EE 396: Lecture 5

Ganesh Sundaramoorthi

Feb. 22, 2011

In this lecture, we learn to solve the Euler-Lagrange equations of our denoising model PDE:

$$\begin{cases} u(x) - \alpha\Delta u(x) = I(x) & x \in int(\Omega) \\ \frac{\partial u}{\partial n}(x) = 0 & x \in \partial\Omega \end{cases}. \tag{1}$$

numerically since in the last lecture, we have understood that the equation has a solution. Note that the solution will be a global minimizer of our energy :

$$E(u) = -\log p(u|I) = \int_\Omega (I(x) - u(x))^2 \,\mathrm{d}x + \alpha \int_\Omega |\nabla u(x)|^2 \,\mathrm{d}x, \tag{2}$$

since the functional is convex.

0.1 Gradient Descent

0.1.1 Steepest Descent in $\mathbb{R}^n$

A generic algorithm for minimizing a smooth energy function is that of *gradient descent* or *steepest descent*. This algorithm for a function $f : \mathbb{R}^n \to \mathbb{R}$ works by making a guess for the minimum, $x_{guess} \in \mathbb{R}^n$ and then correcting the guess by moving slightly in the negative gradient direction (recall from your calculus class that the gradient direction is the direction that maximizes f the quickest), $x_{next} = x_{guess} - \Delta t \nabla f(x_{guess})$ where $\Delta t > 0$ is small. Since the negative gradient direction decreases the function in the steepest possible way, x_{next} should have lower value than x_{guess} provided Δt is small enough. Then one repeats the process on x_{next} and hopes that the procedure converges. That is,

$$x_0 = x_{guess} \tag{3}$$

$$x_{k+1} = x_k - \Delta t \nabla f(x_k). \tag{4}$$

If the algorithm converges, then there is a possibility that the algorithm could become trapped in a local minimum if for example f is not convex (if f is convex, then the algorithm will reach a global minimum). Therefore, the initial guess is quite important; ideally it should be chosen close enough to the global minimum as possible or as close as possible to the *desired local minimum*.

0.1.2 Gradient of Functionals

We now generalize this gradient descent procedure to our denoising model. However, what is meant by gradient of such a functional? Turns out that this is analogous to the multivariable function case. Recall that from previous lectures we have shown that the directional derivative of E is

$$\mathrm{d}E(u) \cdot v = \frac{\mathrm{d}}{\mathrm{d}t} E(u + tv)\bigg|_{t=0} = \int_\Omega v(x) \cdot (u(x) - I(x) - \alpha \Delta u(x))\, \mathrm{d}x \tag{5}$$

for $v \in \mathcal{V}$, the space of permissible perturbations, and $u \in \mathcal{U}$. Note that from this expression of the directional derivative, we can choose the perturbation v of u so that the directional derivative in the direction v is minimum. Note that an sure way to ensure $\mathrm{d}E(u) \cdot v < 0$ is to choose

$$v(x) = -(u(x) - I(x) - \alpha \Delta u(x)); \tag{6}$$

in this case,

$$\mathrm{d}E(u) \cdot v = -\int_\Omega |u(x) - I(x) - \alpha \Delta u(x)|^2\, \mathrm{d}x \leq 0. \tag{7}$$

But is the *steepest direction*? To answer this question, we must first note that $\mathcal{V}$ is an *inner product space* equipped with the inner product [1]

$$\langle v_1, v_2 \rangle_{\mathbb{L}^2} = \int_\Omega v_1(x) v_2(x)\, \mathrm{d}x \text{ for } v_1, v_2 \in \mathcal{V}. \tag{8}$$

In an inner product space, we automatically get for free the *Cauchy-Schwartz* inequality $|\langle v_1, v_2 \rangle| \leq \|v_1\| \|v_2\|$ for the norm induced from the inner product and equality if and only if v_1 and v_2 are such that $v_1 = a v_2$ for some $a \in \mathbb{R}$. Applying this to (5), we find that

$$|\,\mathrm{d}E(u) \cdot v| \leq \|v\|_{\mathbb{L}^2} \|u - I - \alpha \Delta u\|_{\mathbb{L}^2} \tag{9}$$

or

$$\frac{|\,\mathrm{d}E(u) \cdot v|}{\|v\|_{\mathbb{L}^2}} \leq \|u - I - \alpha \Delta u\|_{\mathbb{L}^2} \tag{10}$$

where $\| \cdot \|_{\mathbb{L}^2}$ is the norm induced from the $\mathbb{L}^2$ inner product. Therefore we see that the magnitude of the quotient above will be maximized (equality in the above expression) when

$$v = -(u - I - \alpha \Delta u), \tag{11}$$

thus, this justifies calling

$$v = -(u - I - \alpha \Delta u) = -\nabla E(u) \tag{12}$$

the negative *gradient* direction. This justifies the formal definition :

Definition 1. *Let $\mathcal{E} : \mathcal{U} \to \mathbb{R}$ be a functional. Then the* gradient *of E at u ($u \in \mathcal{U}$), denoted $\nabla E(u) \in \mathcal{V}$, is defined as the perturbation that satisfies*

$$\mathrm{d}E(u) \cdot v = \int_\Omega v(x) \cdot \nabla E(u)\, \mathrm{d}x = \langle v, \nabla E(u) \rangle_{\mathbb{L}^2}\,. \tag{13}$$

[1] Recall that $\langle \cdot, \cdot \rangle$ is a (real) inner product on a vector space V if the properties hold:

1. Linearity: $\langle a_1 v_1 + a_2 v_2, v \rangle = a_1 \langle v_1, v \rangle + a_2 \langle v_2, v \rangle$ for all $v_1, v_2, v \in V$ and $a_1, a_2 \in \mathbb{R}$.
2. Positive-definiteness: $\langle v, v \rangle \geq 0$and $\langle v, v \rangle = 0$ is equivalent to $v = 0$ for all $v \in V$
3. Symmetry: $\langle v_1, v_2 \rangle = \langle v_2, v_1 \rangle$.

Note that a norm can be defined from the inner product: $\|v\| = \sqrt{\langle v, v \rangle}$.

0.1.3 Gradient Descent PDE

Now the steepest descent algorithm for our functionals is

$$u_0 = u_{guess} \tag{14}$$
$$u_{k+1} = u_k - \Delta t \nabla E(u_k) \tag{15}$$

where k represents the iteration number. In continuous time, the gradient descent is the PDE

$$u(0,\cdot) = u_{guess}(0,\cdot)$$
$$u_t(t,\cdot) = -\nabla E(u(t,\cdot))$$

and (15) is a discretization where we have used the *forward Euler difference approximation* :

$$u_t(t,\cdot) = \frac{u(t+\Delta t,\cdot) - u(t,\cdot)}{\Delta t} + \mathcal{O}(\Delta t). \tag{16}$$

For our denoising model, this becomes

$$u_t = I - u + \alpha \Delta u \tag{17}$$

where $\alpha > 0$ [2]

We assume that we sample the function u on a regular grid with spacing Δx in the x direction and Δy in the y direction. We now discretize the PDE [3]. Using central differences [4], we find that

$$u(t, x+\Delta x, y) = u(t,x,y) + \frac{\partial u}{\partial x}(t,x,y)\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x}(t,x,y)(\Delta x)^2 + \mathcal{O}((\Delta x)^2) \tag{18}$$
$$u(t, x-\Delta x, y) = u(t,x,y) - \frac{\partial u}{\partial x}(t,x,y)\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x}(t,x,y)(\Delta x)^2 + \mathcal{O}((\Delta x)^2) \tag{19}$$

where $\mathcal{O}(\Delta x)$ denotes linear dependence on Δx. Adding the above two equations, we find that

$$\frac{\partial^2 u}{\partial x^2}(t,x,y) = \frac{u(t,x+\Delta x,y) - 2u(t,x,y) + u(t,x-\Delta x,y)}{(\Delta x)^2} + \mathcal{O}(\Delta x) \tag{20}$$
$$\frac{\partial^2 u}{\partial y^2}(t,x,y) = \frac{u(t,x,y+\Delta y) - 2u(t,x,y) + u(t,x,y-\Delta y)}{(\Delta y)^2} + \mathcal{O}(\Delta y) \tag{21}$$

Setting $u^k(x,y) = u(k\Delta t, x, y)$ we find that our PDE can be written

$$\begin{aligned} u^{k+1}(x,y) = u^k(x,y) + \Delta t \Big[& I(x,y) - u^k(x,y) + \\ \alpha \frac{u^k(x+\Delta x,y) - 2u^k(x,y) + u^k(x-\Delta x,y)}{(\Delta x)^2} & + \alpha \frac{u^k(x,y+\Delta y) - 2u^k(x,y) + u^k(x,y-\Delta y)}{(\Delta y)^2} \Big] \\ & + \mathcal{O}(\Delta x \Delta t) + \mathcal{O}(\Delta y \Delta t), \end{aligned} \tag{22}$$

thus, we have discretized our PDE. The question remains on how to choose Δt to ensure that the numerical scheme is *stable* and converges.

[2] The equation $u_t = \alpha \Delta u$ is called the heat equation for $\Omega = \mathbb{R}^n$; one can show that $u(t,\cdot)$ is simply a Gaussian of standard deviation $\sqrt{t}$ convolved with the initial guess u_0. Therefore, we can understand that our denoising model will be a combination smoothing with and closeness to the original image I.

[3] The is a subtle art in choosing the discretizing scheme for a PDE; ultimately the approriate numerical scheme requires full understanding on the behavior of the PDE.

[4] Generally, central differences are preferred compared to forward or backward differences because they induce the least error. However, in certain PDE choosing central differences could lead to an unstable PDE. Again, the differencing scheme is dependent on the behavior of the PDE.

0.1.4 Von Neumann Analysis

We apply Von Neumann analysis to analyze the stability of the discretization of the PDE, and this analysis typically can be applied to linear PDE. We compute the Fourier Transform in the spatial variable (x, y) of our discretization scheme :

$$U^{k+1}(\omega_1, \omega_2) = U^k(\omega_1, \omega_2) \left[1 - \Delta t + \alpha \Delta t \left(\frac{e^{i\Delta x \omega_1} - 2 + e^{-i\Delta x \omega_2}}{(\Delta x)^2} + \frac{e^{i\Delta y \omega_2} - 2 + e^{-i\Delta y \omega_2}}{(\Delta y)^2} \right)\right] + \Delta t I(\omega_1, \omega_2) \quad (23)$$

where we have noted that the Fourier transform of $u^k(x + \Delta x, y)$ is $U^k(x, y)e^{i\Delta x\omega_1}$ where U^k denotes Fourier transform of u^k and i is the imaginary unit. We define the $H(\omega_1, \omega_2)$ to be the expression in the brackets above, and then

$$H(\omega_1, \omega_2) = 1 - \Delta t - 4\alpha\Delta t \left(\frac{\sin^2(\Delta x\omega_1/2)}{(\Delta x)^2} + \frac{\sin^2(\Delta y\omega_2/2)}{(\Delta y)^2} \right). \quad (24)$$

Therefore, (23) becomes

$$U^{k+1}(\omega_1, \omega_2) = H(\omega_1, \omega_2)U^k(\omega_1, \omega_2) + \Delta t I(\omega_1, \omega_2). \quad (25)$$

Writting the above as an explicit formula, we have that

$$U^{k+1}(\omega_1, \omega_2) = (H(\omega_1, \omega_2))^k U^0(\omega_1, \omega_2) + \Delta t I(\omega_1, \omega_2) \sum_{i=0}^{k-1} (H(\omega_1, \omega_2))^k \quad (26)$$

where $(\cdot)^i$ denotes ith power. We see that as $k \to \infty$, we must have that

$$|H(\omega_1, \omega_2)| < 1 \text{ for all } \omega_1, \omega_2; \quad (27)$$

that is

$$1 - \Delta t - 4\alpha\Delta t \left(\frac{\sin^2(\Delta x\omega_1/2)}{(\Delta x)^2} + \frac{\sin^2(\Delta y\omega_2/2)}{(\Delta y)^2} \right) < 1 \quad (28)$$

and

$$1 - \Delta t - 4\alpha\Delta t \left(\frac{\sin^2(\Delta x\omega_1/2)}{(\Delta x)^2} + \frac{\sin^2(\Delta y\omega_2/2)}{(\Delta y)^2} \right) > -1. \quad (29)$$

The first yields that $\Delta t > 0$ (noting that $\alpha > 0$) and the second yields that

$$\Delta t < \frac{2}{1 + 4\alpha \left(\frac{\sin^2(\Delta x\omega_1/2)}{(\Delta x)^2} + \frac{\sin^2(\Delta y\omega_2/2)}{(\Delta y)^2} \right)} \text{ for all } \omega_1, \omega_2 \quad (30)$$

which means that Δt must be less than the min of the right hand side over ω_1, ω_2 or

$$\Delta t < \frac{2}{1 + 4\alpha \left(\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} \right)}. \quad (31)$$

In the case that $\Delta x = \Delta y = 1$, we have that

$$\Delta t < \frac{2}{1 + 8\alpha}. \quad (32)$$

0.1.5 Choosing α

Recall that the original energy from our model is

$$E(u) = \frac{1}{2\sigma_n^2}\int_\Omega (I(x)-u(x))^2\,\mathrm{d}x + \frac{1}{2\sigma_p^2}\int_\Omega |\nabla u(x)|^2\,\mathrm{d}x, \tag{33}$$

where σ_n is the standard deviation of the noise and σ_p is the standard deviation of the true noise free image derivatives. We scaled the energy by $2\sigma_n^2$ (which doesn't affect the minimum u). Then α was defined as

$$\alpha = \frac{\sigma_n^2}{\sigma_p^2} = \frac{1}{\text{SNR}}, \tag{34}$$

where SNR can be thought of as a signal to noise ratio. We see that α should be chosen larger as the noise increases. In practice, we do not know these parameters, but you may experiment with estimating these parameters from the noisy image.

Notice that as α becomes larger (as the image becomes more noisy), Δt becomes smaller and thus the algorithm becomes slower.

0.1.6 Algorithm Denoising Results

A key point is the initialization. Theoretically, we have shown that the energy is convex, and the energy only has global minimum, and thus the initialization should be irrelevant since the scheme can only converge at a global minimum. In practice, initialization is important. For these experiments we choose $u_0 = I$. Also of importance is the convergence criteria, and we take that to be

$$\max_{x\in\Omega} |u^k(x) - I(x) - \alpha\Delta u^k(x)| < \varepsilon \tag{35}$$

where $\varepsilon > 0$ is our allowed error.

Fig. 1 shows an original image that is then corrupted by iid Gaussian noise. Fig. 2 shows the result of the algorithm constructed.

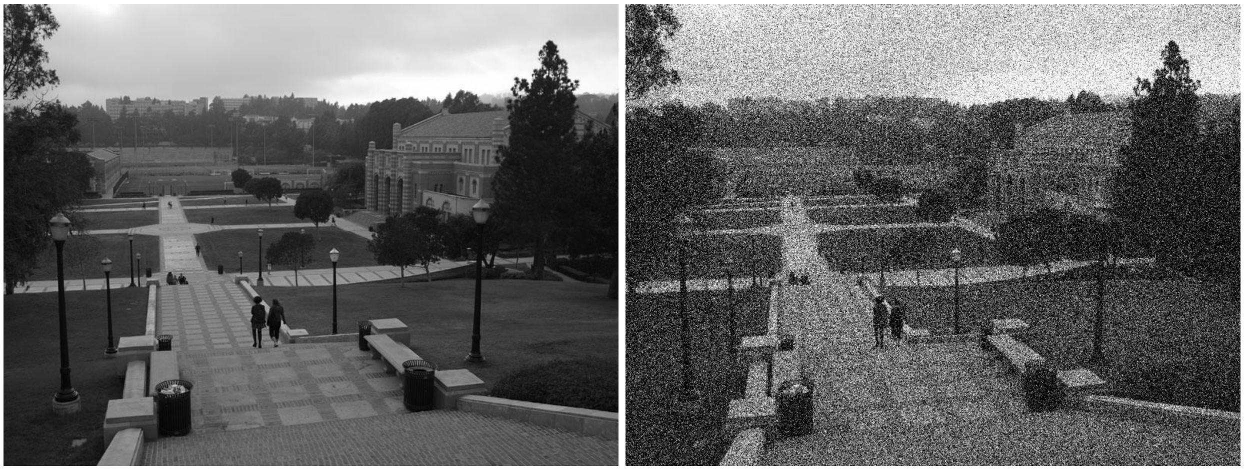

Figure 1: Left: An image of the UCLA campus, and right: the image corrupted synthetically generated noise (iid Gaussian mean zero with $\sigma^2 = 0.05$).

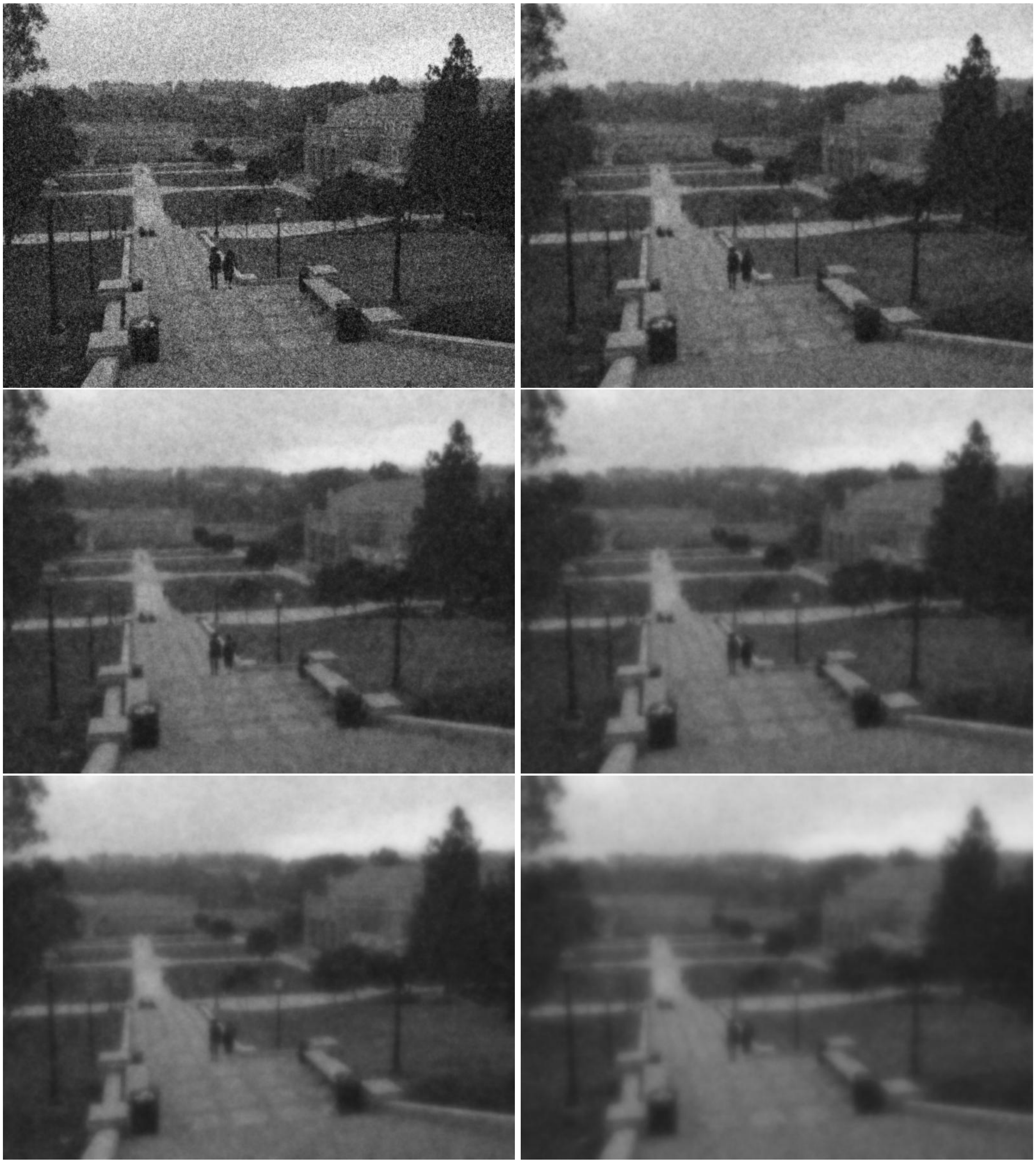

Figure 2: Left to right, top to bottom: denoising results for the noisy image in Fig. 1 with $\alpha = 1, 5, 10, 17, 25, 50$, respectively.