PGMO Lecture: Vision, Learning and Optimization

8. Learning

Thomas Pock

Institute of Computer Graphics and Vision

February 12, 2020
Introduction

So far we have considered mainly convex models based on the total variation, which however only served as a crude approximation to the true image statistics.

In this chapter we will discuss methods to learn better variational models from data.

We will start by learning just the regularization parameter but then will also learn filters and potential functions.

Finally, we will also consider deep-learning inspired architectures that achieve state-of-the-art performance.
Overview

Parameter learning in variational models

The Fields of Experts model

Early stopping

Total Deep Variation

Learning with graphical models
Learning regularization parameters

- In [Kunisch, P. '12] we considered a weighted sum of $\ell_1$ regularizers:

$$\mathcal{R}(u) = \sum_{k=1}^{N_K} \vartheta_k \| K_k u \|_1 = \sum_{k=1}^{N_K} \sum_{i,j} \vartheta_k |(K_k u)_{i,j}|,$$

where $K_k$ are linear operators and $\vartheta_k \geq 0$ are the regularization weights.

- Can be seen as a generalization of the total variation

- Usually, we restrict the linear operators to small convolution kernels $f_k$ with the property that $K_k u \Leftrightarrow f_k \ast u$

- From JPEG compression, it is known that images have a sparse representation in terms of DCT basis functions.

The 24 DCT5 filters $f_k$
Bilevel optimization

- How can we choose optimal weights for the different operators?
Bilevel optimization

- How can we choose optimal weights for the different operators?
- In machine learning a popular approach is empirical risk minimization adopting a loss function.
- We assume we have given training data \((f_s, g_s)_{s=1}^S\) consisting of noisy observations \(f_s\) and ground truth reconstructions \(g_s\).
- Applying this idea to our image reconstruction problems leads to a bilevel optimization problem [Kunisch, P. '12]

\[
\begin{align*}
\min_{\vartheta \geq 0} \quad & \frac{1}{2} \sum_{s=1}^S \| u_s(\vartheta) - g_s \|_2^2 \\
\text{s.t.} \quad & u_s(\vartheta) = \arg\min_u \sum_{k=1}^{N_K} \vartheta_k \| K_k u \|_1 + \frac{1}{2} \| u - f_s \|_2^2.
\end{align*}
\]
Bilevel optimization

- How can we choose optimal weights for the different operators?
- In machine learning a popular approach is empirical risk minimization adopting a loss function.
- We assume we have given training data \((f_s, g_s)_{s=1}^S\) consisting of noisy observations \(f_s\) and ground truth reconstructions \(g_s\).
- Applying this idea to our image reconstruction problems leads to a bilevel optimization problem [Kunisch, P.’12]

\[
\begin{aligned}
\min_{\vartheta \geq 0} \frac{1}{2} \sum_{s=1}^S \| u_s(\vartheta) - g_s \|_2^2 \\
\text{s.t.} \quad u_s(\vartheta) = \arg\min_u \sum_{k=1}^{N_K} \vartheta_k \| K_k u \|_1 + \frac{1}{2} \| u - f_s \|_2^2.
\end{aligned}
\]

- Interpretation: We try to find parameters \(\vartheta\) such that the minimizers of the variational model minimizes the loss function
- Closely related approaches: [Haber and Tenorio, ’02], [Samuel and Tappen ’09], [Peyré and Fadili ’11], [De Los Reyes and Schönlieb ’12], ...
- We developed semi-smooth Newton algorithms to solve the bilevel optimization problem for the optimal parameter vector \(\vartheta\).
Example: Image denoising

Original image

Noisy image
Example: Image denoising

Original image

TV denoised
Example: Image denoising

Original image

DCT5
Overview

Parameter learning in variational models

The Fields of Experts model

Early stopping

Total Deep Variation

Learning with graphical models
The Fields of Experts model

- Recall that the $|x|$ function does not provide a very accurate match to the marginal distributions of zero-mean filters.
- A much better match is obtained by the negative log student’s-t distribution $\log(1 + |x|^2/\mu^2)$ [Huang and Mumford ’99].
- Let us consider the following nonconvex model [Roth, Black ’09], [Samuel, Tappen ’09], called the “Fields of Experts” model:
  \[
  R(u) = \sum_{k=1}^{N_K} \sum_{i,j} \rho_k((K_k u)_{i,j}),
  \]
  where $K_k$ are again linear operators implementing 2D convolutions with small filters $f_k$, that is $f_k * u \Leftrightarrow K_k u$, and $\rho_k(t) = \alpha_k \log(1 + |t|^2)$.
- In contrast to the previous model, also the filters are learned.
We again consider training data consisting of clean and noisy images \((f_s, g_s)_{s=1}^S\).

We again used a bilevel optimization approach to learn the filters and functions \(\vartheta = (f_k, \alpha_k)_{k=1}^{N_K}\)

\[
\begin{cases}
\min_{\vartheta} \quad L(\vartheta) = \frac{1}{2} \sum_{s=1}^{S} \|u_s(\vartheta) - g_s\|^2 + R(\vartheta) \\
\text{s.t.} 

\begin{align*}
u_s(\vartheta) &= \arg \min_u \sum_{k=1}^{N_K} \sum_{i,j} \rho_k((K_k u)_{i,j}) + \frac{1}{2} \|u - f_s\|_2^2, \\
\end{align*}
\end{cases}
\]

where \(R(\vartheta)\) is a regularization term for the learned parameters, for example one could consider the constraints

\[1^T f_k = 0, \quad \alpha_k \geq 0, \quad k = 1, \ldots, K\]
In order to compute gradients of the loss function with respect to $\vartheta$, we replace the lower-level optimization problem by its first-order optimality condition (assuming $s = 1$ and dropping the index):

$$\sum_{k=1}^{N_K} K_k^* \phi_k(K_k u) + u - f = 0, \quad \phi_k(y) = \text{diag}(\rho'_k(y_1), \ldots, \rho'_k(y_n));$$

where $K_k^*$ denotes the adjoint filter and consider the Lagrangian functional

$$\mathcal{L}(u, \vartheta, \lambda) = \| u - g \|^2 + R(\vartheta) + \left( \sum_{k=1}^{N_K} K_k^* \phi_k(K_k u) + u - f \right)^T p,$$

where $p$ is a vector of Lagrange multipliers.

Assuming the existence of a regular local minimum in $(u, \vartheta)$, we can invoke the classical **Lagrange multiplier theorem**, which guarantees the existence of multipliers $p$ such that:

$$\left( \sum_{k=1}^{N_K} K_k^* D\phi_k(K_k u)K_k + I \right)p + u - g \quad \begin{cases} D_\vartheta R(\vartheta) + (D_\vartheta \sum_{k=1}^{N_K} K_k^* \phi_k(K_k u))p \\ \sum_{k=1}^{N_K} K_k^* \phi_k(K_k u) + u - f \end{cases} = 0.$$
Implicit differentiation

For fixed $\vartheta$, the system can be reduced by first solving the lower level problem (last equation) for $u^*$, that is

$$\sum_{k=1}^{N_K} K_k^* \phi_k(K_k u^*) + u^* - f = 0,$$

then one can solve for $p^*$ by solving the linear system

$$p^* = \left(\sum_{k=1}^{N_K} K_k^* D\phi_k(K_k u^*)K_k + I\right)^{-1}(g - u^*),$$

and finally the gradient of the loss function with respect to $\vartheta$ is given by

$$\partial_\vartheta L(\vartheta) = D_\vartheta R(\vartheta) + (D_\vartheta \sum_{k=1}^{N_K} K_k^* \phi_k(K_k u^*))\left(\sum_{k=1}^{N_K} K_k^* D\phi_k(K_k u^*)K_k + I\right)^{-1}(g - u^*),$$

which is nothing else then implicit differentiation.

The loss function can then be minimized using any gradient-based optimization algorithm.
The learned filters and functions

- In [Chen, Ranftl, P. '14] we learned 80 filters of size $9 \times 9$ plus function parameters → 6480 parameters on a database of $\sim 200$ images using bilevel optimization
- ... two weeks later ...
The learned filters and functions

In [Chen, Ranftl, P. ’14] we learned 80 filters of size $9 \times 9$ plus function parameters $\rightarrow 6480$ parameters on a database of $\sim 200$ images using bilevel optimization

... two weeks later ...
Evaluation

- Comparison with five state-of-the-art approaches: K-SVD [Elad and Aharon '06], FoE [Q. Gao and Roth '12], BM3D [Dabov et al. '07], GMM [D. Zoran et al. '12], LSSC [Mairal et al. '09]

- We report the average PSNR on 68 images of the Berkeley image data base [Chen, P. 14]

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>KSVD</th>
<th>FoE</th>
<th>BM3D</th>
<th>GMM</th>
<th>LSSC</th>
<th>BL7x7</th>
<th>BL9x9</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>30.87</td>
<td>30.99</td>
<td>31.08</td>
<td>31.19</td>
<td><strong>31.27</strong></td>
<td>31.18</td>
<td>31.22</td>
</tr>
<tr>
<td>50</td>
<td>25.17</td>
<td>25.35</td>
<td>25.62</td>
<td>25.67</td>
<td>25.72</td>
<td>25.70</td>
<td><strong>25.76</strong></td>
</tr>
</tbody>
</table>

- Performs as well as state-of-the-art
Denoising results for $\sigma = 25$
Denoising results for $\sigma = 25$
Denoising results for $\sigma = 25$

Original image

FoE prior
foe.ipynb
Computing gradients

- Bilevel optimization is heavily time consuming since for implicit differentiation we need to:
  - Solve the lower problems exactly
  - Invert the Hessian of the lower level problem
- Performance strongly depends on the error of the stationary point $u^*$
Computing gradients

- Bilevel optimization is heavily time consuming since for implicit differentiation we need to:
  - Solve the lower problems exactly
  - Invert the Hessian of the lower level problem
- Performance strongly depends on the error of the stationary point $u^*$
- Alternative: Unroll the steps of an iterative algorithm
- The bilevel optimization problem becomes

$$\begin{aligned}
\min_{\vartheta} & \frac{1}{2} \sum_{s=1}^{S} \left\| u_s^T(\vartheta) - g_s \right\|^2 \\
\text{s.t.} & \quad u_s^{t+1} = u_s^t - \tau^t \left( \sum_{k=1}^{N_K} K^*_k \phi_k(K_k u_s) + (u_s^t - f) \right), \\
& \quad t = 0 \ldots T - 1
\end{aligned}$$

- We can compute the exact gradient with respect to the model parameters using the backpropagation algorithm.
- It turns out that taking only a finite number of steps works even better....
Overview

Parameter learning in variational models

The Fields of Experts model

Early stopping

Total Deep Variation

Learning with graphical models
Motivation

As a motivating example, let us step back to a smooth approximation of the TV–$L^2$ (ROF) model

$$E_\epsilon[u, \nu] = \nu \sum_{i,j} \sqrt{|(Du)_{i,j}|^2 + \epsilon^2} + \frac{1}{2} \|u - g\|_2^2,$$

and for various weighting parameters $\nu$ the gradient flow with step size $\tau$

$$u_{s+1} = u_s - \tau \left( \nu D^* \left( \frac{Du_s}{\sqrt{|(Du_s)|^2 + \epsilon^2}} \right) + u_s - g \right)$$
TV-$L^2$ classical
TV–$L^2$ classical
TV–$L^2$ classical

PSNR = 20.53
TV–$L^2$ classical

$u_0$, $u_g$, $u_\infty(\nu = 0.002)$

PSNR = 24.26
TV–$L^2$ classical

PSNR = 26.59
TV–$L^2$ classical

$u_0$, $u_g$, $u_\infty(\nu = 0.01)$

PSNR = 19.61
TV–$L^2$ with early stopping

$u_\infty(\nu = 0.01)$

$u_64(\nu = 0.01)$

$u_0$

$u_g$

PSNR = 25.23
TV–$L^2$ with early stopping

$\nu = 0.01$

$u_0$

$u_{256}(\nu = 0.01)$

$u_{\infty}(\nu = 0.01)$

$u_g$

PSNR = 25.55
TV–$L^2$ with early stopping

\[ u_\infty(\nu = 0.01) \]

\[ u_{128}(\nu = 0.01) \]

\[ u_g \]

PSNR = 27.19

\[ \nu = 0.01 \]
Parameter search

The best performance (red cross) is achieved for early stopping.

When minimizing the energy exactly the solution is “overfitted” to the variational model and gives inferior results.
A gradient flow perspective

Let $u \in \mathbb{R}^n$ be a data vector, then the variational energy is

$$\mathcal{E}[u] = \mathcal{D}[u] + \mathcal{R}[u].$$
A gradient flow perspective

Let $u \in \mathbb{R}^n$ be a data vector, then the variational energy is

$$
\mathcal{E}[u] = \mathcal{D}[u] + \mathcal{R}[u].
$$

Fields-of-Experts regularization:

$$
\mathcal{R}[u] = \sum_{i=1}^{m} \sum_{k=1}^{N_k} \rho_k((K_k u)_i)
$$

with $K_k \in \mathbb{R}^{m \times n}$ and associated nonlinear functions $\rho_k : \mathbb{R} \to \mathbb{R}$
A gradient flow perspective

Let $u \in \mathbb{R}^n$ be a data vector, then the variational energy is

$$\mathcal{E}[u] = \mathcal{D}[u] + \mathcal{R}[u].$$

Fields-of-Experts regularization:

$$\mathcal{R}[u] = \sum_{i=1}^{m} \sum_{k=1}^{N_K} \rho_k((K_k u)_i)$$

with $K_k \in \mathbb{R}^{m \times n}$ and associated nonlinear functions $\rho_k : \mathbb{R} \to \mathbb{R}$ data fidelity:

$$\mathcal{D}[u] = \frac{1}{2} \|Au - b\|_2^2$$

$A \in \mathbb{R}^{l \times n}$ and $b \in \mathbb{R}^l$ fixed
A gradient flow perspective

Gradient flow of energy $\mathcal{E}$ for a time $t \in (0, T)$:

$$\dot{x}(t) = f(\tilde{x}(t), (K_k, \Phi_k)_{k=1}^{N_K}) = -D\mathcal{E}[\tilde{x}(t)]$$

$$= -A^*(A\tilde{x}(t) - b) - \sum_{k=1}^{N_K} K_k^* \Phi_k(K_k \tilde{x}(t)),$$

$$\tilde{x}(0) = x_0,$$

with $\tilde{x} \in C^1([0, T], \mathbb{R}^n)$, $T \in \mathbb{R}^+$ and the functions $\Phi_k \in V^s$ are given by

$$(y_1, \ldots, y_m)^\top \mapsto (\rho'_k(y_1), \ldots, \rho'_k(y_m))^\top,$$

with $V^s$ finite dimensional subspace of $C^s(\mathbb{R}^m, \mathbb{R}^m)$
Optimal control problem

Reparametrization: $x(t) = \tilde{x}(t^T)$
Optimal control problem

Reparametrization: $x(t) = \tilde{x}(t^T)$

optimal control problem:

$$\min_{T \in \mathbb{R}, K_k \in \mathbb{R}^{m \times n}, \Phi_k \in \mathcal{V}} J(T, (K_k, \Phi_k)^{N_K}_{k=1})$$
Reparametrization: $x(t) = \tilde{x}(t^T)$

optimal control problem:

$$\min_{T \in \mathbb{R}, K_k \in \mathbb{R}^{m \times n}, \Phi_k \in \mathcal{V}} J(T, (K_k, \Phi_k)_{k=1}^{N_k})$$

cost functional:

$$J(T, (K_k, \Phi_k)_{k=1}^{N_k}) := \frac{1}{2} \|x(1) - x_g\|^2_2$$
Optimal control problem

Reparametrization: \( x(t) = \tilde{x}(t^T) \)

optimal control problem:

\[
\min_{T \in \mathbb{R}, K_k \in \mathbb{R}^{m \times n}, \Phi_k \in V^s} J(T, (K_k, \Phi_k)_{k=1}^{N_K})
\]

cost functional:

\[
J(T, (K_k, \Phi_k)_{k=1}^{N_K}) := \frac{1}{2} \|x(1) - x_g\|_2^2
\]

constraints:

\[
0 \leq T \leq T_{\text{max}}, \quad \alpha(K_k) \leq 1, \quad \beta(\Phi_k) \leq 1, \quad K_k \mathbf{1} = 0 \in \mathbb{R}^m,
\]

\( \alpha : \mathbb{R}^{m \times n} \to \mathbb{R}_0^+ \) and \( \beta : V^s \to \mathbb{R}_0^+ \) are continuously differentiable, coercive functions
Optimal control problem

Reparametrization: \( x(t) = \tilde{x}(t^T) \)

optimal control problem:

\[
\min_{T \in \mathbb{R}, K_k \in \mathbb{R}^{m \times n}, \Phi_k \in \mathcal{V}_s} J(T, (K_k, \Phi_k)_{k=1}^{N_k})
\]

cost functional:

\[
J(T, (K_k, \Phi_k)_{k=1}^{N_k}) := \frac{1}{2} \| x(1) - x_g \|^2_2
\]

constraints:

\[
0 \leq T \leq T_{\text{max}}, \quad \alpha(K_k) \leq 1, \quad \beta(\Phi_k) \leq 1, \quad K_k \mathbf{1} = 0 \in \mathbb{R}^m,
\]

\( \alpha: \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^+_0 \) and \( \beta: \mathcal{V}_s \rightarrow \mathbb{R}^+_0 \) are continuously differentiable, coercive functions

transformed state equation for \( t \in (0, 1) \):

\[
\dot{x}(t) = T f(x(t), (K_k, \Phi_k)_{k=1}^{N_k}), \quad x(0) = x_0
\]
First order condition

Theorem (First order necessary condition)

Let $s \geq 1$. For each stationary point $(\bar{T}, (\bar{K}_k, \bar{\Phi}_k)_{k=1}^{N_k})$ of $J$ with state $\bar{x}$

$$\int_0^1 \langle \bar{p}(t), \dot{\bar{x}}(t) \rangle \, dt = 0$$

holds true. Here, $\bar{p} \in C^1([0, 1], \mathbb{R}^n)$ denotes the adjoint state of $\bar{x}$, which is given as the solution to the ODE

$$\dot{\bar{p}}(t) = \sum_{k=1}^{N_k} \bar{K}_k^* D\bar{\Phi}_k(\bar{K}_k \bar{x}(t)) \bar{K}_k \bar{p}(t) + A^* A\bar{p}(t)$$

with terminal condition $\bar{p}(1) = x_g - \bar{x}(1)$. 
Time discretization
Time discretization

\[ x(x) \subset (\infty) \times (T) \times (0) = x(0) + S_{24/67} \]
Let $S \geq 2$ be a fixed depth and $\Theta = ((K_k, \Phi_k)_{k=1}^{N_K})$.

**State equation:**

Explicit forward Euler:

$$x_{s+1} = x_s + \frac{T}{S} f(x_s, \Theta) \quad s = 0, \ldots, S - 1$$
Time discretization

Let \( S \geq 2 \) be a fixed depth and \( \Theta = ((K_k, \Phi_k)_{k=1}^{N_k}) \).

**State equation:**

Explicit forward Euler:

\[
x_{s+1} = x_s + \frac{T}{S} f(x_s, \Theta) \quad s = 0, \ldots, S - 1
\]

Explicit 2\textsuperscript{nd}–order Heun:

\[
x_{s+1} = x_s + \frac{T}{2S} \left( f(x_s, \Theta) + f \left( x_s + \frac{T}{S} f(x_s, \Theta) \right) \right)
\]
Time discretization

Let \( S \geq 2 \) be a fixed depth and \( \Theta = ((K_k, \Phi_k)_{k=1}^{N_K}) \).

**Adjoint state equation:**

\[
\dot{p}(t) = g(x(t), p(t), \Theta) = \sum_{k=1}^{N_K} K_k^* D\Phi_k(K_k x(t)) K_k p(t) + A^* A p(t)
\]

Explicit forward Euler:

\[
p_{s} = p_{s+1} - \frac{T}{S} g(x_{s+1}, p_{s+1}, \Theta)
\]

Explicit 2nd–order Heun:

\[
p_{s} = p_{s+1} - \frac{T}{2S} \left( g(x_{s+1}, p_{s+1}, \Theta) + g\left(x_s, p_{s+1} - \frac{T}{S} g(x_{s+1}, p_{s+1}, \Theta), \Theta\right) \right)
\]
For a given training set \((x^i_0, x^i_g)_{i \in I}\), the loss for a batch \(B \subset I\) is given by

\[
J_B(T, (K_k, w_k)_{k=1}^{N_k}) := \frac{1}{2|B|} \sum_{i \in B} ||x^i_S - x^i_g||^2_2
\]
Learning

For a given training set $((x^i_0, x^i_g)_{i \in I}$, the loss for a batch $B \subset I$ is given by

$$J_B(T, (K_k, w_k)_{k=1}^{N_k}) := \frac{1}{2|B|} \sum_{i \in B} \|x^i_s - x^i_g\|_2^2$$

subject to

$$K_k \in \mathcal{K} = \{ K \in \mathbb{R}^{m \times n} : \alpha(K) \leq 1, K1 = 0 \},$$

$$w_k \in \mathcal{W} = \{ w \in \mathbb{R}^{N_w} : \beta(w) \leq 1 \}.$$

Here

$$\rho'(x) = \sum_{j=1}^{N_w} w_j \psi_j(x)$$

with quadratic B–spline basis functions $\psi_j(x) \in C^1(\mathbb{R})$.  
Image restoration

image denoising

\[ b = x_g + n \]

where

\[ n \sim \mathcal{N}(0, \sigma^2 I). \]

Default: \( \sigma = 0.1 \)
Image restoration

image denoising

\[ b = x_g + n \]

where

\[ n \sim \mathcal{N}(0, \sigma^2 I). \]

Default: \( \sigma = 0.1 \)

image deblurring

\[ b = A_\tau x_g + n \]

with blur operator \( A_\tau \)

\[ (x, y) \mapsto \frac{1}{\sqrt{2\pi\tau^2}} \exp \left( -\frac{x^2 + y^2}{2\tau^2} \right). \]

Default: \( \tau = 1.5 \) and \( \sigma = 0.01 \)
Regularization parameters - denoising
Regularization parameters - deblurring
Early stopping

image denoising

\[ S \mapsto \text{PSNR}(x^S_i, x^g_i) \quad i \in \hat{I} \]

Inference speed: 5.694ms for \( S = 20 \)
Early stopping

**image denoising**

\[ S \mapsto \text{PSNR}(x^S_i, x^g_i) \quad i \in \hat{I} \]

- Euler
- Heun

**image deblurring**

\[ S \mapsto \text{PSNR}(x^S_i, x^g_i) \quad i \in \hat{I} \]

- Euler
- Heun

Inference speed: 5.694ms for \( S = 20 \)

Inference speed: 8.687ms for \( S = 20 \)
First order condition

image denoising

\[ T \mapsto J_{(j)}(T, (K_k, w_k)_{k=1}^{N_k}) \]

\[ T \mapsto -\frac{1}{T} \int_0^T \langle p_T, \dot{x}_T \rangle \, dt \]
First order condition

**image denoising**

\[ T \mapsto J_{(i)}(T, (K_k, \bar{w}_k)_{k=1}^N) \]

\[ T \mapsto -\frac{1}{T} \int_0^T \langle p_T, \dot{x}_T \rangle dt \]

**image deblurring**

\[ T \mapsto J_{(i)}(T, (K_k, \bar{w}_k)_{k=1}^N) \]

\[ T \mapsto -\frac{1}{T} \int_0^T \langle p_T, \dot{x}_T \rangle dt \]
First order condition

\[ S = 0 \quad T = 0 \]

PSNR=20.00

PSNR=26.68
First order condition

\[
S = 10 \cdot T = \frac{T}{2}
\]

PSNR = 26.36

PSNR = 28.87
First order condition

$S = 20 \cdot T = \overline{T}$

PSNR = 29.68

PSNR = 29.52
First order condition

\[ S = 30 \]
\[ T = \frac{3T}{2} \]

PSNR = 29.15

PSNR = 25.34
First order condition

PSNR=27.90

\[ S = 1000 \quad T = 50\overline{T} \]

PSNR=5.13
Does the stopping time depend on the noise level $\sigma$?
Stopping time $\iff$ Noise level

$$D[x] = \frac{1}{2\sigma^2}\|x - b\|^2$$
Stopping time $\iff$ Noise level

<table>
<thead>
<tr>
<th>$\sigma = 0.075$</th>
<th>$\sigma = 0.1$</th>
<th>$\sigma = 0.125$</th>
<th>$\sigma = 0.15$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PSNR</strong></td>
<td><strong>$\overline{T}$</strong></td>
<td><strong>PSNR</strong></td>
<td><strong>$\overline{T}$</strong></td>
</tr>
<tr>
<td>full optimization of all controls</td>
<td>30.05</td>
<td>0.724</td>
<td>27.72</td>
</tr>
<tr>
<td>30.00</td>
<td>0.757</td>
<td>28.72</td>
<td>1.082</td>
</tr>
<tr>
<td>optimization only of $\overline{T}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Stopping time \iff \text{Blur strength }

Does the stopping time depend on the blur strength $\tau$?
Stopping time $\iff$ Blur strength

\[
\begin{align*}
\tau &= 1.25 \\
\tau &= 1.50 \\
\tau &= 1.75 \\
\tau &= 2.00
\end{align*}
\]
Stopping time \iff Blur strength

<table>
<thead>
<tr>
<th></th>
<th>( \tau = 1.25 )</th>
<th></th>
<th>( \tau = 1.5 )</th>
<th></th>
<th>( \tau = 1.75 )</th>
<th></th>
<th>( \tau = 2.0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR ( T )</td>
<td>PSNR ( T )</td>
<td>PSNR ( T )</td>
<td>PSNR ( T )</td>
<td>PSNR ( T )</td>
<td>PSNR ( T )</td>
<td></td>
</tr>
<tr>
<td>full optimization of all controls</td>
<td>29.95 ( T )</td>
<td>( \tau = 1.25 )</td>
<td>28.76 ( T )</td>
<td>( \tau = 1.5 )</td>
<td>27.87 ( T )</td>
<td>( \tau = 1.75 )</td>
<td>27.13 ( T )</td>
</tr>
<tr>
<td></td>
<td>29.73 ( T )</td>
<td>( \tau = 1.25 )</td>
<td>27.69 ( T )</td>
<td>( \tau = 1.5 )</td>
<td>27.69 ( T )</td>
<td>( \tau = 1.75 )</td>
<td>26.71 ( T )</td>
</tr>
<tr>
<td>optimization only of ( \overline{T} )</td>
<td>39.86 ( T )</td>
<td>( \tau = 1.25 )</td>
<td>37.78 ( T )</td>
<td>( \tau = 1.5 )</td>
<td>40.60 ( T )</td>
<td>( \tau = 1.75 )</td>
<td>40.01 ( T )</td>
</tr>
</tbody>
</table>
Spectral analysis of the learned regularizers

Nonlinear eigenvalue analysis of FoE regularizers
Spectral analysis of the learned regularizers

Nonlinear eigenvalue analysis of FoE regularizers
Compute generalized eigenpairs \((\lambda_j, v_j) \in \mathbb{R} \times \mathbb{R}^n\) via

\[
\sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \lambda_j v_j
\]
Spectral analysis of the learned regularizers

Nonlinear eigenvalue analysis of FoE regularizers

Compute generalized eigenpairs \((\lambda_j, v_j) \in \mathbb{R} \times \mathbb{R}^n\) via

\[
\sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \lambda_j v_j
\]

→ forward Euler scheme reduces to

\[
v_j - \frac{T}{S} \sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \left(1 - \frac{\lambda_j T}{S}\right) v_j,
\]
Spectral analysis of the learned regularizers

Nonlinear eigenvalue analysis of FoE regularizers
Compute generalized eigenpairs \((\lambda_j, v_j) \in \mathbb{R} \times \mathbb{R}^n\) via

\[
\sum_{k=1}^{N_K} K_k^T \Phi_k(K_k v_j) = \lambda_j v_j
\]

→ forward Euler scheme reduces to

\[
v_j - \frac{T}{S} \sum_{k=1}^{N_K} K_k^T \Phi_k(K_k v_j) = \left(1 - \frac{\lambda_j T}{S}\right) v_j,
\]

▶ contrast factor \((1 - \frac{\lambda_j T}{S})\) determines global contrast change
Spectral analysis of the learned regularizers

Nonlinear eigenvalue analysis of FoE regularizers
Compute generalized eigenpairs \((\lambda_j, v_j) \in \mathbb{R} \times \mathbb{R}^n\) via

\[
\sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \lambda_j v_j
\]

\(\rightarrow\) forward Euler scheme reduces to

\[
v_j - \frac{T}{S} \sum_{k=1}^{N_K} K_k^\top \Phi_k(K_k v_j) = \left(1 - \frac{\lambda_j T}{S}\right) v_j,
\]

\(\triangleright\) contrast factor \(1 - \frac{\lambda_j T}{S}\) determines global contrast change
\(\triangleright\) holds only for one step
Spectral analysis of the learned regularizers

Nonlinear eigenvalue analysis of FoE regularizers
Compute generalized eigenpairs \((\lambda_j, v_j) \in \mathbb{R} \times \mathbb{R}^n\) via

\[
\sum_{k=1}^{N_k} K_k^\top \Phi_k(K_k v_j) = \lambda_j v_j
\]

→ forward Euler scheme reduces to

\[
v_j - \frac{T}{S} \sum_{k=1}^{N_k} K_k^\top \Phi_k(K_k v_j) = \left(1 - \frac{\lambda_j T}{S}\right) v_j,
\]

- contrast factor \((1 - \frac{\lambda_j T}{S})\) determines global contrast change
- holds only for one step
- eigenvalue determines contrast preservation
Estimation of generalized eigenpairs

Compute $N_v$ generalized eigenpairs by solving

$$\min_{\{v_j\}_{j=1}^{N_v}} \left\| \sum_{k=1}^{N_K} K_k^\top \Phi_k (K_k v_j) - \Lambda(v_j) v_j \right\|_2^2,$$

where

$$\Lambda(v) = \frac{\left\langle \sum_{k=1}^{N_K} K_k^\top \Phi_k (K_k v), v \right\rangle}{\|v\|_2^2}$$

is generalized Rayleigh quotient.
Estimation of generalized eigenpairs

Compute $N_v$ generalized eigenpairs by solving

$$
\min_{\{v_j\}_{j=1}^{N_v}} \sum_{j=1}^{N_v} \left\| \sum_{k=1}^{N_k} K_k^T \Phi_k(K_k v_j) - \Lambda(v_j)v_j \right\|_2^2,
$$

where

$$
\Lambda(v) = \frac{\left\langle \sum_{k=1}^{N_k} K_k^T \Phi_k(K_k v), v \right\rangle}{\|v\|_2^2}
$$

is generalized Rayleigh quotient.

Optimization by accelerated gradient method with backtracking.
Nonlinear eigenpairs for image denoising

\[ \lambda_1 = 0.025 \]
\[ \lambda_3 = 0.052 \]
\[ \lambda_5 = 0.070 \]
\[ \lambda_7 = 0.080 \]
\[ \lambda_9 = 0.092 \]
\[ \lambda_{11} = 0.104 \]
\[ \lambda_{13} = 0.111 \]
\[ \lambda_{15} = 0.121 \]
\[ \lambda_{17} = 0.138 \]
\[ \lambda_{19} = 0.139 \]
\[ \lambda_{21} = 0.156 \]
\[ \lambda_{23} = 0.164 \]
\[ \lambda_{25} = 0.184 \]
\[ \lambda_{27} = 0.200 \]
\[ \lambda_{29} = 0.204 \]
\[ \lambda_{31} = 0.214 \]
\[ \lambda_{33} = 0.226 \]
\[ \lambda_{35} = 0.249 \]
\[ \lambda_{37} = 0.268 \]
\[ \lambda_{39} = 0.282 \]
\[ \lambda_{41} = 0.297 \]
\[ \lambda_{43} = 0.297 \]
\[ \lambda_{45} = 0.341 \]
\[ \lambda_{47} = 0.366 \]
\[ \lambda_{49} = 0.398 \]
\[ \lambda_{51} = 0.458 \]
\[ \lambda_{53} = 0.504 \]
\[ \lambda_{55} = 0.587 \]
\[ \lambda_{57} = 1.001 \]
\[ \lambda_{59} = 1.056 \]
\[ \lambda_{61} = 3.093 \]
\[ \lambda_{63} = 5.028 \]
Nonlinear eigenpairs for image deblurring

\[
\lambda_1 = -0.02922 \\
\lambda_3 = -0.00212 \\
\lambda_5 = -0.00127 \\
\lambda_7 = -0.00120 \\
\lambda_9 = -0.00059 \\
\lambda_{11} = -0.00053 \\
\lambda_{13} = -0.00035 \\
\lambda_{15} = 0.00006 \\
\lambda_{17} = 0.00011 \\
\lambda_{19} = 0.00024 \\
\lambda_{21} = 0.00031 \\
\lambda_{23} = 0.00034 \\
\lambda_{25} = 0.00041 \\
\lambda_{27} = 0.00046 \\
\lambda_{29} = 0.00054 \\
\lambda_{31} = 0.00058 \\
\lambda_{33} = 0.00065 \\
\lambda_{35} = 0.00072 \\
\lambda_{37} = 0.00079 \\
\lambda_{39} = 0.00082 \\
\lambda_{41} = 0.00098 \\
\lambda_{43} = 0.00100 \\
\lambda_{45} = 0.00103 \\
\lambda_{47} = 0.00105 \\
\lambda_{49} = 0.00122 \\
\lambda_{51} = 0.00134 \\
\lambda_{53} = 0.00169 \\
\lambda_{55} = 0.00194 \\
\lambda_{57} = 0.00215 \\
\lambda_{59} = 0.00242 \\
\lambda_{61} = 0.00366 \\
\lambda_{63} = 0.00445
\]
Overview

Parameter learning in variational models

The Fields of Experts model

Early stopping

Total Deep Variation

Learning with graphical models
Variational Formulation of Linear Inverse Problems

- $x \in \mathbb{R}^{nC}$ restored image (size $n = n_1 \cdot n_2$, $C$ channels)

\[
x \in \arg\min_{\hat{x} \in \mathbb{R}^{nC}} \left\{ \mathcal{E}(\hat{x}, \theta, z) := \mathcal{D}(\hat{x}, z) + \mathcal{R}(\hat{x}, \theta) \right\}
\]

- data fidelity term $\mathcal{D}(x, z) = \frac{1}{2} \|Ax - z\|_2^2$ for fixed task-dependent $A \in \mathbb{R}^{lC \times nC}$ and observed data $z \in \mathbb{R}^{lC}$

- total deep variation: parametric deep multi-scale regularizer $\mathcal{R}$ depending on learned training parameters $\theta \in \Theta \subset \mathbb{R}^p$
Variational Formulation of Linear Inverse Problems

- $x \in \mathbb{R}^{nC}$ restored image (size $n = n_1 \cdot n_2$, $C$ channels)

$$x \in \arg\min_{\hat{x} \in \mathbb{R}^{nC}} \left\{ \mathcal{E}(\hat{x}, \theta, z) := \mathcal{D}(\hat{x}, z) + \mathcal{R}(\hat{x}, \theta) \right\}$$

- data fidelity term $\mathcal{D}(x, z) = \frac{1}{2} \| Ax - z \|_2^2$ for fixed task-dependent $A \in \mathbb{R}^{lC \times nC}$ and observed data $z \in \mathbb{R}^{lC}$

- total deep variation: parametric deep multi-scale regularizer $\mathcal{R}$ depending on learned training parameters $\theta \in \Theta \subset \mathbb{R}^p$

Gradient flow for $t \in (0, T)$:

$$\dot{\tilde{x}}(t) = f(\tilde{x}(t), \theta, z) := -A^\top (A\tilde{x}(t) - z) - \nabla_1 \mathcal{R}(\tilde{x}(t), \theta),$$

$$\tilde{x}(0) = x_{\text{init}}$$

Reparametrization $x(t) = \tilde{x}(tT)$ results in equivalent gradient flow for $t \in (0, 1)$:

$$\dot{x}(t) = Tf(x(t), \theta, z), \quad x(0) = x_{\text{init}}$$
Total Deep Variation

\[ K \in \mathbb{R}^{nm \times nC} \text{ learned convolution kernel} \]
\[ (\sum_{i=1}^{nC} K_{j,i} = 0 \text{ for } j = 1, \ldots, nm), \]

\[ f : \mathbb{R}^{nm} \rightarrow \mathbb{R}^{nq} \text{ multiscale convolutional neural network,} \]

\[ w \in \mathbb{R}^q \text{ learned weight vector,} \]

\[ \theta = (K, K^l_s t, w) \text{ for } l \in \{1, 2, 3\}, \]
\[ s \in \{1, \ldots, 5\}, t \in \{1, 2\}, \]

\[ r(x, \theta) := w^\top f(Kx), \]
Sampled Optimal Control Problem

Training set: $N \in \mathbb{N}$ triples $(x^i_{\text{init}}, y^i, z^i)_{i=1}^N$

- $x^i_{\text{init}} \in \mathbb{R}^{nC}$ initial image
- $y^i \in \mathbb{R}^{nC}$ ground truth image
- $z^i \in \mathbb{R}^{lC}$ observed data
Sampled Optimal Control Problem

Training set: $N \in \mathbb{N}$ triples $(x_{\text{init}}^i, y^i, z^i)_{i=1}^N$

- $x_{\text{init}}^i \in \mathbb{R}^{nC}$ initial image
- $y^i \in \mathbb{R}^{nC}$ ground truth image
- $z^i \in \mathbb{R}^{lC}$ observed data

Example (additive Gaussian image denoising): $A = I$, ground truth $y^i$ corrupted by noise $n^i \sim \mathcal{N}(0, \sigma^2) \Rightarrow x_{\text{init}}^i = z^i = y^i + n^i$
Sampled Optimal Control Problem

Training set: \( N \in \mathbb{N} \) triples \((x_{init}^i, y^i, z^i)_{i=1}^N\)

- \( x_{init}^i \in \mathbb{R}^{nC} \) initial image
- \( y^i \in \mathbb{R}^{nC} \) ground truth image
- \( z^i \in \mathbb{R}^{lC} \) observed data

Example (additive Gaussian image denoising): \( A = I \), ground truth \( y^i \) corrupted by noise \( n^i \sim \mathcal{N}(0, \sigma^2) \) \( \Rightarrow x_{init}^i = z^i = y^i + n^i \)

Sampled optimal control problem with convex and coercive loss \( l \):

\[
\inf_{\tau \in [0, \tau_{\text{max}}], \theta \in \Theta} \left\{ J(\tau, \theta) := \frac{1}{N} \sum_{i=1}^N l(x^i(1) - y^i) \right\}
\]

s.t. state equation for each sample \((i = 1, \ldots, N\) and \( t \in (0, 1)\))

\[
\dot{x}^i(t) = Tf(x^i(t), \theta, z^i), \quad x^i(0) = x_{init}^i
\]
Sampled Optimal Control Problem

Training set: $N \in \mathbb{N}$ triples $(x^i_{\text{init}}, y^i, z^i)_{i=1}^N$

- $x^i_{\text{init}} \in \mathbb{R}^{nC}$ initial image
- $y^i \in \mathbb{R}^{nC}$ ground truth image
- $z^i \in \mathbb{R}^{lC}$ observed data

Example (additive Gaussian image denoising): $A = I$, ground truth $y^i$ corrupted by noise $n^i \sim \mathcal{N}(0, \sigma^2) \Rightarrow x^i_{\text{init}} = z^i = y^i + n^i$

Sampled optimal control problem with convex and coercive loss $l$:

$$\inf_{T \in [0, T_{\text{max}}], \theta \in \Theta} \left\{ J(T, \theta) := \frac{1}{N} \sum_{i=1}^N l(x^i(1) - y^i) \right\}$$

s.t. state equation for each sample ($i = 1, \ldots, N$ and $t \in (0, 1)$)

$$\dot{x}^i(t) = Tf(x^i(t), \theta, z^i), \quad x^i(0) = x^i_{\text{init}}$$

Theorem

The minimum in the sampled optimal control problem is attained.
Discretized Optimal Control Problem

\[
\inf_{T \in [0, T_{\text{max}}], \theta \in \Theta} \left\{ J_S(T, \theta) := \frac{1}{N} \sum_{i=1}^{N} l(x^i_S - y^i) \right\}
\]

subject to discrete state equation \((s = 0, \ldots, S - 1 \text{ and } i = 1, \ldots, N)\)

\[
x^i_{s+1} = x^i_s - \frac{T}{S} A^\top (A x^i_{s+1} - z^i) - \frac{T}{S} \nabla_1 R(x^i_s, \theta),
\]

\[
x^i_0 = x^i_{\text{init}} \in \mathbb{R}^{nC}
\]

depth \(S \in \mathbb{N}\) a priori fixed
Discretized Optimal Control Problem

\[
\inf_{T \in [0, T_{\text{max}}], \theta \in \Theta} \left\{ J_S(T, \theta) := \frac{1}{N} \sum_{i=1}^{N} l(x_s^i - y^i) \right\}
\]

subject to discrete state equation \((s = 0, \ldots, S - 1 \text{ and } i = 1, \ldots, N)\)

\[
x_{s+1}^i = x_s^i - \frac{T}{S} A^\top (A x_{s+1}^i - z^i) - \frac{T}{S} \nabla_1 R(x_s^i, \theta),
\]

\[
x_0^i = x_{\text{init}}^i \in \mathbb{R}^{nC}
\]

depth \(S \in \mathbb{N}\) a priori fixed

Equivalent state equation: \(x_{s+1}^i = \tilde{f}(x_s^i, T, \theta, z^i)\) with

\[
\tilde{f}(x, T, \theta, z) := (I + \frac{T}{S} A^\top A)^{-1} (x + \frac{T}{S} (A^\top z - \nabla_1 R(x, \theta)))
\]
Discretized Optimal Control Problem

Let \((\overline{T}, \overline{\theta})\) be a pair of optimal control parameters with the corresponding state \(\{\overline{x}_s^i\}_{s=0,\ldots,S}^{i=1,\ldots,N}\).

We define the Hamiltonian

\[
H : \mathbb{R}^{nC} \times \mathbb{R}^{nC} \times [0, T_{\text{max}}] \times \Theta \times \mathbb{R}^{|C|} \to \mathbb{R}
\]

\[
(x, p, T, \theta, z) \mapsto \langle p, \tilde{f}(x, T, \theta, z) \rangle.
\]

If \(\nabla \tilde{f}(\overline{x}_s^i, \overline{T}, \overline{\theta}, z^i)\) has full rank for all \(i = 1, \ldots, N\) and \(s = 0, \ldots, S\), then there exists an adjoint process \(\{\overline{p}_s^i\}_{s=0,\ldots,S}^{i=1,\ldots,N}\) s.t.

\[
\overline{x}_{s+1}^i = \nabla_2 H(\overline{x}_s^i, \overline{p}_{s+1}^i, \overline{T}, \overline{\theta}, z^i), \quad \overline{x}_0^i = x_{\text{init}},
\]

\[
\overline{p}_s^i = \nabla_1 H(\overline{x}_s^i, \overline{p}_{s+1}^i, \overline{T}, \overline{\theta}, z^i), \quad \overline{p}_S^i = -\frac{1}{N} \nabla l(\overline{x}_S^i - y^i).
\]

Finally, the solution is optimal in the sense that

\[
\sum_{i=1}^{N} H(\overline{x}_s^i, \overline{p}_{s+1}^i, \overline{T}, \overline{\theta}, z^i) \geq \sum_{i=1}^{N} H(\overline{x}_s^i, \overline{p}_{s+1}^i, T, \theta, z^i)
\]

for all \(T \in [0, T_{\text{max}}]\) and \(\theta \in \Theta\).
Discretized Optimal Control Problem

discrete adjoint states $p^i_s$ computed via discrete Pontryagin maximum principle:

$$\bar{p}^i_s = \nabla_1 H(x^i_s, p^i_{s+1}, \bar{T}, \bar{\theta}, z^i)$$
$$= (I - \frac{T}{\bar{S}} \nabla_2^2 R(x^i_s, \bar{\theta})) (I + \frac{\bar{T}}{\bar{S}} A^\top A)^{-1}\bar{p}^i_{s+1},$$

$$\bar{p}^i_s = -\frac{1}{N} \nabla l(x^i_s - y^i).$$
Discretized Optimal Control Problem

discrete adjoint states $p^i_s$ computed via discrete Pontryagin maximum principle:

$$p^i_s = \nabla_1 H(\bar{x}^i_s, \bar{p}^i_{s+1}, \bar{T}, \bar{\theta}, z^i)$$

$$= (I - \frac{T}{S} \nabla_1^2 \mathcal{R}(\bar{x}^i_s, \bar{\theta}))(I + \frac{T}{S} A^\top A)^{-1} p^i_{s+1},$$

$$\bar{p}^i_S = -\frac{1}{N} \nabla l(\bar{x}^i_S - y^i).$$

Theorem (Optimality condition)

Let $(\bar{T}, \bar{\theta})$ be a stationary point of $J_S$ with associated states $\bar{x}^i_s$ and adjoint states $\bar{p}^i_s$. We further assume that $\nabla \tilde{f}(\bar{x}^i_s, \bar{T}, \bar{\theta}, z^i)$ has full rank for all $i = 1, \ldots, N$ and $s = 0, \ldots, S$. Then, we have

$$-\frac{1}{N} \sum_{s=0}^{S-1} \sum_{i=1}^{N} \langle \bar{p}^i_{s+1}, (I + \frac{T}{S} A^\top A)^{-1}(\bar{x}^i_{s+1} - \bar{x}^i_s) \rangle = 0.$$
Importance of Early Stopping

ground truth image
Importance of Early Stopping

\[ x_{\text{init}} \]

\[ \text{PSNR} = 20.18 \]
Importance of Early Stopping

PSNR = 24.33
Importance of Early Stopping

\[ \text{PSNR} = 26.61 \]
Importance of Early Stopping

PSNR = 29.63
Importance of Early Stopping

PSNR = 32.91
Importance of Early Stopping

PSNR = 33.92
Importance of Early Stopping

PSNR = 33.04
Importance of Early Stopping

$PSNR = 32.34$
Numerical Results (Gaussian Image Denoising)

- Ground truth image
- Noisy image, $S = 5, T = \frac{T}{2}$, PSNR = 20.19
- Noisy image, $S = 10, T = \bar{T}$, PSNR = 26.66
- Noisy image, $S = 15, T = \frac{3T}{2}$, PSNR = 30.29
- Noisy image, $S = 20, T = 2\bar{T}$, PSNR = 29.47
- Noisy image, $S = 20, T = 2\bar{T}$, PSNR = 28.61
Surface plots of deep variation $[-1, 1] \ni (\xi_1, \xi_2) \mapsto r(\xi_1 x + \xi_2 n)_i$
Numerical Results (Gaussian Image Denoising)

\[ S \mapsto \frac{1}{N} \sum_{i=1}^{N} \text{PSNR}(x^i_S, y^i) \]

\[ S \mapsto \bar{T} \]

Top: \[ S \mapsto \frac{1}{N} \sum_{i=1}^{N} \text{PSNR}(x^i_S, y^i) \]

Bottom: \[ S \mapsto \bar{T} \]
Numerical Results (Gaussian Image Denoising)

\[ T \mapsto \text{PSNR}(x^i_S, y^i) \]

\[ T \mapsto -\sum_{s=0}^{S} \langle p_{s+1}^i, x_{s+1}^i - x_s^i \rangle \]

Top: \[ T \mapsto \text{PSNR}(x^i_S, y^i) \]

Bottom: \[ T \mapsto -\sum_{s=0}^{S} \langle p_{s+1}^i, x_{s+1}^i - x_s^i \rangle \]

Averages across samples are depicted by red curves
### Numerical Results (Gaussian Image Denoising)

<table>
<thead>
<tr>
<th>Data set</th>
<th>$\sigma$</th>
<th>BM3D</th>
<th>TNRD</th>
<th>DnCNN</th>
<th>FFDNet</th>
<th>$N^3$Net</th>
<th>FOCNet</th>
<th>TDV$^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15</td>
<td>32.37</td>
<td>32.50</td>
<td>32.86</td>
<td>32.75</td>
<td>-</td>
<td>33.07</td>
<td>33.01</td>
</tr>
<tr>
<td>Set12</td>
<td>25</td>
<td>29.97</td>
<td>30.05</td>
<td>30.44</td>
<td>30.43</td>
<td>30.55</td>
<td>30.73</td>
<td>30.66</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>26.72</td>
<td>26.82</td>
<td>27.18</td>
<td>27.32</td>
<td>27.43</td>
<td>27.68</td>
<td>27.59</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>31.08</td>
<td>31.42</td>
<td>31.73</td>
<td>31.63</td>
<td>-</td>
<td>31.83</td>
<td>31.82</td>
</tr>
<tr>
<td>BSDS68</td>
<td>25</td>
<td>28.57</td>
<td>28.92</td>
<td>29.23</td>
<td>29.19</td>
<td>29.30</td>
<td>29.38</td>
<td>29.37</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>32.34</td>
<td>31.98</td>
<td>32.67</td>
<td>32.43</td>
<td>-</td>
<td>33.15</td>
<td>32.87</td>
</tr>
<tr>
<td>Urban100</td>
<td>25</td>
<td>29.70</td>
<td>29.29</td>
<td>29.97</td>
<td>29.92</td>
<td>30.19</td>
<td>30.64</td>
<td>30.38</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>25.94</td>
<td>25.71</td>
<td>26.28</td>
<td>26.52</td>
<td>26.82</td>
<td>27.40</td>
<td>27.04</td>
</tr>
<tr>
<td># Parameters</td>
<td></td>
<td>26,645</td>
<td>555,200</td>
<td>484,800</td>
<td>705,895</td>
<td>53,513,120</td>
<td>427,330</td>
<td></td>
</tr>
</tbody>
</table>

TDV$^3$ slightly worse than FOCNet (state-of-the-art), but
- FOCNet only applicable for denoising,
- TDV$^3$ has less than 1% of the parameters of FOCNet,
- rigorous mathematical theory for TDV$^3$ available
Understanding TDV

Nonlinear eigenmode analysis for TDV\(^3\):

\[ \bar{x} \in \arg\min_x \mathcal{R}(x, \theta) \quad \text{s.t.} \quad \|x\|_2 = \|x_{\text{init}}\|_2 \]
Understanding TDV

Nonlinear eigenmode analysis for TDV$^3$:

$$\bar{x} \in \arg\min_x \mathcal{R}(x, \theta) \quad \text{s.t.} \quad \|x\|_2 = \|x_{\text{init}}\|_2$$
Sensitivity Analysis (Gaussian Image Denoising)

- $((T, \theta), (\hat{T}, \hat{\theta}))$ two pairs of control parameters (obtained from two different training datasets)
- $x, \tilde{x} \in (\mathbb{R}^{nC})^{(S+1)}$ two solutions of state equation with same observed data $z$ and initial condition $x_{\text{init}}$, i.e.

$$x_{s+1} = \tilde{f}(x_s, T, \theta, z), \quad \tilde{x}_{s+1} = \tilde{f}(\tilde{x}_s, \hat{T}, \hat{\theta}, z)$$

for $s = 1, \ldots, S - 1$ and $x_0 = \tilde{x}_0 = x_{\text{init}}$

- upper bound estimate by ODE theory
Numerical Results (Single Image Super-Resolution)

- single image super-resolution with scale factor $\gamma \in \{2, 3, 4\}$
- full resolution ground truth image $y_i \in \mathbb{R}^{nC}$
- linear downsampling operator $A$ as implementation of scale factor-dependent interpolation convolution kernel in conjunction with stride
- observed low resolution image $z_i = Ay_i \in \mathbb{R}^{nC/\gamma^2}$

<table>
<thead>
<tr>
<th>$S$</th>
<th>$T$</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>$\frac{T}{2}$</td>
<td>28.32</td>
</tr>
<tr>
<td>10</td>
<td>$T$</td>
<td>31.41</td>
</tr>
<tr>
<td>15</td>
<td>$\frac{3T}{2}$</td>
<td>32.56</td>
</tr>
<tr>
<td>20</td>
<td>$2T$</td>
<td>31.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>28.47</td>
</tr>
</tbody>
</table>
Numerical Results (Single Image Super-Resolution)

ground truth image
Numerical Results (Single Image Super-Resolution)
Numerical Results (Single Image Super-Resolution)
Numerical Results (Single Image Super-Resolution)

TDV (continued)
## Numerical Results (Single Image Super-Resolution)

<table>
<thead>
<tr>
<th>Data set</th>
<th>Scale</th>
<th>MemNet</th>
<th>VDSR</th>
<th>DRRN</th>
<th>OISR-LF-s</th>
<th>TDV$^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set14</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>×2</td>
<td>33.28</td>
<td>33.03</td>
<td>33.23</td>
<td>33.62</td>
<td>33.35</td>
<td></td>
</tr>
<tr>
<td>×3</td>
<td>30.00</td>
<td>29.77</td>
<td>29.96</td>
<td>30.35</td>
<td>29.96</td>
<td></td>
</tr>
<tr>
<td>×4</td>
<td>28.26</td>
<td>28.01</td>
<td>28.21</td>
<td>28.63</td>
<td>28.41</td>
<td></td>
</tr>
<tr>
<td><strong>BSDS100</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>×2</td>
<td>32.08</td>
<td>31.90</td>
<td>32.05</td>
<td>32.20</td>
<td>32.18</td>
<td></td>
</tr>
<tr>
<td>×3</td>
<td>28.96</td>
<td>28.82</td>
<td>28.95</td>
<td>29.11</td>
<td>28.98</td>
<td></td>
</tr>
<tr>
<td>×4</td>
<td>27.40</td>
<td>27.29</td>
<td>27.38</td>
<td>27.60</td>
<td>27.50</td>
<td></td>
</tr>
<tr>
<td><strong># Parameters</strong></td>
<td>585,435</td>
<td>665,984</td>
<td>297,000</td>
<td>1,370,000</td>
<td>428,970</td>
<td></td>
</tr>
</tbody>
</table>
Transfering TDV to medical imaging

2D CT reconstruction for angular undersampling

 undersampled MRI reconstruction
Overview

Parameter learning in variational models

The Fields of Experts model

Early stopping

Total Deep Variation

Learning with graphical models
Learning with graphical models

- Finally, we consider learning parameters with graphical models for image labeling.
- For many years, both the unary terms and binary terms have been computed based on handcrafted functions.
- Cannot compete with recent deep-learning methods.
- We propose to ...
Finally, we consider learning parameters with graphical models for image labeling. For many years, both the unary terms and binary terms have been computed based on handcrafted functions. Cannot compete with recent deep-learning methods.

We propose to ... 

(i) Learn neural networks that compute $\theta = (\theta_i, \theta_{i,j})$ from the input.
(ii) Adopt the graphical model as an inference layer in the network.
Application to Stereo

Given a stereo image pair, compute the disparity (inverse depth)
Application to Stereo

- Given a stereo image pair, compute the disparity (inverse depth)

\[ I_0 \rightarrow \text{Feature layer} \]

\[ I_1 \rightarrow \text{Feature layer} \]
Application to Stereo

- Given a stereo image pair, compute the disparity (inverse depth)

\[ \theta_i(x_i) \]
Application to Stereo

Given a stereo image pair, compute the disparity (inverse depth)

\[ \theta_i(x_i) \quad \theta_{i,j}(x_i, x_j) \]
Given a stereo image pair, compute the disparity (inverse depth)

\[
\sum_{i \in V} \theta_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{i,j}(x_i, x_j)
\]
Given a stereo image pair, compute the disparity (inverse depth)

\[
\begin{align*}
\min_{x \in \mathcal{L}} E(x, \theta) := & \sum_{i \in \mathcal{V}} \theta_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{i,j}(x_i, x_j)
\end{align*}
\]
Application to Stereo

- Given a stereo image pair, compute the disparity (inverse depth)

\[
\min_{x \in \mathcal{L}} E(x, \theta) := \sum_{i \in V} \theta_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{i,j}(x_i, x_j)
\]

- Two main issues:
  (i) Efficient solution of the inference layer
  (ii) End-to-end learning
For learning the parameters $\vartheta$ of the neural network, we consider a loss function $\ell$ that compares the output of the image labeling problem $x(\vartheta)$ with the ground truth labels $x^\dagger$, e.g.

$$\ell(x(\vartheta), x^\dagger) = \|x(\vartheta) - x^\dagger\|_1$$

The learning problem represents a bilevel optimization problem:

$$\min_{\vartheta} \ell(x(\vartheta), g), \quad \text{s.t. } x(\vartheta) \in \arg\min_{x \in L} \in E(x, f(\vartheta)),$$

Hard to solve, because $x(\vartheta)$ does not continuously depend on $\vartheta$.

We approximate the problem by constructing a differentiable upper bound similar to the structured output SVM [Tsochantaridis et al. ’04]
Convex upper bound

We use the following chain of upper bounds:

\[ \max_{x \in \text{arg min}_{x \in \mathcal{L}} E(x, f)} \ell(x, x^\dagger) \leq \max_{x \in \mathcal{L} : E(x, f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) \]
Convex upper bound

- We use the following chain of upper bounds:

\[
\max_{x \in \text{arg min}_{x \in \mathcal{L}} E(x, f)} \ell(x, x^\dagger) \leq \max_{x \in \mathcal{L}: E(x, f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) \\
\leq \max_{x \in \mathcal{L}: E(x, f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f)
\]
We use the following chain of upper bounds:

\[
\max_{x \in \arg \min_{x \in \mathcal{L}} E(x, f)} \ell(x, x^\dagger) \leq \max_{x \in \mathcal{L} : E(x, f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) \\
\leq \max_{x \in \mathcal{L} : E(x, f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\
\leq \max_{x \in \mathcal{L}} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f)
\]
Convex upper bound

We use the following chain of upper bounds:

\[
\max_{x \in \text{arg min}_{x \in \mathcal{L}} E(x,f)} \ell(x, x^\dagger) \leq \max_{x \in \mathcal{L}: E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) \\
\leq \max_{x \in \mathcal{L}: E(x,f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\
\leq \max_{x \in \mathcal{L}} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\
= \psi(\hat{x}, x^\dagger, f)
\]

where \( \hat{x} = \text{arg max}_{x \in \mathcal{L}} \ell(x, x^\dagger) - E(x, f) \).

The function \( \psi \) is linear in \( f \) (in the lifted space), hence it is a maximum over linear functions \( \sim \) convex.
Convex upper bound

- We use the following chain of upper bounds:

\[
\max_{x \in \text{arg min}_{x \in L} E(x, f)} \ell(x, x^\dagger) \leq \max_{x \in L : E(x, f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) \\
\leq \max_{x \in L : E(x, f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\
\leq \max_{x \in L} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\
= \psi(\hat{x}, x^\dagger, f)
\]

where \(\hat{x} = \text{arg max}_{x \in L} \ell(x, x^\dagger) - E(x, f)\).

- The function \(\psi\) is linear in \(f\) (in the lifted space), hence it is a maximum over linear functions \(\leadsto\) convex.

- Computing the upper bound requires to solve the labeling problem but with loss-augmented unary terms.

- The resulting surrogate function is differentiable with respect to the unaries \(f_i\)

\[
D_{f_i} \psi(\hat{x}, x^\dagger, f) = \delta(x_i^\dagger) - \delta(\hat{x}_i) \ \leadsto
\]
Convex upper bound

- We use the following chain of upper bounds:

\[
\max_{x \in \arg \min_{x \in \mathcal{L}}} \ell(x, x^\dagger) \leq \max_{x \in \mathcal{L} : E(x, f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) \\
\leq \max_{x \in \mathcal{L} : E(x, f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\
\leq \max_{x \in \mathcal{L}} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) \\
= \psi(\hat{x}, x^\dagger, f)
\]

where \( \hat{x} = \arg \max_{x \in \mathcal{L}} \ell(x, x^\dagger) - E(x, f) \).

- The function \( \psi \) is linear in \( f \) (in the lifted space), hence it is a maximum over linear functions \( \Rightarrow \) convex.

- Computing the upper bound requires to solve the labeling problem but with loss-augmented unary terms.

- The resulting surrogate function is differentiable with respect to the unaries \( f_i \)

\[
D_{f_i} \psi(\hat{x}, x^\dagger, f) = \delta(x_i^\dagger) - \delta(\hat{x}_i) \Rightarrow D_{f_i} \psi(\hat{y}, y^\dagger, f) = y_i^\dagger - \hat{y}_i
\]

in the lifted space.
Convex upper bound

- We use the following chain of upper bounds:

\[
\max_{x \in \arg\min_{x \in \mathcal{L}} E(x, f)} \ell(x, x^\dagger) 
\leq \max_{x \in \mathcal{L} : E(x, f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) 
\leq \max_{x \in \mathcal{L} : E(x, f) \leq E(x^\dagger, f)} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) 
\leq \max_{x \in \mathcal{L}} \ell(x, x^\dagger) + E(x^\dagger, f) - E(x, f) 
= \psi(\hat{x}, x^\dagger, f)
\]

where \(\hat{x} = \arg\max_{x \in \mathcal{L}} \ell(x, x^\dagger) - E(x, f)\).

- The function \(\psi\) is linear in \(f\) (in the lifted space), hence it is a maximum over linear functions \(\rightarrow\) convex.

- Computing the upper bound requires to solve the labeling problem but with loss-augmented unary terms.

- The resulting surrogate function is differentiable with respect to the unaries \(f_i\)

\[
D_f \psi(\hat{x}, x^\dagger, f) = \delta(x_i^\dagger) - \delta(\hat{x}_i) \quad \sim \quad D_f \psi(\hat{y}, y^\dagger, f) = y_i^\dagger - \hat{y}_i 
\]

- Similar formula for the binary weights \(f_{i,j}\).
Graphical Explanation

True label $x_i^\dagger$  Augmented label $\hat{x}_i$  Loss margin

\[ D_{fi, \psi}(\hat{x}, x^\dagger, f) = \delta(x_i^\dagger) - \delta(\hat{x}_i) \]
Graphical Explanation

True label $x_i^{\dagger}$  
Augmented label $\hat{x}_i$

$D_{f_i}\psi(\hat{x}, x^{\dagger}, f) = \delta(x_i^{\dagger}) - \delta(\hat{x}_i)$
Training

Data bases

- Middlebury Stereo - Version 3
- KITTI 2015

Training

- Learning is performed using stochastic subgradient descent with momentum
- First, we perform a CNN-only pre-training, followed by a joint training

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Method</th>
<th>CNN</th>
<th>+CRF</th>
<th>+Joint</th>
<th>+PW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middlebury</td>
<td>CNN3</td>
<td>23.89</td>
<td>11.18</td>
<td>9.48</td>
<td>9.45</td>
</tr>
<tr>
<td></td>
<td>CNN7</td>
<td>18.58</td>
<td>9.35</td>
<td>8.05</td>
<td>7.88</td>
</tr>
<tr>
<td>Kitti 2015</td>
<td>CNN3</td>
<td>28.38</td>
<td>6.33</td>
<td>6.11</td>
<td>4.75</td>
</tr>
<tr>
<td></td>
<td>CNN7</td>
<td>13.08</td>
<td>4.79</td>
<td>4.60</td>
<td>4.04</td>
</tr>
<tr>
<td></td>
<td>[28]</td>
<td>5.99</td>
<td>4.31</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>[55]</td>
<td>13.56</td>
<td>4.45</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Experiments - Middlebury Stereo

3 layer

24.67

4.25

3.84

7 layer

17.83

3.11

2.69
Experiments - Middlebury Stereo
Experiments - Kitti 2015
Experiments - Kitti 2015

2.11
Kitti 2015 - Quality of groundtruth

Ours  |  MC-CNN  |  ContentCNN

Ours  |  MC-CNN  |  ContentCNN
Extension to motion estimation (Sintel)