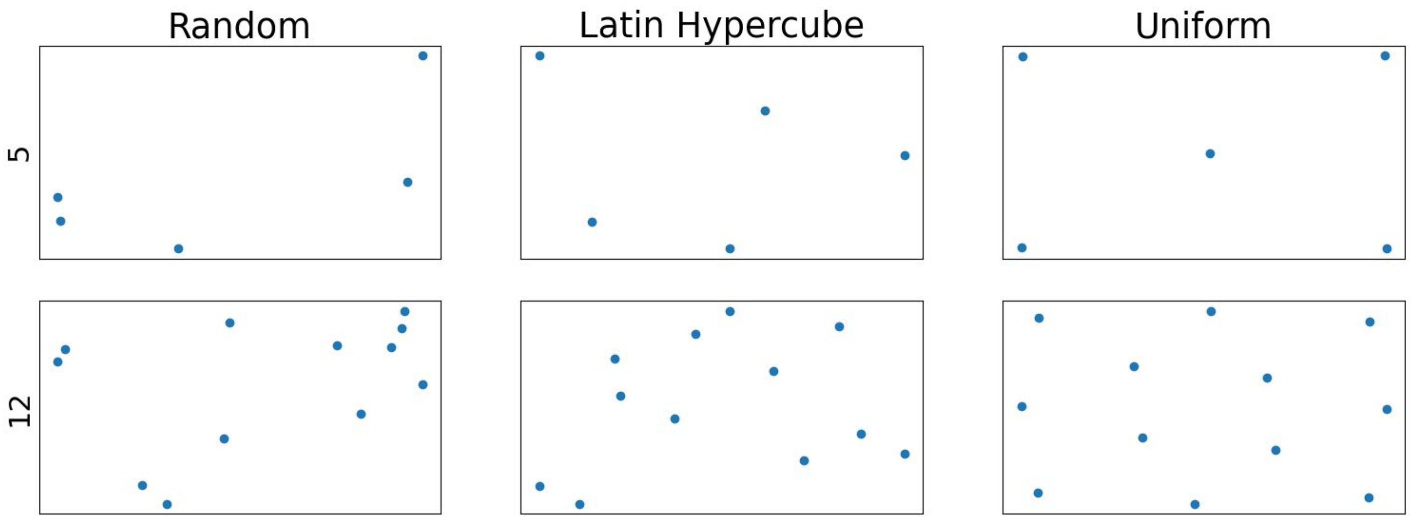
Carnegie Mellon University

Evolving Hyperparameters:
Learning Enhanced Model Training

Braden Eichmeier, Shaun Ryer, Stefan Zhu

Background - Sampling Techniques

- Find hyperparameters using dense sampling:
  - Large holes often exist in the search space



Background - Genetic Algorithms

1. Set-up model for learning task
2. Select initial generation of hyperparameters
   - Domain Knowledge
   - Sparse Sampling
3. Train models on learning task
4. Genetic algorithm selects new batch of parameters
   - Select Parents, Cross-Over, Mutation
5. Repeat steps 2-4 as desired

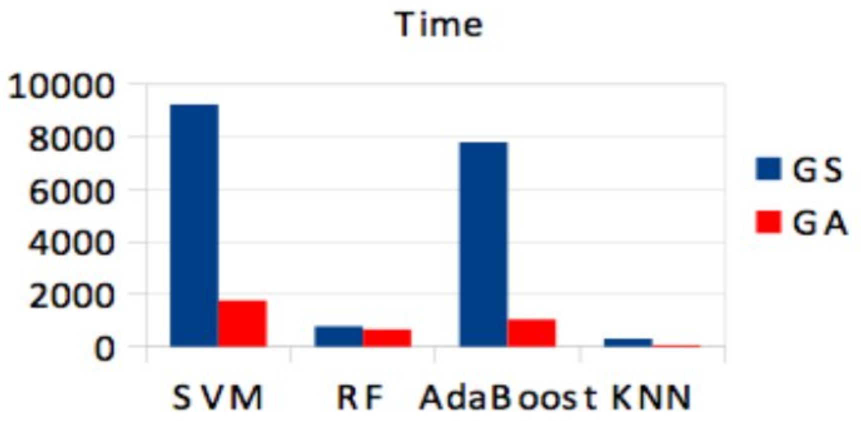
Background - Genetic Algorithms

- Used to generate high-quality solutions to optimization and search problems

- GAs can be used to tune learnable parameters
    - traffic light management
    - hyperparameter selection in Deep Reinforcement Learning for a manipulation task

Related Works - Hyperparameter Tuning

- SVM, KNN, AdaBoost, Random Forests [1]
- Convolutional Networks [2] [3]
- Random Initialization



[1] Hyper Parameter Optimization using Genetic Algorithm on Machine Learning Methods for Online News Popularity Prediction
[2] Efficient Hyperparameter Optimization In Deep Learning Using A Variable Length Genetic Algorithm
[3] Speeding up the Hyperparameter Optimization of Deep Convolutional Neural Networks

Related Works - Training RL with GAs

- Directly train neural network weights in an RL task
- Train 4 million parameter network
- Resultant training is faster than A3C and DQN

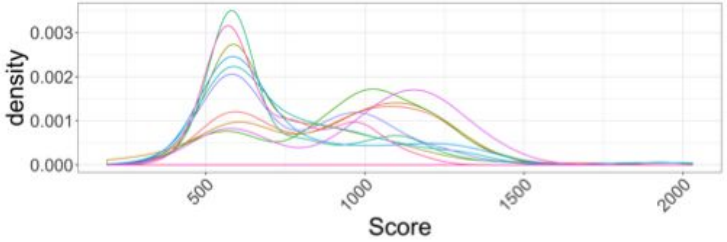
	DQN	ES	A3C	RS	GA	GA
Frames	200M	1B	1B	1B	1B	6B
Time	~7-10d	~ 1h	~ 4d	~ 1h or 4h	~ 1h or 4h	~ 6h or 24h
Forward Passes	450M	250M	250M	250M	250M	1.5B
Backward Passes	400M	0	250M	0	0	0
Operations	1.25B U	250M U	1B U	250M U	250M U	1.5B U

Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O. Stanley, and Jeff Clune.Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning, 2018

RL Environments/Heavy Randomness

Deep reinforcement learning faces substantial and unusual challenges in evaluation and reproducibility

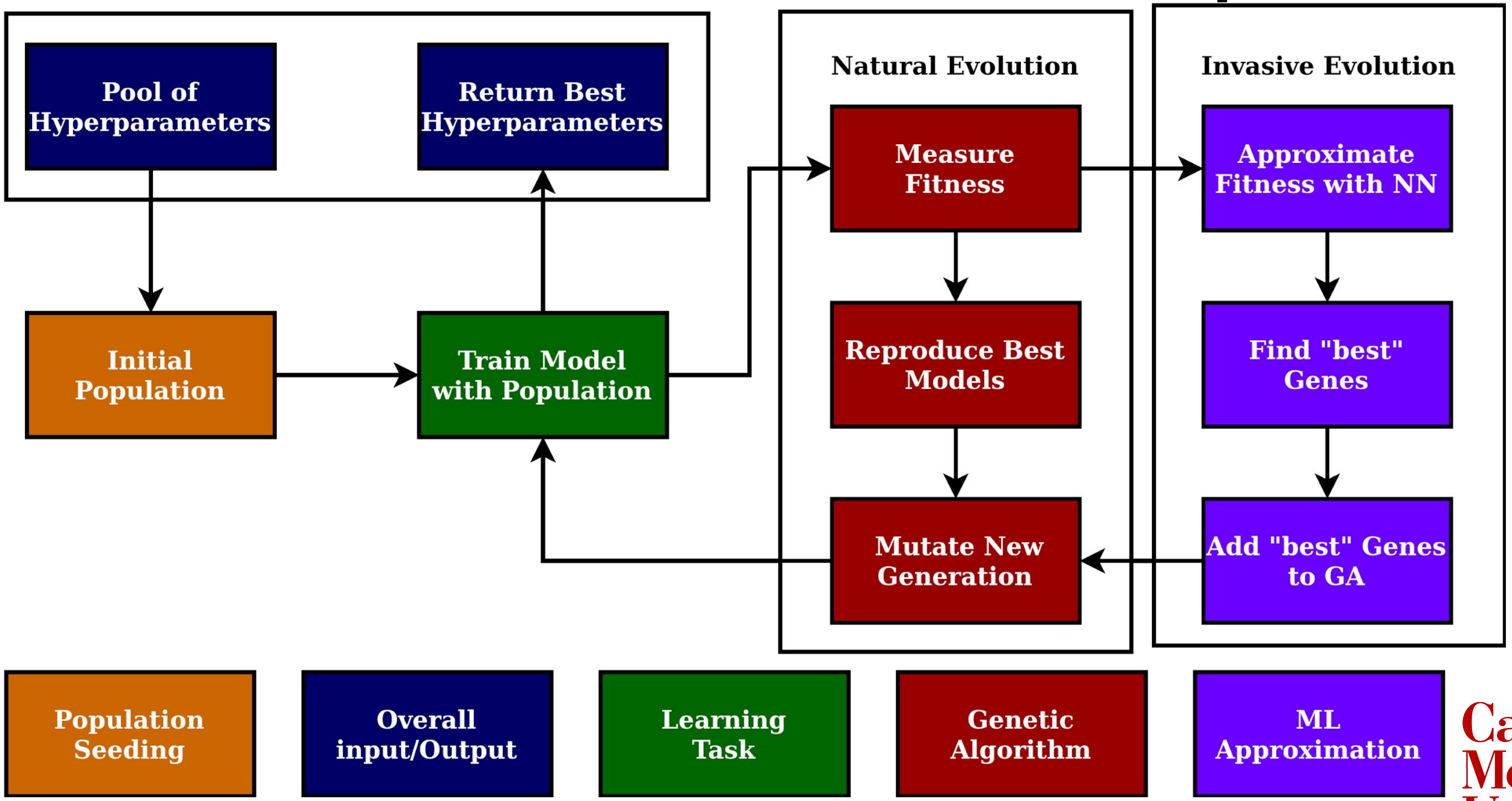
- *Let's Play Again: Variability of Deep Reinforcement Learning Agents in Atari Environments*



Process Overview

1. Use GA to search for best solution (Natural Evolution)
2. Approximate loss function with NN (Invasive Evolution)
    - Densely sample search space
    - Predict which genes will return the lowest loss
    - Use results to create *n* children
3. Create next generation by mixing GA and NN children

Process Overview - GA with Invasive Species



Carnegie Mellon University

Results - Test Case

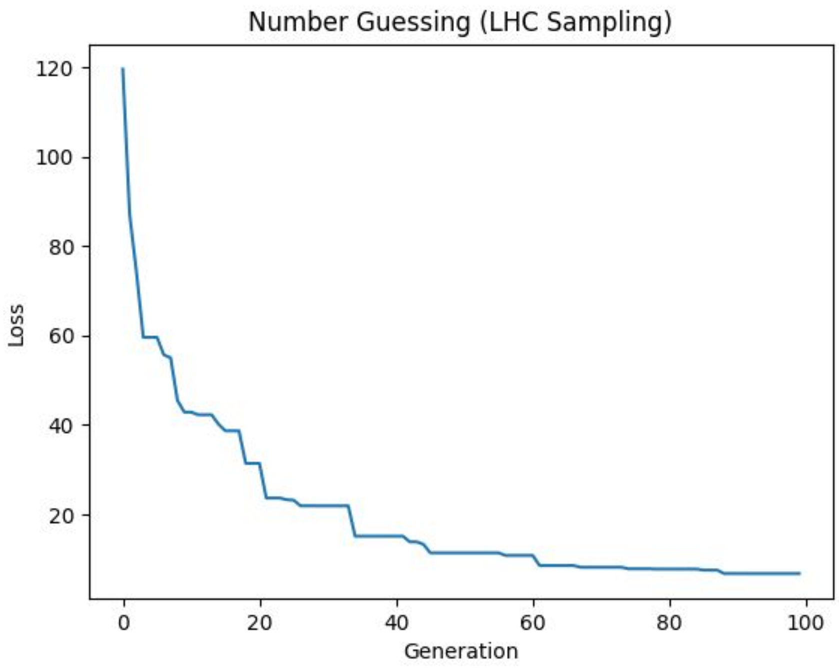
Task: Guess 10 numbers [0,99]

GA setup:

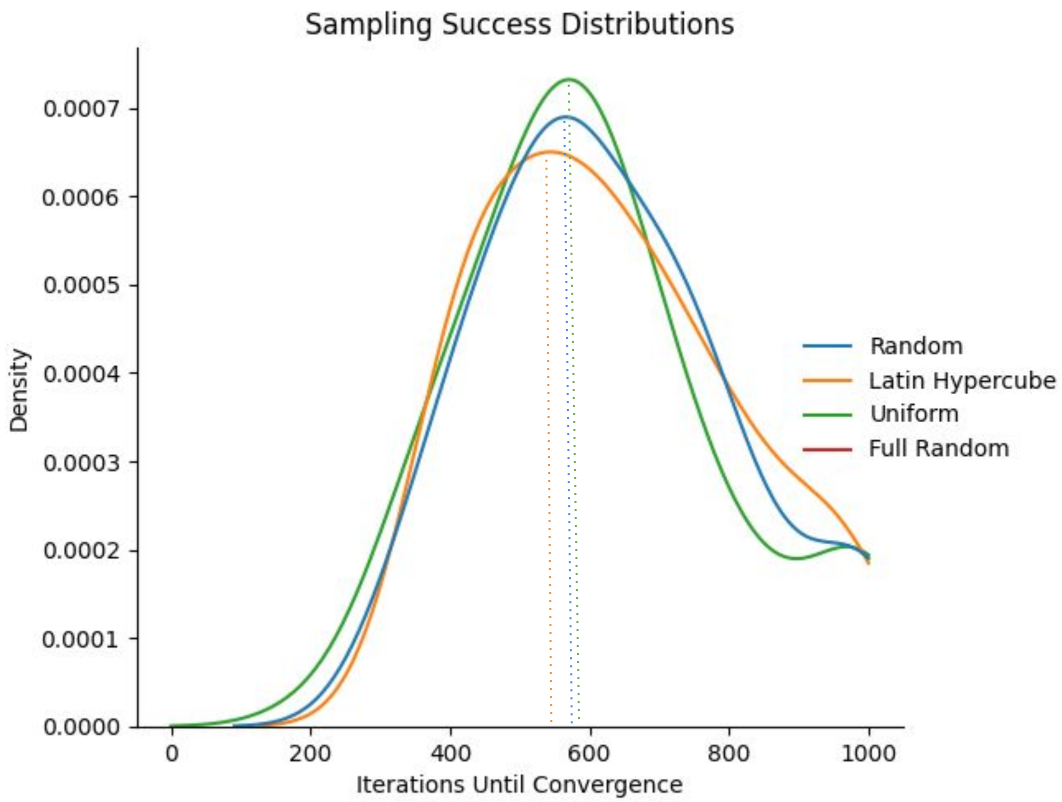
Initial Population: 10

Number Parents: 4

Number Mutations: 1



Results - Different Sampling Methods



Results - Neural Networks

Model: MLP

Env: Handwritten digit classification

Task: Optimize 2 hyperparams

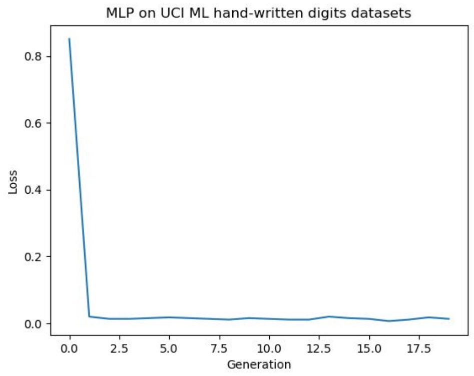
1) Learning Rate
2) Momentum

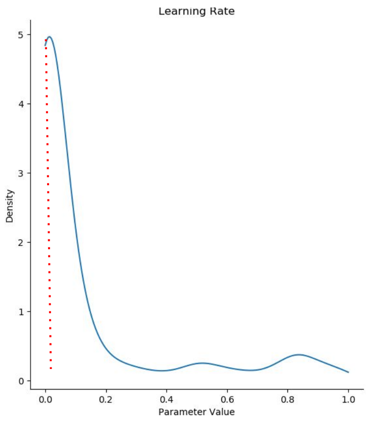
GA setup:

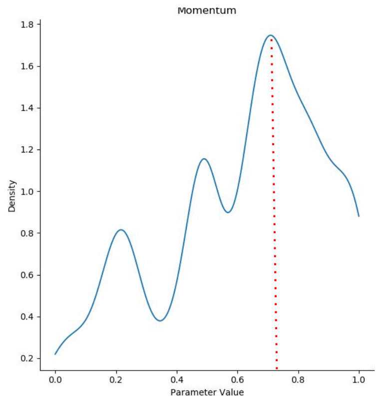
Initial Population: 10

Number Parents: 4

Number Mutations: 1







Results - RL

Model: PPO2

Env: Cart-Pole

Task: Optimize 2 hyperparams

1) Learning Rate
2) Maximum Gradient Norm

GA setup:

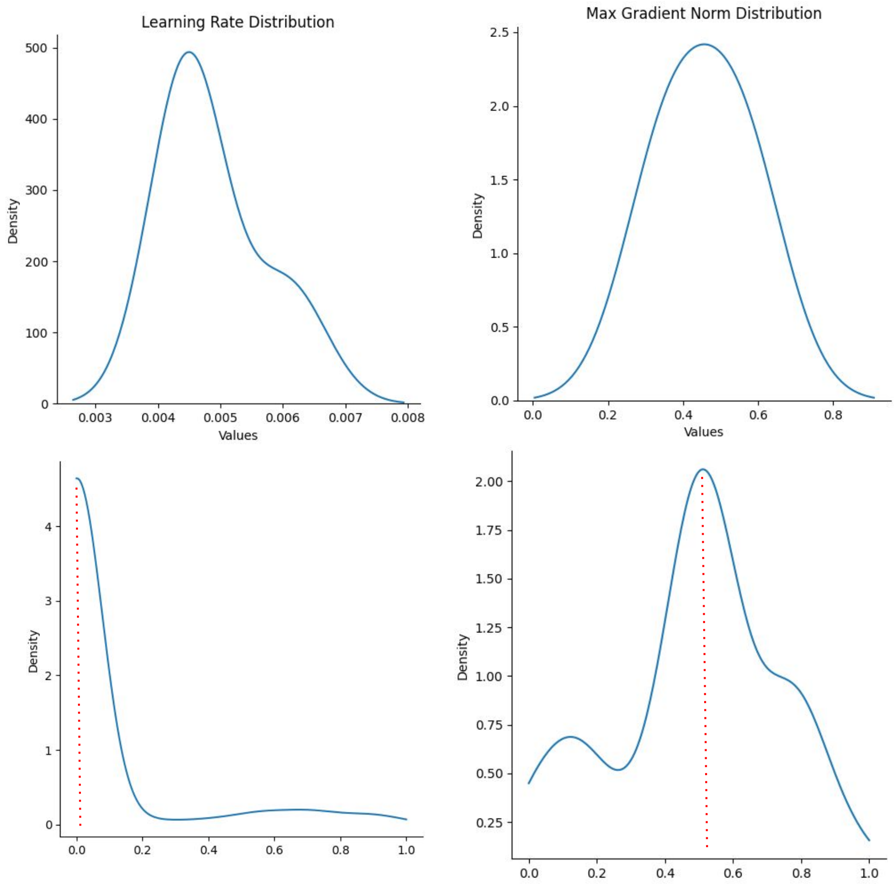
Initial Population: 8

Number Parents: 4

Number Mutations: 1

10 Generations

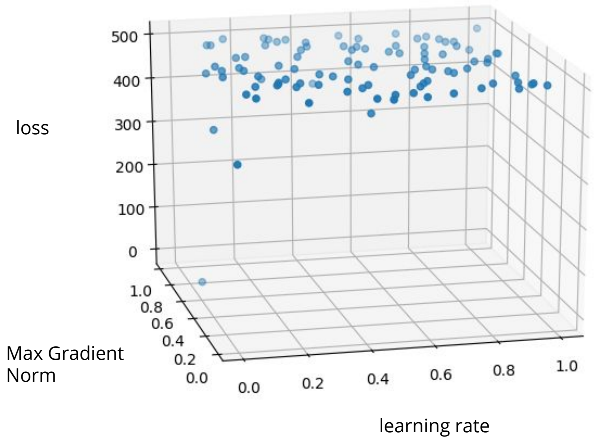
50 Generations



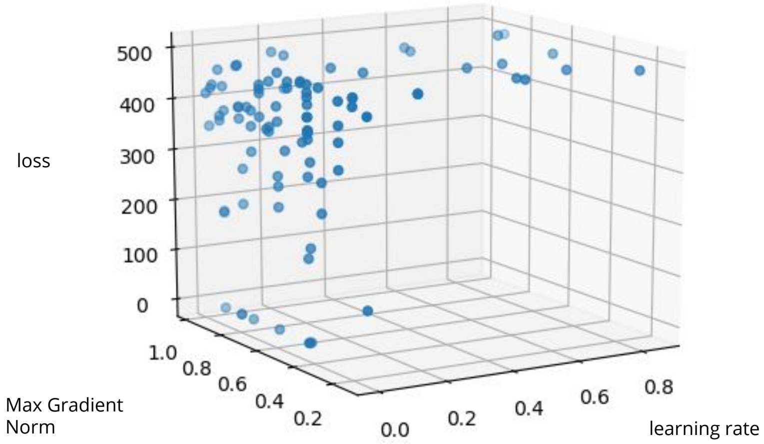
Carnegie Mellon University

Results - Improvement on Random Sampling

= 100 random samples



= 100 samples from GA



The GA focuses sampling in low cost regions, this helps provide better results when compared to random guessing

Conclusion and Future Work

Conclusion:

- Novel GA framework with Invasive Species
- Optimization with deterministic and non-deterministic models
- Significant improvement over random sampling

Future Work:

- Evaluate performance against standard GA
- Explore other ML models for Invasive Species

Questions?