Notes on Learned Proximal Networks

- Let $\underline{x} \in \mathbb{R}^n$, $A: \mathbb{R}^m \to \mathbb{R}^n$, and consider the inverse problem

$$\underline{y} = \underline{\underline{A}}(\underline{x}) + \underline{v}, \tag{1}$$

where $\underline{v}$ is some noise/nuisance term. Our goal is to recover a solution $\hat{\underline{x}} \in \mathbb{R}^n$ to this problem that approximates the real signal $\underline{x}$ or even recover it.

- This is an ill-posed inverse problem. The inverse problem may have an infinite number of solutions all approximating the signal ... To overcome this, we need to regularize the problem.

- One approach: Use a prior to regularize the problem. This is a function $R: \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ that:

  i) Promote certain desirable properties in $\underline{x}$ or
  ii) Encapsulate the knowledge we have about $\underline{x}$ or
  iii) Promote a solution likely under the distribution of $\underline{x}$,

  or any combinations of i) - iii).

⚠ Using a prior function R does NOT mean necessarily that $\underline{x}$ is sampled from a distribution $p \sim e^{-R(\cdot)}$ or is likely a sample from $\sim e^{-R(\cdot)}$ or that samples of $\sim e^{-R(\cdot)}$ are similar to $\underline{x}$. There are counterexamples to this. (E.g., $R = \|\cdot\|_1$.)

- Assume now that the noise $\underline{v}$ is Gaussian. Then an appropriate data fidelity term is the quadratic norm $\frac{1}{2}\|\cdot\|_2^2$.

- We can recover a solution with the data fidelity term and the prior R by minimizing their weighted sum:

$$\hat{\underline{x}}_R(\underline{y}, t) \in \operatorname*{argmin}_{\underline{x} \in \operatorname{dom} R} \frac{1}{2t}\|\underline{y} - \underline{\underline{A}}(\underline{x})\|_2^2 + R(\underline{x}), \qquad (2)$$

where $t > 0$ is a hyperparameter.

(Note: We minimize over $\operatorname{dom} R$ to allow for the prior R to take the extended value $+\infty$ on some subset of $\mathbb{R}^n$. This allows for, e.g., priors defined on bounded domains.)

- Key to our work is the proximal operator of prior R, which we will denote by prox_R:

$$\operatorname{prox}_R(\underline{y}) \in \operatorname*{argmin}_{\underline{x} \in \operatorname{dom} R} \frac{1}{2}\|\underline{y} - \underline{x}\|_2^2 + R(\underline{x}). \qquad (3)$$

When R is proper (i.e., not identically $+\infty$ and nowhere equal to $-\infty$), lower semicontinuous and convex, then prox_R is single-valued.

When R is non-convex, then we do not have uniqueness of solutions in (3)...

- Following Gribonval and Nikolova (2020), we define the proximal operator of R as a selection over the solutions of (3).

- More precisely, a function $f: \operatorname{dom} R \to \mathbb{R}^n$ is a proximity operator of a prior $R: \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ if

$$f(y) \in \underset{x \in \operatorname{dom} R}{\operatorname{argmin}} \; \frac{1}{2}\|y - x\|_2^2 + R(x) \qquad (4)$$

for each $y \in \operatorname{dom} R$.

- For short, we will write (4) as $f(y) \in \operatorname{prox}_R(y)$.

- Theorem 1 + Corollary 1 from Gribonval and Nikolova (2020) yield the following:

Let $\operatorname{dom} R$ be non-empty and open. Then f is a (continuous) proximal operator of R if and only if there exists a convex, differentiable function Ψ on $\operatorname{dom} R$ such that

$$f(y) = \nabla_y \Psi(y) \quad \text{for each } y \in \operatorname{dom} R.$$

Moreover, we have the identity

$$\Psi(y) + \left(R(\nabla_y \Psi(y)) + \frac{1}{2}\|\nabla_y \Psi(y)\|_2^2 \right) = \langle y, \nabla_y \Psi(y) \rangle. \qquad (5)$$

Neural Networks for parametrizing gradients of convex functions

- We seek a neural network (NN) $\psi_\theta : \mathbb{R}^n \to \mathbb{R}$ parametrized by some parameters θ such that ψ_θ is convex w.r.t. its input and such that

$$f_\theta = \nabla \psi_\theta$$

is a "good" approximation to the (continuous) proximal operator of the prior R.

- Candidates: NNs whose i) Nonlinear activation functions are convex and non-decreasing.
  ii) Network weights are non-negative.

One way of achieving this is as follows:

- Pick some $g : \mathbb{R} \to \mathbb{R}$ that is convex, non-decreasing and C^2.
- Define $\underline{g} : \mathbb{R}^n \to \mathbb{R}^n$ via $\underline{g}(\underline{z}) = (g(z_1), g(z_2), \dots, g(z_n))$.
- Let $\theta = \{\underline{w}, b, \{\underline{\underline{W}}_k\}_{k=2}^K, \{\underline{\underline{H}}_k, \underline{b}_k\}_{k=1}^K\}$ be a set of learnable parameters. Here,

$$\underline{w} \in \mathbb{R}^n,\ b \in \mathbb{R},\ \underline{\underline{W}}_k \in \mathbb{R}^{n \times n} \text{ for } k = 2, \dots, K,$$

$$(\underline{\underline{H}}_k, \underline{b}_k) \in \mathbb{R}^{n \times n} \times \mathbb{R}^n \text{ for } k = 1, \dots, K.$$

- Let $\boxed{\psi_\theta(y) = \langle \underline{w}, \underline{z}_K \rangle + b}$, where

$$\underline{z}_1 = \underline{g}(\underline{\underline{H}}_1 y + \underline{b}_1) \text{ and } \underline{z}_k = \underline{g}(\underline{\underline{W}}_k \underline{z}_{k-1} + \underline{\underline{H}}_k y + \underline{b}_k) \text{ for } k \in \{2, \dots, K\}.$$

- **Note 1:** The conditions i) and ii) ensure that the function $y \mapsto \tau_\theta(y)$ is convex. This follows from the condition needed to build a convex function from other convex functions, including:

a) Positive combinations of convex functions is convex (all weights are non-negative).

b) Pre-composition of a convex function with an affine mapping is convex (domain of the convex function and the image of the mapping must have non-zero intersection).

c) Post-composition with a non-decreasing convex function is convex.

See Hiriart-Urruty and Lemaréchal (1996), IV, Section 2.1.

- **Note 2:** Other convexity-preserving operations (e.g., supremum of convex functions, conjugations, distances / perspectives) could be used, in principle.

Moreover different / multiples convex, non increasing C^2 functions could be used in the layers instead of just one (i.e., $(g_1, g_2, \ldots, g_k)$ instead of g.)

Connections between the convex function ψ and the original minimization problem

Let $S: \mathbb{R}^n \to \mathbb{R}$ denote the value of the minimization problem underlying $f \in \text{prox}_R$:

$$S(y) := \min_{x \in \text{dom} R} \frac{1}{2}\|x - y\|_2^2 + R(x).$$

$$\equiv \frac{1}{2}\|f(y) - y\|_2^2 + R(f(y)). \tag{6}$$

- What is the connection between the convex function ψ from $f = \nabla_y \psi$ and the function S?

- Expand the quadratic term in (6) and rearrange:

$$R(f(y)) + \frac{1}{2}\|f(y)\|_2^2 + \frac{1}{2}\|y\|_2^2 - S(y) = \langle y, f(y) \rangle$$

Comparing this with identity (5), we find

$$\psi(y) = \frac{1}{2}\|y\|_2^2 - S(y)$$

$\Rightarrow$ This connects the convex function ψ to the minimal value function S. In particular, if we are given sample observations $\{y_i, S(y_i)\}_{i=1}^n$, then we effectively know the true value of ψ at the points $\{y_i\}_{i=1}^n$.

- Suppose we receive samples $\{y_i, S(y_i)\}_{i=1}^{L}$ (e.g., observations / measurements from sensors).
We can set $\psi(y_i) = \frac{1}{2}\|y_i\|_2^2 - S(y_i)$ and train a neural network by minimizing the loss function

$$\min_{\theta} \frac{1}{L}\sum_{i=1}^{L} \left(\psi_\theta(y_i) - \psi(y_i)\right)^2$$

- After training, we obtain the function $\psi_{\hat\theta}(y)$ with trained parameters $\hat\theta$. In particular, we have

$$R(\nabla_y \psi_{\hat\theta}(y_i)) = S(y_i) - \frac{1}{2}\|\nabla_y \psi_{\hat\theta}(y_i) - y_i\|_2^2$$

- What if we want to evaluate the prior R at x?
Write

$$R(x) = R(\nabla_y \psi_{\hat\theta}(y^*)) \text{ such that } \nabla_y \psi_{\hat\theta}(y^*) = x.$$

Fang, Buchanan and Sulam (2024) suggests to solve this as follows: Set $\sigma > 0$ and minimize

$$\min_{y \in \mathbb{R}^n} \left\{ \psi_{\hat\theta}(y) - \langle y, x\rangle + \frac{\sigma}{2}\|y\|_2^2 \right\}$$

This is σ-strongly convex with a global, unique minimum $y^*(x)$ characterized by the first-order conditions

$$\nabla_y \psi_{\hat\theta}(y^*(x)) + \sigma y^*(x) = x.$$

Then write $R(x) \approx R(\nabla_y \psi_{\hat\theta}(y^*(x)) + \sigma y^*(x))$.

Note 1: This possibly can be done sequentially with a monotonically decreasing sequence $\{x_j\}_{j=1}^{N}$, using $y^*(x_j)$ as a warm start for the subsequent minimum $y^*(x_{j+1})$.

Note 2: GPL thinks there is a rigorous basis to this procedure; he will try to find and cite the correct result to help with this.

TBC.