Convolutional neural networks for image denoising

DS-GA 1013 / MATH-GA 2824 Mathematical Tools for Data Science

Carlos Fernandez-Granda

Prerequisites

Linear algebra (singular value decomposition)

Thresholding

Wiener filtering

Image denoising

- Linear translation-invariant estimation: Wiener filtering
- Nonlinear estimation: Thresholding in transform domain
- Nonlinear translation-invariant estimation: Convolutional neural networks (CNNs)

Deep learning for image denoising

- Gather dataset of natural images
- Simulate noise
- Train CNN to estimate clean image minimizing mean squared error

Wiener filtering

Linear translation invariant estimator, i.e. *convolutional* filter

$$y_{\text{est}} := w * x = Wx$$

If the noise is additive and independent, at each frequency

$$\text{DFT filter coefficient} = \frac{\text{image variance}}{\text{image variance } + \text{noise variance}}$$

Component preserved if signal energy is large with respect to noise

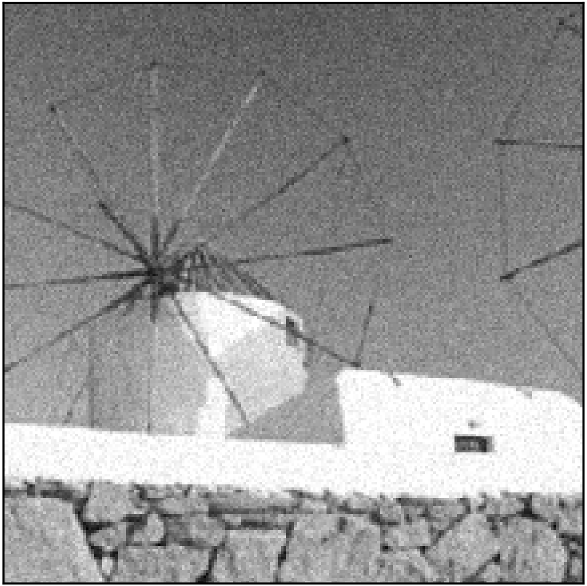
Convolutional neural network

Large number of convolutional filters combined with pointwise nonlinearity

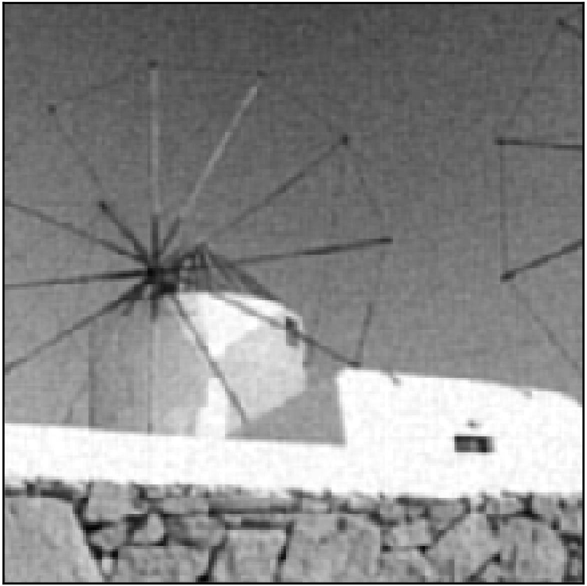
$$f(x) = W_L r(W_{L-1} \cdots r(W_2 r(W_1 x)))$$

$r(v)[i] := \max\{0, v[i]\}$

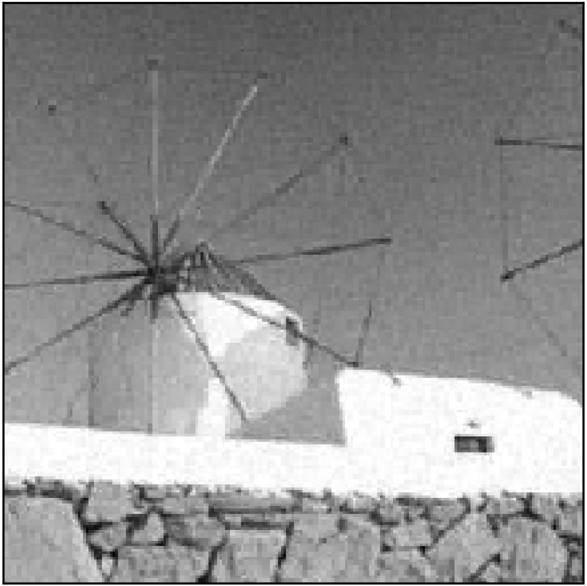
Noisy image



Wiener filtering

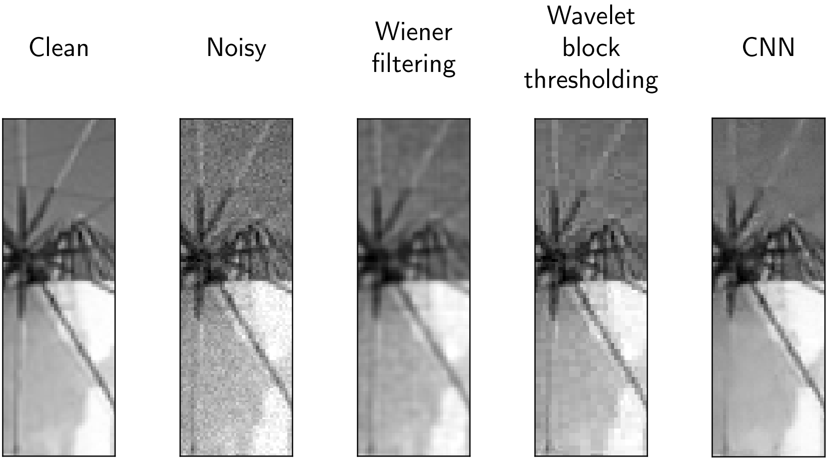


Wavelet block thresholding



Convolutional neural network

Comparison


Clean
Noisy
Wiener filtering
Wavelet block thresholding
CNN

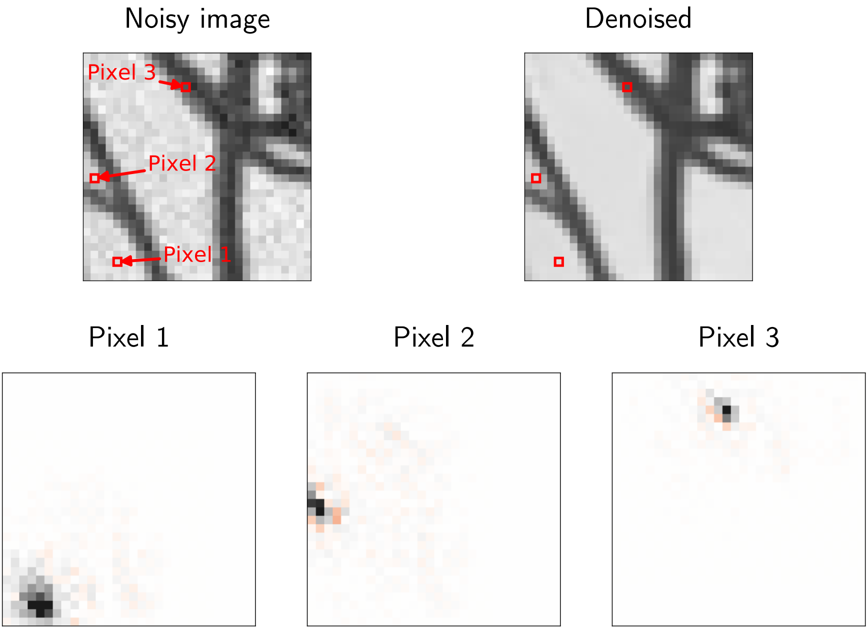
Jacobian

For fixed input x, matrix J_x such that

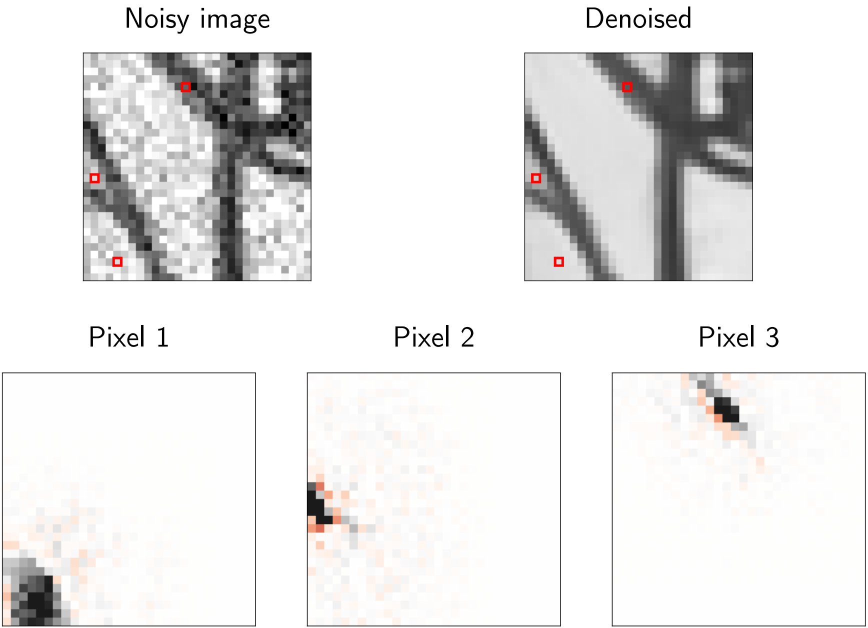
$$J_x x \approx f(x)$$

Rows can be interpreted as filters adapted to specific image

Low noise

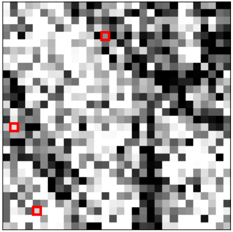

Noisy image
Denoised
Pixel 3
Pixel 2
Pixel 1
Pixel 1
Pixel 2
Pixel 3

Medium noise

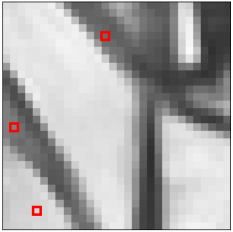


High noise

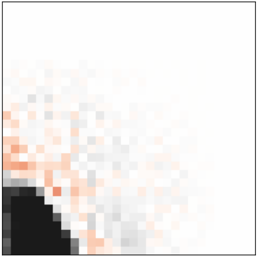
Noisy image



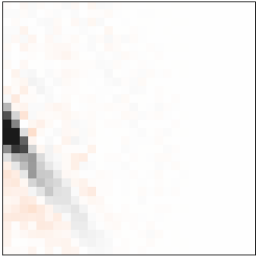
Denoised



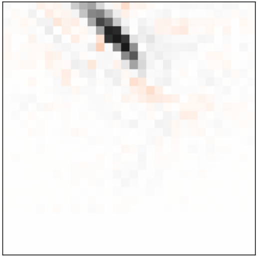
Pixel 1



Pixel 2



Pixel 3



Wiener filter

Frequency domain: Approximate projection onto low-pass 2D sinusoids

Problem: Same projection for each image

Blurs edges and other fine-scale features

Sparsity-based denoising

Sparsity-based methods implement adaptive projection:

1. Learn/design basis functions
2. Select sparse subset for each image/patch through thresholding/optimization
3. Project on span of sparse subset

Projection onto subspace that *depends on the input*

SVD analysis of Jacobian

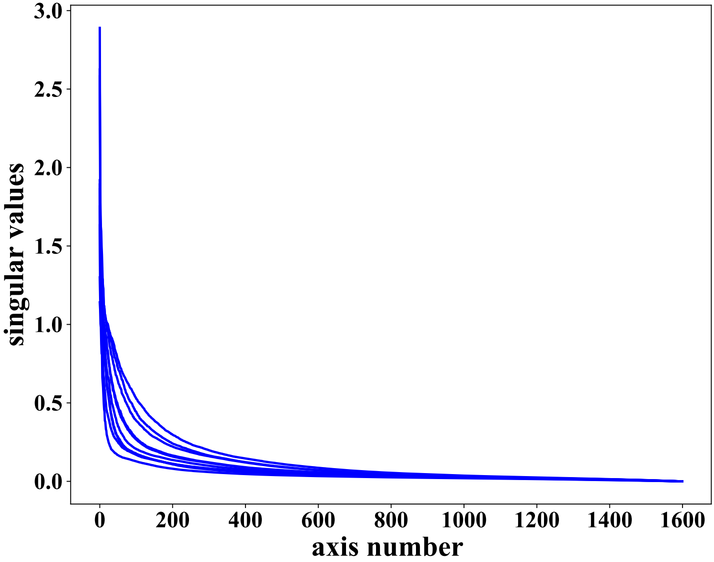
$$J_x = U\,S\,V^T$$

Empirical observations:

- Matrix is approximately symmetric $U \approx V$
- Matrix is approximately low-rank

$$f(x) \approx J_x x = \sum_j s_j u_j \left\langle v_j, x \right\rangle$$

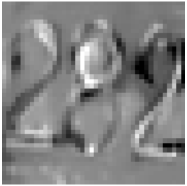
Singular values



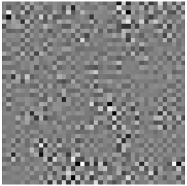
Singular vectors computed from noisy image

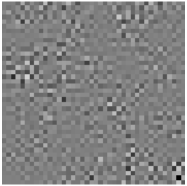
Clean image

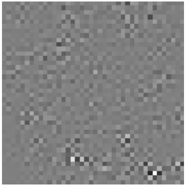
Large singular values



Small singular values







What have we learned?

Convolutional neural networks are nonlinear translation-invariant models

Analysis of the Jacobian shows that they learn to adapt to individual signals

Many of their properties are not well understood!