Nonconvex optimization

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

https://cims.nyu.edu/~cfgranda/pages/MTDS_spring19/index.html

Carlos Fernandez-Granda

Structured matrix factorization

Deep learning for image denoising

Low-rank bilinear model

$$y[i,j] \approx \sum_{l=1}^{r} a_l[i] b_l[j], \quad 1 \le i \le m, \ 1 \le j \le n$$

$$Y \approx AB, \qquad A \in \mathbb{R}^{m \times r}, \quad B \in \mathbb{R}^{r \times n}$$

(A, B) and $(AC, C^{-1}B)$ yield same model for any invertible C

Low-rank bilinear model

$$y[i,j] \approx \sum_{l=1}^{r} a_l[i] b_l[j], \quad 1 \leq i \leq m, \; 1 \leq j \leq n$$

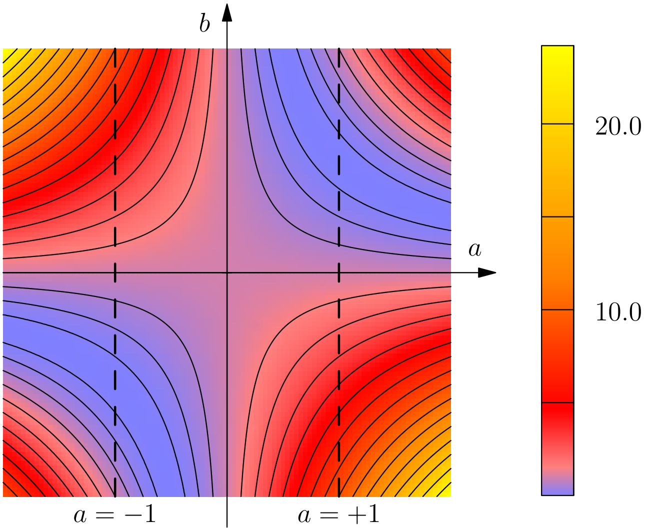
$$Y \approx AB, \qquad A \in \mathbb{R}^{m \times r}, \quad B \in \mathbb{R}^{r \times n}$$

(A, B) and $(AC, C^{-1}B)$ yield same model for any invertible C

Nonconvex cost function with nonconvex constraints:

$$\min_{A \in \mathbb{R}^{m \times r}, B \in \mathbb{R}^{r \times n}} ||Y - A\,B||_{\mathrm{F}} \qquad \text{subject to} \quad \left|\left|\tilde{A}_i\right|\right|_2 = 1, \qquad 1 \leq i \leq r$$

$(1 - ab)^2$



Singular value decomposition

Let USV^T be the SVD of Y

The truncated SVD $U_{:,1:r}S_{1:r,1:r}V_{:,1:r}^T$ is the best rank-r approximation

$$U_{:,1:r}S_{1:r,1:r}V_{:,1:r}^T = \arg\min_{\{\widetilde{A} \mid \text{rank}(\tilde{A})=r\}} \left|\left|A - \widetilde{A}\right|\right|_{\text{F}}$$

Singular value decomposition

Let USV^T be the SVD of Y

The truncated SVD $U_{:,1:r} S_{1:r,1:r} V_{:,1:r}^T$ is the best rank-r approximation

$$U_{:,1:r} S_{1:r,1:r} V_{:,1:r}^T = \arg\min_{\{\widetilde{A} \mid \text{rank}(\tilde{A})=r\}} \left|\left| A - \widetilde{A} \right|\right|_{\text{F}}$$

$A^* := U_{:,1:r}$ and $B := S_{1:r,1:r} V_{:,1:r}^T$ is a solution to

$$\min_{A \in \mathbb{R}^{m \times r}, B \in \mathbb{R}^{r \times n}} ||Y - A\,B||_{\text{F}} \qquad \text{subject to} \quad \left|\left| \tilde{A}_i \right|\right|_2 = 1, \qquad 1 \leq i \leq r$$

Nonnegative matrix factorization

Forcing the factors to be nonnegative factors can improve interpretability

$$X \approx A\, B, \quad A_{i,j} \geq 0, \; B_{i,j} \geq 0, \; \text{for all } i, j$$

Nonconvex optimization problem:

$$\begin{aligned} \text{minimize} \quad & \left|\left| X - \tilde{A}\, \tilde{B} \right|\right|_{\mathrm{F}}^2 \\ \text{subject to} \quad & \tilde{A}_{i,j} \geq 0, \\ & \tilde{B}_{i,j} \geq 0, \qquad \text{for all } i, j \end{aligned}$$

$\tilde{A} \in \mathbb{R}^{m \times r}$ and $\tilde{B} \in \mathbb{R}^{r \times n}$

Nonnegative matrix factorization

Forcing the factors to be nonnegative factors can improve interpretability

$$X \approx A\, B, \quad A_{i,j} \geq 0, \; B_{i,j} \geq 0, \; \text{for all } i, j$$

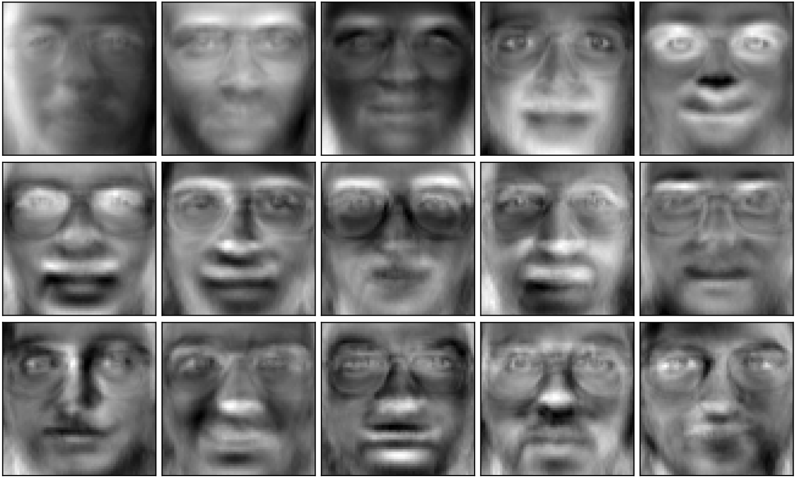
Nonconvex optimization problem:

$$\begin{aligned} \text{minimize} \quad & \left|\left| X - \tilde{A}\, \tilde{B} \right|\right|_{\mathrm{F}}^2 \\ \text{subject to} \quad & \tilde{A}_{i,j} \geq 0, \\ & \tilde{B}_{i,j} \geq 0, \qquad \text{for all } i, j \end{aligned}$$

$\tilde{A} \in \mathbb{R}^{m \times r}$ and $\tilde{B} \in \mathbb{R}^{r \times n}$

No longer solved by SVD

Faces dataset: SVD



Faces dataset: NMF

Topic modeling

$$A := \begin{pmatrix} 6 & 1 & 1 & 0 & 0 & 1 & 9 & 0 & 8 \\ 1 & 0 & 9 & 5 & 8 & 1 & 0 & 1 & 0 \\ 8 & 1 & 0 & 1 & 0 & 0 & 9 & 1 & 7 \\ 0 & 7 & 1 & 0 & 0 & 9 & 1 & 7 & 0 \\ 0 & 5 & 6 & 7 & 5 & 6 & 0 & 7 & 2 \\ 1 & 0 & 8 & 5 & 9 & 2 & 0 & 0 & 1 \end{pmatrix}$$

singer	GDP	senate	election	vote	stock	bass	market	band	Articles
6	1	1	0	0	1	9	0	8	a
1	0	9	5	8	1	0	1	0	b
8	1	0	1	0	0	9	1	7	c
0	7	1	0	0	9	1	7	0	d
0	5	6	7	5	6	0	7	2	e
1	0	8	5	9	2	0	0	1	f

SVD

$$A = USV^T = U \begin{bmatrix} 23.64 & 0 & 0 & 0 & & \\ 0 & 18.82 & 0 & 0 & 0 & 0 \\ 0 & 0 & 14.23 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.63 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.03 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.36 \end{bmatrix} V^T$$

Left singular vectors

	a	b	c	d	e	f
U_1 =	$(-0.24$	-0.47	-0.24	-0.32	-0.58	$-0.47)$
U_2 =	$(\ 0.64$	-0.23	0.67	-0.03	-0.18	$-0.21)$
U_3 =	$(-0.08$	-0.39	-0.08	0.77	0.28	$-0.40)$

Right singular vectors

	singer	GDP	senate	election	vote	stock	bass	market	band
$V_1 =$	-0.18	-0.24	-0.51	-0.38	-0.46	-0.34	-0.2	-0.3	-0.22
$V_2 =$	0.47	0.01	-0.22	-0.15	-0.25	-0.07	0.63	-0.05	0.49
$V_3 =$	-0.13	0.47	-0.3	-0.14	-0.37	0.52	-0.04	0.49	-0.07

Nonnegative matrix factorization

$$X \approx W\ H$$

$$W_{i,j} \geq 0,\ H_{i,j} \geq 0,\ \text{for all } i, j$$

Right nonnegative factors

	singer	GDP	senate	election	vote	stock	bass	market	band
$H_1 =$	(0.34	0	3.73	2.54	3.67	0.52	0	0.35	0.35)
$H_2 =$	(0	2.21	0.21	0.45	0	2.64	0.21	2.43	0.22)
$H_3 =$	(3.22	0.37	0.19	0.2	0	0.12	4.13	0.13	3.43)

Interpretations:

- Count basis function: Counts for each doc are weighted sum of H_1, H_2, H_3
- Coefficients: They cluster words into politics, music and economics

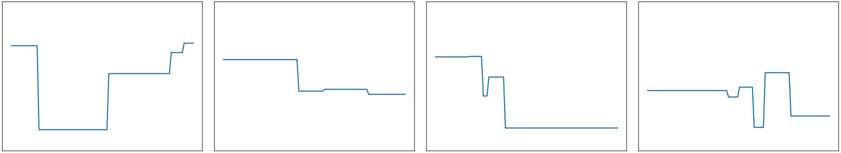
Left nonnegative factors

		a	b	c	d	e	f
W_1	=	(0.03	2.23	0	0	1.59	2.24)
W_2	=	(0.1	0	0.08	3.13	2.32	0)
W_3	=	(2.13	0	2.22	0	0	0.03)

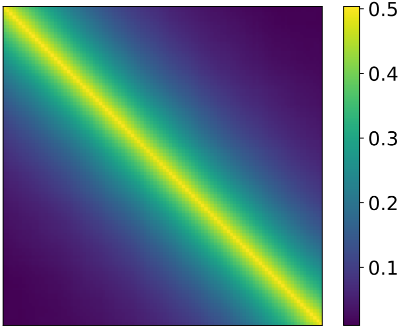
Interpretations:

- Count basis function: Counts for each word are weighted sum of W_1, W_2, W_3
- Coefficients: They cluster docs into politics, music and economics

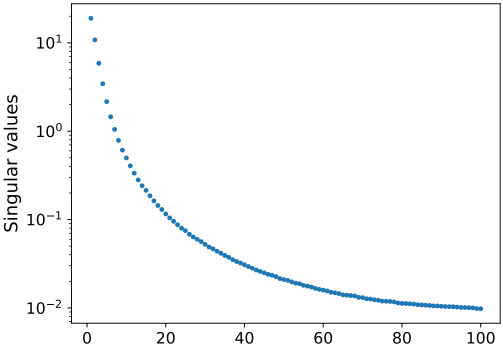
PCA of translation-invariant signals



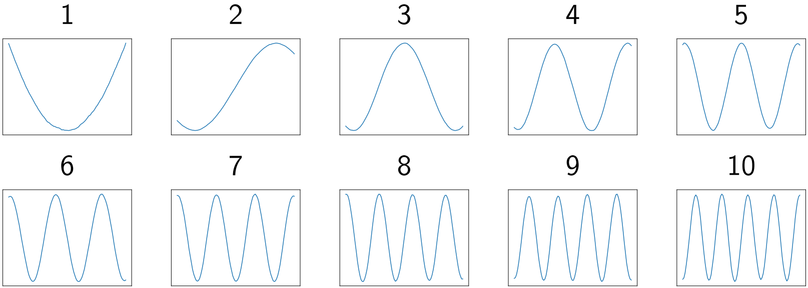
Sample covariance matrix



Singular values



Principal directions



Principal directions

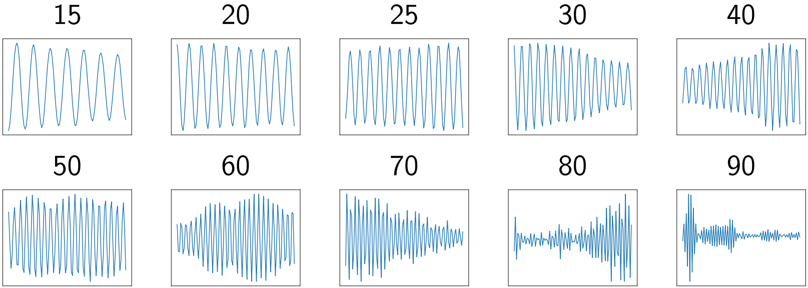
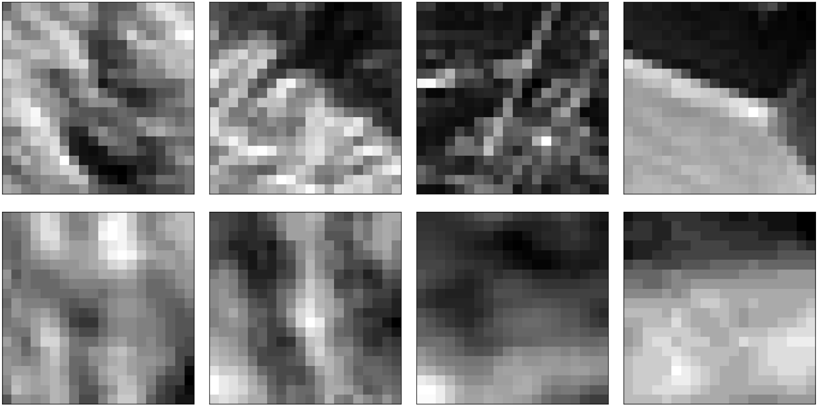
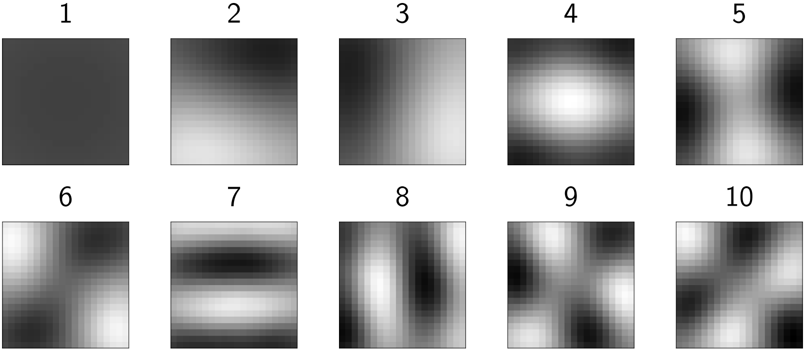


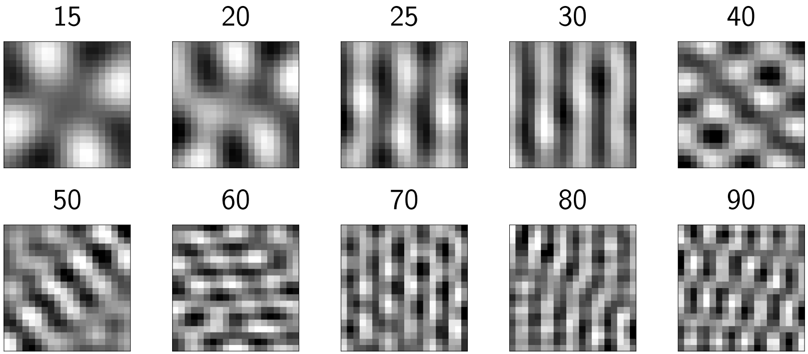
Image patches



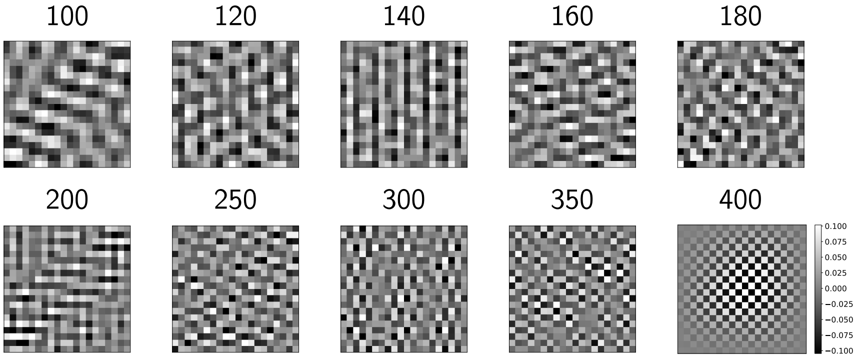
Principal directions



Principal directions



Principal directions



Sparse coding

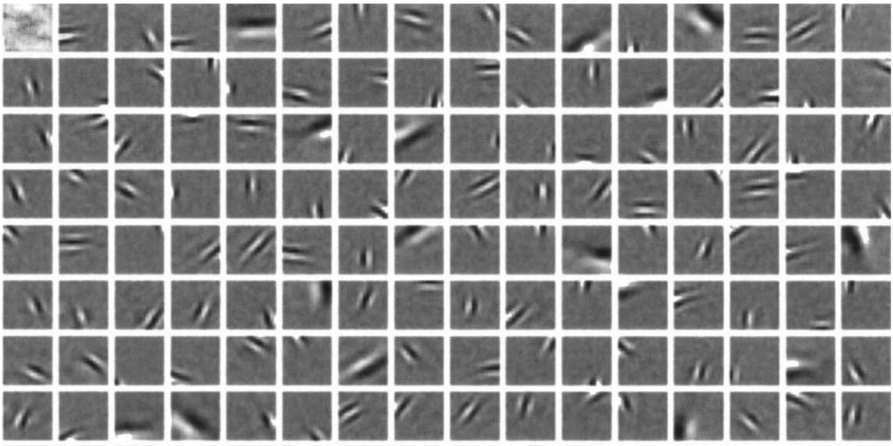
Aim: Find basis functions that represent data parsimoniously

Equivalently coefficients/codes B should be sparse

ℓ_1-norm regularization is used to enforce sparsity

$$\begin{aligned} \text{minimize} \quad & \left|\left|X - \tilde{A}\,\tilde{B}\right|\right|_2^2 + \lambda \sum_{i=1}^{r} \left|\left|\tilde{B}_i\right|\right|_1 \\ \text{subject to} \quad & \left|\left|\tilde{A}_i\right|\right|_2 = 1, \qquad 1 \leq i \leq r \end{aligned}$$

Sparse coding (Olshausen and Field 1996)



Structured matrix factorization

Deep learning for image denoising

Image denoising

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

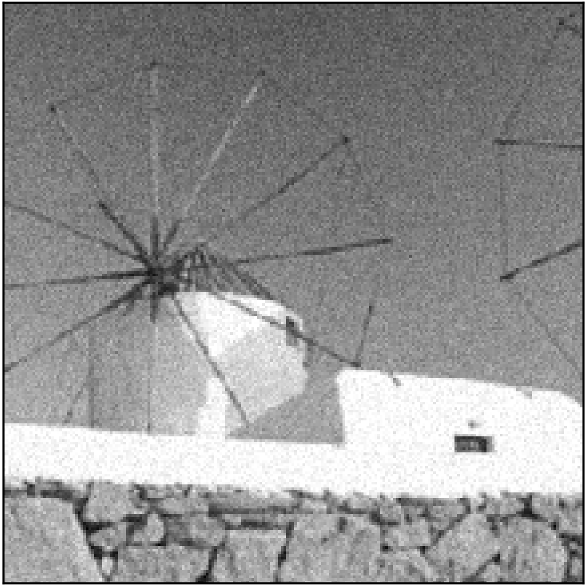
Convolutional neural networks

Large number of learned filters combined with pointwise nonlinearity

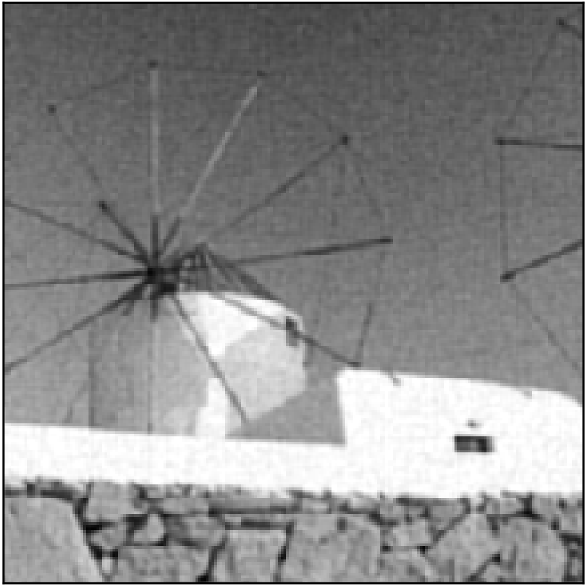
$$\min_{W_1,\ldots,W_L} \sum_{j=1}^{n} \left|\left|\vec{x}^{[j]} - W_L \rho\left(W_{L-1}\rho\left(\ldots W_2 \rho\left(W_1 \vec{y}^{\,[j]}\right)\right)\right)\right|\right|_2^2$$

$$\rho(\vec{v})[i] := \max\left\{0, \vec{v}[i]\right\}$$

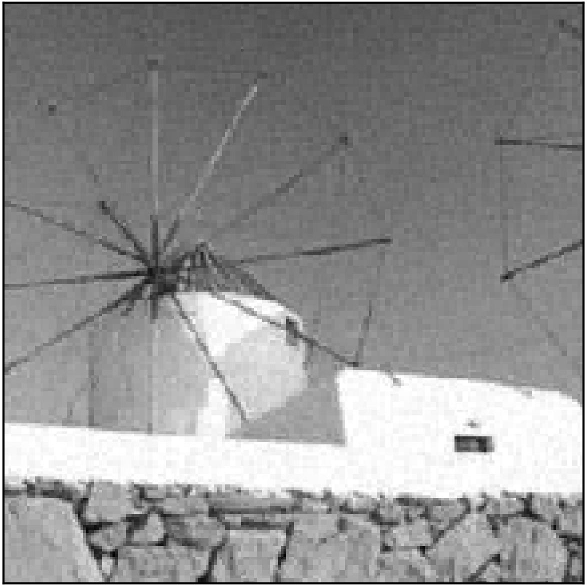
Noisy image



Wiener filtering

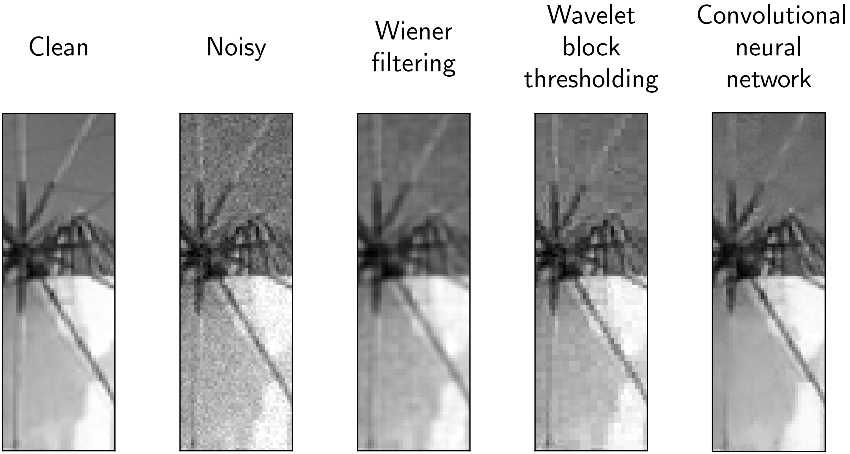


Wavelet block thresholding



Convolutional neural network

Comparison


Clean
Noisy
Wiener filtering
Wavelet block thresholding
Convolutional neural network

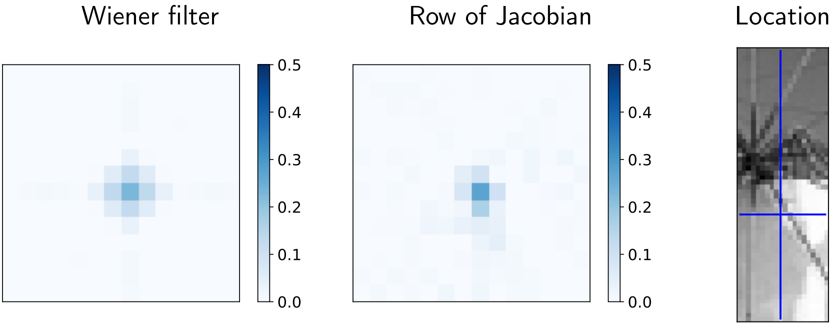
Jacobian

For fixed input, matrix J such that

$$J\vec{y} = W_L \rho \left(W_{L-1} \rho \left(\ldots W_2 \rho \left(W_1 \vec{y} \right) \right) \right)$$

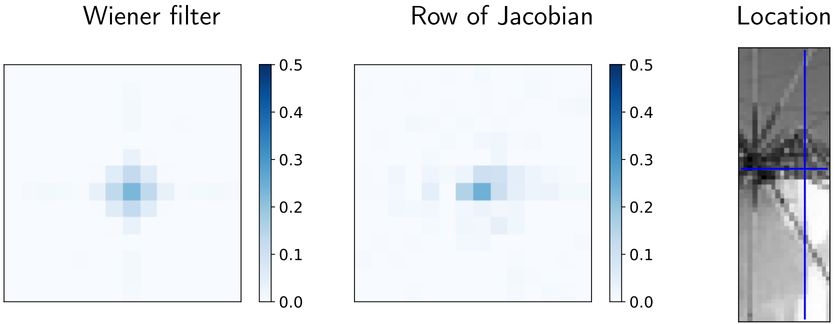
Rows can be interpreted as filters adapted to specific image

Jacobian



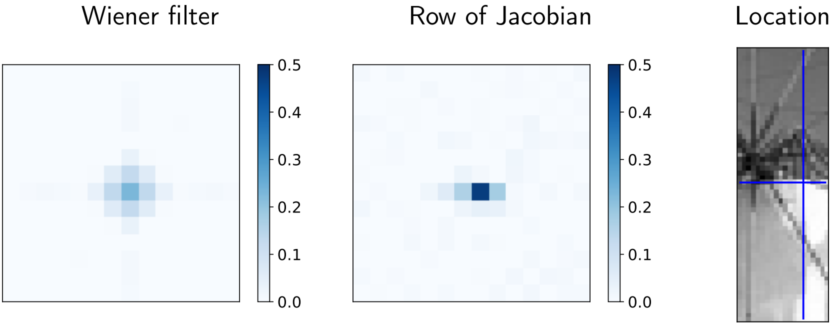
Wiener filter
0.5
0.4
0.3
0.2
0.1
0.0
Row of Jacobian
0.5
0.4
0.3
0.2
0.1
0.0
Location

Jacobian

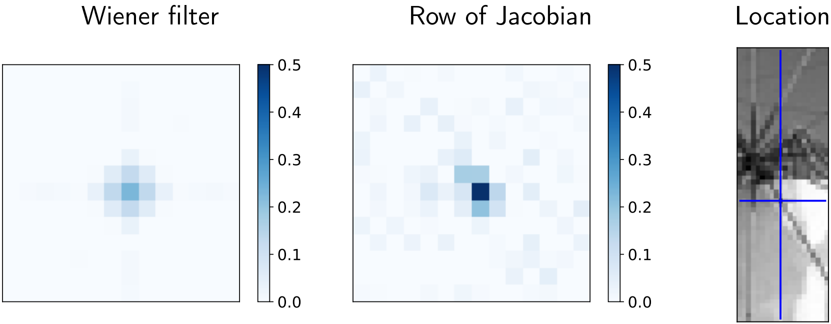


Wiener filter
0.5
0.4
0.3
0.2
0.1
0.0
Row of Jacobian
0.5
0.4
0.3
0.2
0.1
0.0
Location

Jacobian


Wiener filter
Row of Jacobian
Location
0.5
0.4
0.3
0.2
0.1
0.0
0.5
0.4
0.3
0.2
0.1
0.0

Jacobian


Wiener filter
Row of Jacobian
Location
0.5
0.4
0.3
0.2
0.1
0.0
0.5
0.4
0.3
0.2
0.1
0.0