Neural Fixed-Point Acceleration for Convex Optimization

Shobha Venkataraman* Brandon Amos*
Facebook AI Research

* Equal Contribution

Overview

Fixed-point problems are often computational bottleneck in large systems

Acceleration of fixed-point iterations: use knowledge of prior iterates to improve future iterates

Classically done without machine learning, e.g. Anderson Acceleration (AA)

Fixed-point problems repeatedly solved in application likely share structure

Neural fixed-point acceleration: leverage shared structure to accelerate distribution over problem instances using learning

Challenge: Complex fixed-point mappings result in subtleties with interweaving model updates with fixed-point computations, differentiating through mappings

Main Application: Convex Cone Programming

Accelerate Splitting Conic Solver, called SCS (state-of-the-art cone solver)

Neural SCS Design

Model:

- MLPs for initial prediction
- LSTMs/GRUs for prediction of sequence iterates

Differentiating through SCS Fixed-Point Iterations:

- Implicit differentiation for linear system solve
- Cone projection derivatives for zero, non-negative, second-order and PSD cones based on prior work.

Loss function:

- Tau normalization: Removing iterate-scaling factor from loss essential for good solution

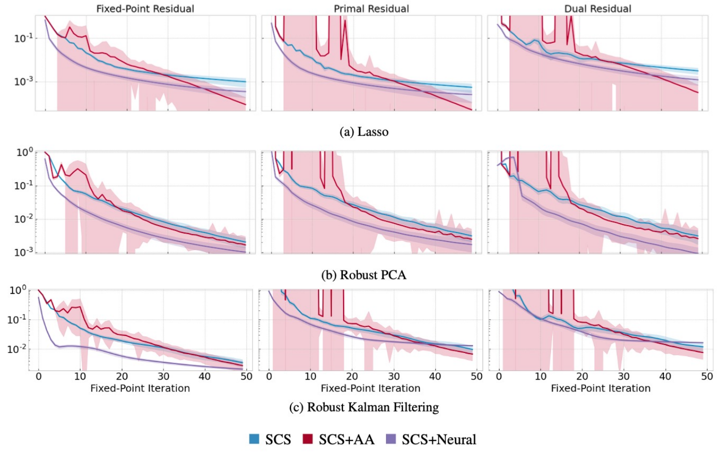
Neural Fixed-Point Acceleration Framework

Algorithm 1 Neural fixed-point acceleration augments standard fixed-point computations with a learned initialization and updates to the iterates.

Inputs: Context ϕ, parameters θ, and fixed-point map f.
$[x_1, h_1] = g_\theta^{\text{init}}(\phi)$ ▷ Initial hidden state and iterate
for fixed-point iteration $t = 1..T$ **do**
  $\tilde{x}_{t+1} = f(x_t; \phi)$ ▷ Original fixed-point iteration
  $x_{t+1}, h_{t+1} = g_\theta^{\text{acc}}(x_t, \tilde{x}_{t+1}, h_t)$ ▷ Acceleration
end for

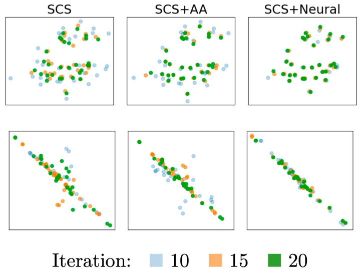
Experiments and Visualizations

Faster convergence to a good solution than SCS and SCS + Anderson Acceleration (default acceleration) on 3 problems: Lasso, Robust PCA, Robust Kalman Filtering

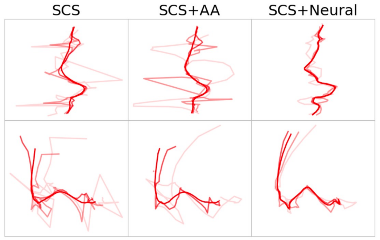


Visualizing iterates also shows that Neural-accelerated SCS stabilizes to a solution much faster than SCS/SCS+AA

Robust PCA



Robust Kalman Filtering



Solutions visualized after 5, 10, 20, 50 iterations