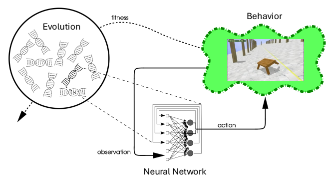
Neuroevolution Tutorial

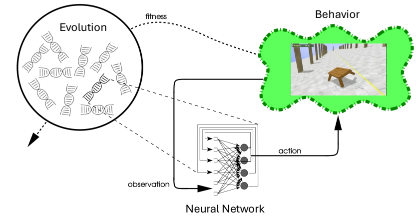
IJCAI 2025

Risto Miikkulainen
The University of Texas at Austin and Cognizant AI Lab

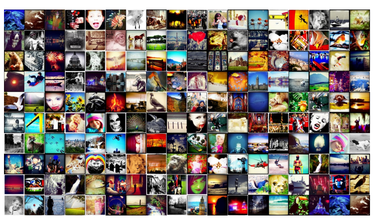
With Sebastian Risi, David Ha, and Yujin Tang

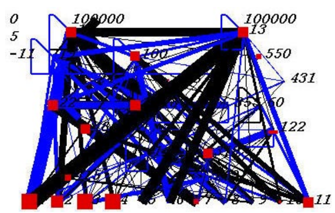


Outline



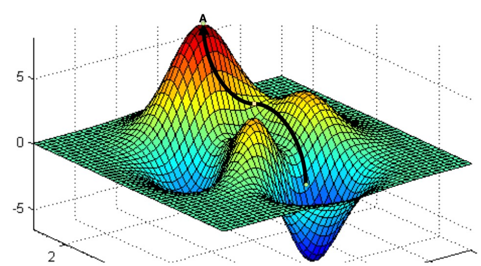
1. Motivation: From Imitation to Creativity





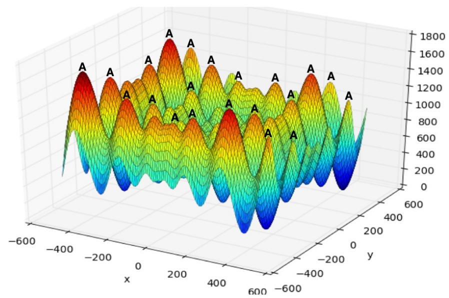
- Much of AI so far focuses on imitation
  - I.e. gradient descent on labeled datasets
  - Powerful in prediction: object recognition, diagnosis, forecasting, etc.
- Agentic AI focuses on behavior
  - Gradients not available
  - Needs to be discovered
- How can we create novel behaviors?

Reinforcement Learning is One Approach



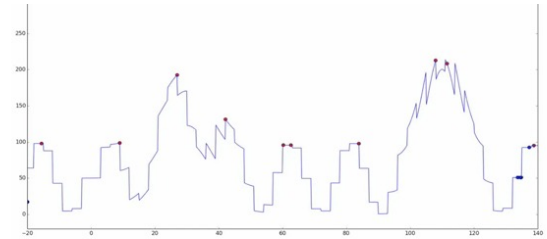
- Approximating gradient descent
  - Explore around the current solution
  - Improve it gradually
  - Can climb the nearest hill well

...but Creativity in RL is Limited



- Space is too large
  - Multiple starts won't help
- Space is too high-dimensional
  - Little improvement from one step
- Space is deceptive
  - Can only find the nearest hill

Solution: Population-based Search



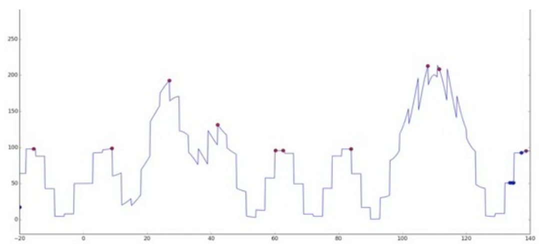
A.k.a Evolutionary Computation

- Many individuals spread out, sharing information
- Not limited to differentiable domains: configurations, choices ok

Not limited to incremental improvement

- Large jumps possible, can be more creative

Scaling up through Evolution



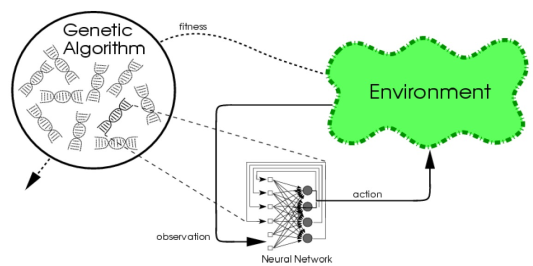
Works in large scales

- Structured search works in large spaces (e.g. 2^2^70; Hodjat & Shahrzad 2016)
- Multiple variables optimized at once (e.g. up to 1B; Deb et al. 2017)
- Multiple objectives and novelty get around deception (Shahrzad and Hodjat 2020)

Neuroevolution uses population-based search to optimize neural netwoks

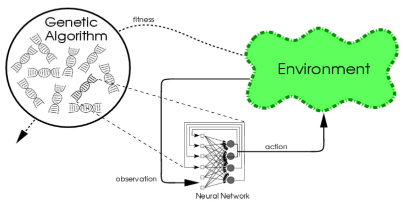
- Weights, topologies, designs

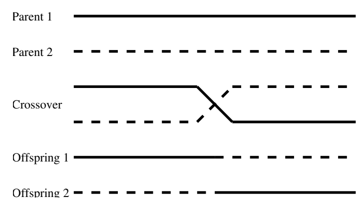
2.1 Evolution Basics: Encoding, Evaluation, and Selection



- A population of encodings (e.g. lists or trees)
- Decoded into individuals that are evaluated in the domain
- Good individuals retained, bad thrown away

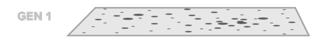
Creating Variation



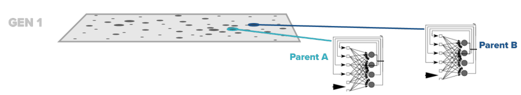


- New individuals generated from the parent encodings
  - Crossover: combine building blocks from two parents
  - Mutation: create new building blocks

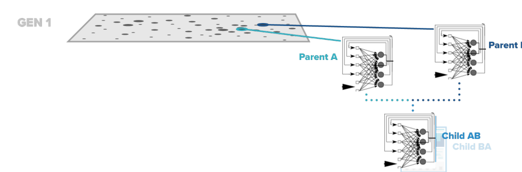
Population-based Search



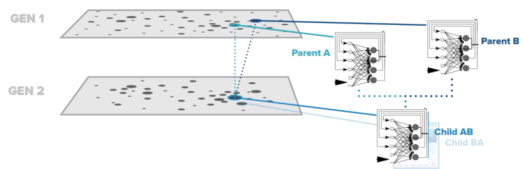
Population-based Search



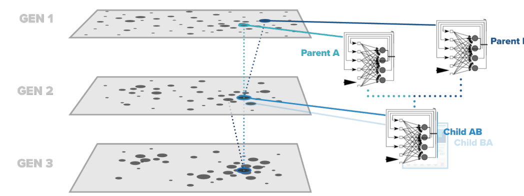
Population-based Search



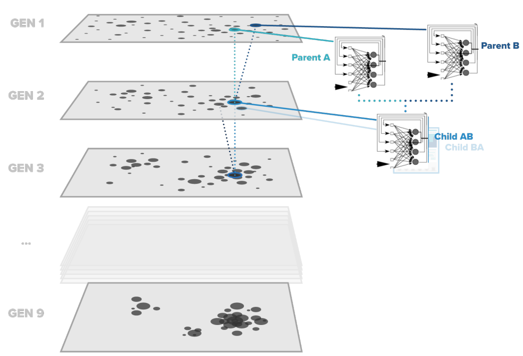
Population-based Search



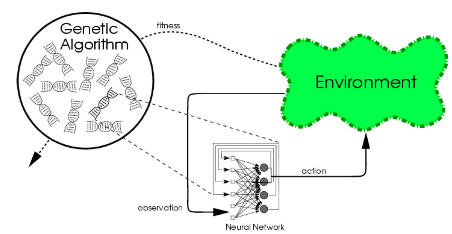
Population-based Search



Population-based Search

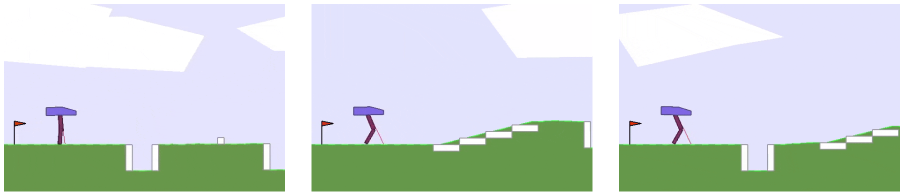


2.2 Basic Neuroevolution



- Evolving connection weights in a population of networks
- Chromosomes are strings of connection weights (bits or real)
  - E.g. 10010110101100101111001
  - Usually fully connected, fixed, initially random topology
- A natural mapping between genotype and phenotype
  - GA and NN are a good match!

Example: Learning to Walk

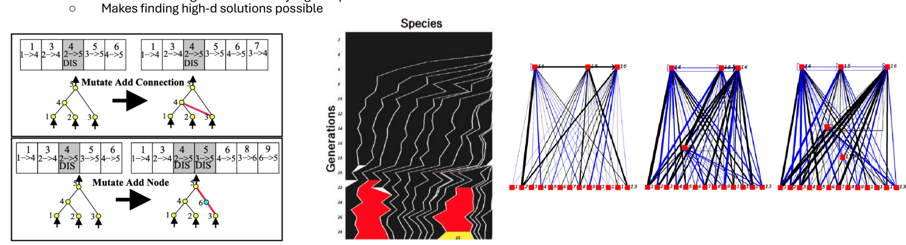


Demo

Advanced Neuroevolution

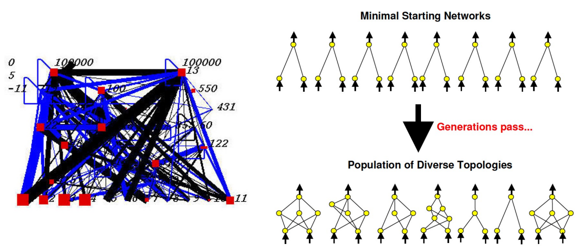
E.g. Neuroevolution of Augmenting Topologies (NEAT)

- *Historical markings* match up different structures
- Speciation
  - Keeps incompatible networks apart
  - Protects innovation
- Incremental growth from minimal structure, i.e. *complexification*
  - Avoids searching in unnecessarily high-d space
  - Makes finding high-d solutions possible



(Stanley and Miikkulainen, 2002)

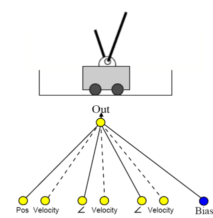
Why Is It a Good Idea?

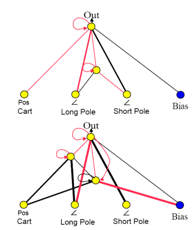


- NN search space is complex with nonlinear interactions
- Complexification keeps the search tractable
  - Start simple, add more sophistication
- Incremental discovery of complex solutions

Discovering Compact, Interpretable Structure

- E.g. in double pole balancing
  - Easy when position, velocity of both poles and the cart are given
  - Hard when only positions: need to figure out how they are moving
- Discovers recurrent structure
  - Either representing velocities separately
  - Or simply the derivative of the difference of the poles!
- Big improvement from other approaches
  - Standard value-function RL unsuccessful





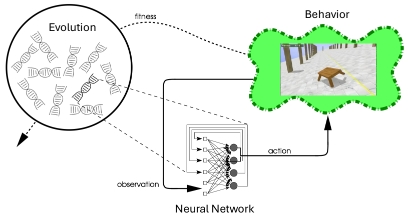
Method	Evaluations
Gruau et al. 1996 Cellular Encoding	840,000
Gomez and Miikkulainen 1999 ESP	169,466
NEAT	33,184

20 trials

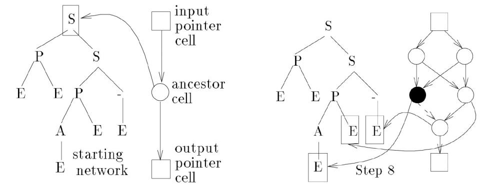
Outline

1. Motivation for Neuroevolution
2. Basics
   1. Fundamentals of Evolution
   2. Fundamentals of Neuroevolution
3. Advances
   1. Taking Advantage of Indirect Encodings
   2. Taking Advantage of Diversity
4. Evolving Intelligent Agents
   1. Control
   2. Strategy
   3. Decision-making
   4. Collective behavior
5. Synergies with other ML
   1. Deep Learning
   2. Reinforcement Learning
   3. LLMs
6. Insights into Biology
7. Conclusion

Hands-on exercise (off-line)

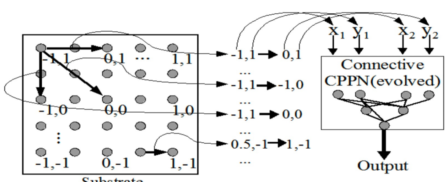


3.1 Indirect Encoding: (A) Development



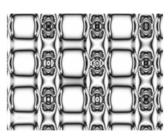
- Instructions for constructing the network evolved
  - Instead of specifying each unit and connection
- E.g. Cellular Encoding (CE Gruau & Whitley 1993)
- Grammar tree describes construction
  - Sequential and parallel cell division
  - Changing thresholds, weights
  - A "developmental" process that results in a network

3.1 Indirect Encoding: (B) Hypernetworks

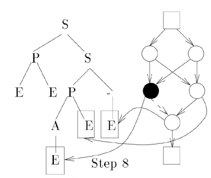


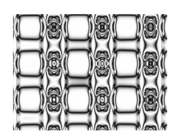
(Stanley et al. 2009)

- Encode the networks as spatial patterns
- E.g. Hypercube-based NEAT (HyperNEAT)
- Evolve a neural network (CPPN) to generate spatial patterns
  - 2D CPPN: (x, y) input → grayscale output
  - 4D CPPN: (x_1, y_1, x_2, y_2) input → w output
  - Connectivity and weights can be evolved indirectly
  - Works with very large networks (millions of connections)



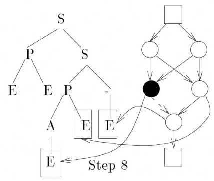
Why are Indirect Encodings a Good Idea?

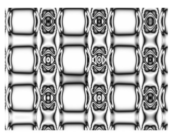




- Describes structure efficiently
  - Recurrency symbol in CE: XOR → parity
  - Repetition with variation in CPPNs
- Useful for evolving topology
  - E.g. large structured networks
  - E.g. repetition of motifs

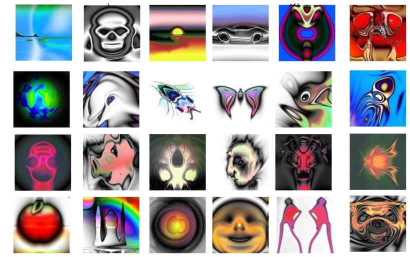
Future Opportunities





- Several possible directions
  - More general L-systems; developmental codings; embryogeny
  - Scaling up spatial coding
  - Genetic Regulatory Networks
  - Evolution of symmetries
- Theory starting to emerge
  - Expressive Encodings Simple GAs are universal probability approximators

3.2 Diversity: (A) Searching for Novelty

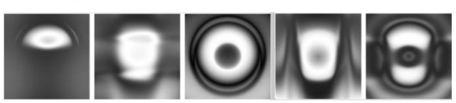


- Motivated by humans as fitness functions
- E.g. picbreeder.com, endlessforms.com (Secretan et al. 2011; Clune et all 2011)
  - CPPNs evolved; Human users select parents
- No specific goal
  - Interesting solutions preferred
  - Similar to biological evolution?

Novelty Search

Gen 12 Gen 20 Gen 36 Gen49 Gen 74

(a) Intermediate images in the evolution of the skull image

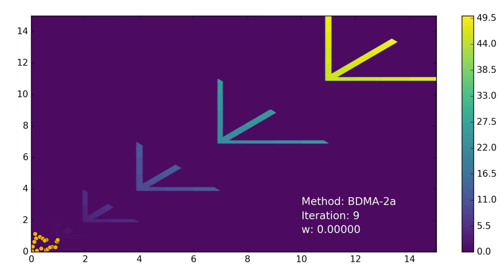


Run 1 Run 3 Run 7 Run 15 Run 17

(b) Attempts to evolve the skull image directly

- Evolutionary algorithms maximize a performance objective
  - But sometimes hard to achieve it step-by-step
- Novelty search rewards candidates that are simply different
  - Stepping stones for constructing complexity

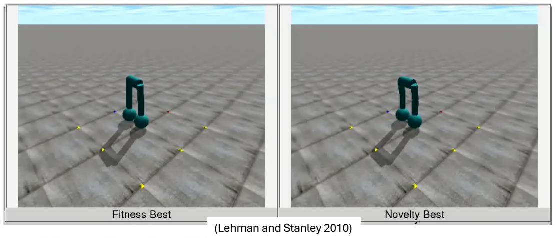
Novelty Search Demo 1



(Meyerson & Miikkulainen 2017)

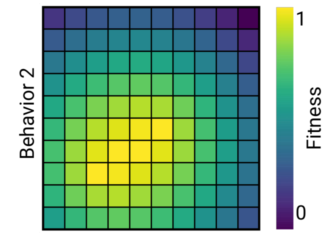
- Illustration of stepping stones
  - Nonzero fitness on “feet” only; stepwise increase
  - Top and right “toes” are stepping stones to next “foot”
  - Difficult for fitness based search; novelty can do it
- DEMO

Novelty Search Demo 2



- Fitness-based evolution is rigid
  - Requires gradual progress
- Novelty-based evolution is more innovative, natural
  - Allows building on stepping stones
- How to guide novelty search towards useful solutions?
  - Quality Diversity methods
- DEMO

(B) Combining Quality with Diversity



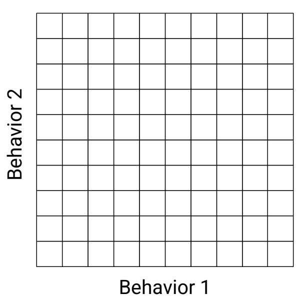
(Mouret and Clune 2015)

Goal: Find a set of different high-quality solutions for a given problem
Ex: Find fast walking gaits for a legged robot for every direction

Popular QD Method: Multi-dimensional Archive of Phenotypic Elites[1] (MAP-Elites)

Idea: Evolve an archive of different high-quality solutions (= elites)

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)

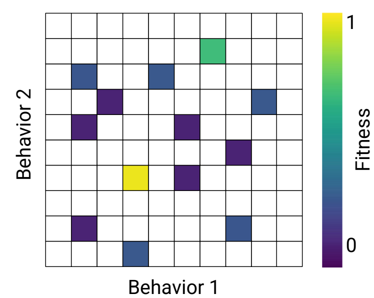


MAP-Elites algorithm

Initialization
1. **Divide the behavior space into cells (= the Archive)**
2. Initialize with random solutions until n_{init} elites are found

Main loop
1. Pick two elites from the Archive
2. Apply crossover and mutation
3. Evaluate new solution
4. If new behavior or better fitness the solution becomes an elite

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)

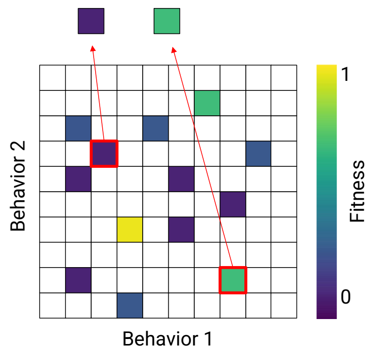


MAP-Elites algorithm

Initialization
1. Divide the behavior space into cells (= the Archive)
2. **Initialize with random solutions until n_{init} elites are found**

Main loop
1. Pick two elites from the Archive
2. Apply crossover and mutation
3. Evaluate new solution
4. If new behavior or better fitness the solution becomes an elite

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)



MAP-Elites algorithm

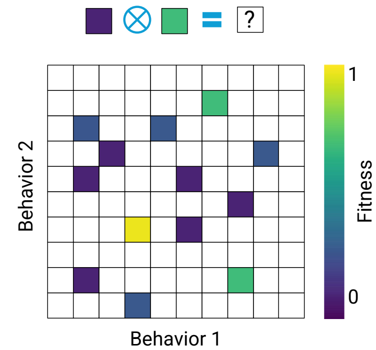
Initialization

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until n_{init} elites are found

Main loop

1. **Pick two elites from the Archive**
2. Apply crossover and mutation
3. Evaluate new solution
4. If new behavior or better fitness the solution becomes an elite

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)



MAP-Elites algorithm

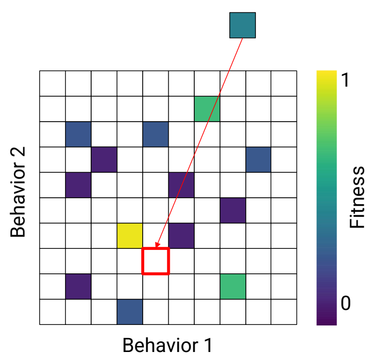
Initialization

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until n_{init} elites are found

Main loop

1. Pick two elites from the Archive
2. **Apply crossover and mutation**
3. Evaluate new solution
4. If new behavior or better fitness the solution becomes an elite

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)



MAP-Elites algorithm

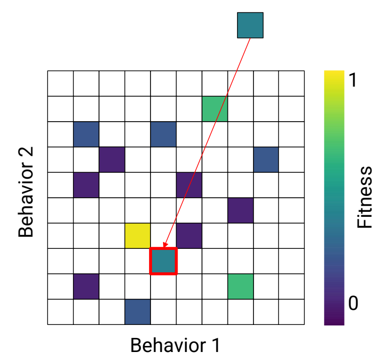
Initialization

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until n_{init} elites are found

Main loop

1. Pick two elites from the Archive
2. Apply crossover and mutation
3. **Evaluate new solution**
4. If new behavior or better fitness the solution becomes an elite

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)



MAP-Elites algorithm

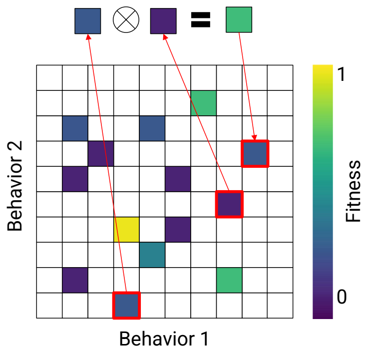
Initialization

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until n_{init} elites are found

Main loop

1. Pick two elites from the Archive
2. Apply crossover and mutation
3. Evaluate new solution
4. **If new behavior** or better fitness **the solution becomes an elite**

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)



MAP-Elites algorithm

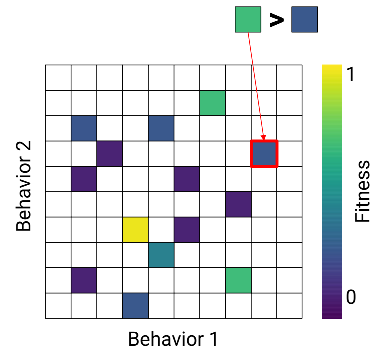
Initialization

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until n_{init} elites are found

Main loop

1. **Pick two elites from the Archive**
2. **Apply crossover and mutation**
3. **Evaluate new solution**
4. If new behavior or better fitness the solution becomes an elite

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)



MAP-Elites algorithm

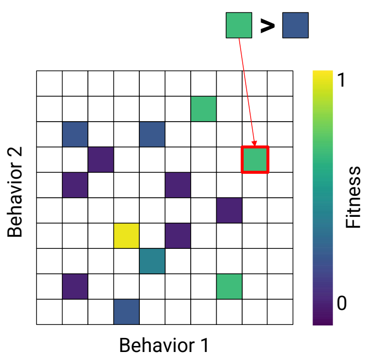
Initialization

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until n_{init} elites are found

Main loop

1. Pick two elites from the Archive
2. Apply crossover and mutation
3. Evaluate new solution
4. **If** new behavior or **better fitness the solution becomes an elite**

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)



MAP-Elites algorithm

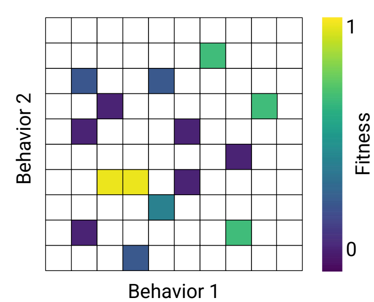
Initialization

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until n_{init} elites are found

Main loop

1. Pick two elites from the Archive
2. Apply crossover and mutation
3. Evaluate new solution
4. **If** new behavior or **better fitness the solution becomes an elite**

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)



MAP-Elites algorithm

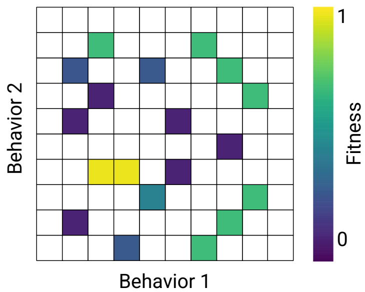
Initialization

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until n_{init} elites are found

Main loop

1. **Pick two elites from the Archive**
2. **Apply crossover and mutation**
3. **Evaluate new solution**
4. **If new behavior or better fitness the solution becomes an elite**

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)

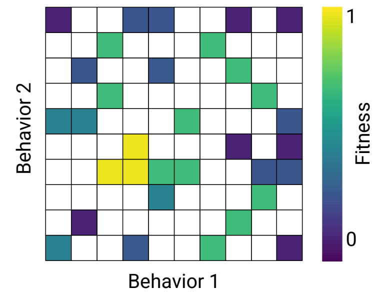


MAP-Elites algorithm

Initialization
1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until n_{init} elites are found

Main loop
1. **Pick two elites from the Archive**
2. **Apply crossover and mutation**
3. **Evaluate new solution**
4. **If new behavior or better fitness the solution becomes an elite**

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)

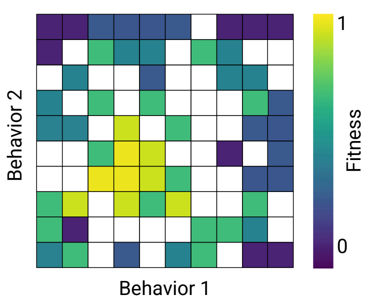


MAP-Elites algorithm

Initialization
1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until n_{init} elites are found

Main loop
1. **Pick two elites from the Archive**
2. **Apply crossover and mutation**
3. **Evaluate new solution**
4. **If new behavior or better fitness the solution becomes an elite**

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)

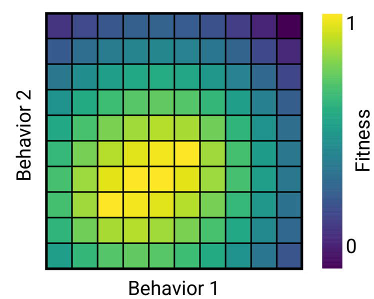


MAP-Elites algorithm

Initialization
1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until n_{init} elites are found

Main loop
1. **Pick two elites from the Archive**
2. **Apply crossover and mutation**
3. **Evaluate new solution**
4. **If new behavior or better fitness the solution becomes an elite**

Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)



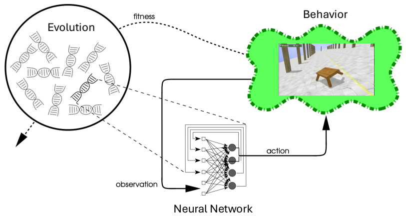
MAP-Elites algorithm

Initialization
1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until n_{init} elites are found

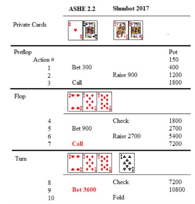
Main loop
1. Pick two elites from the Archive
2. Apply crossover and mutation
3. Evaluate new solution
4. If new behavior or better fitness the solution becomes an elite

Hands-on Exercise!

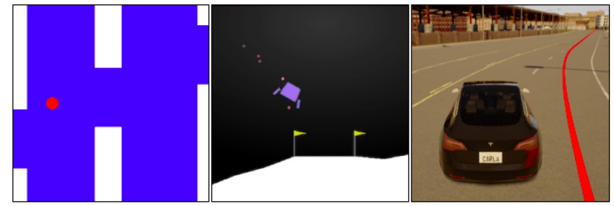
Outline



4.1 Evolving Controllers



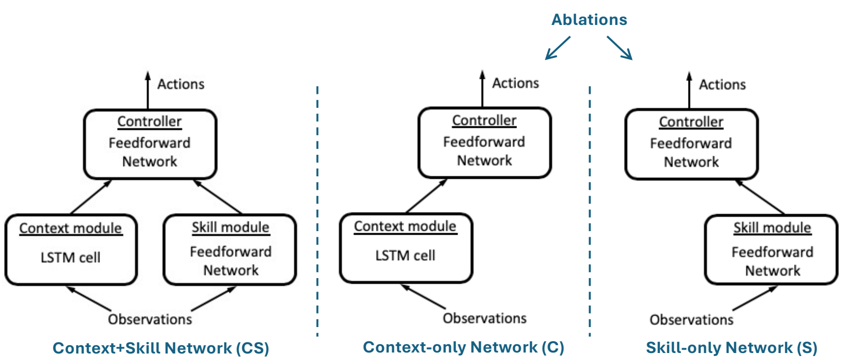
(Li et al. 2018)



(Tutum et al. 2021)

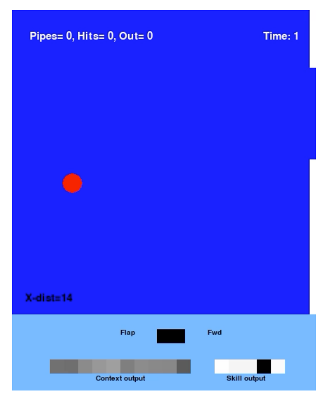
- The main challenge for evolved controllers: Extrapolation
- Need to generalize immediately to unseen situations
  - Games, robotics, process control, healthcare, finance

Solution: Modeling Context Explicitly



(Li et al 2018; Tutum et al. 2021)

FlappyBall Domain

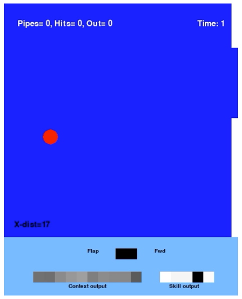


Demo

- Extension of Flappy Bird: FlapFwd, Drag
- **Inputs**: 6 numerical state values
  - Vertical position, distance to next pipe
  - Horizontal and vertical velocity
  - Height of the upper and lower pipe
- **Outputs**: select FlapUp, FlapFwd, glide
- **Objectives**:
  - Safety: Don't hit pipes, ceiling, ground
  - Performance: Fly fast
- **Task Variation:**
  - Strength of Gravity, Drag, FlapUp, FlapFwd
- **Extrapolation:**
  - Evaluated with 4x range in all combinations

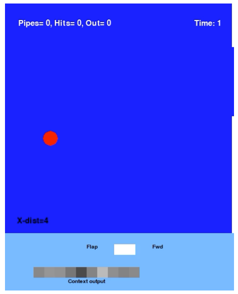
Example Behaviors in FlappyBall

- Extrapolated conditions: F=-7.0, G=0.58, Fwd=8.75, D=0.58



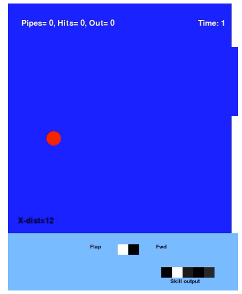
Context+Skill Network (CS)

Pipes = 21
Hits = 0



Context-only Network (C)

Pipes = 15
Hits = 6



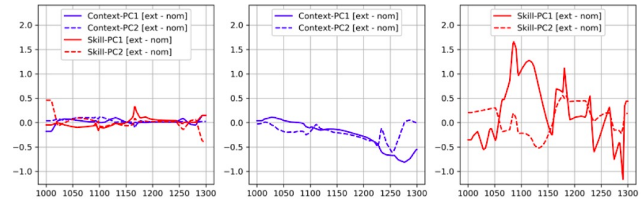
Skill-only Network (S)

Pipes = 16
Hits = 5

TEXAS

DEMO

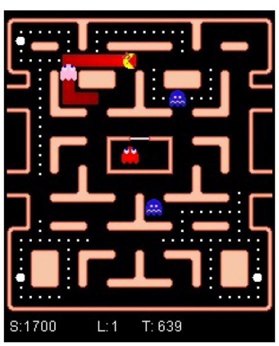
Modulation by Context



- Output of Context and Skill modules mapped to 2D with PCA
- Difference in an extrapolated task and the nominal task plotted
- Differences are smaller in CS than in C-only and S-only
  - Decision network needs to deal with less variance
  - Easier to generalize
  - CS evolves to make new tasks look more familiar
- Allows coping robustly in novel situations

TEXAS

4.2 Evolving Behavioral Strategies

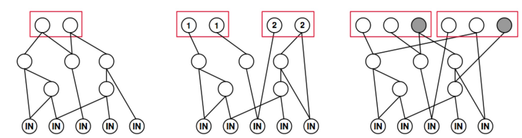


(Schrum and Miikkulainen 2015)

Strategy: different behaviors at different times

Ms. Pac-Man:
- Agents perform many different tasks
  - E.g. eat pills, avoid ghosts, eat powerpills, eat ghosts
  - Sometimes clearly separate in time
  - Sometimes multiple tasks at once
- How can we evolve them into a single network?

MM-NEAT: Modular Multiobjective Approach



(a) Single-module Network (b) Multitask Network (c) Preference Neuron Network

- Evolution discovers modules and when to use them
  - Vs. human-designed division with multitasking
- Multiple modules with preference neurons
  - Modules implement different behaviors
  - Preference neurons used to choose among them
  - Module-mutation adds new modules
- Evolved towards multiple objectives
  - Correspond to dimensions of game play
  - E.g. pills and ghosts in Ms. Pac-Man

Human-Designed Task Division

- Multitask approach
  - One module for threat ghosts
  - Another module for edible ghosts
  - Works ok, but...
  - DEMO

Evolution-Discovered Task Division

- One module used 95% of the time
  - Eat pills, avoid ghosts, chase ghosts
  - Different behaviors with a common base
- A second module 5% of the time
  - Luring ghosts near a power pill
  - Escaping from tight spaces
- A different multimodal perspective
- Not as obvious, but more powerful
- DEMO

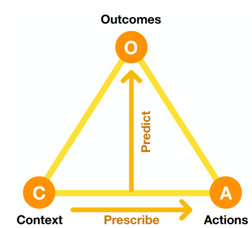
4.3 Optimizing Decision-Making

Organizations have lots of data
- Can build predictive models of patients, customers, students...

Such models do not specify how to make decisions
- Optimal decisions not known

Need to search for decision strategies
- But testing strategy candidates in the real world is costly

Surrogate Optimization Approach



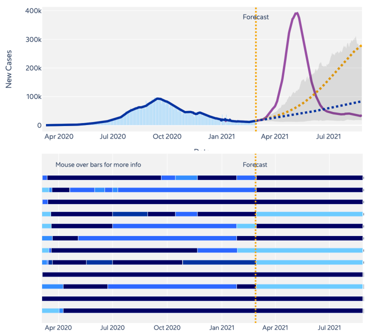
(Francon et al. 2020)

Use a predictive model as a surrogate for the world

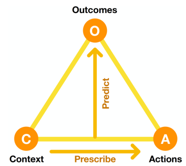
Train model with historical data: Context+Actions → Outcomes
- Phenomenological model (based on data)
- Not a simulation from first principles

Search for a good decision strategy (i.e. policy): Context → Actions
- Use the model to evaluate strategies
- Evolve a neural network to represent the strategy

Optimizing Interventions in COVID-19



(Miikkulainen et al. IEEE TEC 2021)

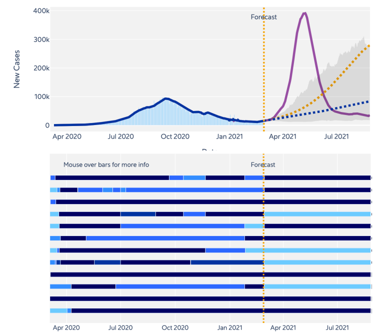


Based on two models:

1. A predictive model
   - Given a history of cases and nonpharma interventions (NPIs)
   - Predict number of cases daily
2. A prescriptive model
   - Given a history of cases and NPIs
   - Prescribe NPIs daily to minimize cases and restrictions

Not just what will happen, but what we should do about it

COVID-19 Predictions and Prescriptions

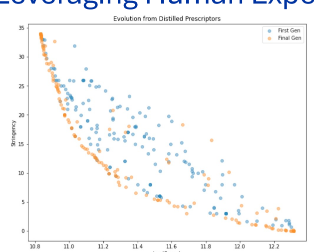


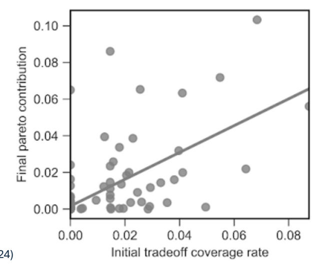
Retrained daily May 2020 – December 2022
- Based on data from Oxford University
- Adapting to the different stages of the pandemic
- Generalizing from experiences across the world

Recommendations about two weeks in advance, e.g.
- May 2020: Focus on schools and workplaces (i.e. indoors)
- Sept 2020: Focus on gatherings, travel restrictions
- March 2021: Delta surge: India lockdown
- August 2021: Recommendations for schools (Iceland)
- Dec 2021: Missed omicron surge; everywhere at once
- March 2022: Masking to avoid a second omicron surge

Interactive demo:
- https://evolution.ml/demos/npidashboard/

Leveraging Human Expertise with AI





(Meyerson et al. 2024)

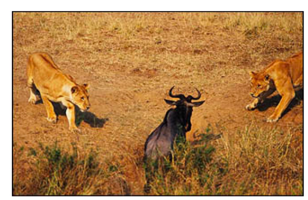
XPRIZE competition resulted in 169 human expert strategies
- Many useful, diverse ideas

Can be used as an initial population for search
- Improve further; better than search from scratch

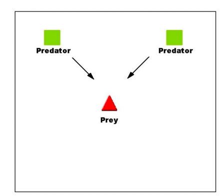
Can realize latent potential of ideas
- How much DNA in the final Pareto front

Technology to bring the community effort together
- There is power in diversity

4.4 Evolving Collective Behavior



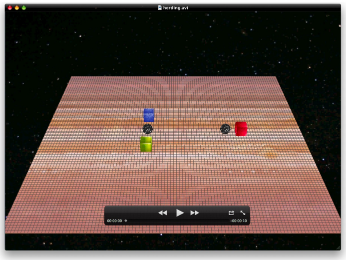
Natural predators and prey



Formalization of behavior

- Complex cooperation observed in pursuit and evasion
  - Motivated by biology, esp. hyenas vs. zebras (Kay Holekamp, MSU)
  - Largely innate, possible to see behaviors and their evolution
- Such behaviors evolve together, in coevolutionary environment
  - Simultaneous competitive and cooperative coevolution

Experimental Setup

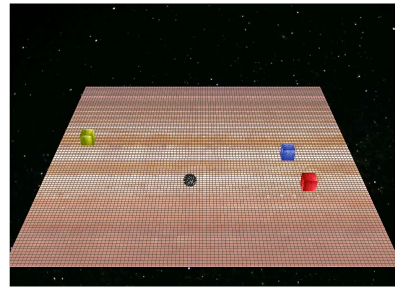


(Rawal et al. 2010)

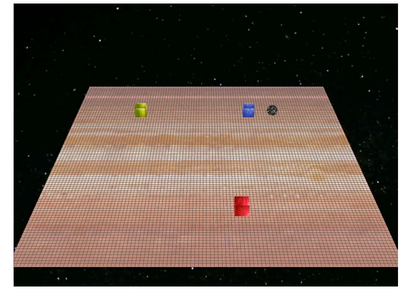
- Toroidal grid world
- Predators, prey move with same speed in 4 directions
- No direct communication between team members
  - Communication still possible through stigmergy
- Does a coevolutionary arms race result?

DEMO

Evolutionary Arms Race



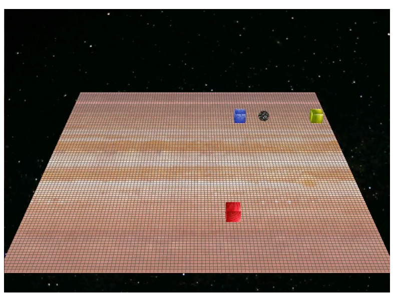
50-75: Single predator catches the prey



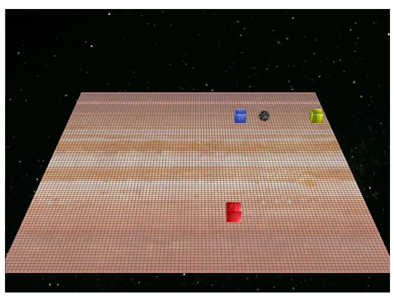
75-100: Prey evades by circling

Predators and prey populations develop increasingly sophisticated behaviors
- Each improvement provides a challenge to the other population

Evolutionary Arms Race

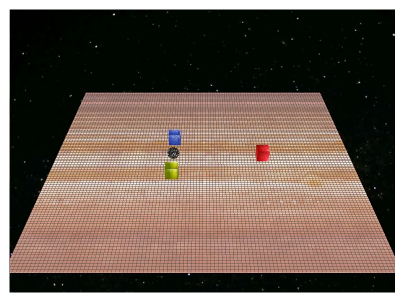


100-150: Two predators cooperate



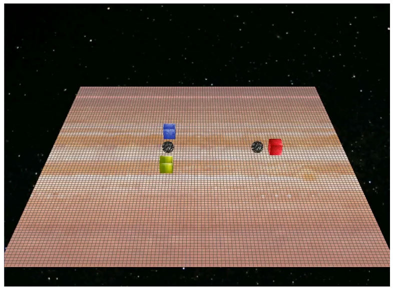
150-180: Prey baits and escapes

Evolutionary Arms Race

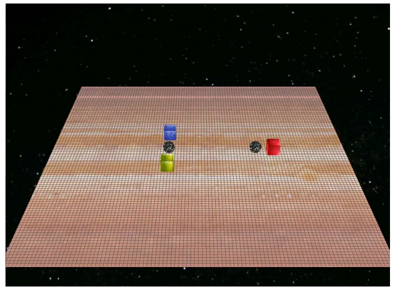


180-200: All predators cooperate

Evolutionary Arms Race



200-250: Predators herd two prey

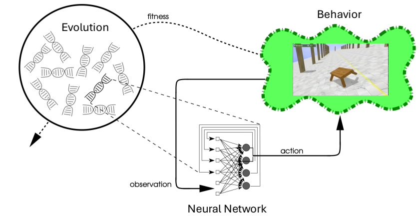


250-300: Prey evade by scattering

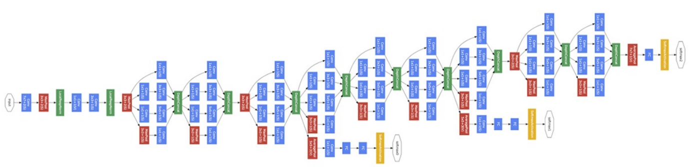
Outline

1. Motivation for Neuroevolution
2. Basics
   1. Fundamentals of Evolution
   2. Fundamentals of Neuroevolution
3. Advances
   1. Taking Advantage of Indirect Encodings
   2. Taking Advantage of Diversity
4. Evolving Intelligent Agents
   1. Control
   2. Strategy
   3. Decision-making
   4. Collective behavior
5. Synergies with other ML
   1. Deep Learning
   2. Reinforcement Learning
   3. LLMs
6. Insights into Biology
7. Conclusion

Hands-on exercise (off-line)



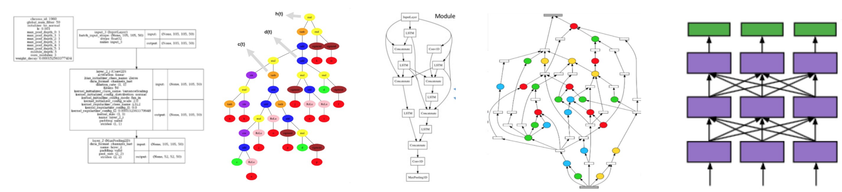
5.1 Neuroevolution Synergies with Deep Learning



(A) Fundamental: Neural Architecture Search
- Optimizing structure and hyperparameters
- Takes advantage of exploration in EC

(B) Extended: Data and training
- Loss functions, activation functions, data augmentation, initialization, learning algorithm
- Takes advantage of flexibility of EC

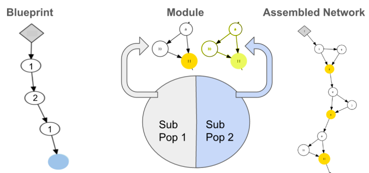
(A) Evolutionary Neural Architecture Search



Evolution is a natural fit:
- Population-based search covers the space
- Crossover between structures discovers principles

Moreover,
- Can build on Neuroevolution work since the 1990s:
  partial solutions, complexification, indirect encoding, novelty search
- Applies to continuous values; discrete choices; graph structures; combinations
- Can evolve hyperparameters; nodes; modules; topologies; multiple tasks

E.G. NAS with CoDeepNEAT

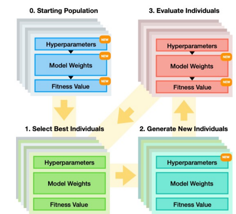


(Liang et al. 2016)

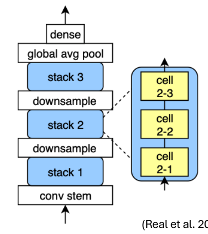
Evolution at three levels
- Module subpopulations optimize building blocks
- Blueprint population optimizes their combinations
- Hyperparameter evolution optimizes their instantiation

Fitness of the complete network drives evolution
- Candidates need to be evaluated through training
- Expensive; use partial training, surrogates...

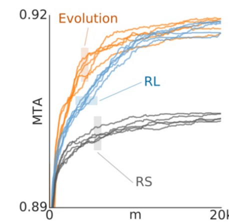
Making NAS Evaluations Practical



(Liang et al. 2021)



(Real et al. 2019)



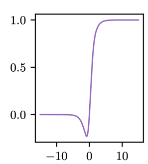
Population-based training (Jaderberg et al 2017; Liang et al. 2021)
- Continual training and evolution

NAS benchmarks created to help evaluate (Ying et al. 2019; Dong et al. 2020; Siems et al. 2021)
- Collections of known architecture evaluations, surrogates

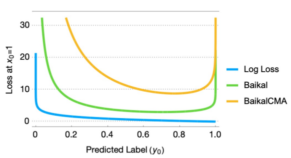
Scaling and regularization (Such et al. 2017; Real et al. 2019)
- State-of-the art at the time in CIFAR-10, CIFAR-100, ImageNet

Specialized crossover operators (Qiu and Miikkulainen 2023)

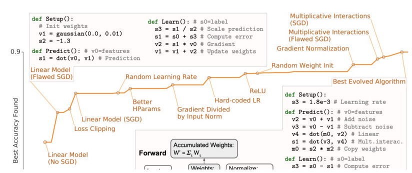
(B) Optimizing Other Aspects of Deep Learning Design



(Bingham and Miikkulainen, 2022)



(Gonzalez and Miikkulainen, 2020)



(Real et al. 2020)

Optimizing activation functions and loss functions (Bingham and Miikkulainen, 2022; Neural Networks Best Paper Award at IJCNN-25!) (Gonzalez and Miikkulainen, 2020)
- Regularization and refinement

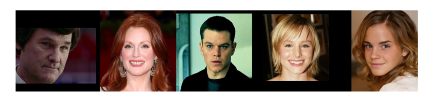
Designing machine learning algorithms with GP (Real et al. 2020)
- Adapts to different task types
- Discovering new layer types

Coevolution of multiple aspects of network design?

Example: Evolving Age-Estimation Networks

Parameter	Possible Values	Type	Class
Algorithm	[adam, rmsprop]	Enum	Opt
Initial Learning Rate (LR)	[1e-5, 1e-3]	Float	Opt
Momentum	[0.7, 0.99]	Float	Opt
(Weight Decay) / LR [26]	[1e-7, 1e-3]	Float	Opt
Patience (Epochs)	[1, 20]	Int	Opt
SWA Epochs [21]	[1, 20]	Int	Opt
Rotation Range (Degrees)	[1, 60]	Int	Aug
Width Shift Range	[0.01, 0.3]	Float	Aug
Height Shift Range	[0.01, 0.3]	Float	Aug
Shear Range	[0.01, 0.3]	Float	Aug
Zoom Range	[0.01, 0.3]	Float	Aug
Horizontal Flip	{True, False}	Bool	Aug
Vertical Flip	{True, False}	Bool	Aug
Cutout Probability [7]	[0.01, 0.999]	Float	Aug
Cutout Max Proportion [7]	[0.05, 0.5]	Float	Aug
Pretrained Base Model	Keras App. [5]	Enum	Arch
Base Model Output Blocks	{B0, B1, B2, B3}	Subset	Arch
Loss function λ in Eq. 5	[0, 1]	Float	Arch

(Miikkulainen et al. 2021)

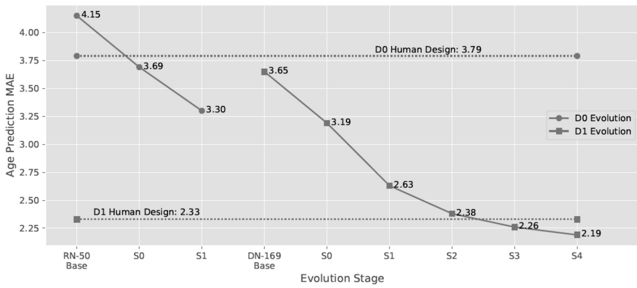


Estimate age from a facial image

Evolving multiple design aspects
- Learning, data augmentation hyperparameters
- Seeded architecture search
- Loss-function optimization: Combination of MAE and CE

Also
- Population-based training
- Ensembling of evolved solutions

Age-Estimation Results



- D0 stages: ResNet-50, DenseNet-121
- D1 stages: DenseNet-169, DenseNet-201, more epochs. EfficientNet-B6, ensembling
- Human optimization of ResNet-50 (D0), EfficientNet-B6 (D1)

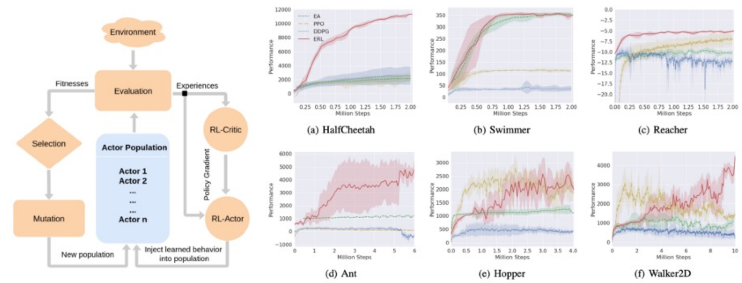
Evolution improves significantly over SotA image models
- Fit the design to the task

Optimizes better than humans can
- Many more parameters simultaneously

Performance exceeds that of humans: 2.19 vs. 3-4 years

5.2 Neuroevolution Synergies with RL

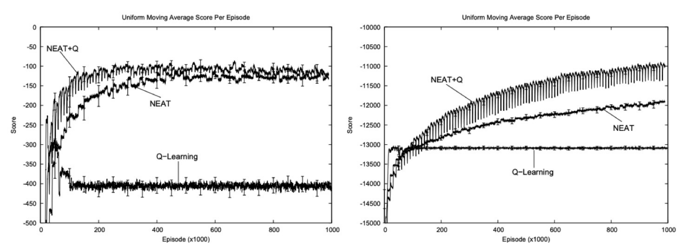
(A) Combining population-based search and RL-based search



(Khadka and Tumer 2018)

Evolutionary Reinforcement Learning:
- A population of networks evolved to maximize rewards
- Evaluations create off-policy training data for Deep RL
- Trained networks periodically injected into the population
- ERL outperforms both EA and Deep RL alone

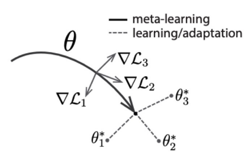
(B) Evolving Value Function Networks



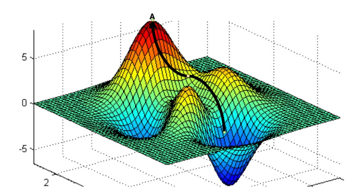
(Whiteson 2006)

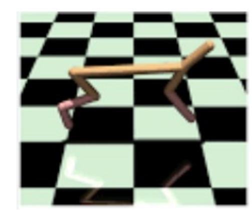
NEAT+Q
- Many RL methods rely on value functions to estimate rewards
- NEAT evolves both weights and topologies for better value-function networks
- NEAT+Q outperforms other value-function approximation methods and manually designed networks in e.g. mountain car and server job scheduling

(C) Evolving Starting Points for RL



(Finn et al. 2017)





MAML-Baldwin, ES-MAML (Fernando et al. 2018; Song et al. 2019)
- Model-agnostic Meta-Learning (MAML) finds good starting points for learning
- Evolutionary methods like MAML-Baldwin and ES-MAML improve by evolving the starting points Evolve initial weights that adapt to different tasks during the agent's lifetime.
- E.g. in half-cheetah task, adapts to changing direction rewards within seconds

5.3 Neuroevolution Synergies with LLMs

(A) Neuroevolution through Large Language Models

LM prompt (Parents)	11101111 11110111 10100111	x^2 + 2.1*x sin x^2 + 7 3*sin x + 6.6	the moon is bad the moon is boring the moon is cold
LM output (Children)	11111111 10110111	x^2 sin x + 6 cos x^2 + 2.1*x	the moon is zen the sky has a moon

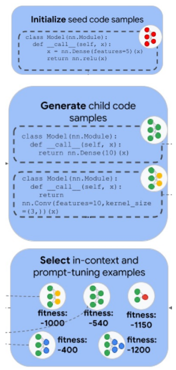
(Meyerson et al. 2024)

Better evolution through LLMs?

- Evolution through large models (ELM)
- Language model crossover (LMX)
- Level generation for Mario (MarioGPT)

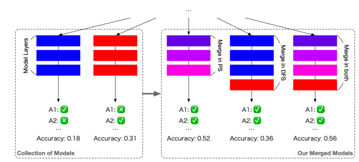
E.g. Evolutionary prompting for NAS (EvoPrompting)

- Existing architectures as prompts; generate new
- Tune the prompts based on performance

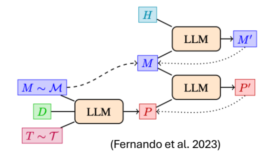


(Chen et al. 2023)

(B) Neuroevolution of Large Language Models



(Akiba et al. 2024)



(Fernando et al. 2023)

Better LLMs through evolution?

Model merging: combine multiple fine-tuned LLMs to one

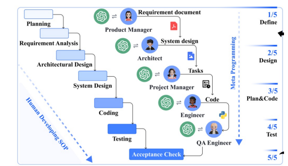
- E.g. Japanese LLM with Math

Evolving prompts: Promptbreeder

- Evolving mutation prompts to improve task prompts

Evolving multi-LLM interactions

- E.g. roles for collaborative problem solving

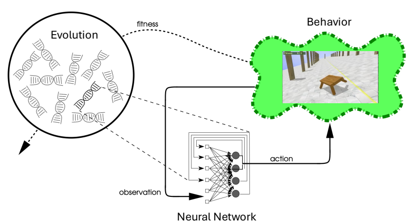


(Hong et al. 2023)

Outline

1. Motivation for Neuroevolution
2. Basics
   1. Fundamentals of Evolution
   2. Fundamentals of Neuroevolution
3. Advances
   1. Taking Advantage of Indirect Encodings
   2. Taking Advantage of Diversity
4. Evolving Intelligent Agents
   1. Control
   2. Strategy
   3. Decision-making
   4. Collective behavior
5. Synergies with other ML
   1. Deep Learning
   2. Reinforcement Learning
   3. LLMs
6. Insights into Biology
7. Conclusion

Hands-on exercise (off-line)



6. Example: Evolution of Intelligent Coordinated Behavior

Stealing a kill from lions

- Succeeds in an otherwise impossible task (sometimes)
- More sophisticated than other hyena behaviors
- Highly rewarding compared to normal hunting
- Largely genetically determined
- A breakthrough in evolution of intelligence?

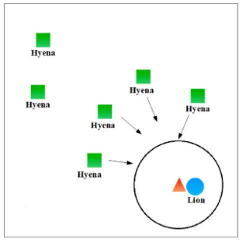
A collaboration with Kay Holekamp's lab (MSU)

- Studying hyenas in Masai Mara since 1982

(Rajagopalan et al. 2021)

DEMO

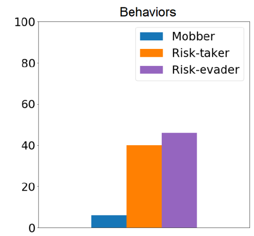
Simulation Setup

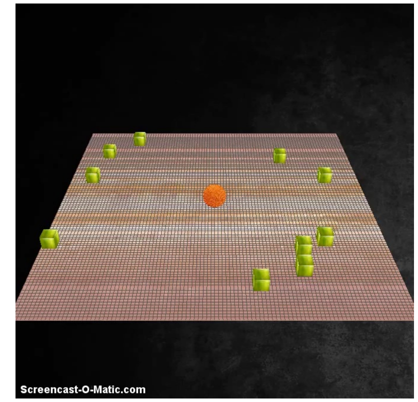


Lion at a kill, with an interaction circle around it
Ten hyenas chosen and placed randomly in the field
If four or more hyenas enter the circle simultaneously, they get the kill
- Otherwise they die

Does mobbing behavior evolve?
- What are the stepping stones for it?

Initial Behaviors



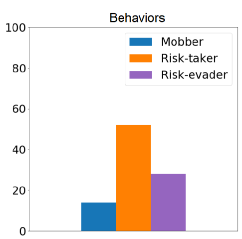


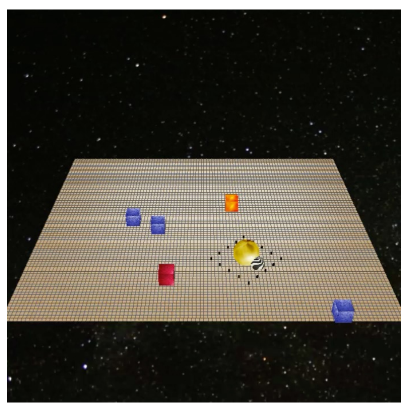
Risk evasion is common
- Never reach the circle; Medium fitness

Risk taking is common
- Charge the circle; Frequent low fitness
- Occasional high fitness by accident

DEMO

Early Behaviors



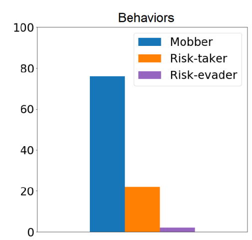


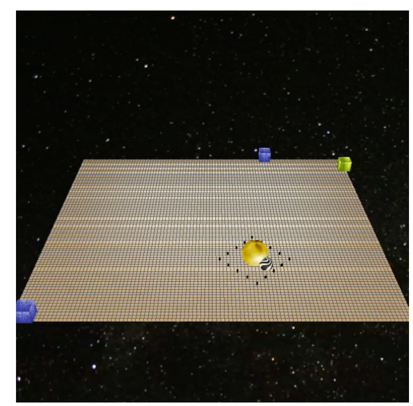
Risk taking grows
- As long as it is successful often enough

Risk evasion also persists
Evasion at the circle starts to emerge
Is mostly detrimental, but an important stepping stone

DEMO

Later Behaviors



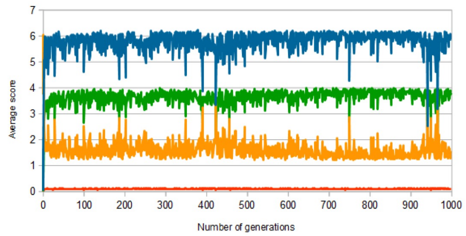


Mobbing emerges
- Not just coincidence of risk takers
- Hyenas wait until there’s enough of them

Risk-evaders evolve into latecomers
Simple risk-taking and risk-evasion still exist

DEMO

These Behaviors Persist in Prolonged Evolution



Risk taking and risk evasion never go away completely

- They serve a role in maintaining the mobbing behavior
- If mobbing starts to get lost, it can be reintroduced

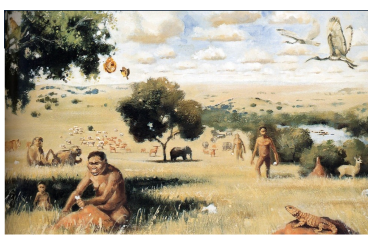
Insight into Real-life Behaviors

These behaviors are observed in real-life hyenas as well

A computational explanation of why they are there:

- Stepping stones in discovery
- Safeguards in maintaining

Future Challenge: Evolution of Language

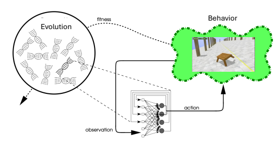


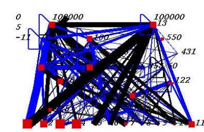
Signaling is possible to evolve in ecological simulations
Structured language is much harder
Perhaps language evolved not from signaling, but cognition

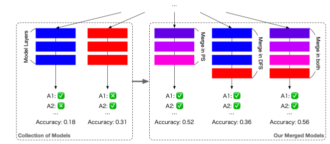
- Complex social structure, with modifiable roles
- Language structure can reuse the same conceptual structures

Enough compute, complex simulations to study now?

Conclusion







AI is progressing from imitation to creativity; from models to agents

Neuroevolution is a powerful approach to discovering behavior

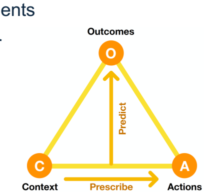
- Control, strategy, collective behavior, decision-making

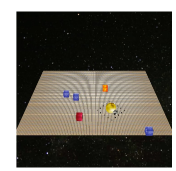
Neuroevolution can provide a boost to ML

- Deep learning designs; RL exploration; LLM optimization
- Automatic design of learning machines

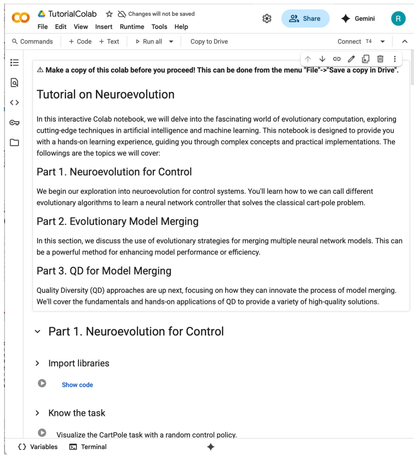
Neuroevolution can provide insight into biological evolution

- Evolutionary origins of circuits, behavior, intelligence
- Evolution of language as a current challenge
- A possible path to AGI





A Hands-on Exercise

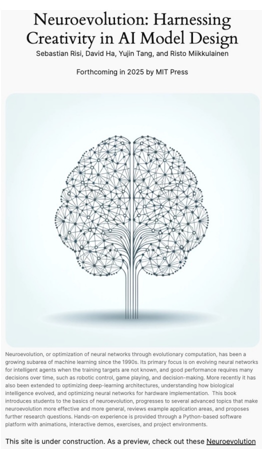


- By Yujin Tang
- Direct link:
  https://colab.research.google.com/drive/1OQFhqJHrV8qpepYRNxHUyauGhmEVB_kx
  Also under:
  https://neuroevolutionbook.com/code-exercises
- Three parts:
  - Neuroevolution for Control
  - Evolutionary Model Merging
  - QD for Model Merging

Further Material

- www.cs.utexas.edu/~risto/talks/enn-tutorial
  - Slides, references, demos, video
- nn.cs.utexas.edu/?miikkulainen:science25
  - A review of neuroevolution insights into neuroscience
- Neuroevolution sessions at conferences (e.g. GECCO)
- And….

The Neuroevolution Book!



- MIT Press, 2025
- A comprehensive overview
- Software platform: Demos, exercises
- Open access
- https://neuroevolutionbook.com