MASTER SIF: Representation Edition and Perception of images (REP)

Written Exam

November 7, 2022

Lecture notes are authorized during the exam. You have to put your name and numbers on each of the answer sheets.

In order to help you, the level of difficulty for each question is indicated with the following code :

$\star$ easy or strongly linked with what has been seen during the lecture

$\star\star$ question of averaged difficulty requiring an interpretation of the concepts seen during the lecture.

$\star\star\star$ difficult or research-oriented question

Each question owes 1.25 points (whatever the level of difficulty).

In this exam, we will study the following paper:

[Mohan20] *S Mohan, Z Kadkhodaie, E P Simoncelli and C Fernandez-Granda, "Robust and interpretable blind image denoising via bias-free convolutional neural networks" Int. Conf. on Learning Representations (ICLR), Apr 2020.*

Here is the abstract of [Mohan20]:

Abstract - *We study the generalization properties of deep convolutional neural networks for image denoising in the presence of varying noise levels. We provide extensive empirical evidence that current state-of-the-art architectures systematically overfit to the noise levels in the training set, performing very poorly at new noise levels.*
We show that strong generalization can be achieved through a simple architectural modification: removing all additive constants. The resulting "bias-free" networks attain state-of-the-art performance over a broad range of noise levels, even when trained over a narrow range. They are also locally linear, which enables direct analysis with linear-algebraic tools. We show that the denoising map can be visualized locally as a filter that adapts to both image structure and noise level. In addition, our analysis reveals that deep networks implicitly perform a projection onto an adaptively-selected low-dimensional subspace, with dimensionality inversely proportional to noise level, that captures features of natural images.

Here is an extract of the introduction that also states the goal of the paper:

Introduction - *The problem of denoising consists of recovering a signal from measurements corrupted by noise (...). In the past decade, convolutional neural networks have achieved state-of-the-art results in image denoising. Despite their success, these solutions are mysterious: we lack both intuition and formal understanding of the mechanisms they implement.*
The goal of this work is to advance our understanding of deep-learning models for denoising. Our contributions are twofold: First, we study the generalization capabilities of deep-learning models across different noise levels. Second, we provide novel tools for analyzing the mechanisms implemented by neural networks to denoise natural images.

Part I - about *denoising* problem

As it can be understood from the extracts above, the paper deals with image denoising. Here is how the authors define the denoising problem in [Mohan20]:

We assume a measurement model in which images are corrupted by additive noise: $y = x + n$, where $x \in \mathbb{R}^N$ is the original image, containing N pixels, n is an image of i.i.d. samples of Gaussian noise with variance σ^2, and y is the noisy observation. The denoising problem consists of finding a function $f : \mathbb{R}^N \to \mathbb{R}^N$, that provides a good estimate of the original image, x. Commonly, one minimizes the mean squared error : $f = argmin_g E||x - g(y)||^2$, where the expectation is taken over some distribution over images, x, as well as over the distribution of noise realizations.

1. $\star$ During a denoising operation, explain what is the noisy image that has to be corrected: x, y or n?
2. $\star$ Explain why the function f has to be designed such that it satisfies $f = \text{argmin}_g E||x - g(y)||^2$.
3. $\star$ If we assume that an image usually has an energy compacted to low frequencies, give an example of *linear* filtering that could be used for denoising.
4. $\star$ Alternatively, *non-linear* filters can be used. Explain their differences with linear filter and give one example of non-linear filter that could be used for denoising.

Part II - about CNN

Authors propose to understand CNN's behavior when it is trained to perform a denoising task. Here is how the authors define a CNN:

Feedforward neural networks with rectified linear units (ReLUs) are piecewise affine: for a given activation pattern of the ReLUs, the effect of the network on the input is a cascade of linear transformations (convolutional or fully connected layers, W_k), additive constants (b_k), and pointwise multiplications by a binary mask corresponding to the fixed activation pattern (R). Since each of these is affine, the entire cascade implements a single affine transformation. For a fixed noisy input image $y \in \mathbb{R}^N$ with N pixels, the function $f : \mathbb{R}^N \to \mathbb{R}^N$ computed by a denoising neural network may be written

$$f(y) = W_L R(W_{L-1} ... R(W_1 y + b_1) + ... b_{L-1}) + b_L = A_y y + b_y, \tag{1}$$

where $A_y \in \mathbb{R}^{N\times N}$ is the Jacobian of $f()$ evaluated at input y, and $b_y \in \mathbb{R}^N$ represents the net bias. The subscripts on A_y and b_y serve as a reminder that both depend on the ReLU activation patterns, which in turn depend on the input vector y.

5. $\star$ In the paragraph above, authors write the CNN equations as a cascade of elementary operations (called layers), each of them following the equation: $output_k = W_k R(input_{k-1}) + b_k$. Is this operation linear or non-linear? Justify your answer.
6. $\star\star$ Explain why the matrices W_k can both model a convolution and a fully connected layer? (it is not necessary to detail how the matrix W_k is built).
7. $\star\star\star$ Authors propose to gather all layers in the formula $A_y y + b_y$. Explain why A_y and b_y depend on y, and how these elements could be calculated for a given input y.

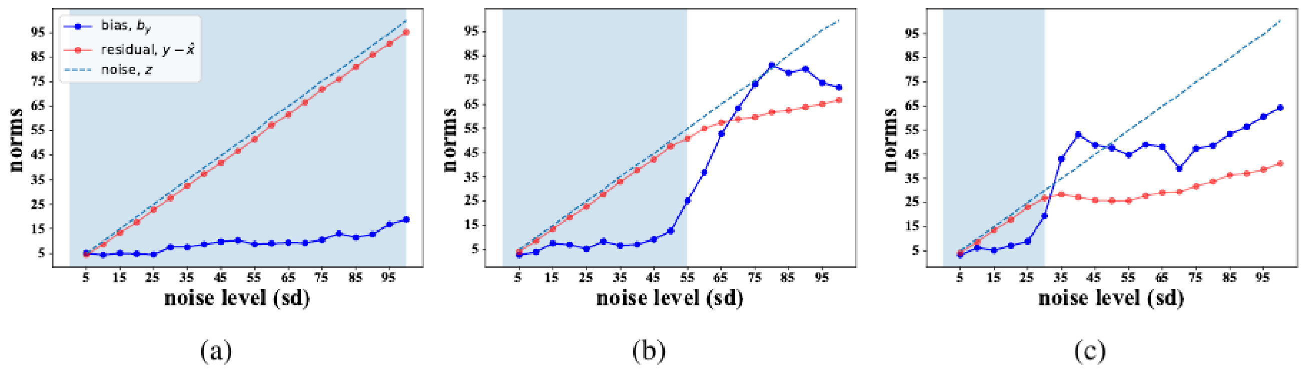

Figure 1: *First-order analysis of the residual of a denoising convolutional neural network as a function of noise level. The plots show the norms of the residual and the net bias averaged over 100* 20×20 *natural-image patches for networks trained over different training ranges. The range of noises used for training is highlighted in blue. (a) When the network is trained over the full range of noise levels (*$\sigma \in [0, 100]$*) the net bias is small, growing slightly as the noise increases. (b-c) When the network is trained over the a smaller range (*$\sigma \in [0, 55]$ *and* $\sigma \in [0, 30]$*), the net bias grows explosively for noise levels beyond the training range. This coincides with a dramatic drop in performance, reflected in the difference between the magnitudes of the residual and the true noise. The CNN used for this example is DnCNN; using alternative architectures yields similar results (...).* .

Part III - Bias free denoising

The authors propose to study the behavior of a CNN-based denoiser. Here is an extract:

Based on equation (1) we can perform a first-order decomposition of the error or residual of the neural network for a specific input: $y - f(y) = (I - A_y)y - b_y$ *. Figure 1 shows the magnitude of the residual and the constant, which is equal to the net bias* b_y *, for a range of noise levels. Over the training range, the net bias is small, implying that the linear term is responsible for most of the denoising (...). However, when the network is evaluated at noise levels outside of the training range, the norm of the bias increases dramatically, and the residual is significantly smaller than the noise, suggesting a form of overfitting. Indeed, network performance generalizes very poorly to noise levels outside the training range.*

8. $\star\star$ Explain why, in Figure 1, we should expect that the residual curve (in red) follows the noise curve (in dashed blue)?

9. $\star$ Authors speak of "overfitting", what does it mean for CNN? Explain from the Figure 1 (b) and (c) why the trained network overfits?

Here is another extract

Figure 1 shows that CNNs overfit to the noise levels present in the training set, and that this is associated with wild fluctuations of the net bias b_y*. This suggests that the overfitting might be ameliorated by removing additive (bias) terms from every stage of the network, resulting in a bias-free CNN (BF-CNN).*
(...)
In order to evaluate the effect of removing the net bias in denoising CNNs, we compare several state-of-the-art architectures to their bias-free counterparts, which are exactly the same except for the absence of any additive constants within the networks (note that this includes the batch-normalization additive parameter).
(...)
Figure 2 shows our results.

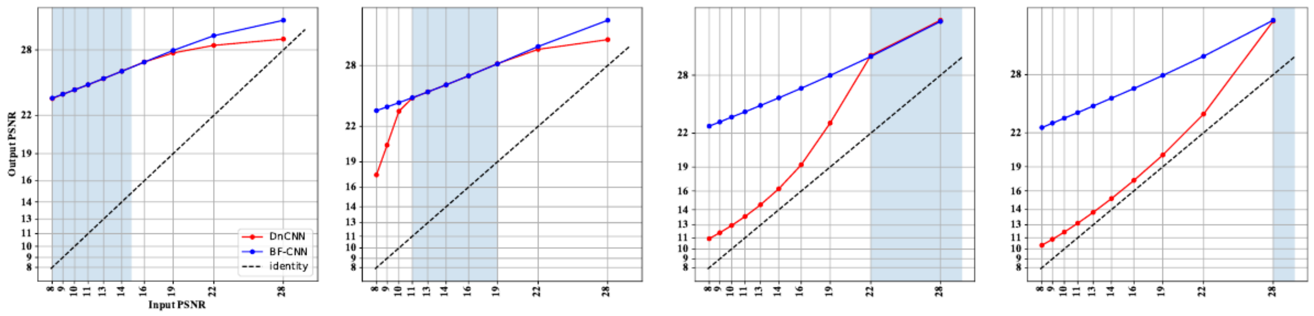

Figure 2: *Comparison of the performance of a CNN and a BF-CNN with the same architecture for the experimental design described in Figure 1. The performance is quantified by the PSNR of the denoised image as a function of the input PSNR. Both networks are trained over a fixed ranges of noise levels indicated by a blue background. In all cases, the performance of BF-CNN generalizes robustly beyond the training range, while that of the CNN degrades significantly. The CNN used for this example is DnCNN; using alternative architectures yields similar results*.

10. ⋆ From the results in Figure 2, explain why the authors can claim the following statement: *Our results provide strong evidence that removing net bias in CNN architectures results in effective generalization to noise levels out of the training range.*

Part IV - Interpretation on A_y

Here is another extract from the paper:

We perform a local analysis of BF-CNN networks, which reveals the underlying denoising mechanisms learned from the data. A bias-free network is strictly linear, and its net action can be expressed as:

$$f_{BF}(y) = W_L R(W_{L_1} ... R(W_1 y)) = A_y y, \tag{2}$$

where A_y is the Jacobian of $f_{BF}()$ evaluated at y. The Jacobian at a fixed input provides a local characterization of the denoising map.
(...)
The linear representation of the denoising map given by equation (3) implies that the ith pixel of the output image is computed as an inner product between the ith row of A_y , denoted $a_y(i)$, and the input image:

$$f_{BF}(y)(i) = \sum_{j=1}^{N} A_y(i,j) y(j) = a_y(i)^\top y. \tag{3}$$

The vectors $a_y(i)$ can be interpreted as adaptive filters that produce an estimate of the denoised pixel via a weighted average of noisy pixels. Examination of these filters reveals their diversity, and their relationship to the underlying image content: they are adapted to the local features of the noisy image, averaging over homogeneous regions of the image without blurring across edges. This is shown for two separate examples and a range of noise levels in Figure 3.

11. ⋆⋆ Explain why the authors say that ”The vectors $a_y(i)$ (of Equation (3)) can be interpreted as adaptive filters” (adaptative filter=non-linear filter)?

12. ⋆⋆ In Figure 3, authors show the weights of the adaptative filter $a_y(i)$ for three different pixels i. How does the filter support evolve when the noise is increasing? Explain why?

13. ⋆⋆ Again in Figure 3, is the shape of the filter's support the same for the different pixels? Explain why it is different and why such behavior leads to efficient denoising?

14. ⋆⋆⋆ What's the strength of a CNN architecture compared to other non-linear filters seen during the course (and that are “not deep”)?

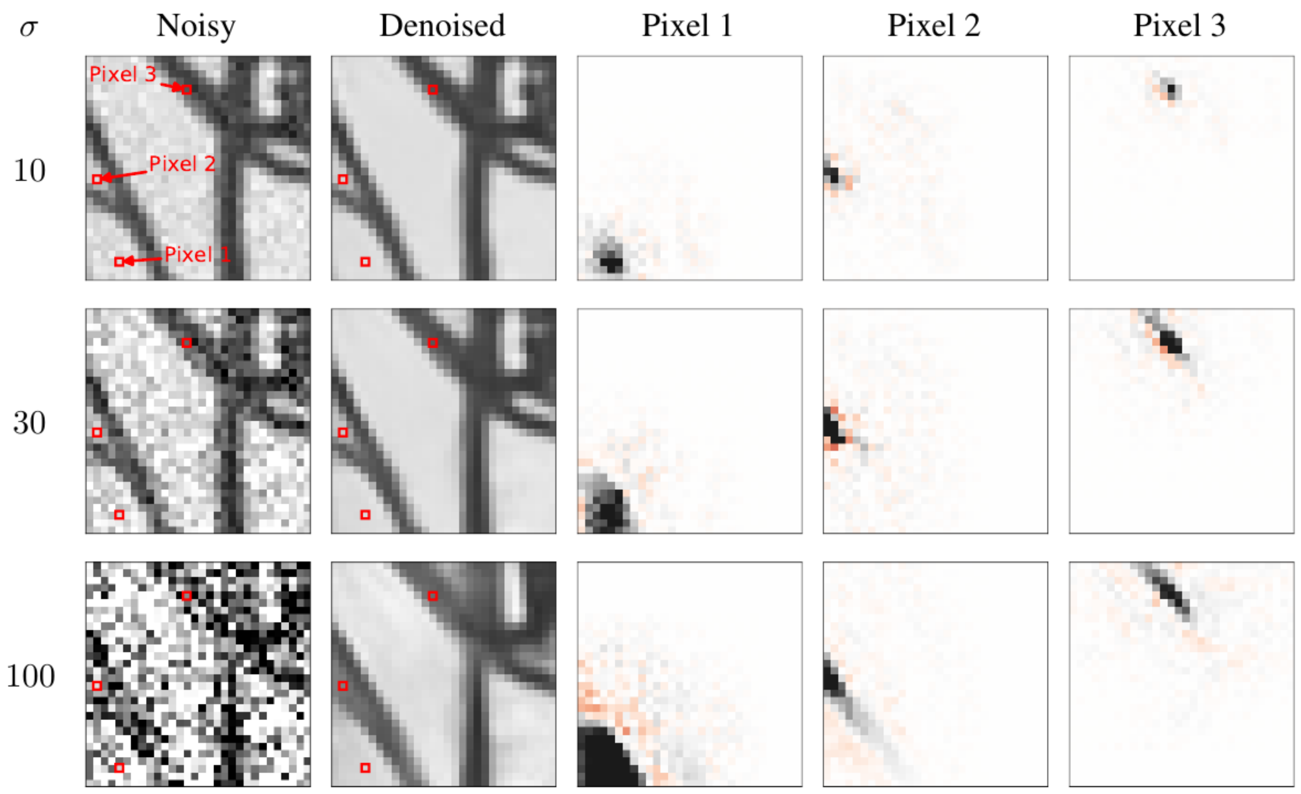

Figure 3: *Visualization of the linear weighting functions (rows of A_y in equation (3)) of a BF-CNN for three example pixels of an input image, and three levels of noise. The images in the three rightmost columns show the weighting functions used to compute each of the indicated pixels (red squares). All weighting functions sum to one, and thus compute a local average (note that some weights are negative, indicated in red). Their shapes vary substantially, and are adapted to the underlying image content. As the noise level σ increases, the spatial extent of the weight functions increases in order to average out the noise, while respecting boundaries between different regions in the image, which results in dramatically different functions for each pixel. The CNN used for this example is DnCNN; using alternative architectures yields similar results.* .

Here is another extract:

The local linear structure of a BF-CNN facilitates analysis of its functional capabilities via the singular value decomposition (SVD). For a given input y, we compute the SVD of the Jacobian matrix: $A_y = USV^T$, with U and V orthogonal matrices, and S a diagonal matrix. We can decompose the effect of the network on its input in terms of the left singular vectors $U_1, U_2..., U_N$ (columns of U), the singular values $s_1, s_2..., s_N$ (diagonal elements of S), and the right singular vectors $V_1, V_2, ...V_N$ (columns of V):

$$f_{BF}(y) = A_y y = USV^\top y = \sum_{i=1}^{N} s_i (V_i^\top y) U_i \tag{4}$$

The output is a linear combination of the left singular vectors, each weighted by the projection of the input onto the corresponding right singular vector, and scaled by the corresponding singular value. Analyzing the SVD of a BF-CNN on a set of ten natural images reveals that most singular values are very close to zero (...). The network is thus discarding all but a very low-dimensional portion of the input image. We also observe that the left and right singular vectors corresponding to the singular values with non-negligible amplitudes are approximately the same (...). This means that the Jacobian is (approximately) symmetric, and we can interpret the action of the network as projecting the noisy signal onto a low-dimensional subspace, as is done in wavelet thresholding schemes. This is confirmed by visualizing the singular vectors as images (Figure 4).

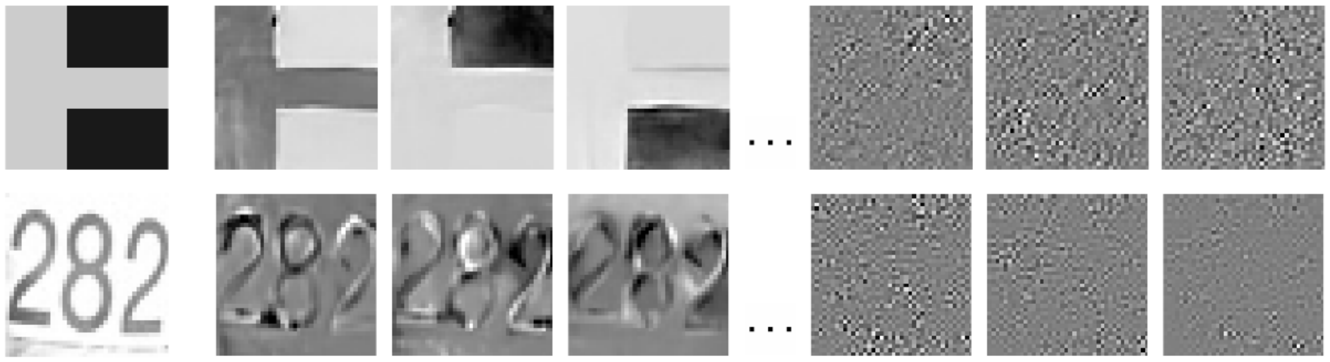

Figure 4: *Visualization of left singular vectors of the Jacobian of a BF-CNN, evaluated on two different images (top and bottom rows), corrupted by noise with standard deviation $\sigma = 50$. The left column shows original (clean) images. The next three columns show singular vectors corresponding to non-negligible singular values. The vectors capture features from the clean image. The last three columns on the right show singular vectors corresponding to singular values that are almost equal to zero. These vectors are noisy and unstructured. The CNN used for this example is DnCNN; using alternative architectures yields similar results.* .

15. $\star\star\star$ Explain why the authors say that the function $f_{BF}(y)$ can be seen as a wavelet thresholding (or any kind of transform-based thresholding algorithm, *e.g.,* ideal low-pass filter)?

16. $\star\star\star$ From Figure 4, explain what is the strength of CNN-based approach compared to traditional low-pass filters (answer should be different than in Question 14)?