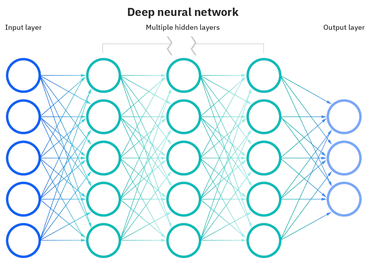
IF: Iterative Fractional Optimization

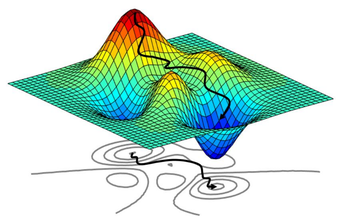
Sarthak Chatterjee[1], Subhro Das[2], and Sérgio Pequito[3]

[1]Department of ECSE, Rensselaer Polytechnic Institute, Troy, NY, USA; [2]MIT-IBM Watson AI Lab, Cambridge, MA, USA; [3]Delft Center for Systems and Control, Delft University of Technology, The Netherlands

Introduction and Motivation

- Many problems modeled as optimization problems seeking to find the minimum of an objective function $f: R^n \to R$
- For supervised learning, e.g., minimize a loss index that measures the performance of a neural network on a certain data set.
- In general, absence of numerically viable closed-form solutions for such optimization problems.
- Our go-to methods for solving these problems are, therefore, iterative optimization algorithms.





$$x^\star = \text{argmin}_{x \in R^n} f(x)$$

Fractional Calculus and ARFIMA Processes

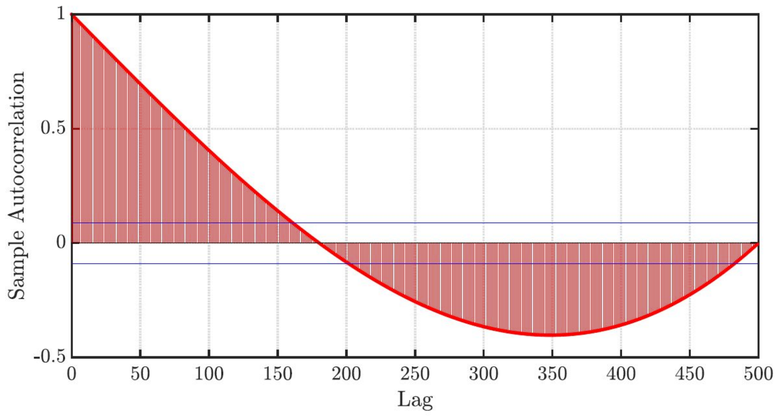
- In this work, we develop an iterative algorithm based on fractional calculus to solve unconstrained optimization problems.
- Fractional calculus: Generalization of ordinary calculus but to non-integer orders.
- Seek to answer questions such as, "What is the half-derivative of a function"?
- Widely used to model phenomena with long-term memory and power law dependence of trajectories.
- ARFIMA (Auto-Regressive Fractionally Integrated Moving Average) processes are a type of time series process that use the fractional derivative to model long-term memory.
- B is the backward shift operator, and the expansion $(1-B)^d = \sum_{j=0}^{\infty} \pi_j B^j$ where $\pi_0 = 1$ and $\pi_j = \frac{\Gamma(j-d)}{\Gamma(j+1)\Gamma(-d)}$
- Although the above is an infinite sum, in practice, we always consider a finite truncation.

ARFIMA: **A**uto-**R**egressive **F**ractionally **I**ntegrated **M**oving **A**verage

$$\underbrace{\left(1 - \sum_{i=1}^{p} \varphi_i B^i\right)}_{\text{auto-regressive}} \underbrace{(1-B)^{\overbrace{d}^{\text{fractional if } d \in \mathbb{R}}}}_{\text{integrated}} \underbrace{X_t}_{\text{generated timeseries}} = \underbrace{\left(1 - \sum_{i=1}^{q} \theta_i B^i\right)}_{\text{moving average}} \underbrace{\varepsilon_t}_{\text{white noise}}$$

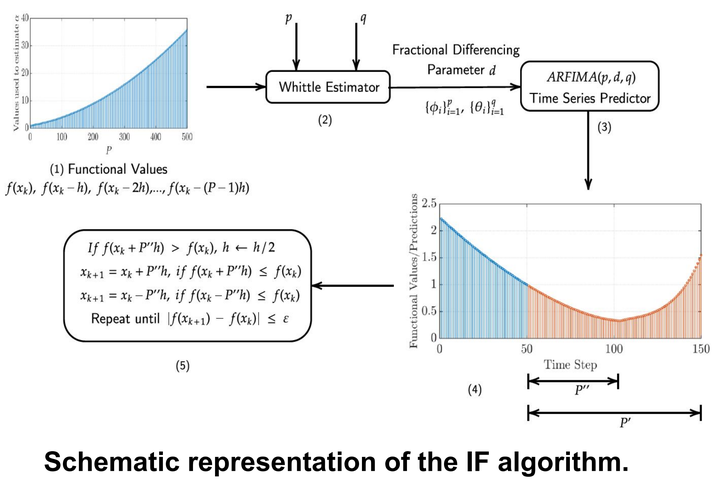
IF: Iterative Fractional Optimization Algorithm

- For the purpose of illustration, we consider the function $f(x) = x^2$, the grid discretization step $h \in R$, the number of steps of memory $P \in N$, and its corresponding functional values $f(x_0), f(x_0 - h), \dots, f(x_0 - (P-1)h)$
- First, we notice that the sample autocorrelation function (sACF) obtained from the aforementioned values suggests slower than exponential algebraic decay and statistically significant (for a significance level of 5%) dependency on past lags, with a large area enclosed by the composite sACF curve and the horizontal axis.
- This suggests that the ARFIMA processes described above can successfully predict the behavior of the functional values obtained.



sACF plot of the functional values $f(x_0), f(x_0 - h), \dots, f(x_0 - (P-1)h)$, with $f(x) = x^2, x_0 = -1, P = 500$, and $h = 0.01$.

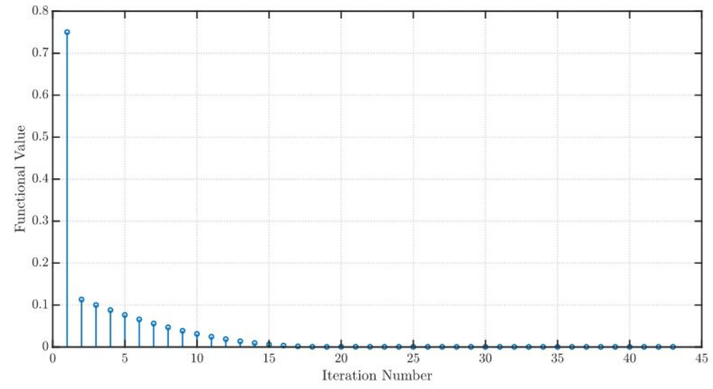
- First, we consider pre-specified p and q and find the fractional-order differencing parameter d from the Whittle estimation procedure.
- We predict using ARFIMA time series P' steps into the future to get the predicted function values $y_1, \dots, y_{P'}$
- Since the prediction capabilities are limited, we can only capture local behavior of evolution of functional values up to a certain number of time steps into the future.
- The IF algorithm works in a descent framework, i.e., we need to satisfy $f(x_{k+1}) \le f(x_k)$, so we select the turning point $P'' \le P$ where $y_1 \ge y_2 \ge \cdots \ge y_{P''} \le y_{P''+1}$
- Update the current iterate as $x_{k+1} = \begin{cases} x_k + P''h, & f(x_k + P''h) \le f(x_k) \\ x_k - P''h, & f(x_k - P''h) \le f(x_k) \end{cases}$



Schematic representation of the IF algorithm.

Result on a 2-D Problem

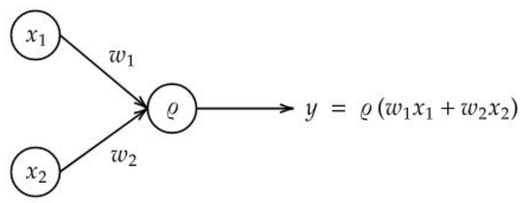
- We first minimize the function $f(x) = x_1^2 + 0.001x_2^2$. This objective function has a Hessian matrix $H = \begin{bmatrix} 2 & 0 \\ 0 & 0.002 \end{bmatrix}$ with a condition number of 1000. The starting point is chosen to be $\begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix}^T$.
- We use $P = 100$ steps of memory, an initial grid discretization step of $h = 0.01$ with $P' = 100$ steps ahead ARFIMA(4,d,0) time series predictions. The IF algorithm is able to attain convergence in 43 iterations while the gradient descent algorithm with inexact backtracking line search (to tune the step size) takes 3453 iterations to converge.
- This suggests the significant advantages of using the IF algorithm in problems where the Hessian is ill-conditioned.



Convergence profile showing the evolution of functional values with iteration number when the IF algorithm is used to find the minimizer of $f(x) = x_1^2 + 0.001x_2^2$.

Result on a Feedforward Neural Network

- Feedforward neural nets often possess ill-conditioned Hessians and thus constitute an interesting test bed of problems.
- We consider the following single-layer perceptron



- Assume that the activation function is the Gaussian Error Linear Unit (GELU) given by $\varrho(s) = \frac{s}{2}\left(1 + \text{erf}\left(\frac{s}{\sqrt{2}}\right)\right)$.
- The weights are arbitrarily initialized such that $w_1^2 + w_2^2 = 1$.
- Consider the arrival of a single training sample $(x_1, x_2, t) = (1,1,0)$, where t is the true output to be produced as a result of proper training.
- Assuming a squared error loss function to be minimized using IF, we obtain convergence in 18 iterations with $w^\star = [-1.0144 \quad 0.9997]^T$ and the optimal value of the loss function $L^\star = 5.2488 \times 10^{-5}$.
- In contrast, gradient descent with inexact backtracking line search takes 60 iterations to converge for the same initialization of the weights.

References:

[1] Figure from: https://juliahub.com/docs/ARFIMA/P6HqW/0.4.0/_packagesource/ARFIMA.png

[2] Chatterjee, S., Das, S. and Pequito, S., NEO: NEuro-inspired Optimization-A Fractional Time Series Approach. *Frontiers in Physiology*, p.1551.