weights incur a similar small change of order $\mathcal{O}(c^{-1/2})$ during training with Adam, suggesting that each weight induces a similar (small) contribution to the network output, in contrast with convolutional sparse coding [35] interpretations, where only a few weights are non-negligible.

Finally, fig. 5 shows the leading eigenvectors of the preactivations of the last hidden layer after 500 iterations of training with GD and Adam. As explained in Section 5, GD does not modify the distribution of the preactivations during training, hence they remain non-informative (low-pass [6, 11]) if noise is placed at the input of the network. However, they carry non-local information when the image is placed at the input. On the other hand, Adam, even with noise at the input, modifies the initial non-informative distribution with non-local features extracted from the target.

7. Discussion

Fully-convolutional trained networks, such as DnCNN [39], do not perform fully global denoising, as the filter is constrained by the size of the receptive field, which does not cover the full image. There has been recent efforts to construct networks which explicitly exploit non-local self-similarities in a fully global fashion, either via non-local networks [32] or using architectures that explicitly operate on noisy patches [17, 30]. The setting studied here, i.e., training a network with a single corrupted image, corresponds to global filtering [28], as correlations between all patches in the image are considered. Our framework has the potential to combine both training data and the exploitation of self-similarities, e.g., through global filtering and Nyström. Finally, it is worth noting that the twicing framework in eq. (9) can also handle general inverse problems with training loss $\mathcal{L}(w) = \frac{1}{2}\|y - Az(w)\|^2$ where A is an ill-conditioned forward operator. We leave the analysis of this general setting for future work.

8. Conclusions

We introduced a novel analysis of CNN denoisers trained with a single corrupted image, using the recent discovery of the neural tangent kernel to elucidate the strong links with non-local patch-based filtering methods. As the number of channels of the network tends to infinity, the associated pixel affinity function is available in closed form, thus we can study the properties of the induced filter and understand the denoising through the NTK's low rank approximation. These results bring insight about the inductive bias of CNNs in image processing problems: The effective degrees of freedom are significantly smaller than the actual number of weights in the network, being fully characterized by the architecture and initialization of the network.

While the NTK theory accurately predicts the behaviour of networks trained with standard gradient descent, we show that it fails to describe the induced filter when training with the popular Adam optimizer. Interestingly, while Adam and other adaptive gradient optimizers are known to provide worse results than stochastic gradient descent in random features models [34], they play a key role here by adapting the filter with non-local information about the target image in the context of the deep image prior. We believe that understanding better the dynamics and hence the inductive bias of these optimizers, will be a very important step for improving our understanding of CNN models, both for denoising and more general imaging and image analysis problems.

Acknowledgements

This work is supported by the ERC Advanced Grant 694888, C-SENSE, and a Royal Society Wolfson Research Merit Award.

References

[1] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems 32*, pages 8141–8150. Curran Associates, Inc., 2019. 2, 4

[2] Yu Bai, Ben Krause, Huan Wang, Caiming Xiong, and Richard Socher. Taylorized Training: Towards Better Approximation of Neural Network Training at Finite Width. *arXiv e-prints*, page arXiv:2002.04010, Feb. 2020. 5

[3] Lukas Balles and Philipp Hennig. Dissecting adam: The sign, magnitude and variance of stochastic gradients. *arXiv preprint arXiv:1705.07774*, 2017. 5

[4] Joshua Batson and Loic Royer. Noise2Self: Blind denoising by self-supervision. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 524–533, Long Beach, California, USA, 09–15 Jun 2019. PMLR. 1, 3

[5] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 60–65. IEEE, 2005. 1, 2

[6] Z. Cheng, M. Gadelha, S. Maji, and D. Sheldon. A Bayesian perspective on the deep image prior. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5438–5446, 2019. 2, 3, 5, 8

[7] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 1, 2, 4, 6

[8] Michael Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010. 1

[9] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. volume 80, pages 1832–1841, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 5

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 3

[11] Reinhard Heckel and Mahdi Soltanolkotabi. Denoising and regularization via exploiting the structural bias of convolutional generators. In *International Conference on Learning Representations*, 2020. 2, 3, 5, 8

[12] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems 31*, pages 8571–8580. Curran Associates, Inc., 2018. 2, 4

[13] Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Universal statistics of fisher information in deep neural networks: Mean field approach. In *Prof. of Mach. Learning Research*, volume 89, pages 1032–1041. PMLR, 16–18 Apr 2019. 4

[14] A. Krull, T. Buchholz, and F. Jug. Noise2void - learning denoising from single noisy images. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2124–2132, 2019. 1, 3

[15] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems 32*, pages 8572–8583. Curran Associates, Inc., 2019. 4

[16] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 2965–2974, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 3

[17] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *Advances in Neural Information Processing Systems*, pages 1673–1682, 2018. 8

[18] J. Liu, Y. Sun, C. Eldeniz, W. Gan, H. An, and U. S. Kamilov. Rare: Image reconstruction using deep priors learned without ground truth. *IEEE Journal of Selected Topics in Signal Processing*, pages 1–1, 2020. 1

[19] Morteza Mardani, Qingyun Sun, David Donoho, Vardan Papyan, Hatef Monajemi, Shreyas Vasanawala, and John Pauly. Neural proximal gradient descent for compressive imaging. In *Advances in Neural Information Processing Systems 31*, pages 9573–9583. Curran Associates, Inc., 2018. 1

[20] Peyman Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE Signal Processing Magazine*, 30(1):106–128, 2012. 2, 3, 4

[21] Sreyas Mohan, Zahra Kadkhodaie, Eero P. Simoncelli, and Carlos Fernandez-Granda. Robust and interpretable blind image denoising via bias-free convolutional neural networks. In *International Conference on Learning Representations*, 2020. 3

[22] Radford M Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995. 2, 3

[23] Roman Novak, Lechao Xiao, Yasaman Bahri, Jaehoon Lee, Greg Yang, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-dickstein. Bayesian deep convolutional networks with many channels are Gaussian processes. In *International Conference on Learning Representations*, 2019. 2

[24] S. Ramani, T. Blu, and M. Unser. Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms. *IEEE Transactions on Image Processing*, 17(9):1540–1554, 2008. 7

[25] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (RED). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017. 1

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 1

[27] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Transactions on Image Processing*, 16(2):349–366, 2007. 4

[28] H. Talebi and P. Milanfar. Global image denoising. *IEEE Transactions on Image Processing*, 23(2):755–768, 2014. 5, 8

[29] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition, pages 9446–9454, 2018. 1, 3, 5, 7

[30] Gregory Vaksman, Michael Elad, and Peyman Milanfar. Lidia: Lightweight learned image denoising with instance adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 8

[31] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013. 1

[32] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 8

[33] Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001. 2, 5

[34] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems 30*, pages 4148–4158. Curran Associates, Inc., 2017. 8

[35] B. Wohlberg. Efficient algorithms for convolutional sparse representations. *IEEE Transactions on Image Processing*, 25(1):301–315, 2016. 8

[36] Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel S Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. *arXiv preprint arXiv:1806.05393*, 2018. 3, 5, 6

[37] Greg Yang. Scaling Limits of Wide Neural Networks with Weight Sharing: Gaussian Process Behavior, Gradient Independence, and Neural Tangent Kernel Derivation. *arXiv e-prints*, page arXiv:1902.04760, Feb. 2019. 2, 4, 6

[38] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016. 2

[39] 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*, 26(7):3142–3155, 2017. 8