Evolution of Neural Networks

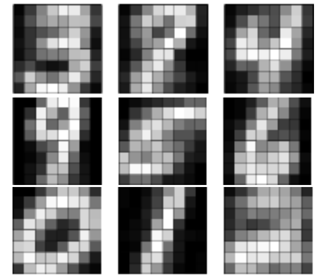
Risto Miikkulainen

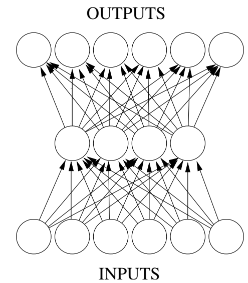
The University of Texas at Austin
and Cognizant AI Labs

GECCO '22 Companion, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9268-6/22/07...$15.00
https://doi.org/10.1145/3520304.3533656

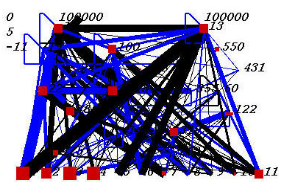
Why Use Neural Networks?

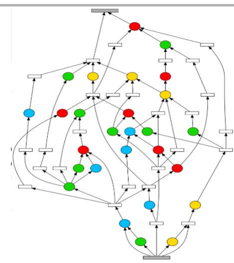


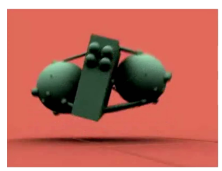


- Neural nets powerful in many statistical domains
  - E.g. control, pattern recognition, prediction, decision making
  - Where no good theory of the domain exists
- Good supervised training algorithms exist
  - Learn a nonlinear function that matches the examples
  - Utilize big datasets

Why Evolve Neural Networks?

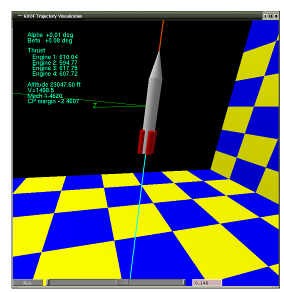






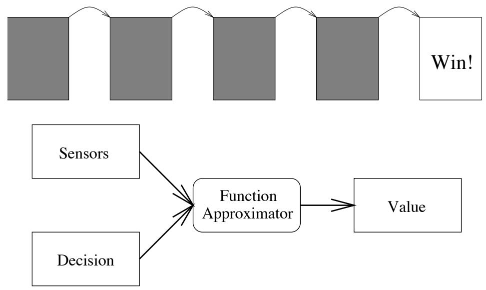
- I. Original role (since 1990s): RL Tasks & especially POMDP
  - Both the structure and the weights evolved (no training)
  - Power from recurrency; behavior
- II. A new role (since 2016): Optimization of Deep Learning Nets
  - Architecture, hyperparameters, functions evolved; weights trained
  - Power from complexity
- III. A possible future role: Emergence of intelligence
  - Body/brain co-evolution; Competitive co-evolution
  - Evolution of memory, language, learning

I. Reinforcement Learning / POMDP Tasks



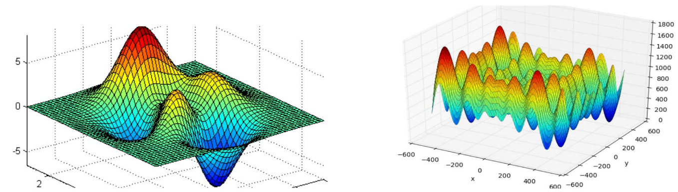
- A sequence of decisions creates a sequence of states
  - States are only partially known
  - Optimal outputs are not known
  - We can only tell how well we are doing
- Exist in many important real-world domains
  - Robot/vehicle/traffic control
  - Computer/manufacturing/process optimization
  - Game playing; Artificial Life; Biological Behavior

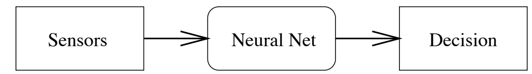
Value-Function Reinforcement Learning



- E.g. Q-learning, Temporal Differences
  - Generate targets through prediction errors
  - Learn when successive predictions differ
- Predictions represented as a value function
  - Values of alternatives at each state
- Difficult with large/continuous state and action spaces
- Difficult with hidden states

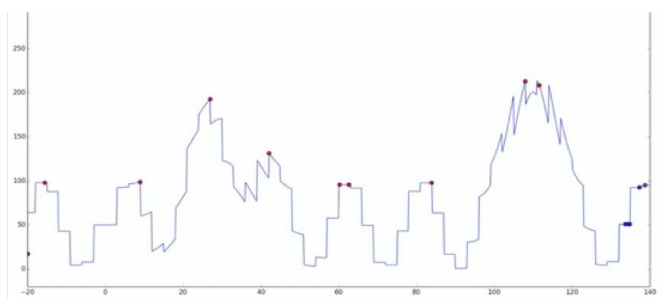
Policy-Search Reinforcement Learning





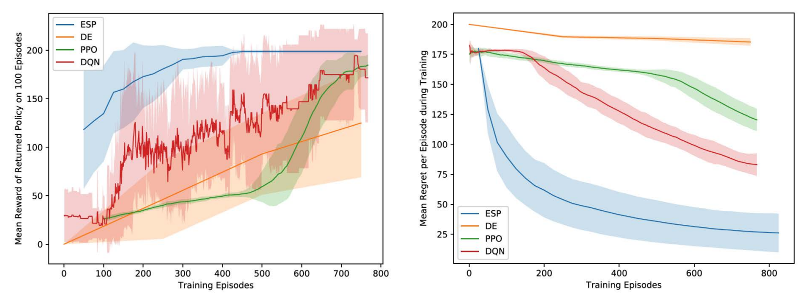
- E.g. REINFORCE, policy gradients
- The policy is optimized directly through hill climbing
- Works well in simple cases
  - Large/continuous states and actions possible
  - Hidden states (in POMDP) disambiguated through memory
  - Does not scale well

Neuroevolution Reinforcement Learning



- Takes advantage of population-based search
  - In essence, multiple interacting searches
  - Each discover building blocks that are combined
  - Extensive exploration possible
- Makes it possible to scale up:
  - to large spaces (e.g. $2^{2^{70}}$ states[49])
  - to high dimensionality (e.g. up to 1B[10])
  - to deceptive landscapes (with e.g. multiobj and novelty[74])

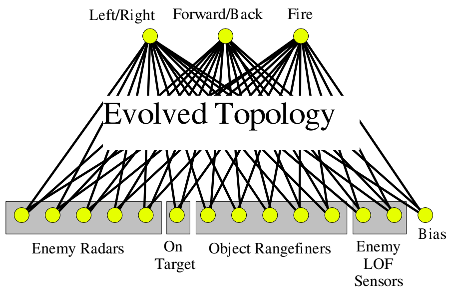
How Well Does It Work?



(a) True Performance (b) Regret

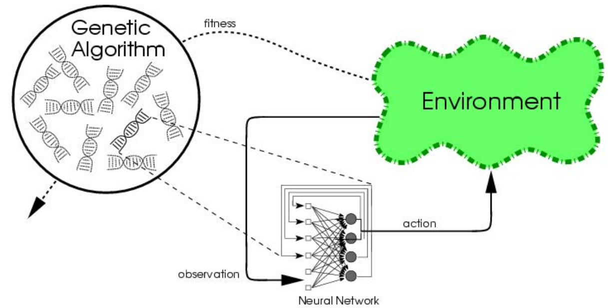
- In the OpenAI Gym CartPole-v0 benchmark vs. PPO, DQN
  - NE converges faster, has lower variance, lower regret
  - NE is more efficient, reliable, and safer[15]
- In a double-pole benchmark vs. Sarsa, Q-MLP, etc.
  - The only method that can find solutions to 1m, 0.1m, POMDP[20]
- The fundamental difference is exploration
  - Evolution provides more exploration than gradients do[30,68,85]

Neuroevolution for RL/POMDP



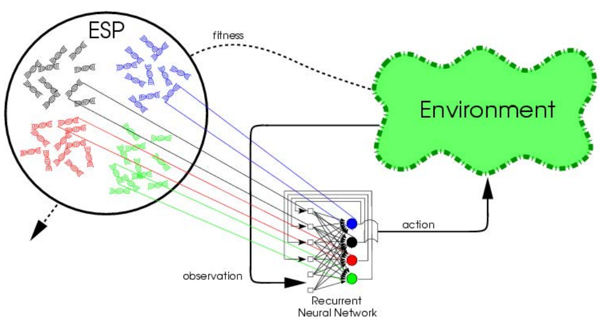
- Input variables describe the state observed through sensors
- Output variables describe actions
- Network between input and output:
  - Recurrent connections implement memory
  - Memory helps with POMDP

Basic Neuroevolution



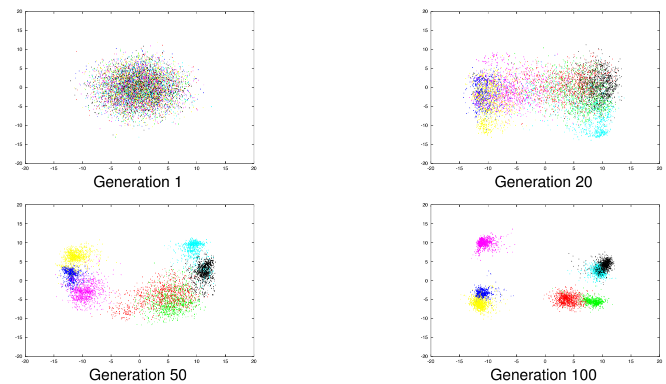
- Evolving connection weights in a population of networks [56,69,96,97]
- Chromosomes are strings of connection weights (bits or real)
  - E.g. 10010110101100101111001
  - Usually fully connected, fixed, initially random topology
- A natural mapping between genotype and phenotype
  - GA and NN are a good match!

Advanced NE 1: Evolving Partial Networks



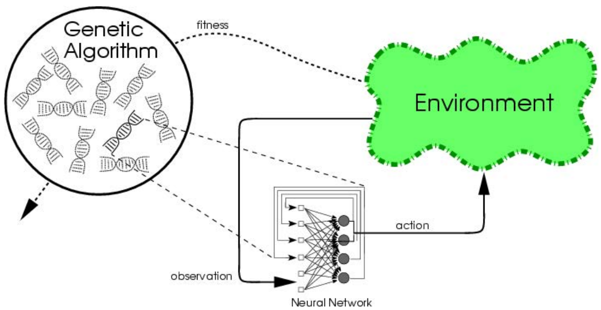
- Evolving individual neurons to cooperate in networks [1,57,60]
- E.g. Enforced Sub-Populations (ESP [18])
  - Each (hidden) neuron in a separate subpopulation
  - Fully connected; weights of each neuron evolved
- Can be applied at the level of weights, and modules [20]

Why Is It a Good Idea?



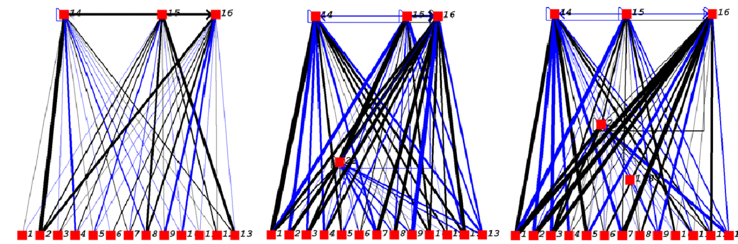
- E.g. evolving neurons for robotic control
  - Simulated Kheperas running a maze
- Subpopulations discover & optimize compatible subtasks
  - E.g. slow down with obstacle on front
    veer left with obstacle at right, etc.
- Each neuron part of 2-3 subtasks
  - Robust coding of behavior during search

Advanced NE 2: Evolutionary Strategies



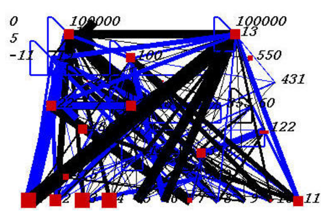
- Evolving complete networks with ES (CMA-ES[26])
- Small populations, no crossover
- Instead, intelligent mutations
  - Adapt covariance matrix of mutation distribution
  - Take into account correlations between weights
- Why is it a good idea?
  - Discovers good weight combinations → CM

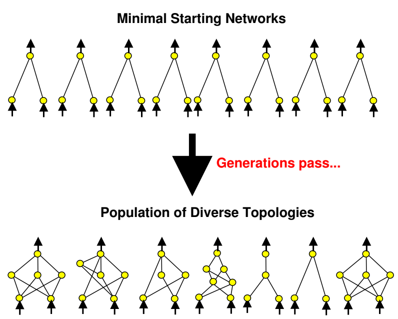
Advanced NE 3: Evolving Network Structure



- Optimizing connection weights and network topology[3,12,16,98]
- E.g. Neuroevolution of Augmenting Topologies (NEAT[78,82])
- Based on *Complexification*
- Of networks:
  - Mutations to add nodes and connections
- Of behavior:
  - Elaborates on earlier behaviors

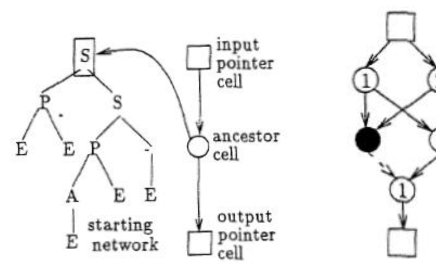
Why Is It a Good Idea?





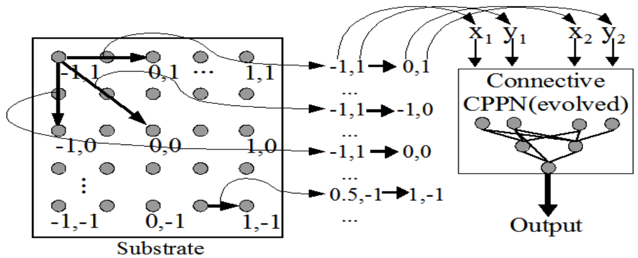
- NN search space is complex with nonlinear interactions
- Complexification keeps the search tractable
  - Start simple, add more sophistication
- Incremental discovery of complex solutions

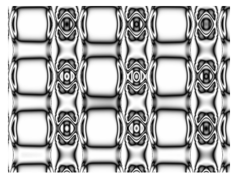
Advanced NE 4: Indirect Encodings (1)



- Instructions for constructing the network evolved
  - Instead of specifying each unit and connection[3,12,55,75,98]
- E.g. Cellular Encoding (CE[23])
- Grammar tree describes construction
  - Sequential and parallel cell division
  - Changing thresholds, weights
  - A “developmental” process that results in a network

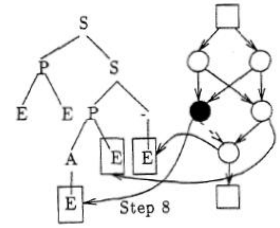
Indirect Encodings (2)

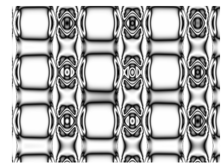




- Encode the networks as spatial patterns
- E.g. Hypercube-based NEAT (HyperNEAT[8])
- Evolve a neural network (CPPN) to generate spatial patterns
  - 2D CPPN: (x, y) input → grayscale output
  - 4D CPPN: (x_1, y_1, x_2, y_2) input → w output
  - Connectivity and weights can be evolved indirectly
  - Works with very large networks (millions of connections)

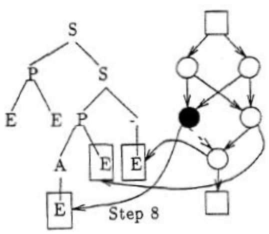
Why Is It a Good Idea?

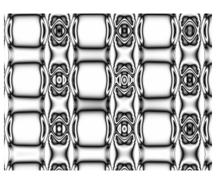




- Describes structure efficiently
  - Recurrency symbol in CE: XOR → parity
  - Repetition with variation in CPPNs
- Useful for evolving topology
  - E.g. large structured networks
  - E.g. repetition of motifs

Future Opportunities



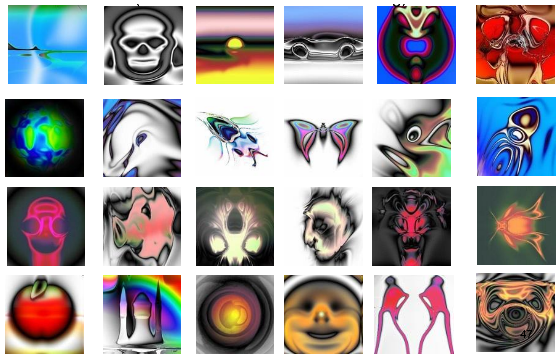


- Several possible directions
  - More general L-systems; developmental codings; embryogeny[83]
  - Scaling up spatial coding[9,17]
  - Genetic Regulatory Networks[65]
  - Evolution of symmetries[91]
- Theory starting to emerge
  - Expressive Encodings[48]: Simple GAs are universal probability approximators (Meyerson et al. GECCO'22)

Further NE Techniques

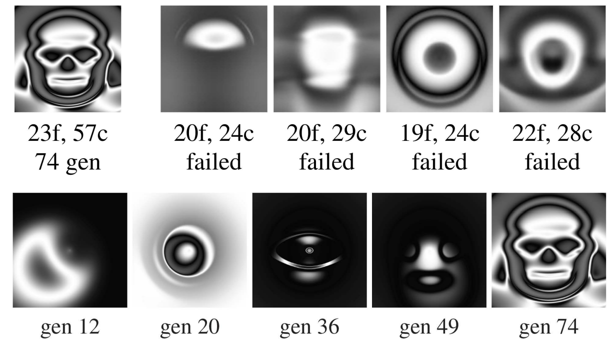
- Incremental and multiobjective evolution[19,72,90,97]
- Utilizing population culture[4,42,87]
- Utilizing evaluation history[41]
- Evolving NN ensembles and modules[28,40,59,66,94]
- Evolving transfer functions and learning rules[7,67,84]
- Bilevel optimization of NE[38]
- Evolving LSTMs for strategic behavior[34]
- Extrapolation with Context+Skill modules[89]
- Combining learning and evolution[6,13,42,58,79,87,95]
- Evolving for novelty

Evolving for Novelty



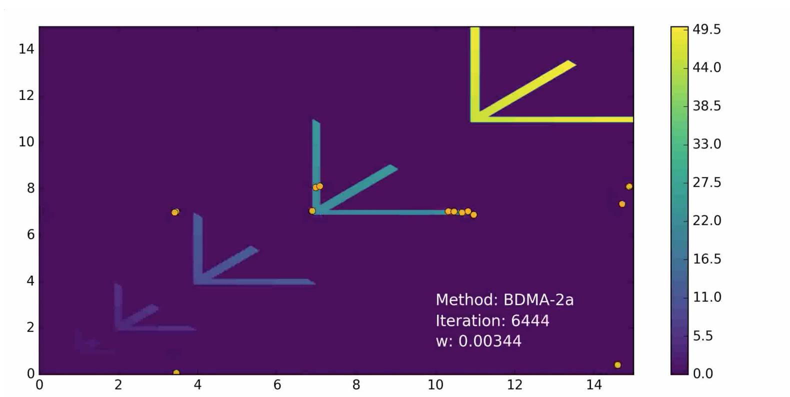
- Motivated by humans as fitness functions
- E.g. picbreeder.com, endlessforms.com[73]
  - CPPNs evolved; Human users select parents
- No specific goal
  - Interesting solutions preferred
  - Similar to biological evolution?

Novelty Search



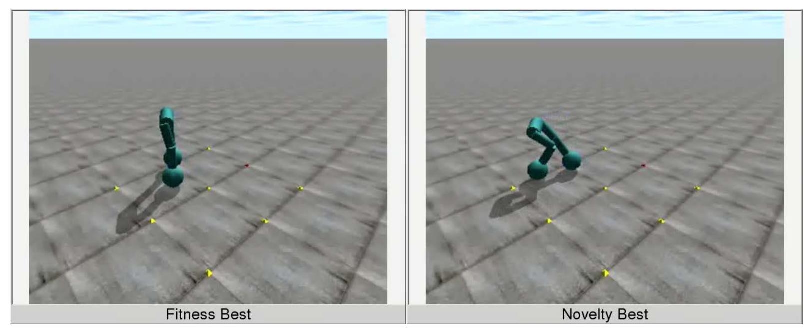
- Evolutionary algorithms maximize a performance objective
  - But sometimes hard to achieve it step-by-step
- Novelty search rewards candidates that are simply different[31,81]
  - Stepping stones for constructing complexity

Novelty Search Demo (1)



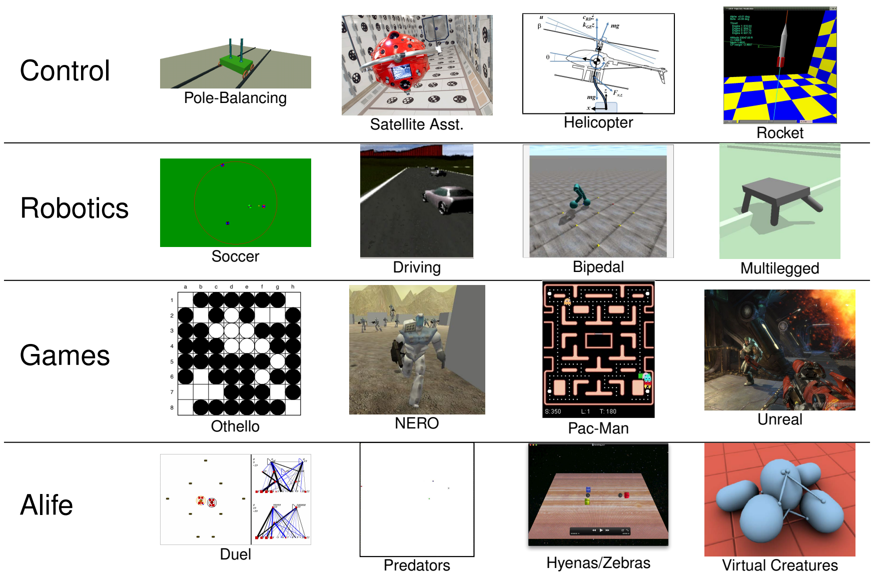
- Illustration of stepping stones[43,44]
  - Nonzero fitness on “feet” only; stepwise increase
  - Top and right “toes” are stepping stones to next “foot”
  - Difficult for fitness based search; novelty can do it
- DEMO

Novelty Search Demo (2)



- Fitness-based evolution is rigid
  - Requires gradual progress
- Novelty-based evolution is more innovative, natural[31,81]
  - Allows building on stepping stones
- How to guide novelty search towards useful solutions?
  - Quality Diversity methods[14,61]
- DEMO

Neuroevolution Applications



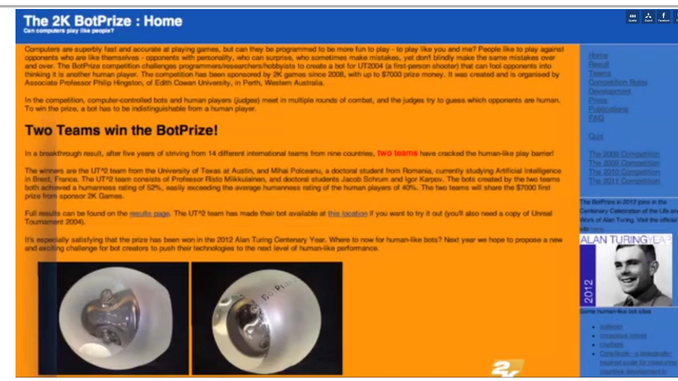
Example 1: Evolving Humanlike Behavior

- Botprize competition, 2007-2012
  - Turing Test for game bots ($10,000 prize)
- Three players in Unreal Tournament 2004:
  - Human confederate: tries to win
  - Software bot: pretends to be human
  - Human judge: tries to tell them apart!

Evolving an Unreal Bot

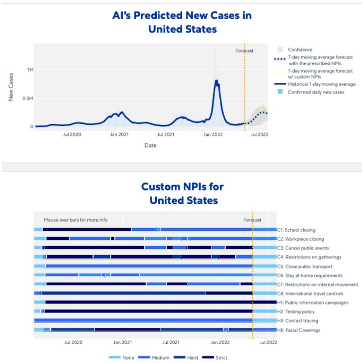
- Wandering, unstuck etc. based on scripts & learning from humans
- Evolve effective fighting behavior[71]
- Persistent gap: 30% vs. 80% human
  - Evolving to win results in unnatural behaviors
  - Human judges do not understand their expertise

After Five Years, Success!!!



- Human-like behavior with resource limitations (speed, accuracy...
  - Best bot better than 50% of the humans
  - Two teams human 50% of the time
- Fascinating challenges remain:
  - Judges can still differentiate in seconds
  - Judges lay cognitive, high-level traps
  - Team competition: collaboration as well
- DEMO

Example 2: Optimizing COVID-19 NPIs



Train a NN to predict COVID-19 cases

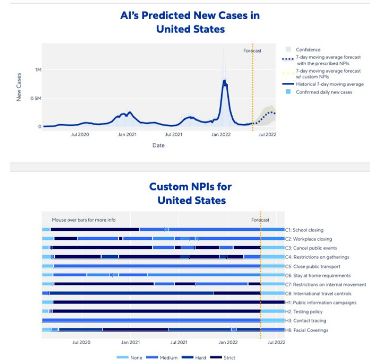
- Based on number of cases in different countries over time
- And non-pharmaceutical interventions (NPIs) over time

Using the predictive model as a surrogate,
evolve a NN to recommend NPIs

- Resulting in smallest number of cases
- With minimal economic cost

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

Example 2: Optimizing COVID-19 NPIs (2)



Retrained daily since May 2020 [15,52]

- Based on data from Oxford University[24]
- Adapting to the different stages of the pandemic
- Generalizing from experiences across the world

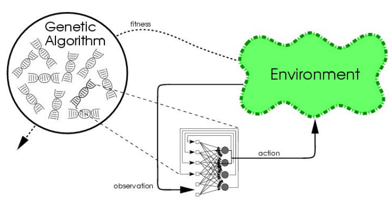
Recommendations about two weeks in advance, e.g.

- May 2020: Focus on schools and workplaces (i.e. indoors)
- Sept 2020: Focus on gatherings, travel restrictions
- March 2021: India lockdown
- July 2021: Delta surge on countries with low rates so far
- March 2022: Masking to avoid a second Omicron surge

Interactive demo:

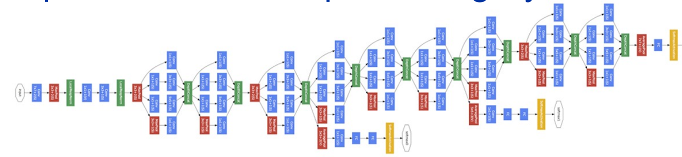
- https://evolution.ml/demos/npidashboard

Part I Conclusion: Neuroevolution RL



- A powerful way to train networks when gradients not available
  - E.g. recurrency in POMDP domains
- Many evolutionary techniques are a good match with NE
  - Partial solutions, CMA, Complexification, Indirect, Novelty, Constrained
- Can discover surprising, believable, effective solutions

II. Optimization of Deep Learning Systems



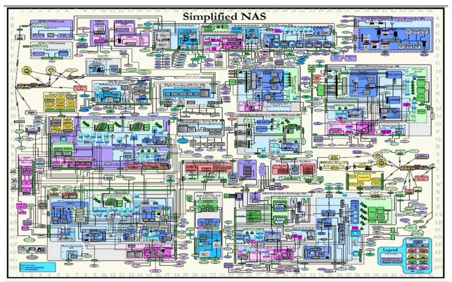
Szegedy et al. 2015 [86]

Deep learning systems operate at a much larger scale

- 10^6 - 10^{12} parameters
- Overparameterized; trained by gradient descent

A new problem: How to configure such systems?

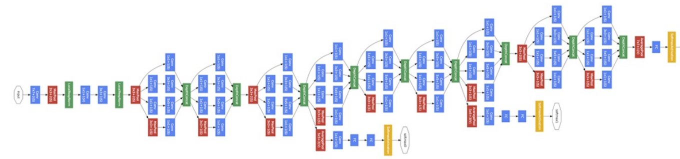
Configuring Complex Systems



A new general approach to engineering
- Humans design just the framework
- Machines optimize the details

Programming by optimization[25]

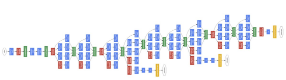
Configuring Deep Learning with Evolution



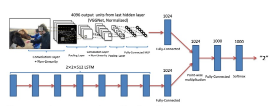
(A) Fundamental: Neural Architecture Search
- Optimizing structure and hyperparameters
- Takes advantage of exploration in EC

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

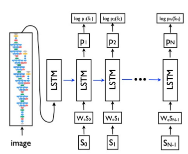
Neural Architecture Search (NAS)



Szegedy et al. 2015[86]



Agrawal et al. 2017[2]



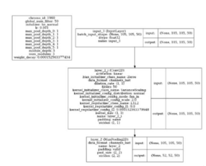
Vinyals et al. 2015[92]

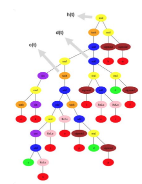
Different architectures work best in different tasks
- Structure matters!

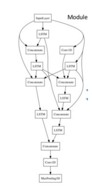
Too complex to be optimized by hand
- How to discover principles of organization?
- How to cover enough of the search space?

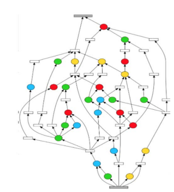
Several possible ML methods: Bayesian optimization, gradient descent, RL, evolution…

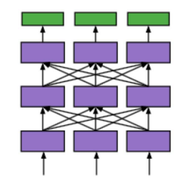
Evolutionary NAS







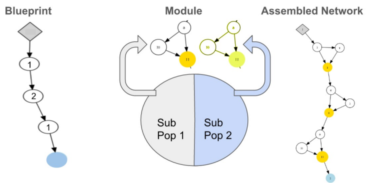




Evolution is a natural fit:
- Population-based search covers the space
- Crossover between structures discovers principles

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

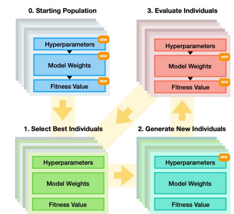
E.G. NAS with CoDeepNEAT

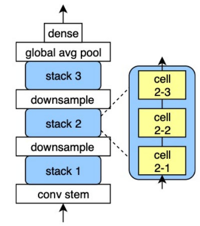


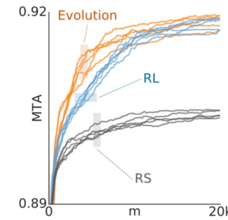
Evolution at three levels[53]
- Module subpopulations optimize building blocks
- Blueprint population optimizes their combinations
- Hyperparameter evolution optimizes their instantiation

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

Making NAS Evaluations Practical





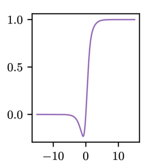


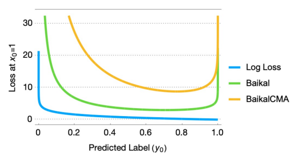
Population-based training (DeepMind, Cognizant)[27,35]
- Continual training and evolution

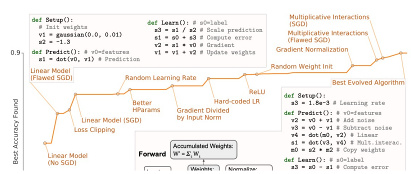
NAS benchmarks created to help evaluate (Google, Baidu, Freiburg)[11,99,100]
- Collections of known architecture evaluations, surrogates

Scaling and regularization (Google)[63]
- State-of-the art at the time in CIFAR-10, CIFAR-100, ImageNet

Optimizing Other Aspects of Deep Learning Design







Optimizing activation functions and loss functions (Cognizant)[5,21,22,35]
- Regularization and refinement

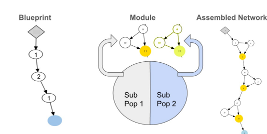
Designing machine learning algorithms with GP (Google)[39,64]
- Adapts to different task types
- Discovering new layer types

Coevolution of multiple aspects of network design?

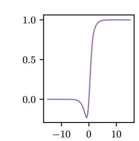
Evolutionary AutoML

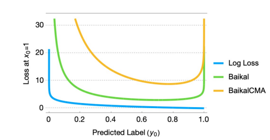
Current AutoML: Mostly hyperparameter optimization
Future Evolutionary AutoML: Many design aspects

Performance
1. Improve state of the art
   With sufficient compute



Applicability
2. Improve over naïve baseline
   Service makes broadly available
3. Minimize network resources
   Train and run networks faster
4. Extend small datasets
   Multitasking with related datasets





1 & 2 in 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

Estimate age from a facial image

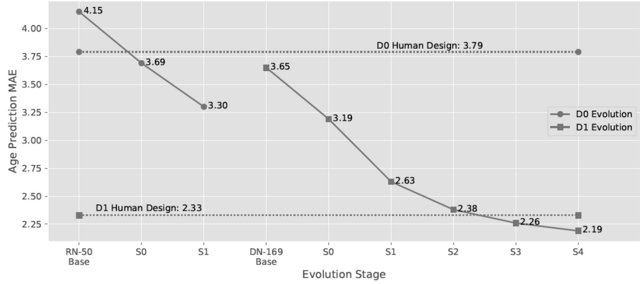
Evolving multiple design aspects [54]

- 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

3. Minimize Network Resources

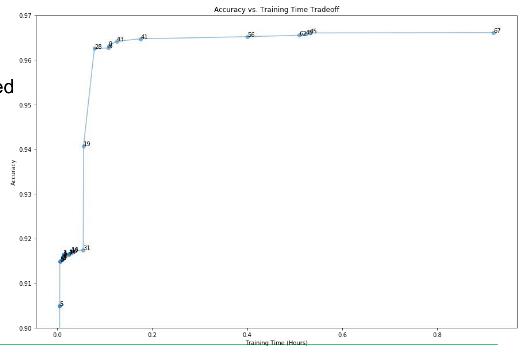
Evolution adds complexity only if needed

- Favors minimal solutions
- Over evolution a range of sizes explored
- Approximation of the Pareto front

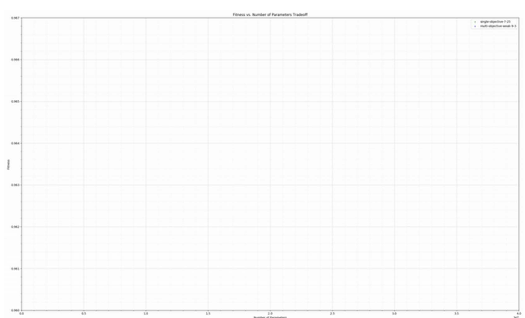
Small networks found that perform well

- Minimization with little cost
- E.g. 0.38% drop with 1/12th of the size

Could we optimize for size directly?

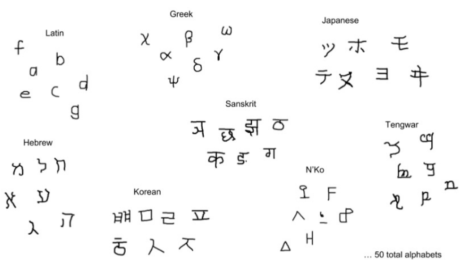


Multiobjective Minimization



- Animation: Pareto front by generation for single-objective (green) vs. multi-objective (blue)
- Single-objective focuses on improving *largest networks*
- Multi-objective focuses on improving the *entire curve*
- Result: Multi-objective finds much smaller models for the majority of performance values [36]
- Evolution can find solutions that fit design constraints

4. Extend Small Datasets



Lake et al. 2015 [29]

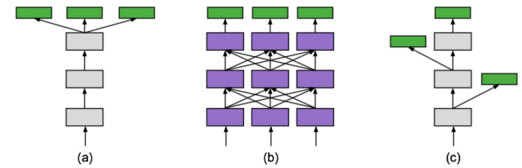
Recognize handwritten characters in a given alphabet

Not enough samples to learn well

- A common problem in deep learning

Could we learn from multiple alphabets?

Evolution of Multitask Architectures



- Learning in multiple tasks at once
  - More generalizable embeddings [37,46]
  - Each task can learn better
- Network structure can have a large effect
  - A good domain to test neuroevolution of structure

Multitasking Benchmarks

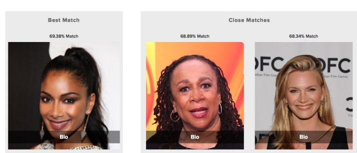
State-of-the-art in two ML benchmarks:

- Omniglot multialphabet character recognition [37]
  - Improved state-of-the-art 31%
  - Demo: evolution.ml/demos/omnidraw
- CelebA multiattribute face classification [45]
  - Improved state-of-the-art 0.75%
  - Demo: evolution.ml/demos/celebmatch

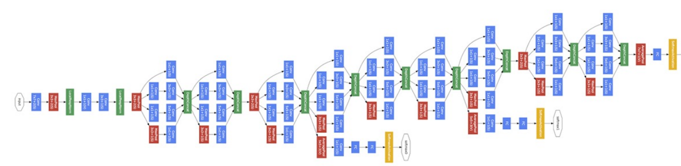
Improves learning in each task

- Even when little data available

Extend small datasets with multiple tasks

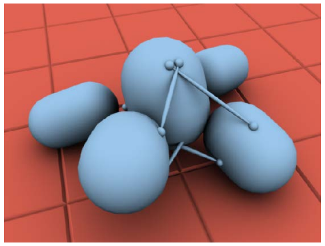


Part II Conclusion: Optimizing Deep Learning Designs

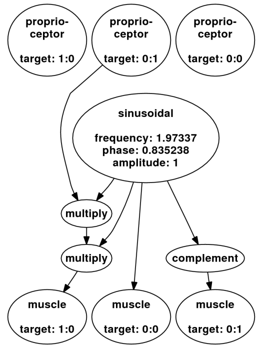


- Deep learning designs are too complex for humans to optimize
- Evolutionary techniques are a good fit
  - Large, structured space; continuous, discrete, and structured
- Can be applied to multiple aspects of the design
  - How to utilize their interactions?
  - How to evaluate candidates efficiently?

III. Emergence of Intelligence



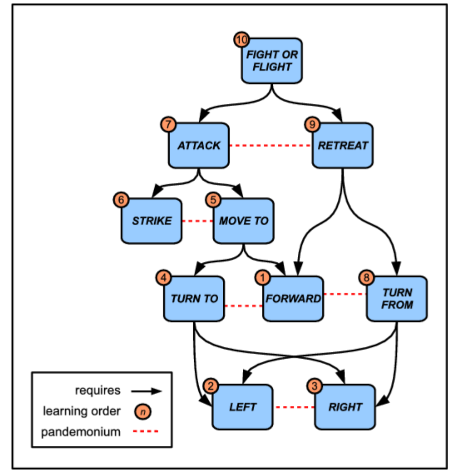
Body



Brain

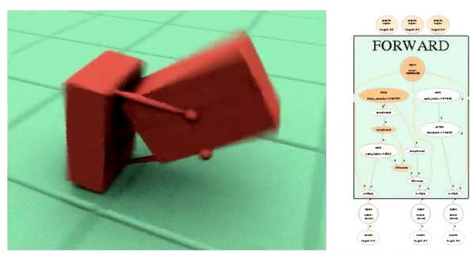
- Origins of intelligence: Embodied optimization
- Body-Brain Coevolution[32,33,77]
  - Body: Blocks, muscles, joints, sensors
  - Brain: A neural network (with general nodes)
  - Evolved together in a physical simulation

Syllabus



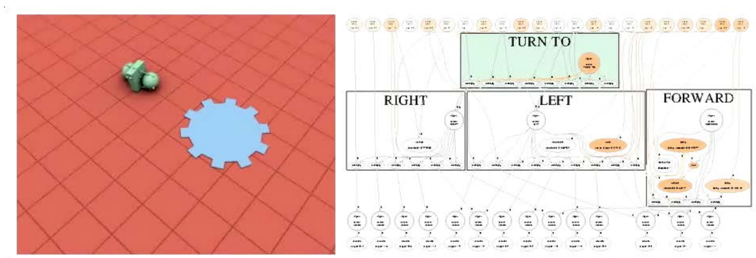
- Step-by-step construction of complex behavior
- Primitives and three levels of complexity
- Constructed by hand; body and brain evolved together

Encapsulation



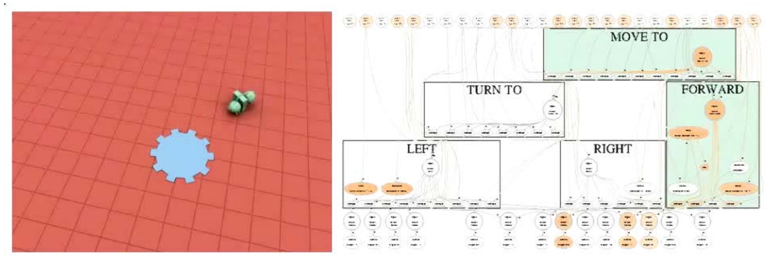
- Once evolved, a trigger node is added
- DEMO

Turn to Light



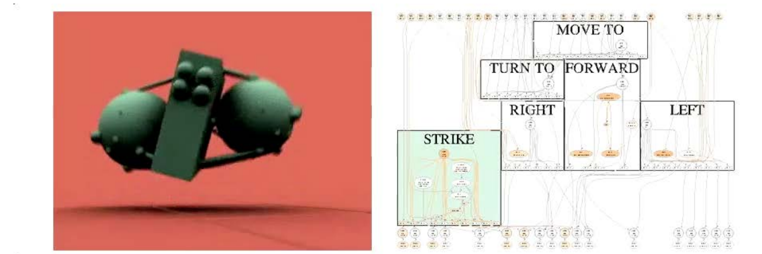
- First level of complexity
- Selecting between alternative primitives

Move to light



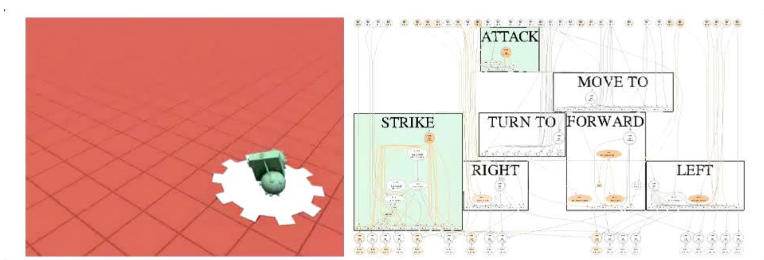
- First level of complexity (Sims 1994)
- Selecting between alternative primitives

Strike



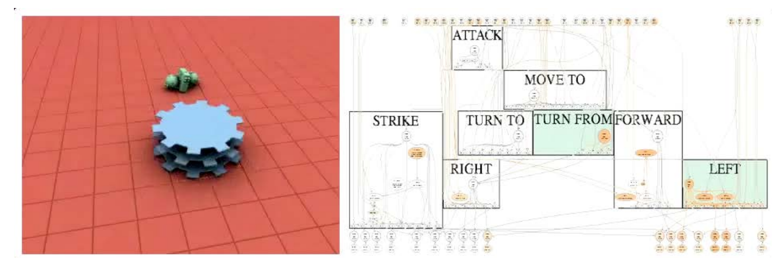
- Alternative behavior primitive

Attack



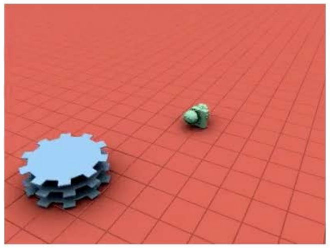
- Second level of complexity (beyond Sims and others)

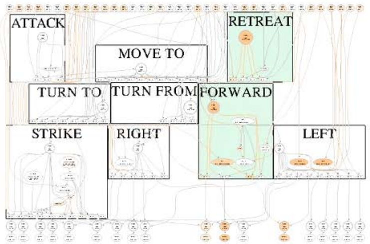
Turn from Light



- Alternative first-level behavior

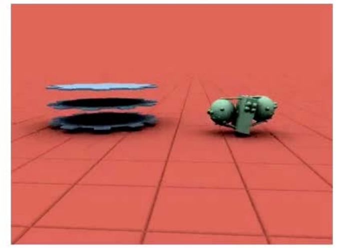
Retreat

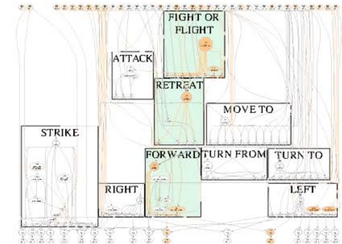




- Alternative second-level behavior

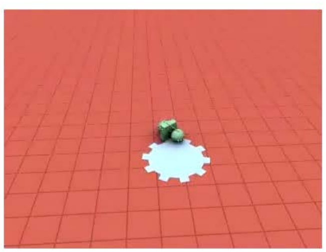
Fight or Flight

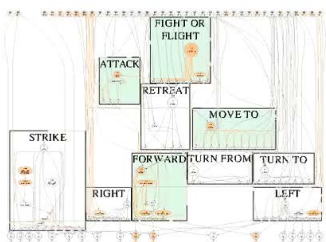




- Third level of complexity

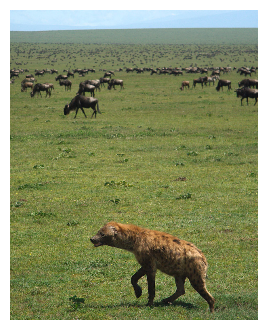
Insight: Body/Brain Coevolution





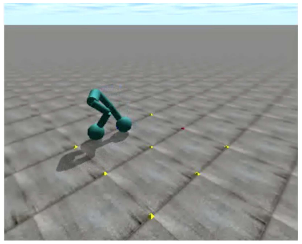
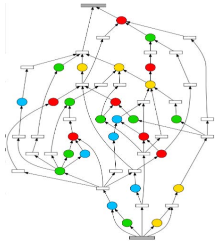
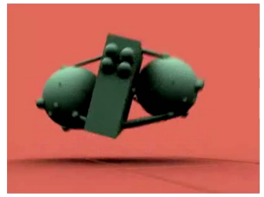
- Evolving body and brain together poses strong constraints
  - Behavior appears believable
  - Worked well also in BotPrize (Turing test for game bots)[71]
- Possible to construct innovative, situated behavior

Constructing Intelligent Systems



- Believable, complex behavior in embedded environments
  - Open-ended “arms race”[62]
- Similar to self-play e.g. in AlphaGo Zero
  - Complexity beyond human ability to design it
- If we can build open ended environments, we should be able to build more complex solutions
  - Co-evolve environments and behaviors? (e.g. POET[93], EUREQA[70])
  - Challenge: Establish major transitions[51]

Conclusion

- Neuroevolution is a powerful approach for POMDPs
  - Discovers surprising, believable, effective behavior
  - Games, robotics, control, alife, decision-making...
- Makes complex DL architectures possible
  - Structure, components, hyperparameters, etc. fit to the task
  - Automatic design of learning machines
- A possible future focus: Emergence of intelligence
  - Body/brain co-evolution; Competitive co-evolution
  - Evolution of memory, language, learning; AGI

Further Material

- Neuroevolution sessions at GECCO!
- www.cs.utexas.edu/users/risto/talks/enn-tutorial
  - Slides and references
  - Demos
  - A step-by-step neuroevolution exercise (evolving behavior in the NERO game)
- nn.cs.utexas.edu/?miikkulainen:encyclopedia20-ne [50]
  - A short summary of neuroevolution
- www.nature.com/articles/s42256-018-0006-z [80]
  - Nature Machine Intelligence survey on Neuroevolution

References I

[1] A. Agogino, K. Tumer, and R. Miikkulainen, Efficient credit assignment through evaluation function decomposition, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2005).

[2] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Parikh, and D. Batra, Vqa: Visual question answering, *International Journal of Computer Vision*, 123(1):4–31 (2017).

[3] P. J. Angeline, G. M. Saunders, and J. B. Pollack, An evolutionary algorithm that constructs recurrent neural networks, *IEEE Transactions on Neural Networks*, 5:54–65 (1994).

[4] R. K. Belew, Evolution, learning and culture: Computational metaphors for adaptive algorithms, *Complex Systems*, 4:11–49 (1990).

[5] G. Bingham, W. Macke, and R. Miikkulainen, Evolutionary optimization of deep learning activation functions, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2020).

[6] B. D. Bryant and R. Miikkulainen, Acquiring visibly intelligent behavior with example-guided neuroevolution, in: *Proceedings of the Twenty-Second National Conference on Artificial Intelligence*, AAAI Press, Menlo Park, CA (2007).

[7] D. J. Chalmers, The evolution of learning: An experiment in genetic connectionism, in: Touretzky et al. [88], 81–90.

[8] D. B. D'Ambrosio and K. O. Stanley, A novel generative encoding for exploiting neural network sensor and output geometry, in: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07)*, 974–981, ACM, New York, NY, USA (2007).

[9] D. B. D'Ambrosio and K. O. Stanley, Generative encoding for multiagent learning, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2008).

[10] K. Deb and C. Myburgh, A population-based fast algorithm for a billion-dimensional resource allocation problem with integer variablesbreaking the billion-variable barrier in real-world, *European Journal of Operational Research*, 261:460–474 (2017).

[11] X. Dong and Y. Yang, Nas-bench-201: Extending the scope of reproducible neural architecture search, in: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net (2020).

References II

[12] D. Floreano, P. Dürr, and C. Mattiussi, Neuroevolution: From architectures to learning, *Evolutionary Intelligence*, 1:47–62 (2008).

[13] D. Floreano and J. Urzelai, Evolutionary robots with on-line self-organization and behavioral fitness, *Neural Networks*, 13:431–4434 (2000).

[14] M. Fontaine and S. Nikolaidis, Differentiable quality diversity, in: *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds., volume 34, 10040–10052, Curran Associates, Inc. (2021).

[15] O. Francon, S. Gonzalez, B. Hodjat, E. Meyerson, R. Miikkulainen, X. Qiu, and H. Shahrzad, Effective reinforcement learning through evolutionary surrogate-assisted prescription, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2020)*, 814–822 (2020).

[16] B. Fullmer and R. Miikkulainen, Using marker-based genetic encoding of neural networks to evolve finite-state behaviour, in: *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, F. J. Varela and P. Bourgine, eds., 255–262, MIT Press, Cambridge, MA (1992).

[17] J. J. Gauci and K. O. Stanley, A case study on the critical role of geometric regularity in machine learning, in: *Proceedings of the Twenty-Third National Conference on Artificial Intelligence*, AAAI Press, Menlo Park, CA (2008).

[18] F. Gomez, *Robust Non-Linear Control Through Neuroevolution*, Ph.D. thesis, Department of Computer Sciences, The University of Texas at Austin (2003).

[19] F. Gomez and R. Miikkulainen, Incremental evolution of complex general behavior, *Adaptive Behavior*, 5:317–342 (1997).

[20] F. Gomez, J. Schmidhuber, and R. Miikkulainen, Accelerated neural evolution through cooperatively coevolved synapses, *Journal of Machine Learning Research*, 9:937–965 (2008).

[21] S. Gonzalez and R. Miikkulainen, Improved training speed, accuracy, and data utilization through loss function optimization, in: *Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC)* (2020).

[22] S. Gonzalez and R. Miikkulainen, Optimizing loss functions through multivariate taylor polynomial parameterization, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2021).

References III

[23] F. Gruau and D. Whitley, Adding learning to the cellular development of neural networks: Evolution and the Baldwin effect, *Evolutionary Computation*, 1:213–233 (1993).

[24] T. Hale, S. Webster, A. Petherick, T. Phillips, and B. Kira, Oxford COVID-19 government response tracker, Blavatnik School of Government (2020). https://www.bsg.ox.ac.uk/research/research-projects/coronavirus-government-response-tracker, accessed 5/26/2020.

[25] H. Hoos, Programming by optimization, *Communications of the ACM*, 55:70–80 (2012).

[26] C. Igel, Neuroevolution for reinforcement learning using evolution strategies, in: *Proceedings of the 2003 Congress on Evolutionary Computation*, R. Sarker, R. Reynolds, H. Abbass, K. C. Tan, B. McKay, D. Essam, and T. Gedeon, eds., 2588–2595, IEEE Press, Piscataway, NJ (2003).

[27] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan, C. Fernando, and K. Kavukcuoglu, Population based training of neural networks, *arXiv:1711.09846* (2017).

[28] A. Jain, A. Subramoney, and R. Miikkulainen, Task decomposition with neuroevolution in extended predator-prey domain, in: *Proceedings of Thirteenth International Conference on the Synthesis and Simulation of Living Systems*, East Lansing, MI, USA (2012).

[29] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, Human-level concept learning through probabilistic program induction, *Science*, 350(6266):1332–1338 (2015).

[30] J. Lehman, J. Chen, J. Clune, and K. O. Stanley, ES is more than just a traditional finite-difference approximator, *arXiv:1712.06568* (2017).

[31] J. Lehman and K. O. Stanley, Abandoning objectives: Evolution through the search for novelty alone, *Evolutionary Computation*, 19:189–223 (2010).

[32] D. Lessin, D. Fussell, and R. Miikkulainen, Open-ended behavioral complexity for evolved virtual creatures, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2013).

[33] D. Lessin, D. Fussell, and R. Miikkulainen, Trading control intelligence for physical intelligence: Muscle drives in evolved virtual creatures, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014)*, Vancouver, BC, Canada (July 2014).

References IV

[34] X. Li and R. Miikkulainen, Evolving adaptive poker players for effective opponent exploitation, in: *AAAI-17 Workshop on Computer Poker and Imperfect Information Games*, San Francisco, CA, USA (2017).

[35] J. Liang, S. Gonzalez, H. Shahrzad, and R. Miikkulainen, Regularized evolutionary population-based training, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2021).

[36] J. Liang, E. Meyerson, B. Hodjat, D. Fink, K. Mutch, and R. Miikkulainen, Evolutionary neural AutoML for deep learning, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2019)* (2019).

[37] J. Liang, E. Meyerson, and R. Miikkulainen, Evolutionary architecture search for deep multitask networks, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2018).

[38] J. Z. Liang and R. Miikkulainen, Evolutionary bilevel optimization for complex control tasks, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2015)*, Madrid, Spain (July 2015).

[39] H. Liu, A. Brock, K. Simonyan, and Q. V. Le, Evolving normalization-activation layers, *arXiv:2004.02967* (2020).

[40] Y. Liu, X. Yao, and T. Higuchi, Evolutionary ensembles with negative correlation learning, *IEEE Transactions on Evolutionary Computation*, 4:380–387 (2000).

[41] A. Lockett and R. Miikkulainen, Neuroannealing: Martingale-driven learning for neural network, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2013).

[42] P. McQuesten, *Cultural Enhancement of Neuroevolution*, Ph.D. thesis, Department of Computer Sciences, The University of Texas at Austin, Austin, TX (2002). Technical Report AI-02-295.

[43] E. Meyerson, J. Lehman, and R. Miikkulainen, Learning behavior characterizations for novelty search, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2016)*, ACM, New York, NY (2016).

[44] E. Meyerson and R. Miikkulainen, Discovering evolutionary stepping stones through behavior domination, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2017)*, 139–146, Berlin, Germany (2017).

[45] E. Meyerson and R. Miikkulainen, Pseudo-task augmentation: From deep multitask learning to intratask sharing—and back, in: *Proceedings of the 35th International Conference on Machine Learning* (2018).

References V

[46] E. Meyerson and R. Miikkulainen, Modular universal reparameterization: Deep multi-task learning across diverse domains, in: *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alche-Buc, E. Fox, and R. Garnett, eds., Curran Associates, Inc. (2019).

[47] E. Meyerson and R. Miikkulainen, The traveling observer model: Multi-task learning through spatial variable embeddings, in: *International Conference on Learning Representations* (2021).

[48] E. Meyerson, X. Qiu, and R. Miikkulainen, Simple genetic operators are universal approximators of probability distributions (and other advantages of expressive encodings), in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2022).

[49] R. Miikkulainen, Creative AI through evolutionary computation, in: *Evolution in Action: Past, Present and Future*, W. Banzhaf, B. Cheng, K. Deb, K. Holekamp, R. E. Lenski, C. Ofria, R. Pennock, B. Punch, and D. Whittaker, eds., Springer, New York (2020).

[50] R. Miikkulainen, Neuroevolution, in: *Encyclopedia of Machine Learning and Data Science, 3rd Edition*, D. Phung, C. Sammut, and G. I. Webb, eds., Springer, New York (in press).

[51] R. Miikkulainen and S. Forrest, A biological perspective on evolutionary computation, *Nature Machine Intelligence*, 3:9–15 (2021).

[52] R. Miikkulainen, O. Francon, E. Meyerson, X. Qiu, D. Sargent, E. Canzani, and B. Hodjat, From prediction to prescription: Evolutionary optimization of non-pharmaceutical interventions in the COVID-19 pandemic, *IEEE Transactions on Evolutionary Computation*, 25:386–401 (2021).

[53] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, and B. Hodjat, Evolving deep neural networks, in: *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, C. F. Morabito, C. Alippi, Y. Choe, and R. Kozma, eds., Elsevier, New York (2020).

[54] R. Miikkulainen, E. Meyerson, X. Qiu, U. Sinha, R. Kumar, K. Hofmann, Y. M. Yan, M. Ye, J. Yang, D. Caiazza, and S. M. Brown, Evaluating medical aesthetics treatments through evolved age-estimation models, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2021).

[55] E. Mjolsness, D. H. Sharp, and B. K. Alpert, Scaling, machine learning, and genetic neural nets, *Advances in Applied Mathematics*, 10:137–163 (1989).

References VI

[56] D. J. Montana and L. Davis, Training feedforward neural networks using genetic algorithms, in: *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, 762–767, San Francisco: Morgan Kaufmann (1989).

[57] D. E. Moriarty and R. Miikkulainen, Forming neural networks through efficient and adaptive co-evolution, *Evolutionary Computation*, 5:373–399 (1997).

[58] S. Nolfi and D. Parisi, Good teaching inputs do not correspond to desired responses in ecological neural networks, *Neural Processing Letters*, 1(2):1–4 (1994).

[59] D. Pardoe, M. Ryoo, and R. Miikkulainen, Evolving neural network ensembles for control problems, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2005).

[60] M. A. Potter and K. A. D. Jong, Cooperative coevolution: An architecture for evolving coadapted subcomponents, *Evolutionary Computation*, 8:1–29 (2000).

[61] J. K. Pugh, L. B. Soros, and K. O. Stanley, Quality diversity: A new frontier for evolutionary computation, *Frontiers in Robotics and AI*, 3 (2016).

[62] A. Rawal, P. Rajagopalan, and R. Miikkulainen, Constructing competitive and cooperative agent behavior using coevolution, in: *IEEE Conference on Computational Intelligence and Games (CIG 2010)*, Copenhagen, Denmark (2010).

[63] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, Regularized evolution for image classifier architecture search, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI (2019).

[64] E. Real, C. Liang, D. So, and Q. Le, AutoML-zero: Evolving machine learning algorithms from scratch, in: *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, eds., volume 119, 8007–8019 (2020).

[65] J. Reisinger and R. Miikkulainen, Acquiring evolvability through adaptive representations, in: *Proceeedings of the Genetic and Evolutionary Computation Conference*, 1045–1052 (2007).

[66] J. Reisinger, K. O. Stanley, and R. Miikkulainen, Evolving reusable neural modules, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2004).

References VII

[67] T. P. Runarsson and M. T. Jonsson, Evolution and design of distributed learning rules, in: *Proceedings of The First IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, 59–63, IEEE, Piscataway, NJ (2000).

[68] T. Salimans, J. Ho, X. Chen, and I. Sutskever, Evolution strategies as a scalable alternative to reinforcement learning, *arXiv:1703.03864* (2017).

[69] J. D. Schaffer, D. Whitley, and L. J. Eshelman, Combinations of genetic algorithms and neural networks: A survey of the state of the art, in: *Proceedings of the International Workshop on Combinations of Genetic Algorithms and Neural Networks*, D. Whitley and J. Schaffer, eds., 1–37, IEEE Computer Society Press, Los Alamitos, CA (1992).

[70] M. Schmidt and H. Lipson, Distilling free-form natural laws from experimental data, *Science*, 324(5923):81–85 (2009).

[71] J. Schrum, I. Karpov, and R. Miikkulainen, Humanlike combat behavior via multiobjective neuroevolution, in: *Believable Bots*, P. Hingston, ed., Springer, New York (2012).

[72] J. Schrum and R. Miikkulainen, Evolving agent behavior in multiobjective domains using fitness-based shaping, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2010).

[73] J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, J. T. Folsom-Kovarik, and K. O. Stanley, Picbreeder: A case study in collaborative evolutionary exploration of design space, *Evolutionary Computation*, 19:345–371 (2011).

[74] H. Shahrzad, B. Hodjat, C. Dolle, A. Denissov, S. Lau, D. Goodhew, J. Dyer, and R. Miikkulainen, Enhanced optimization with composite objectives and novelty pulsation, in: *Genetic Programming Theory and Practice XVII*, W. Banzhaf, E. Goodman, L. Sheneman, L. Trujillo, and B. Worzel, eds., 275–293, Springer, New York (2020).

[75] A. A. Siddiqi and S. M. Lucas, A comparison of matrix rewriting versus direct encoding for evolving neural networks, in: *Proceedings of IEEE International Conference on Evolutionary Computation*, 392–397, IEEE, Piscataway, NJ (1998).

[76] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, Mastering the game of go without human knowledge, *Nature*, 550:354–359 (2017).

References VIII

[77] K. Sims, Evolving 3D morphology and behavior by competition, in: *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems (Artificial Life IV)*, R. A. Brooks and P. Maes, eds., 28–39, MIT Press, Cambridge, MA (1994).

[78] K. O. Stanley, *Efficient Evolution of Neural Networks Through Complexification*, Ph.D. thesis, Department of Computer Sciences, The University of Texas at Austin, Austin, TX (2003).

[79] K. O. Stanley, B. D. Bryant, and R. Miikkulainen, Evolving adaptive neural networks with and without adaptive synapses, in: *Proceedings of the 2003 Congress on Evolutionary Computation*, IEEE, Piscataway, NJ (2003).

[80] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, Designing neural networks through evolutionary algorithms, *Nature Machine Intelligence*, 1:24–35 (2019).

[81] K. O. Stanley and J. Lehman, *Why Greatness Cannot Be Planned: The Myth of the Objective*, Springer, Berlin (2015).

[82] K. O. Stanley and R. Miikkulainen, Evolving Neural Networks Through Augmenting Topologies, *Evolutionary Computation*, 10:99–127 (2002).

[83] K. O. Stanley and R. Miikkulainen, A taxonomy for artificial embryogeny, *Artificial Life*, 9(2):93–130 (2003).

[84] D. G. Stork, S. Walker, M. Burns, and B. Jackson, Preadaptation in neural circuits, in: *International Joint Conference on Neural Networks* (Washington, DC), 202–205, IEEE, Piscataway, NJ (1990).

[85] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning, *arXiv:1712.06567* (2017).

[86] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going deeper with convolutions, in: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9 (2015).

[87] W. Tansey, E. Feasley, and R. Miikkulainen, Accelerating evolution via egalitarian social learning, in: *Proceedings of the 14th Annual Genetic and Evolutionary Computation Conference (GECCO 2012)*, Philadelphia, Pennsylvania, USA (July 2012).

[88] D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, eds., *Proceedings of the 1990 Connectionist Models Summer School*, San Francisco: Morgan Kaufmann (1990).

References IX

[89] C. C. Tutum, S. Abdulquddos, and R. Miikkulainen, Generalization of agent behavior through explicit representation of context, in: *Proceedings of the 3rd IEEE Conference on Games (COG-2021)* (2021).

[90] J. Urzelai, D. Floreano, M. Dorigo, and M. Colombetti, Incremental robot shaping, *Connection Science*, 10:341–360 (1998).

[91] V. K. Valsalam and R. Miikkulainen, Evolving symmetric and modular neural networks for distributed control, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2009*, 731–738, ACM, New York, NY, USA (2009).

[92] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, Show and tell: A neural image caption generator, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3156–3164 (2015).

[93] R. Wang, J. Lehman, J. Clune, and K. O. Stanley, POET: open-ended coevolution of environments and their optimized solutions, in: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019, Prague, Czech Republic, July 13-17, 2019*, 142–151 (2019).

[94] S. Whiteson, N. Kohl, R. Miikkulainen, and P. Stone, Evolving keepaway soccer players through task decomposition, *Machine Learning*, 59:5–30 (2005).

[95] S. Whiteson and P. Stone, Evolutionary function approximation for reinforcement learning, *Journal of Machine Learning Research*, 7:877–917 (2006).

[96] D. Whitley, S. Dominic, R. Das, and C. W. Anderson, Genetic reinforcement learning for neurocontrol problems, *Machine Learning*, 13:259–284 (1993).

[97] A. P. Wieland, Evolving controls for unstable systems, in: Touretzky et al. [88], 91–102.

[98] X. Yao, Evolving artificial neural networks, *Proceedings of the IEEE*, 87(9):1423–1447 (1999).

[99] C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, and F. Hutter, NAS-bench-101: Towards reproducible neural architecture search, in: *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, eds., volume 97, 7105–7114 (2019).

[100] A. Zela, J. N. Siems, L. Zimmer, J. Lukasik, M. Keuper, and F. Hutter, Surrogate NAS benchmarks: Going beyond the limited search spaces of tabular NAS benchmarks, in: *International Conference on Learning Representations* (2022).