Evolution of Neural Networks

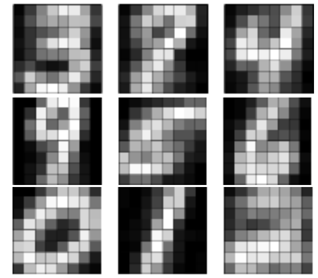
Risto Miikkulainen

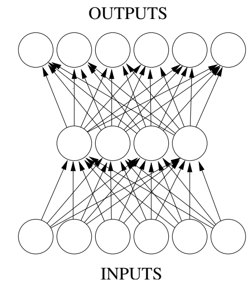
The University of Texas at Austin
and Cognizant Technology Solutions

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07...$15.00
https://doi.org/10.1145/3449726.3461432

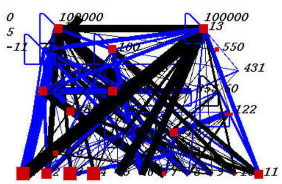
Why Use Neural Networks?

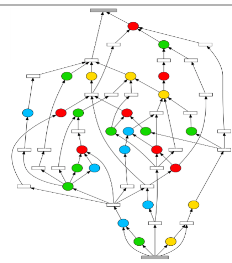


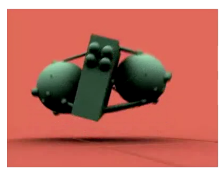


- Neural nets powerful in many statistical domains
  - E.g. control, pattern recognition, prediction, decision making
  - Where no good theory of the domain exists
- Good supervised training algorithms exist
  - Learn a nonlinear function that matches the examples
  - Utilize big datasets

Why Evolve Neural Networks?

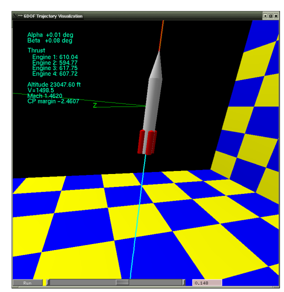






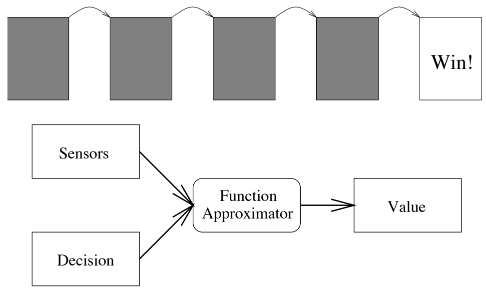
- I. Original role (since 1990s): Sequential Decision Tasks
  - Both the structure and the weights evolved (no training)
  - Power from recurrency: POMDP tasks; behavior
- II. A new role (since 2016): Optimization of Deep Learning Nets
  - Architecture, hyperparameters, functions evolved; weights trained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

I. Sequential Decision Tasks



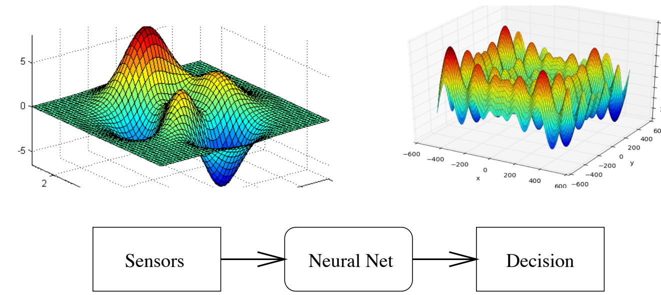
- A sequence of decisions creates a sequence of states
  - States are only partially known
  - Optimal outputs are not known
  - We can only tell how well we are doing
- Exist in many important real-world domains
  - Robot/vehicle/traffic control
  - Computer/manufacturing/process optimization
  - Game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

Value-Function Reinforcement Learning



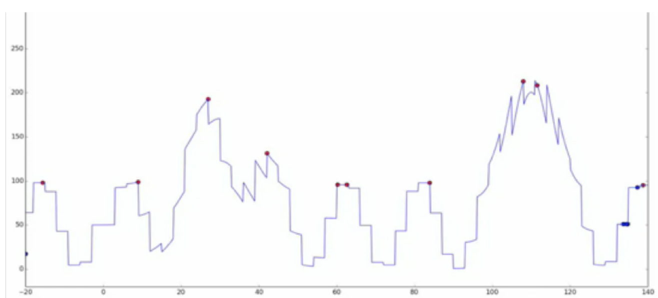
- E.g. Q-learning, Temporal Differences
  - Generate targets through prediction errors
  - Learn when successive predictions differ
- Predictions represented as a value function
  - Values of alternatives at each state
- Difficult with large/continuous state and action spaces
- Difficult with hidden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

Policy-Search Reinforcement Learning



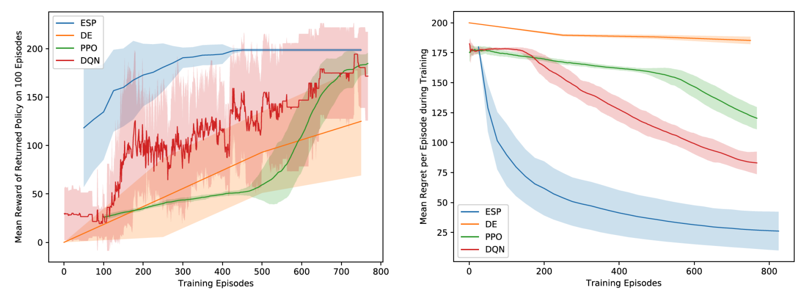
- E.g. REINFORCE, policy gradients
- The policy is optimized directly through hill climbing
- Works well in simple cases
  - Large/continuous states and actions possible
  - Hidden states (in POMDP) disambiguated through memory
  - Does not scale well

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[45])
  - to high dimensionality (e.g. up to 1B[9])
  - to deceptive landscapes (with e.g. multiobj and novelty[66])

How Well Does It Work?

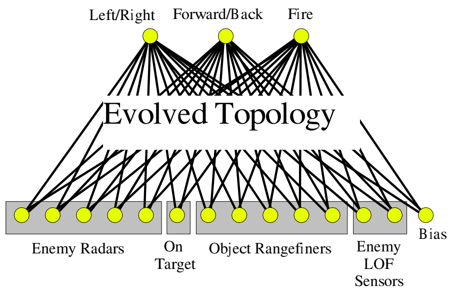


(a) True Performance

(b) Regret

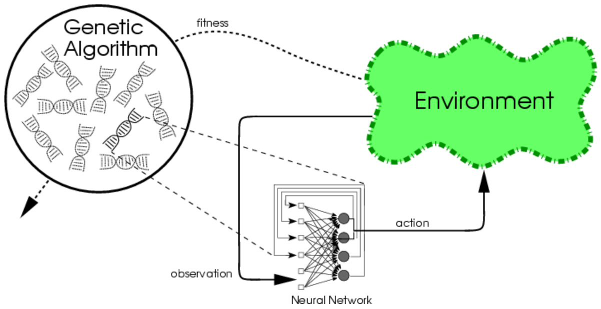
- In the OpenAI Gym CartPole-v0 benchmark vs. PPO, DQN
  - NE converges faster, has lower variance, lower regret
  - NE is more efficient, reliable, and safer[12]
- In a double-pole benchmark vs. Sarsa, Q-MLP, etc.
  - The only method that can find solutions to 1m, 0.1m, POMDP[17]

Neuroevolution for POMDP



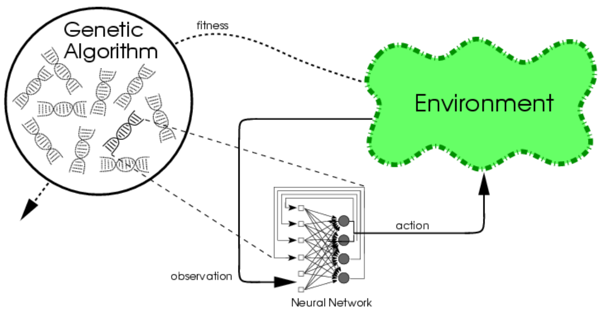
- Input variables describe the state observed through sensors
- Output variables describe actions
- Network between input and output:
  - Recurrent connections implement memory
  - Memory helps with POMDP

Basic Neuroevolution (1)



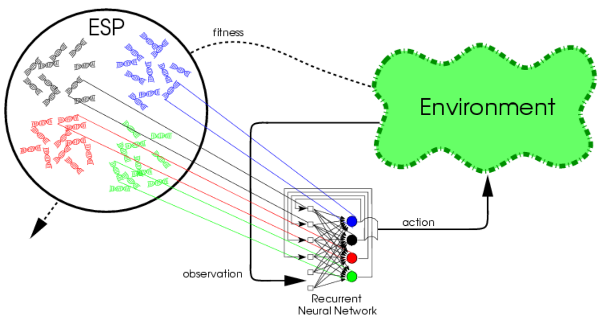
- Evolving connection weights in a population of networks [49,61,84,85]
- Chromosomes are strings of connection weights (bits or real)
  - E.g. 10010110101100101111001
  - Usually fully connected, fixed topology
  - Initially random

Basic Neuroevolution (2)



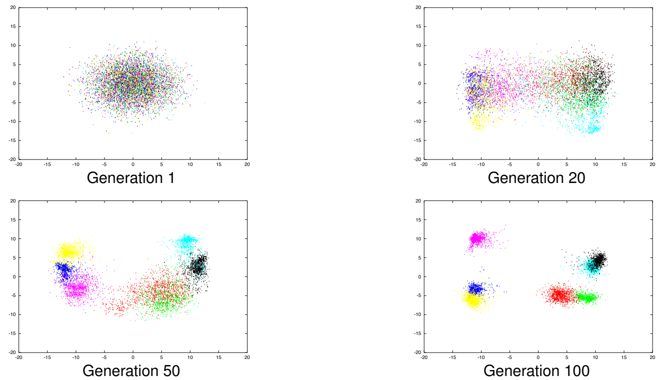
- Parallel search for a solution network
  - Each NN evaluated in the task
  - Good NN reproduce through crossover, mutation
  - Bad thrown away
- Natural mapping between genotype and phenotype
  - GA and NN are a good match!

Advanced NE 1: Evolving Partial Networks



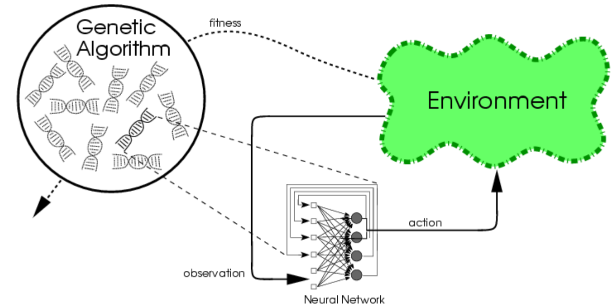
- Evolving individual neurons to cooperate in networks [1,50,53]
- E.g. Enforced Sub-Populations (ESP [15])
  - Each (hidden) neuron in a separate subpopulation
  - Fully connected; weights of each neuron evolved
  - Populations learn compatible subtasks
- Can be applied at the level of weights, and modules [17]

Evolving Neurons with ESP



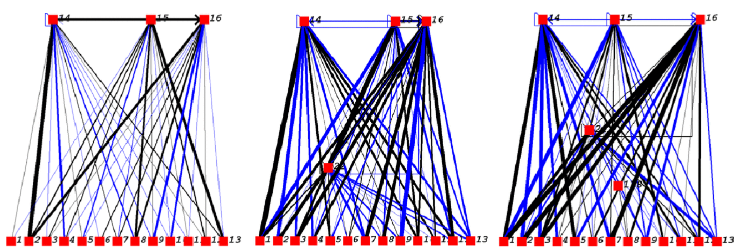
- Evolution encourages diversity automatically
  - Good networks require different kinds of neurons
- Evolution discourages competing conventions
  - Neurons optimized for compatible roles
- Large search space divided into subtasks
  - Optimize compatible neurons

Advanced NE 2: Evolutionary Strategies



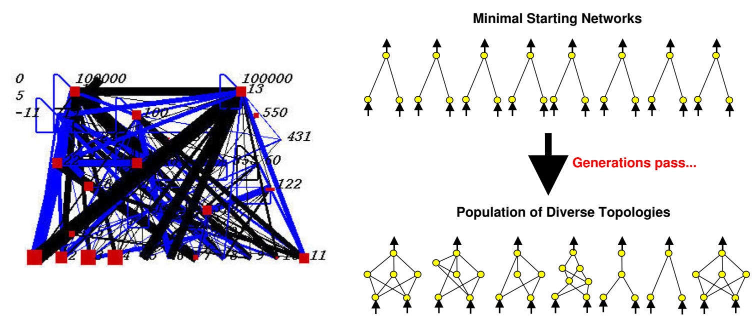
- Evolving complete networks with ES (CMA-ES[22])
- Small populations, no crossover
- Instead, intelligent mutations
  - Adapt covariance matrix of mutation distribution
  - Take into account correlations between weights
- Smaller space, less convergence, fewer conventions

Advanced NE 3: Evolving Network Structure



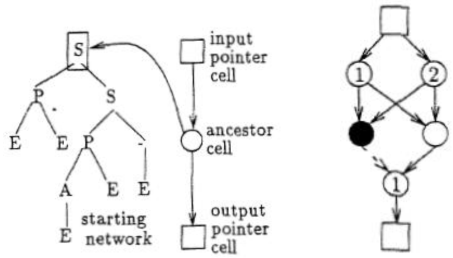
- Optimizing connection weights and network topology[2,10,13,86]
- E.g. Neuroevolution of Augmenting Topologies (NEAT[69,72])
- Based on *Complexification*
- Of networks:
  - Mutations to add nodes and connections
- Of behavior:
  - Elaborates on earlier behaviors

Why Complexification?



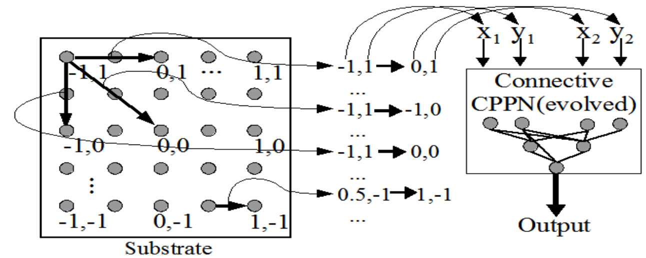
- Challenge with NE: Search space is very large
- Complexification keeps the search tractable
  - Start simple, add more sophistication
- Incremental construction of intelligent agents

Advanced NE 4: Indirect Encodings (1)

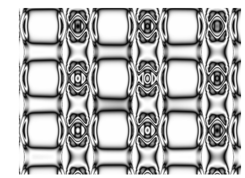


- Instructions for constructing the network evolved
  - Instead of specifying each unit and connection[2,10,48,67,86]
- E.g. Cellular Encoding (CE[20])
- Grammar tree describes construction
  - Sequential and parallel cell division
  - Changing thresholds, weights
  - A "developmental" process that results in a network

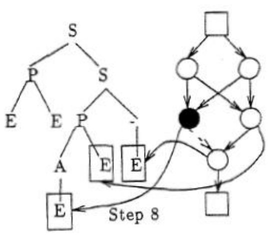
Indirect Encodings (2)

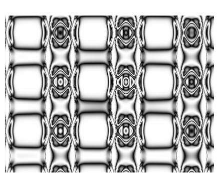


- Encode the networks as spatial patterns
- E.g. Hypercube-based NEAT (HyperNEAT[7])
- Evolve a neural network (CPPN) to generate spatial patterns
  - 2D CPPN: (x, y) input $\rightarrow$ grayscale output
  - 4D CPPN: (x_1, y_1, x_2, y_2) input $\rightarrow$ w output
  - Connectivity and weights can be evolved indirectly
  - Works with very large networks (millions of connections)



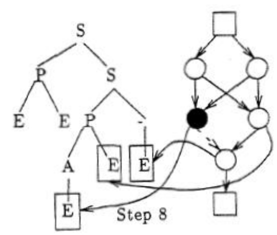
Properties of Indirect Encodings (1)

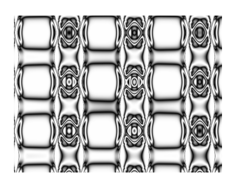




- Smaller search space
- Avoids competing conventions
- Describes classes of networks efficiently
- Modularity, reuse of structures
  - Recurrency symbol in CE: XOR $\rightarrow$ parity
  - Repetition with variation in CPPNs
  - Useful for evolving morphology

Properties of Indirect Encodings (2)





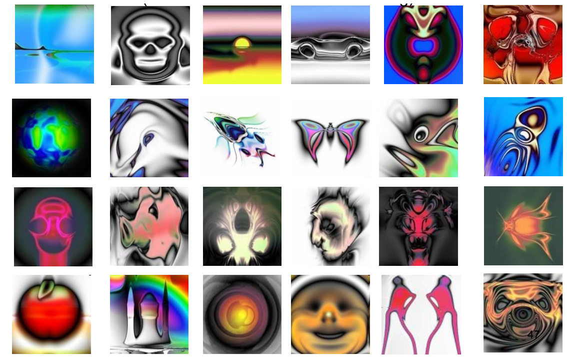
- Not fully explored (yet)
  - See e.g. CS track at GECCO
- Promising current work
  - More general L-systems; developmental codings; embryogeny[73]
  - Scaling up spatial coding[8,14]
  - Genetic Regulatory Networks[57]
  - Evolution of symmetries[79]

Further NE Techniques

- Incremental and multiobjective evolution[16,64,78,85]
- Utilizing population culture[3,39,76]
- Utilizing evaluation history[38]
- Evolving NN ensembles and modules[24,37,52,58,82]
- Evolving transfer functions and learning rules[6,59,74]
- Bilevel optimization of NE[35]
- Evolving LSTMs for strategic behavior[30]
- Combining learning and evolution[5,11,39,51,70,76,83]

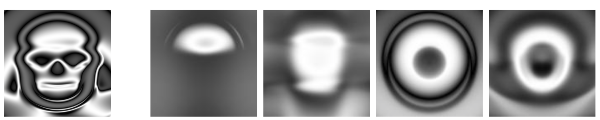
- Evolving for novelty

Evolving for Novelty



- Motivated by humans as fitness functions
- E.g. picbreeder.com, endlessforms.com[65]
  - CPPNs evolved; Human users select parents
- No specific goal
  - Interesting solutions preferred
  - Similar to biological evolution?

Novelty Search

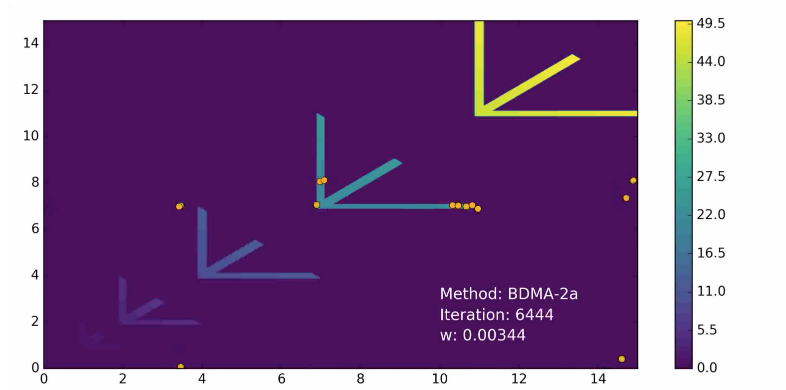


23f, 57c 74 gen | 20f, 24c failed | 20f, 29c failed | 19f, 24c failed | 22f, 28c failed

gen 12 | gen 20 | gen 36 | gen 49 | gen 74

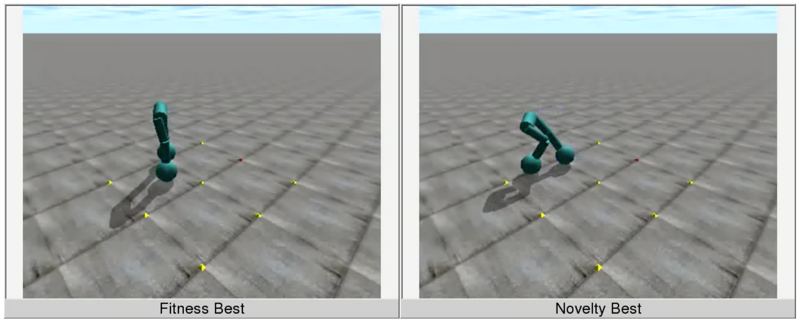
- Evolutionary algorithms maximize a performance objective
  - But sometimes hard to achieve it step-by-step
- Novelty search rewards candidates that are simply different[27,71]
  - Stepping stones for constructing complexity

Novelty Search Demo (1)



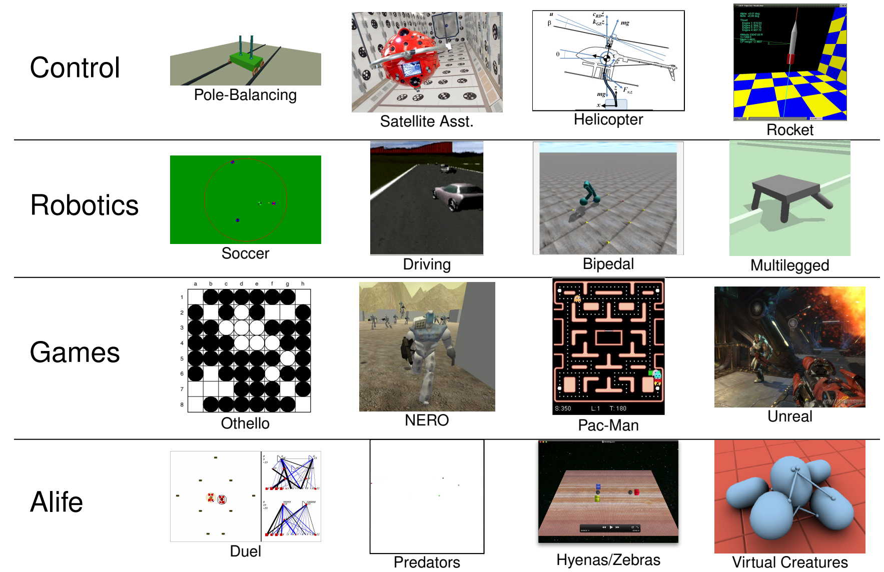
- Illustration of stepping stones[40,41]
  - Nonzero fitness on “feet” only; stepwise increase
  - Top and right “toes” are stepping stones to next “foot”
  - Difficult for fitness based search; novelty can do it
- DEMO

Novelty Search Demo (2)

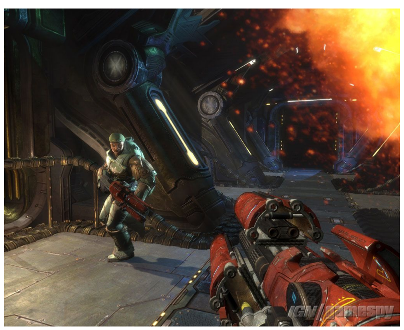


- Fitness-based evolution is rigid
  - Requires gradual progress
- Novelty-based evolution is more innovative, natural[27,71]
  - Allows building on stepping stones
  - As a secondary objective—or even the only one!
- DEMO

Neuroevolution Applications



Games: Evolving Humanlike Behavior

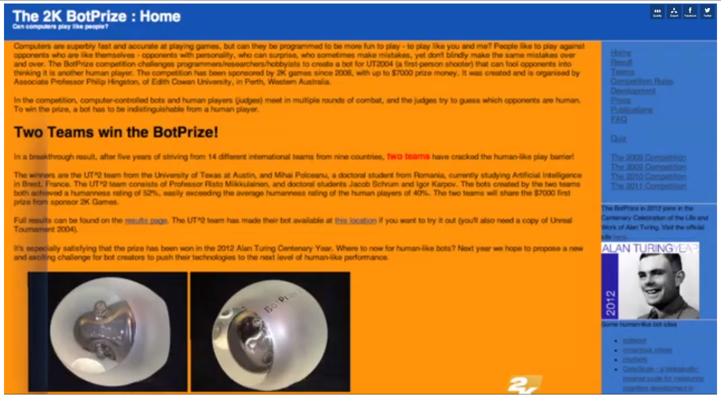


- Botprize competition, 2007-2012
  - Turing Test for game bots ($10,000 prize)
- Three players in Unreal Tournament 2004:
  - Human confederate: tries to win
  - Software bot: pretends to be human
  - Human judge: tries to tell them apart!

Evolving an Unreal Bot

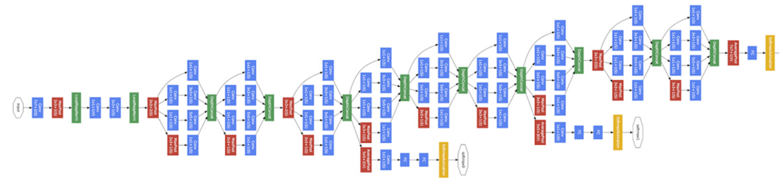
- Evolve effective fighting behavior[63]
  - Human-like with resource limitations (speed, accuracy...)
- Also scripts & learning from humans (unstuck, wandering...)
- 2007-2011: bots 25-30% vs. humans 35-80% human
- 6/2012 best bot better than 50% of the humans
- 9/2012...?

Success!!!



- In 2012, two teams reach the 50% mark!
- Fascinating challenges remain:
  - Judges can still differentiate in seconds
  - Judges lay cognitive, high-level traps
  - Team competition: collaboration as well
- DEMO

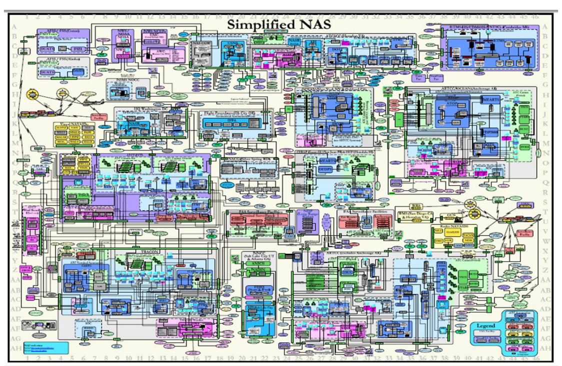
II. Optimization of DL Architectures



Szegedy et al. 2014

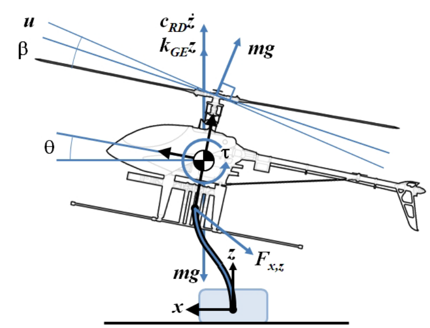
- Big Data and Big Compute available since 2000s
  - Machine learning systems have scaled up
- E.g. Deep Learning ideas existed since the 1990s
  - With million times more data & compute, they now work!
- A new problem: How to configure such systems?

Configuring Complex Systems



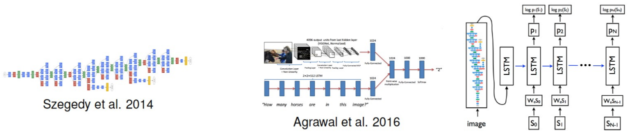
- A new general approach to engineering
  - Humans design just the framework
  - Machines optimize the details
- Programming by optimization[21]

E.g. Optimizing NE in Helicopter Hovering



- A challenging benchmark
  - RL, NE solutions exist
- Eight parameters optimized by hand[25]
  - Hard for a human designer to do more
- With EA, increased to 15
  - →Significantly better performance[35]

Designing Deep Learning Architectures



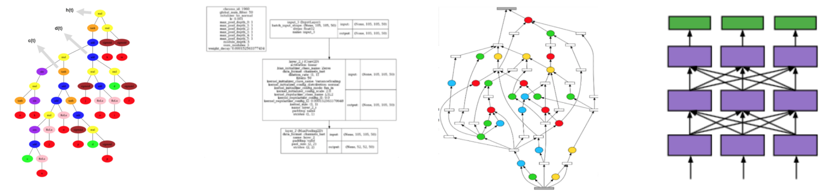
Szegedy et al. 2014

Agrawal et al. 2016

(Vinyals et al. 2016)

- Different architectures work better in different tasks
  - Structure matters!
- Too complex to be discovered by hand
  - How to discover principles of organization?
  - How to cover enough of the space?

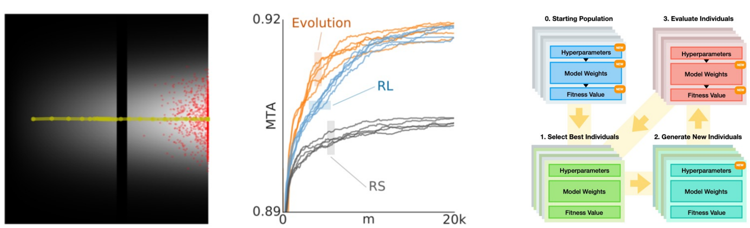
How to Discover Network Structure?



Evolutionary Neural Architecture Search is a natural fit:
- Population-based search covers the space
- Crossover between structures discovers principles
- Novelty search maximizes exploration

Building on Neuroevolution work since the 1990s
Hyperparameters; nodes; modules; topologies; multiple tasks

Progress in Evolutionary Deep Learning

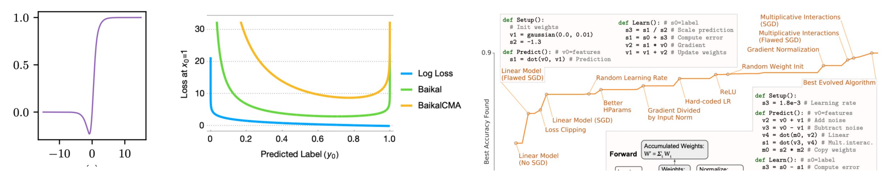


Understanding ES and GAs in RL (Uber, OpenAI)[26,60,75]
- ES provides more exploration than gradients; GA more than ES

Scaling and regularization (Google)[55]
- State-of-the art in CIFAR-10, CIFAR-100, ImageNet

Population-based training (DeepMind, Cognizant)[23,32]
- Continual training and evolution

Progress in Evolutionary Deep Learning (2)

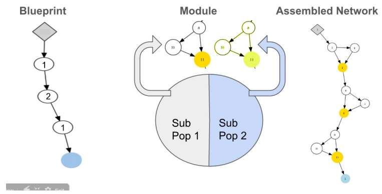


Optimizing activation functions and loss functions (Cognizant)[4,18,19,32]
- Regularization and refinement (Gonzalez et al. GECCO'21; Liang et al. GECCO21)

Designing machine learning algorithms with GP (Google)[36,56]
- Adapts to different task types
- Discovering new layer types

Coevolution of multiple aspects of network design?

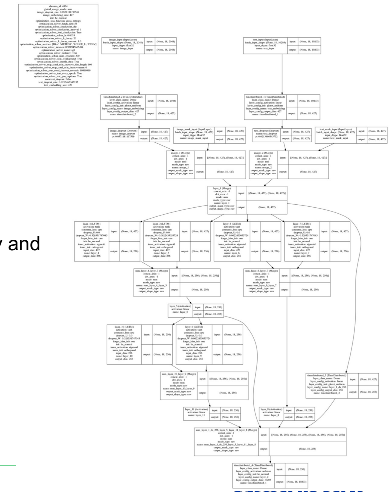
E.G. Neural Architecture Search with CoDeepNEAT



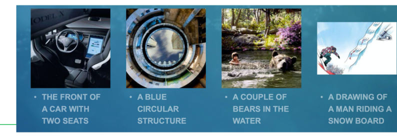
Evolution at three levels[46]
- Module subpopulations optimize building blocks
- Blueprint population optimizes their combinations
- Hyperparameter evolution optimizes their instantiation

Fitness of the complete network drives evolution
Applies to both CNN (vision), LSTM (language) networks

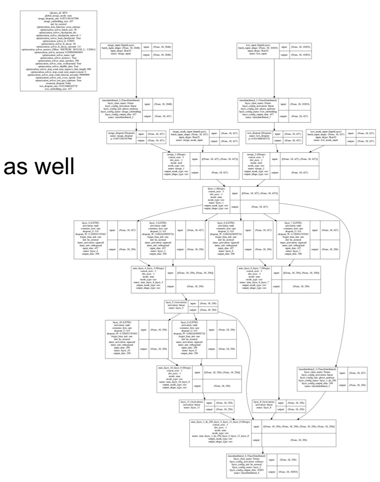
Improve Human Design



- E.g. image captioning:
  - Start with a state-of-the art design: Show&Tell[80]
  - Search in the space of similar elements
  - 5% improvement
  - A prototype service on the web
- Best-performing AI defies human notions of symmetry and patterns of organization
- AI designing AI: could we automate it?



Evolutionary AutoML

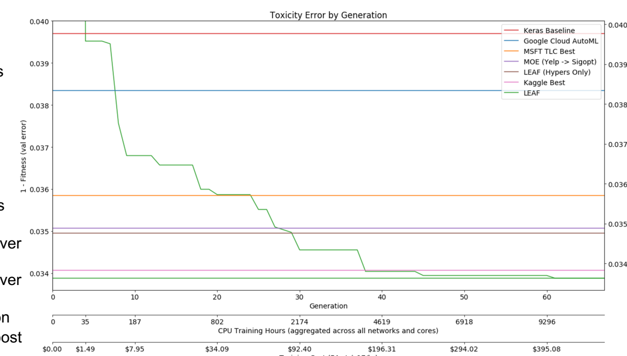


Current AutoML: Hyperparameter optimization
Evolutionary AutoML: Architectures and modules as well

1. Improve over naïve baseline
   20% or more with little effort
2. Improve state of the art
   With more expertise & compute
3. Minimize network resources
   Train and run networks faster
4. Extend small datasets
   Multitasking with related datasets

1 and 2: Improve Performance

- Domain: Wikipedia Toxic Comment Identification
  - Why: Toxicity is bad for business
  - Data: 160K labeled comments
  - Challenge: highly diverse vocabulary, style, and length
- Layer Types: Conv1D, LSTM, GRU
- LEAF Results:[33]
  - With minimal compute: Improves over naïve Keras baseline
  - With more compute: Improves over other AutoML methods
  - With more compute: Improves over SOTA hand-designed model.
  - LEAF Hyperparameter Search on final architecture gives a final boost
- Similar results on Age Estimation[47]
  (Miikkulainen et al. GECCO-21)

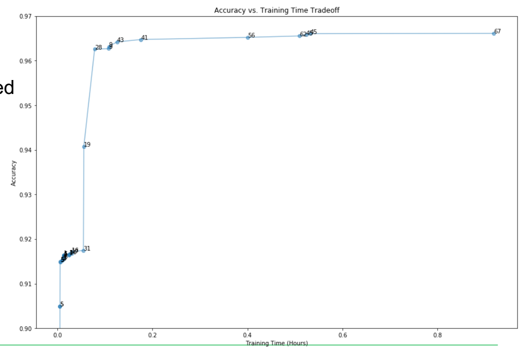


3. Minimize Network Resources

Evolution adds complexity only if needed
- Favors minimal solutions
- Over evolution a range of sizes explored
- Approximation of the Pareto front

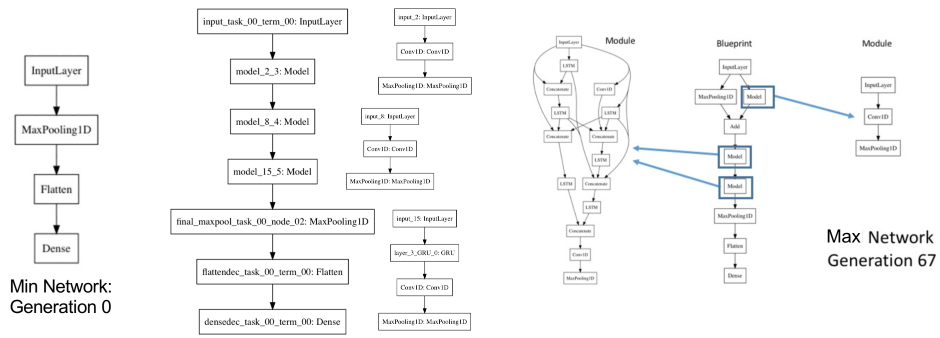
Small networks found that perform well
- Minimization with little cost
- E.g. 0.38% drop with 1/12th of the size

Adding a size objective will explore more[33]



Cognizant

Example Performance/Size Tradeoffs



Tradeoff Network: Generation 28
- Sequential, GRUs after 2 pooling layers

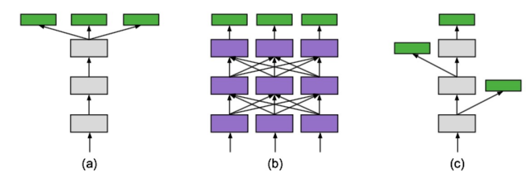
Cognizant

4. Extend Small Datasets

- Recognize characters in 50 alphabets

Cognizant

Harnessing Multiple Datasets through Multitasking



- Learning in multiple tasks at once
  - More generalizable embeddings[34,43]
  - Each task can learn better
- Network structure can have a large effect
  - A good domain to test neuroevolution of structure

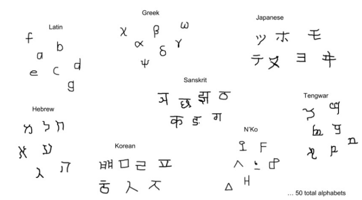
Cognizant

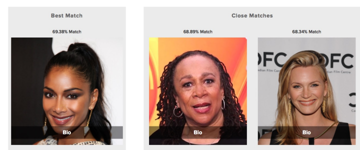
Multitasking Benchmarks

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

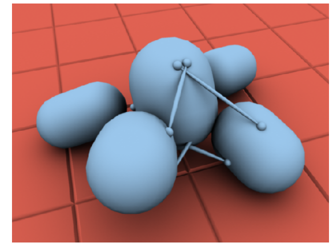
- Omniglot multialphabet character recognition [34]
  - Improved state-of-the-art 31%
  - Demo: evolution.ml/demos/omnidraw
- CelebA multiattribute face classification [42]
  - Improved state-of-the-art 0.75%
  - Demo: evolution.ml/demos/celebmatch

Improves learning in each task
- Even when little data available

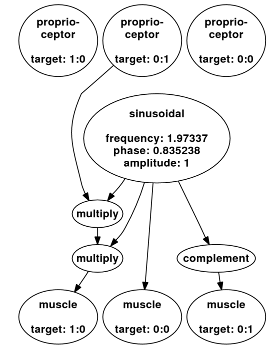




III. Emergence of Intelligence



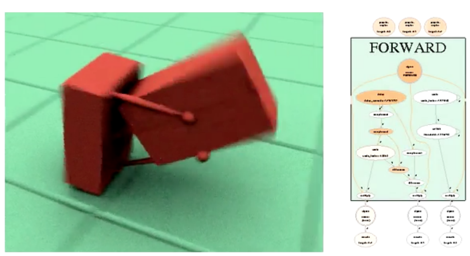
Body



Brain

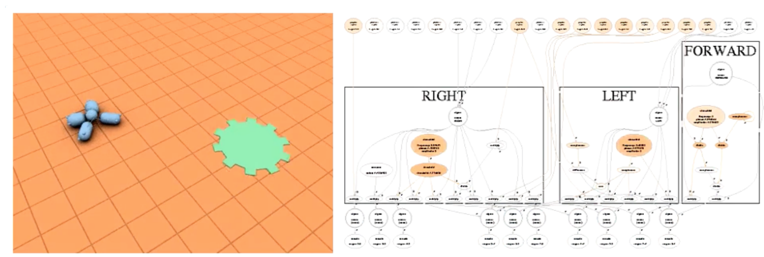
- Origins of intelligence: Embodied optimization
- Body-Brain Coevolution [28,29,68]
  - Body: Blocks, muscles, joints, sensors
  - Brain: A neural network (with general nodes)
  - Evolved together in a physical simulation
- Encapsulation, Pandemodium, Syllabus

Encapsulation



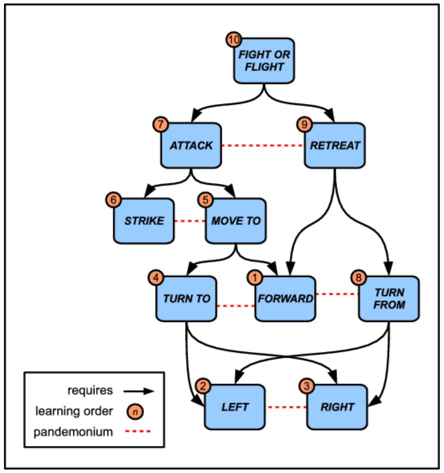
- Once evolved, a trigger node is added
- DEMO

Pandemonium



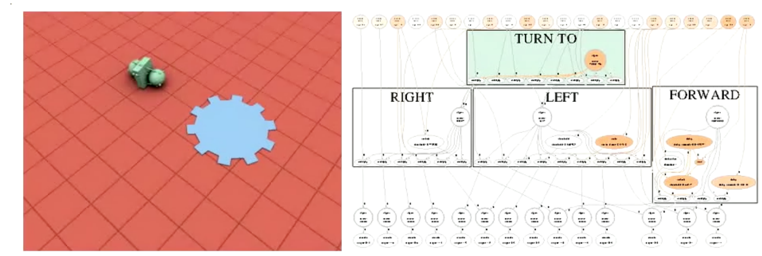
- Conflicting behaviors: Highest trigger wins
- DEMO

Syllabus



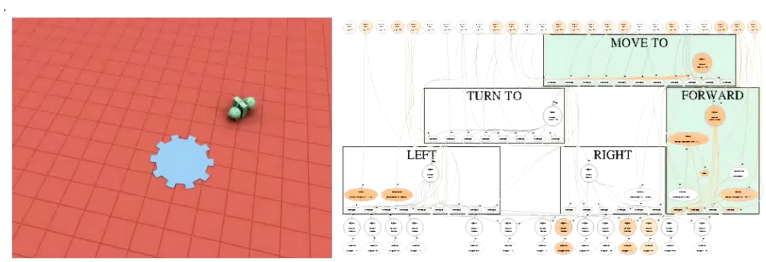
- Step-by-step construction of complex behavior
- Primitives and three levels of complexity
- Constructed by hand; body and brain evolved together

Turn to Light



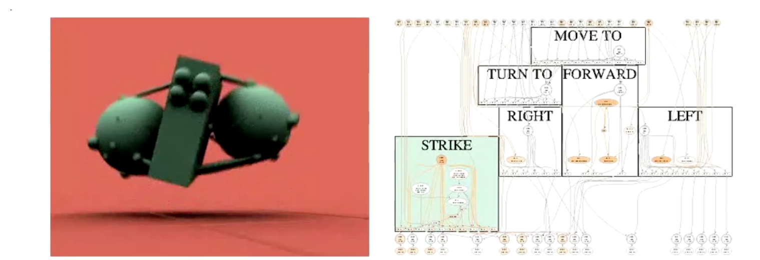
- First level of complexity
- Selecting between alternative primitives

Move to light



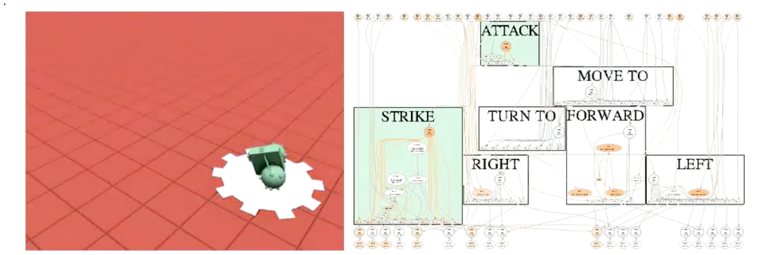
- First level of complexity (Sims 1994)
- Selecting between alternative primitives

Strike



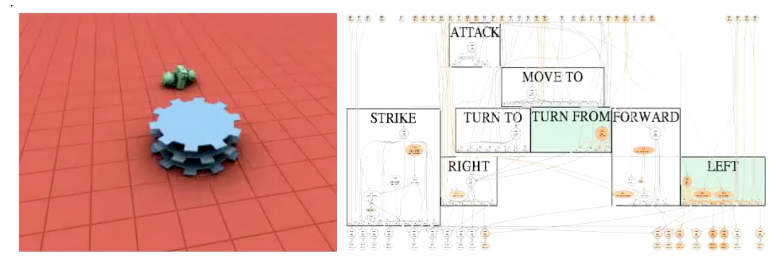
- Alternative behavior primitive

Attack



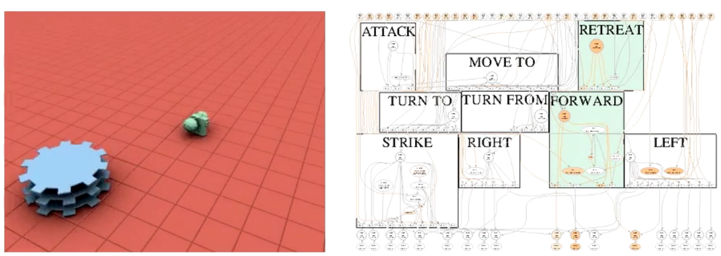
- Second level of complexity (beyond Sims and others)

Turn from Light



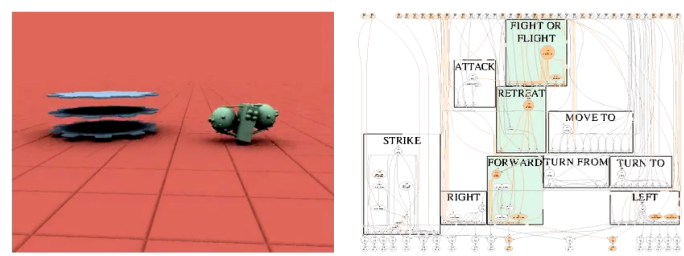
- Alternative first-level behavior

Retreat



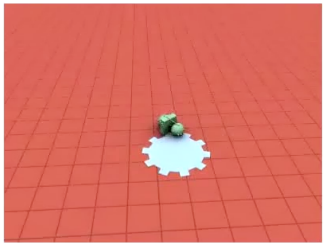
- Alternative second-level behavior

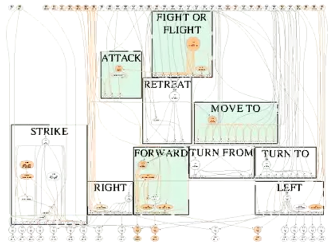
Fight or Flight



- Third level of complexity

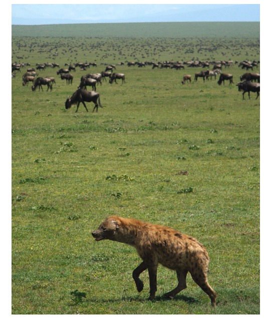
Insight: Body/Brain Coevolution





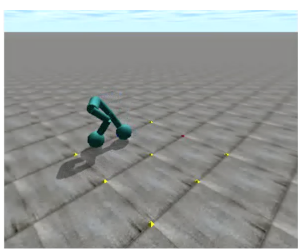
- Evolving body and brain together poses strong constraints
  - Behavior appears believable
  - Worked well also in BotPrize (Turing test for game bots)[63]
- Possible to construct innovative, situated behavior

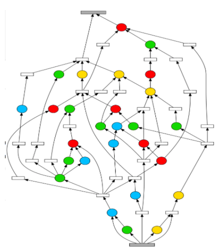
Constructing Intelligent Systems

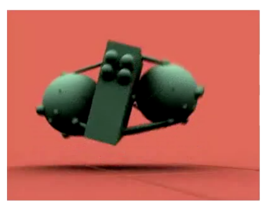


- Believable, complex behavior in embedded environments
  - Open-ended “arms race”[54]
- Similar to self-play e.g. in AlphaGo Zero
  - Complexity beyond human ability to design it
- If we can build open ended environments, we should be able to build more complex solutions
  - Co-evolve environments and behaviors? (e.g. POET[81], EUREQA[62])

Conclusion







- AI extending from prediction to creativity
  - i.e. from modeling to optimization
  - i.e. from Deep Learning to Evolution/RL
- Evolutionary optimization of neural networks can
  - Discover novel and strategic behavior
  - Discover useful complexity for Deep Learning
  - Gain insight into origins of intelligence

Further Material

- 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)
- http://nn.cs.utexas.edu/?miikkulainen:encyclopedia20-new
  - A short summary of neuroevolution
- www.nature.com/articles/s42256-018-0006-z
  - Nature Machine Intelligence survey on Neuroevolution
- arxiv.org/abs/1902.09635
  - Proposal for NAS benchmark

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] 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).

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

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

[5] 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).

[6] D. J. Chalmers, The evolution of learning: An experiment in genetic connectionism, in: Touretzky et al.[77], 81–90.

[7] 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).

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

[9] 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).

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

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

References II

[12] 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).

[13] 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).

[14] 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).

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

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

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

[18] 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).

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

[20] 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).

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

[22] 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).

References III

[23] 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, *CoRR*, abs/1711.09846 (2017).

[24] 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).

[25] R. Koppejan and S. Whiteson, Neuroevolutionary reinforcement learning for generalized control of simulated helicopters, *Evolutionary Intelligence*, 4:219–241 (2011).

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

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

[28] 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).

[29] 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).

[30] 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).

[31] J. Liang, S. Gonzalez, and R. Miikkulainen, Population-based training for loss function optimization, *arXiv:2002.04225* (2020).

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

[33] 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).

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

References IV

[35] 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).

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

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

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

[39] 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.

[40] 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).

[41] 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).

[42] 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).

[43] 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).

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

[45] 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).

References V

[46] 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).

[47] 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).

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

[49] 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).

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

[51] 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).

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

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

[54] 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).

[55] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, Regularized evolution for image classifier architecture search, *arXiv:1802.01548* (2018).

[56] E. Real, C. Liang, D. R. So, and Q. V. Le, Automl-zero: Evolving machine learning algorithms from scratch, *arXiv:2003.03384* (2020).

References VI

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

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

[59] 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).

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

[61] 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).

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

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

[64] 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).

[65] 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).

[66] 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).

References VII

[67] 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).

[68] 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).

[69] 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).

[70] 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).

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

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

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

[74] 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).

[75] 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).

[76] 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).

[77] 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).

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

References VIII

[79] 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).

[80] 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).

[81] 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).

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

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

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

[85] A. P. Wieland, Evolving controls for unstable systems, in: Touretzky et al. [77], 91–102.

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