Neural Network Accelerated Implicit Filtering: Integrating Neural Network Surrogates With Provably Convergent Derivative Free Optimization Methods

Brian Irwin [1]   Eldad Haber [1]   Raviv Gal [2]   Avi Ziv [2]

[1] Department of Earth, Ocean and Atmospheric Sciences,
The University of British Columbia (UBC)

[2] Hybrid Cloud Quality Technologies, IBM Research Haifa

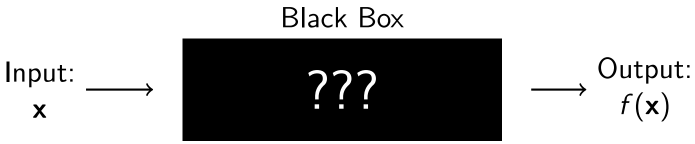
Black Box Optimization

Black Box Optimization

- *Black box optimization* (BBO), also referred to as *derivative free optimization* (DFO) and *zeroth order* (ZO) optimization, is an important subfield with important practical applications in business, chemistry, computer science, economics, engineering design, mathematics, medicine, operations research, and physics.

Black Box Optimization

- *Black box optimization* (BBO), also referred to as *derivative free optimization* (DFO) and *zeroth order* (ZO) optimization, is an important subfield with important practical applications in business, chemistry, computer science, economics, engineering design, mathematics, medicine, operations research, and physics.



Black Box Optimization

- *Black box optimization* (BBO), also referred to as *derivative free optimization* (DFO) and *zeroth order* (ZO) optimization, is an important subfield with important practical applications in business, chemistry, computer science, economics, engineering design, mathematics, medicine, operations research, and physics.

Black Box

Input: $\mathbf{x}$ $\longrightarrow$??? $\longrightarrow$ Output: $f(\mathbf{x})$

- BBO problems naturally arise when the structure of the objective function is *unknown* (e.g. someone else's computer simulation) or *very laborious* to differentiate (e.g. your legacy FORTRAN code).

- What if the black box output $f(\mathbf{x})$ is a *noise corrupted* version of a smooth function $f_s(\mathbf{x})$ that can be decomposed as

$$f(\mathbf{x}) = f_s(\mathbf{x}) + \phi(\mathbf{x}) \tag{1}$$

where $\phi(\mathbf{x})$ is noise?

- What if the black box output $f(\mathbf{x})$ is a *noise corrupted* version of a smooth function $f_s(\mathbf{x})$ that can be decomposed as

$$f(\mathbf{x}) = f_s(\mathbf{x}) + \phi(\mathbf{x}) \tag{1}$$

where $\phi(\mathbf{x})$ is noise?

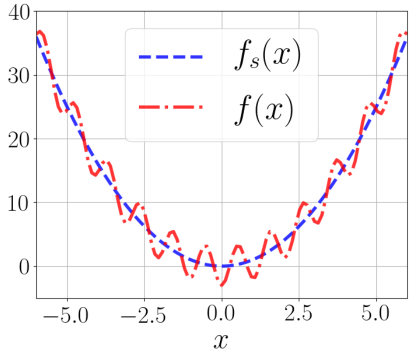
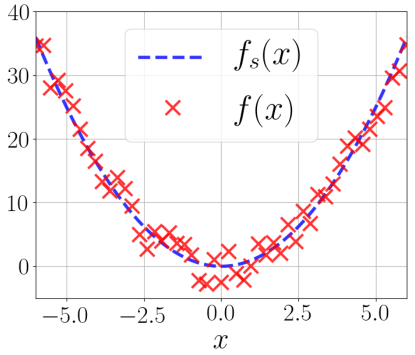
- Our goal is to solve

$$\min_{\mathbf{x}\in\mathbb{R}^{N_\mathbf{x}}} \left\{ f_s(\mathbf{x}) \right\} \tag{2}$$

using *only* samples from the noisy black box given in (1) *instead of the inaccessible* smooth function $f_s(\mathbf{x})$.

Examples Of Noise $\phi(\mathbf{x})$ With $f_s(\mathbf{x}) = \|\mathbf{x}\|_2^2$ And $N_{\mathbf{x}} = 1$

Examples Of Noise $\phi(\mathbf{x})$ With $f_s(\mathbf{x}) = \|\mathbf{x}\|_2^2$ And $N_{\mathbf{x}} = 1$	
Deterministic $\phi(\mathbf{x})$:	*Stochastic* $\phi(\mathbf{x})$:
High frequency periodic function	Draw IID uniform samples
$\phi(\mathbf{x}) = -3\cos\left(6\,\|\mathbf{x}\|_2\right)$	$\phi(\mathbf{x}) \sim \mathcal{U}(-3, 3)$

Examples Of Noise $\phi(\mathbf{x})$ With $f_s(\mathbf{x}) = \|\mathbf{x}\|_2^2$ And $N_\mathbf{x} = 1$	
Deterministic $\phi(\mathbf{x})$:	*Stochastic* $\phi(\mathbf{x})$:
High frequency periodic function	Draw IID uniform samples
$\phi(\mathbf{x}) = -3\cos\left(6\,\|\mathbf{x}\|_2\right)$	$\phi(\mathbf{x}) \sim \mathcal{U}(-3,3)$
	

- Minimize the Rosenbrock function

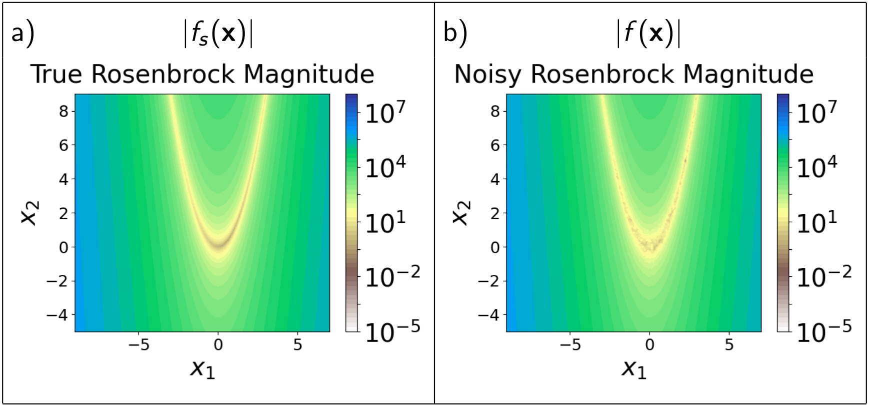
$$f_s(x_1, x_2) = \big(1 - x_1\big)^2 + 100\big(x_2 - (x_1)^2\big)^2 \tag{3}$$

$$f(x_1, x_2) = f_s(x_1, x_2) + \mathcal{N}(0, 100) \tag{4}$$

- Minimize the Rosenbrock function

$$f_s(x_1, x_2) = \left(1 - x_1\right)^2 + 100\left(x_2 - (x_1)^2\right)^2 \tag{3}$$

$$f(x_1, x_2) = f_s(x_1, x_2) + \mathcal{N}(0, 100) \tag{4}$$



- Implicit filtering (IF) methods were designed to handle noisy objective function evaluations and *only sample* $f(\mathbf{x})$ directly.

- Implicit filtering (IF) methods were designed to handle noisy objective function evaluations and *only sample* $f(\mathbf{x})$ directly.

- IF methods have established *convergence theory*.

- Implicit filtering (IF) methods were designed to handle noisy objective function evaluations and *only sample* $f(\mathbf{x})$ directly.

- IF methods have established *convergence theory*.

- IF methods are *memoryless* and do not explicitly keep track of *previous values* of $f(\mathbf{x})$.

- Implicit filtering (IF) methods were designed to handle noisy objective function evaluations and *only sample* $f(\mathbf{x})$ directly.

- IF methods have established *convergence theory*.

- IF methods are *memoryless* and do not explicitly keep track of *previous values* of $f(\mathbf{x})$.

- NNAIF *keeps track of previous* values of $f(\mathbf{x})$ to build a relatively cheap *nonlinear* surrogate model $\widehat{f_s}(\mathbf{x}, \mathbf{\Theta}_k)$.

- Implicit filtering (IF) methods were designed to handle noisy objective function evaluations and *only sample* $f(\mathbf{x})$ directly.

- IF methods have established *convergence theory*.

- IF methods are *memoryless* and do not explicitly keep track of *previous values* of $f(\mathbf{x})$.

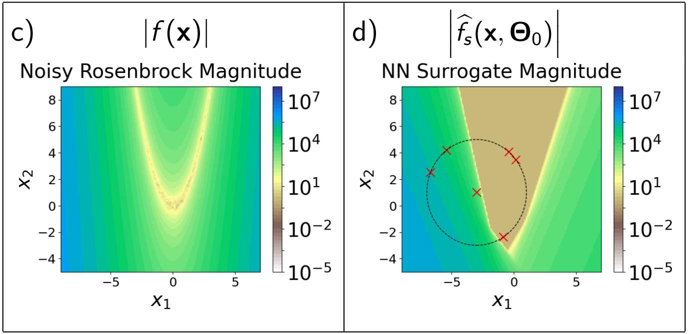
- NNAIF *keeps track of previous* values of $f(\mathbf{x})$ to build a relatively cheap *nonlinear* surrogate model $\widehat{f_s}(\mathbf{x}, \mathbf{\Theta}_k)$.

- Derivatives of $\widehat{f_s}$ can be obtained via automatic differentiation.

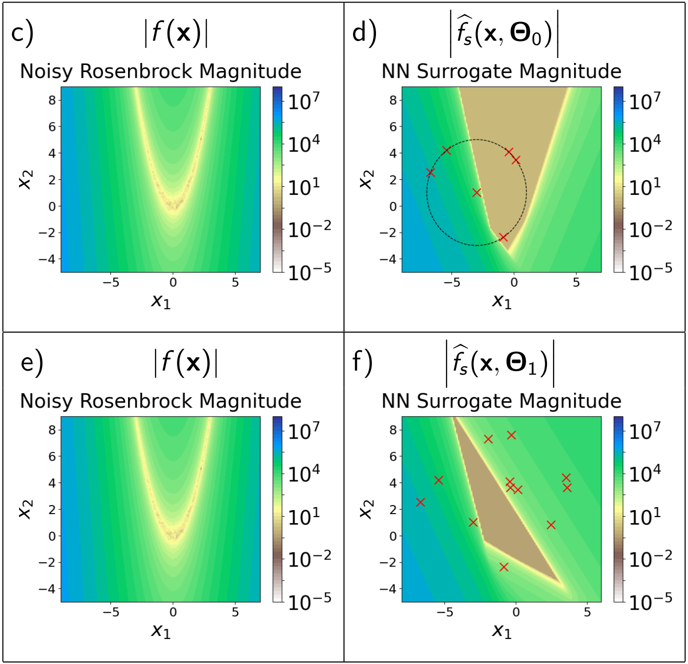
- Implicit filtering (IF) methods were designed to handle noisy objective function evaluations and *only sample* $f(\mathbf{x})$ directly.
- IF methods have established *convergence theory*.
- IF methods are *memoryless* and do not explicitly keep track of *previous values* of $f(\mathbf{x})$.
- NNAIF *keeps track of previous* values of $f(\mathbf{x})$ to build a relatively cheap *nonlinear* surrogate model $\widehat{f_s}(\mathbf{x}, \mathbf{\Theta}_k)$.
- Derivatives of $\widehat{f_s}$ can be obtained via automatic differentiation.
- Use surrogate model $\widehat{f_s}$ to obtain an *approximate minimizer of* $f_s(\mathbf{x})$.

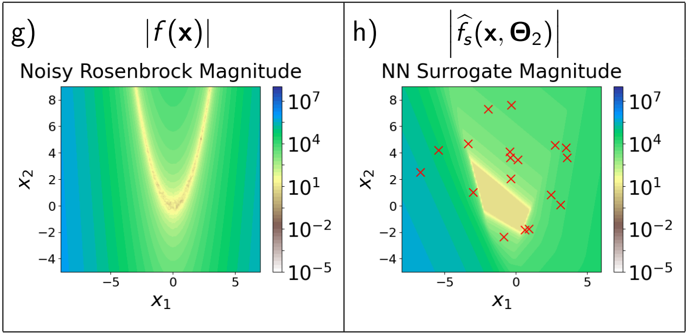
- Implicit filtering (IF) methods were designed to handle noisy objective function evaluations and *only sample* $f(\mathbf{x})$ directly.
- IF methods have established <u>*convergence*</u> *theory*.
- IF methods are *memoryless* and do <u>not</u> explicitly keep track of *previous values* of $f(\mathbf{x})$.
- NNAIF *keeps track of previous* values of $f(\mathbf{x})$ to build a relatively cheap *nonlinear* surrogate model $\widehat{f_s}(\mathbf{x}, \mathbf{\Theta}_k)$.
- Derivatives of $\widehat{f_s}$ can be obtained via automatic differentiation.
- Use surrogate model $\widehat{f_s}$ to obtain an *approximate minimizer of* $f_s(\mathbf{x})$.
- Use surrogate model $\widehat{f_s}$ to *propose points likely to decrease* $f_s(\mathbf{x})$.

- Implicit filtering (IF) methods were designed to handle noisy objective function evaluations and *only sample* $f(\mathbf{x})$ directly.
- IF methods have established *convergence theory*.
- IF methods are *memoryless* and do not explicitly keep track of *previous values* of $f(\mathbf{x})$.
- NNAIF *keeps track of previous* values of $f(\mathbf{x})$ to build a relatively cheap *nonlinear* surrogate model $\widehat{f_s}(\mathbf{x}, \mathbf{\Theta}_k)$.
- Derivatives of $\widehat{f_s}$ can be obtained via automatic differentiation.
- Use surrogate model $\widehat{f_s}$ to obtain an *approximate minimizer of* $f_s(\mathbf{x})$.
- Use surrogate model $\widehat{f_s}$ to *propose points likely to decrease* $f_s(\mathbf{x})$.
- If surrogate model $\widehat{f_s}$ is bad, fall back to convergence of IF.

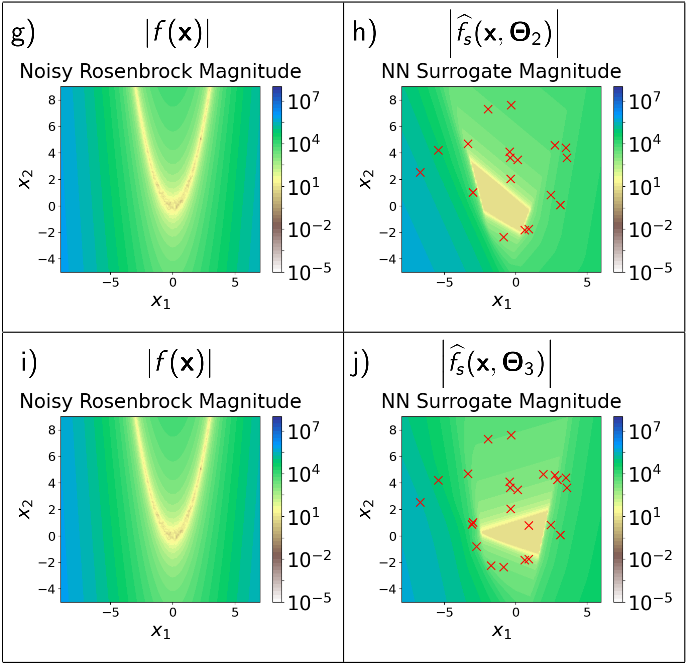


c)
$|f(\mathbf{x})|$
Noisy Rosenbrock Magnitude
d)
$\left|\widehat{f_s}(\mathbf{x}, \mathbf{\Theta}_0)\right|$
NN Surrogate Magnitude
x_1
x_2
10^7
10^4
10^1
10^{-2}
10^{-5}


c)
$|f(\mathbf{x})|$
Noisy Rosenbrock Magnitude
x_2
x_1
10^7
10^4
10^1
10^{-2}
10^{-5}
d)
$\left|\widehat{f_s}(\mathbf{x}, \mathbf{\Theta}_0)\right|$
NN Surrogate Magnitude
x_2
x_1
10^7
10^4
10^1
10^{-2}
10^{-5}
e)
$|f(\mathbf{x})|$
Noisy Rosenbrock Magnitude
x_2
x_1
10^7
10^4
10^1
10^{-2}
10^{-5}
f)
$\left|\widehat{f_s}(\mathbf{x}, \mathbf{\Theta}_1)\right|$
NN Surrogate Magnitude
x_2
x_1
10^7
10^4
10^1
10^{-2}
10^{-5}



g)
$|f(\mathbf{x})|$
Noisy Rosenbrock Magnitude
h)
$\left|\widehat{f_s}(\mathbf{x}, \mathbf{\Theta}_2)\right|$
NN Surrogate Magnitude
x_1
x_2
10^7
10^4
10^1
10^{-2}
10^{-5}


g)
$|f(\mathbf{x})|$
Noisy Rosenbrock Magnitude
h)
$\left|\widehat{f_s}(\mathbf{x}, \mathbf{\Theta}_2)\right|$
NN Surrogate Magnitude
i)
$|f(\mathbf{x})|$
Noisy Rosenbrock Magnitude
j)
$\left|\widehat{f_s}(\mathbf{x}, \mathbf{\Theta}_3)\right|$
NN Surrogate Magnitude
x_1
x_2
10^7
10^4
10^1
10^{-2}
10^{-5}

Summary

Summary

- NNAIF is a novel accelerated DFO method for unconstrained optimization problems.

Summary

- NNAIF is a novel accelerated DFO method for unconstrained optimization problems.
- NNAIF demonstrates how even crude neural network surrogate models of the objective function $f_s(\mathbf{x})$ can be used to improve the performance of IF optimization methods.

Summary

- NNAIF is a novel accelerated DFO method for unconstrained optimization problems.
- NNAIF demonstrates how even crude neural network surrogate models of the objective function $f_s(\mathbf{x})$ can be used to improve the performance of IF optimization methods.
- NNAIF maintains established convergence properties of IF methods.

Summary

- NNAIF is a novel accelerated DFO method for unconstrained optimization problems.
- NNAIF demonstrates how even crude neural network surrogate models of the objective function $f_s(\mathbf{x})$ can be used to improve the performance of IF optimization methods.
- NNAIF maintains established convergence properties of IF methods.
- Code and documentation available at: `https://github.com/0x4249/NNAIF`