FractionalNets - A Neural Network for Approximating Fractional Derivatives

Dhanush Biligiri[1], Shivayogi Akki[1], Jack Anders Smitterberg[1], Sujan Kumar Roy[2], Tan Chen[1]*

[1] are with the Department of Electrical and Computer Engineering and [2] are with the Department of Computer Science ,Michigan Technological University, Houghton, MI 49931, USA.

**Partial support for this work under the ORAU Ralph E. Powe Junior Faculty Enhancement Award is gratefully acknowledged*

INTRODUCTION

- What are Fractional derivatives(FD)?
  - Derivative for any arbitrary non–integer order (i.e 0.5, 0.25)
  - Do not have a simple interpretation like integer-order derivatives.
  - Fractional derivatives measure memory effects – Past values influence present derivatives (Difficult to interpret)
- How do we calculate FD?
  - Notably **Riemann-Liouville (RL)** and **Caputo definition**

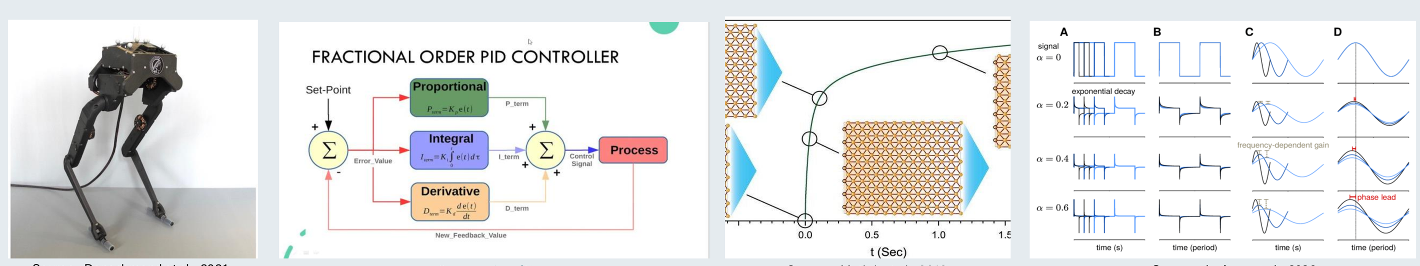
$$D_a^{\alpha}(f)(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_0^t \frac{f(x)}{(t-x)^{\alpha-n+1}} dx$$

Riemann-Liouville

$$D_a^{\alpha}(f)(t) = \frac{1}{\Gamma(n-\alpha)} \int_0^t \frac{f^{(n)}(x)}{(t-x)^{\alpha-n+1}} dx$$

Caputo

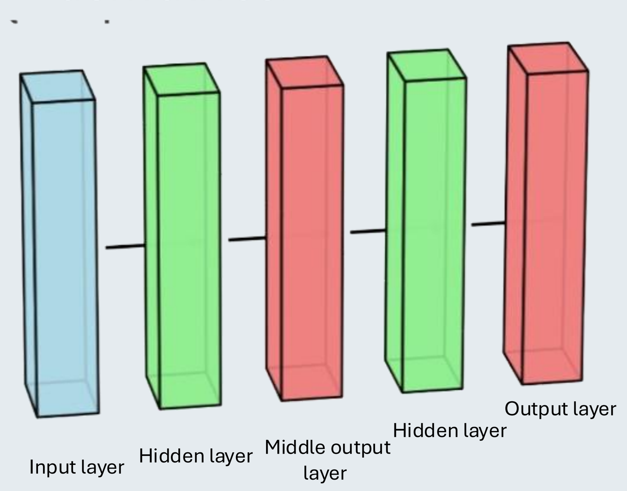
- Fixed-formula FD computation limits accuracy and adaptability.
- Why are FDs important –
  - FDs accurately describe memory effects in materials and fluid dynamics improving predictive models.
  - Used in robotics and circuit design for better stability, noise reduction, and adaptive control in engineering applications.



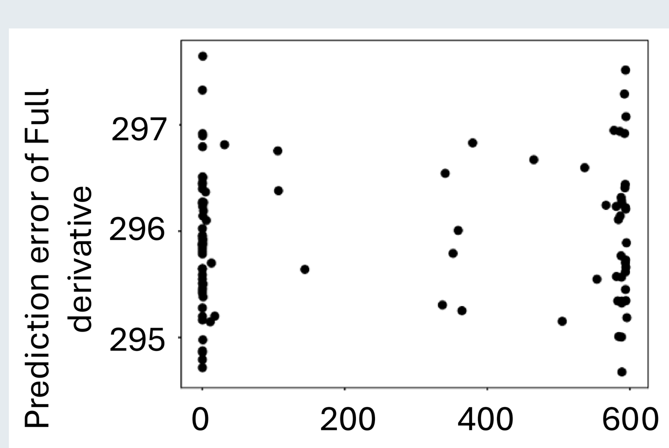
Real-world applications of fractional order derivatives in robotics, control systems, material modeling, and signal processing.

RELATED WORK

- Existing Method - **A Symmetric Neural Network to Compute Fractional Derivatives by Training with Integer Derivatives**[1]
  •A **symmetric neural network** was trained using **integer derivatives only**.
  •The **middle layer learns the half-derivative implicitly**.
  •The **middle layer** in the network acts as an interpolator in the derivative space, naturally approximating the half-order derivative by learning the transition between integer derivatives.



Model structure

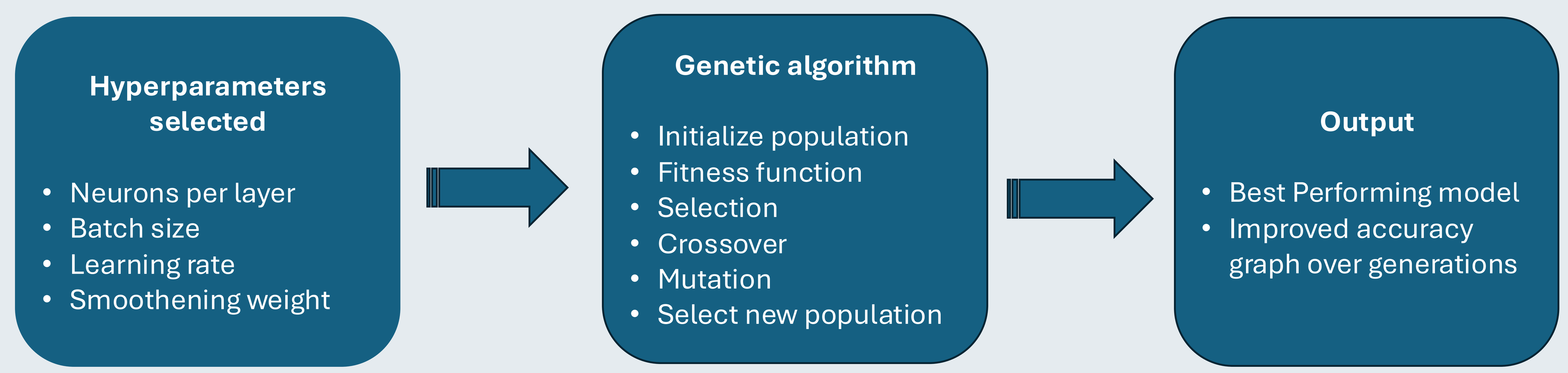


Models' performance

•**Limitations:**
  •Only **80% accuracy** on polynomials.
  •Fixed hyperparameters – **not optimized**.
  •The **effect of weight initialization** was **not studied**.

PROPOSAL

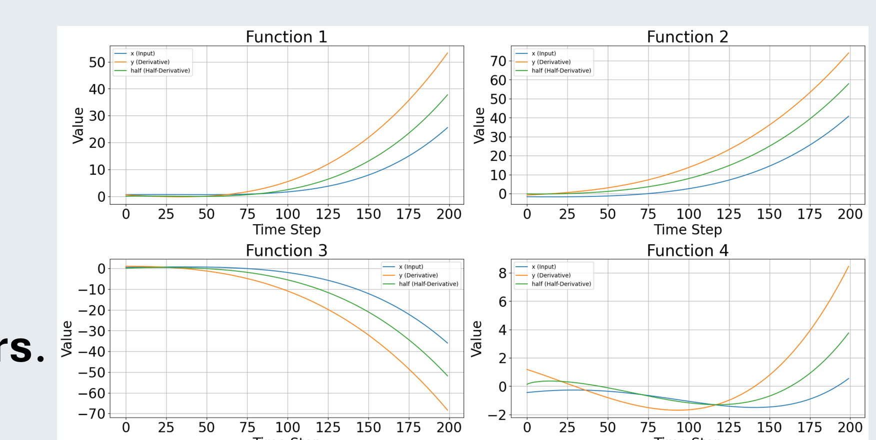
- Goal for FractionalNets –
  •**Improve FD prediction beyond 80%** using **Genetic Algorithms (GA)** and **Weight Initialization**.
  •**Optimize hyperparameters dynamically** instead of using fixed values.



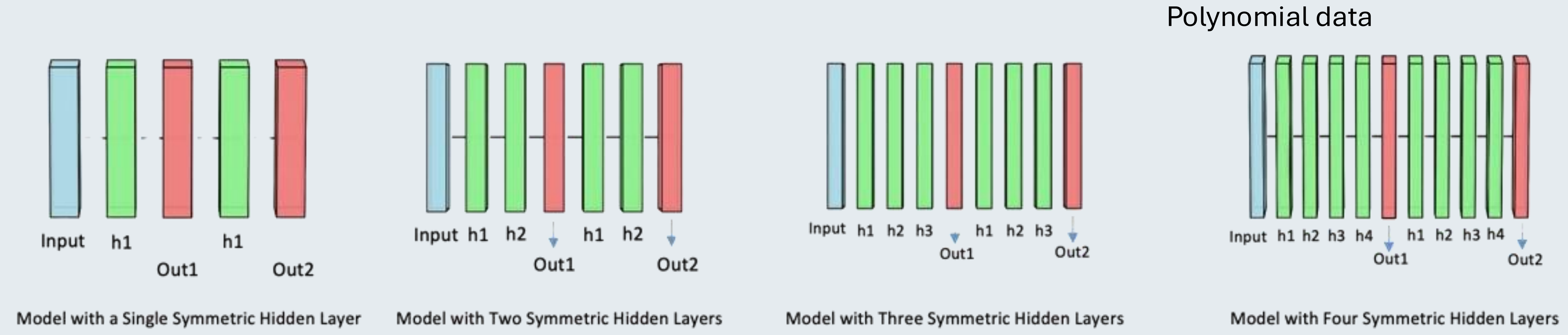
•Key Assumptions –
  •**Hyperparameter Dependence:** Model performance depends on **optimized hyperparameters via GA**.
  •**Weight Initialization Dependence:** Performance improves with **better initial weight selection**.

METHODOLOGY

- **Data Generation**–
  - **10,000 polynomials** generated.
  - Random coefficients in **[-2,2]**, Order = **4**.
- **Model Architecture** –
  - Tested **1, 2, 3, and 4 symmetric hidden layers**.
  - **Middle layer represents the half-derivative**



Polynomial data



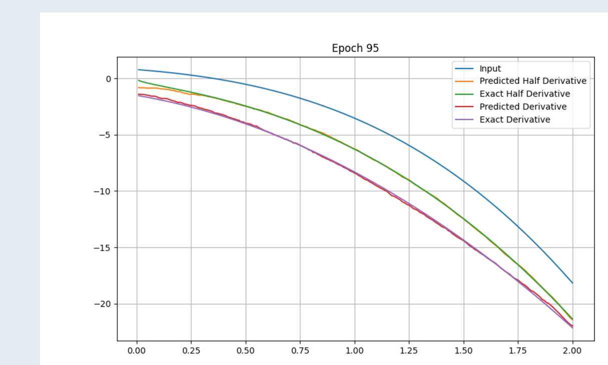
- **Genetic Algorithm (GA) for Hyperparameter Optimization –**
  - Genetic Algorithm is an optimization technique inspired by **natural selection.**
  - It iteratively evolves solutions to find the best one over multiple generations.

GA Process

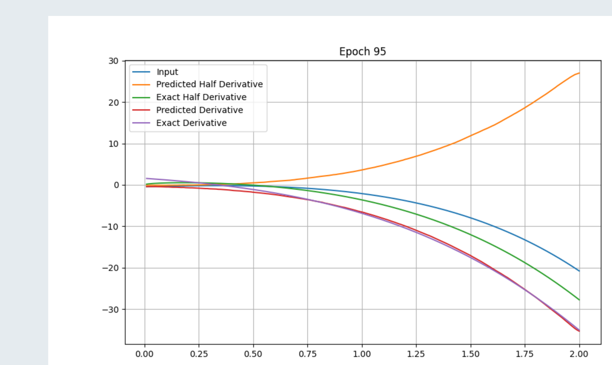
[1] Chen, T., & Goodwine, B. (2022, January). A symmetric neural network to compute fractional derivatives by training with integer derivatives. In *2022 IEEE/SICE International Symposium on System Integration (SII)* (pp. 291-296). IEEE.
[2] Daneshmand, E., Khadiv, M., Grimminger, F., & Righetti, L. (2021). Variable horizon mpc with swing foot dynamics for bipedal walking control. *IEEE Robotics and Automation Letters*, *6*(2), 2349-2356.
[3] Images from - https://www.youtube.com/watch?v=UZmGyASK7Zw
[4] Latimer, K. W., & Fairhall, A. L. (2020). Capturing multiple timescales of adaptation to second-order statistics with generalized linear models: gain scaling and fractional differentiation. *Frontiers in systems neuroscience*, *14*, 60.
[5] Matlob, M. A., & Jamali, Y. (2019). The concepts and applications of fractional order differential calculus in modeling of viscoelastic systems: A primer. *Critical Reviews™ in Biomedical Engineering*, *47*(4).

DISCUSSIONS & RESULTS

- Inverse FD Predictions Are Valid –
  - Some polynomials **have valid inverse half-derivatives. Applying the inverse still results in the correct first derivative.**



Direct Prediction

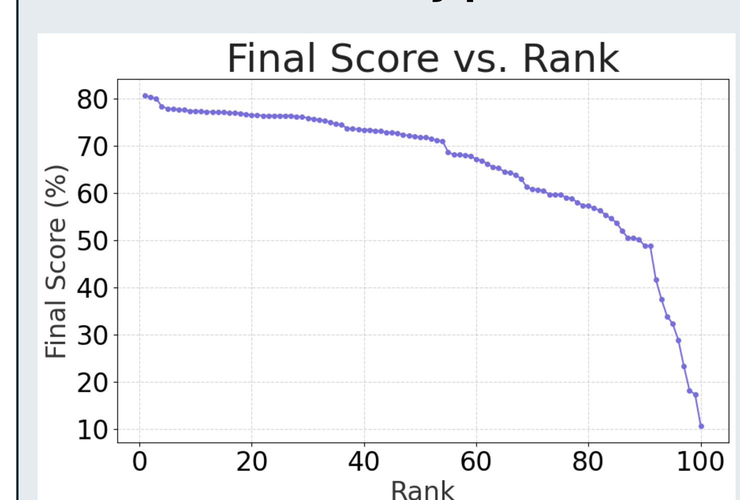


Inverted Prediction

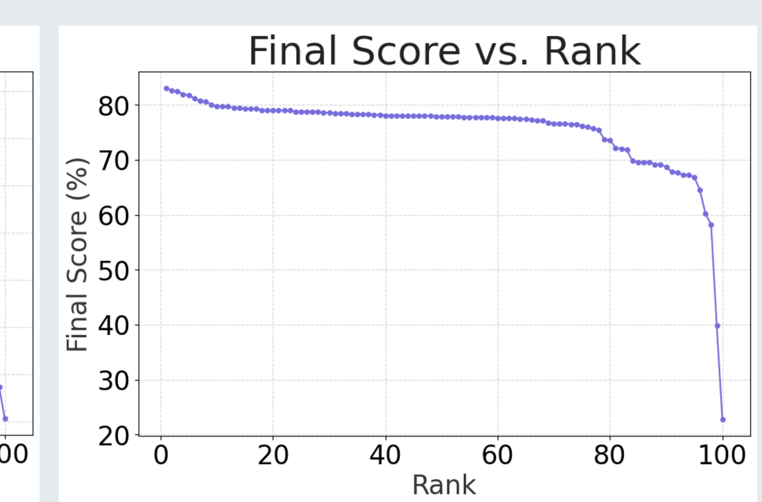
- Calculation Metrics – **Final Score**
  - We define accuracy based on how well the predicted half-derivative matches the actual one using AUC error ratios.

$$\text{Error Ratio} = \frac{\text{Selected AUC}}{\text{Total Area of Actual Half-Derivative}}$$

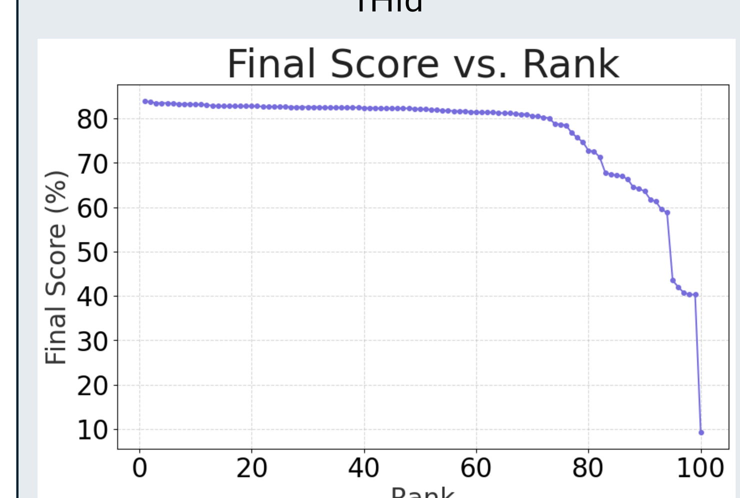
- Genetic Algorithm Optimization –
  - Ran **100 generations** of different models.
  - Found that **3 Hidden Layers (400, 600, 800 neurons) consistently performed best.**



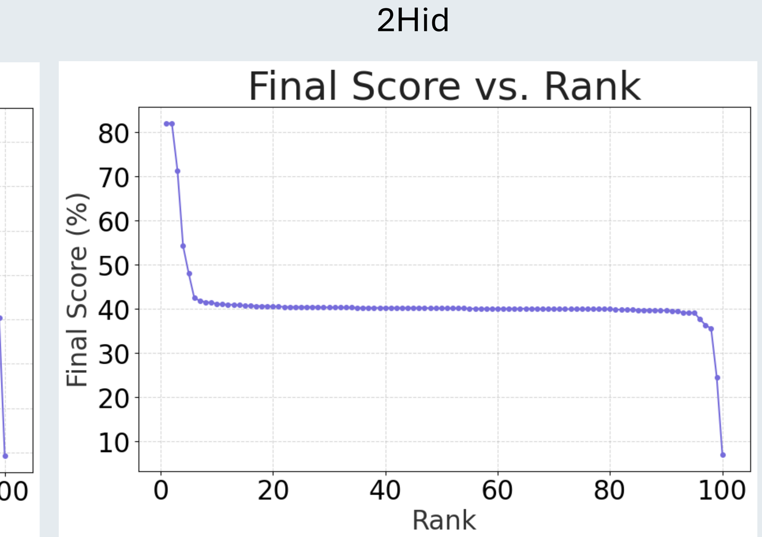
1Hid



2Hid



3Hid



4Hid

Weight Initialization and Future work

PyTorch defaults to Kaiming He Uniform Initialization. Testing Kaiming Normal, Xavier Glorot, and Orthogonal Initialization to push FD approximation accuracy beyond 80% using neural networks + Gas

Future work –

- Expand dataset: sinusoids, exponentials functions.
- Investigate finer fractional derivatives: **0.25, 0.1.**
- Develop an **API for FD computation** for broader accessibility.