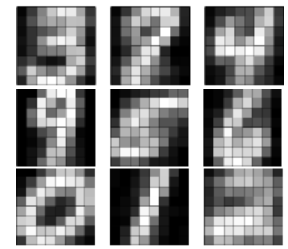
Evolution of Neural Networks

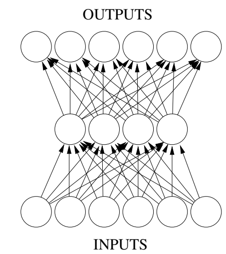
Risto Miikkulainen

The University of Texas at Austin and
Sentient Technologies, Inc.

GECCO '17 Companion, July 15-19, 2017, Berlin, Germany
ACM 978-1-4503-4939-0/17/07.
http://dx.doi.org/10.1145/3067695.3067716

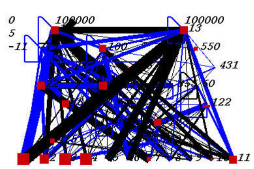
Why Use Neural Networks?

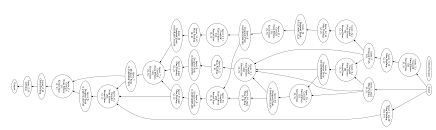




- Neural nets powerful in many statistical domains
  - E.g. control, pattern recognition, prediction, decision making
  - Where no good theory of the domain exists
- Good supervised training algorithms exist
  - Learn a nonlinear function that matches the examples
  - Utilize big datasets

Why Evolve Neural Networks?



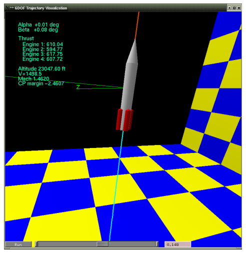


- Traditional role (since 1990s): Solving POMDP tasks
  - Both the structure and the weights evolved (no training)
  - Power from recurrency
- A new role: Optimization of Deep Learning Architectures
  - Components, topology, hyperparameters evolved; weights trained
  - Power from complexity
- Allows solving more challenging tasks with neural networks

Outline

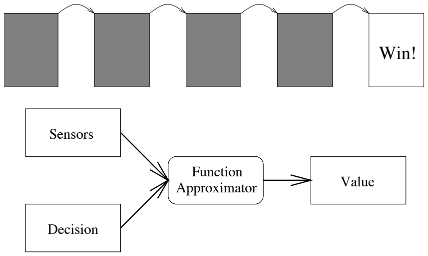
- I. Neuroevolution for POMDP tasks
  - NE vs. traditional RL
  - Basic and advanced NE techniques; Novelty search
  - Applications: Control, Robotics, Games, Alife
- II. Optimization of Deep Learning Architectures
  - Deep neural networks, Autoencoders, LSTMs
  - Computational requirements
  - Applications: Vision, language modeling

Sequential Decision Tasks



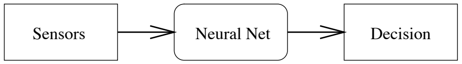
- A sequence of decisions creates a sequence of states
  - States are only partially known
  - Optimal outputs are not known
  - We can only tell how well we are doing
- Exist in many important real-world domains
  - Robot/vehicle/traffic control
  - Computer/manufacturing/process optimization
  - Game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

Standard Reinforcement Learning



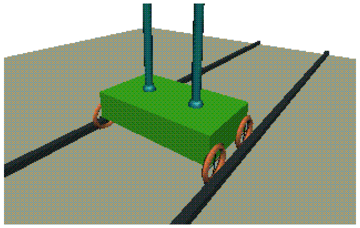
- AHC, Q-learning, Temporal Differences
  - Generate targets through prediction errors
  - Learn when successive predictions differ
- Predictions represented as a value function
  - Values of alternatives at each state
- Difficult with large/continuous state and action spaces
- Difficult with hidden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

Neuroevolution (NE) Reinforcement Learning



- NE = constructing neural networks with evolutionary algorithms
- Direct nonlinear mapping from sensors to actions
- Large/continuous states and actions easy
  - Generalization in neural networks
- Hidden states (in POMDP) disambiguated through memory
  - Recurrency in neural networks[73]
  - Deep Reinforcement Learning[52,59]

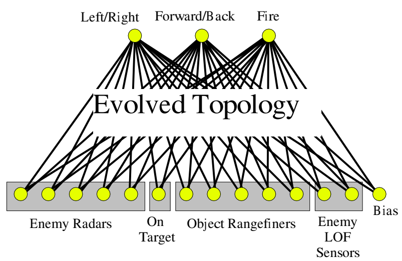
How Well Does It Work?



Poles	Method	Evals	Succ.
One	VAPS	(500,000)	0%
	SARSA	13,562	59%
	Q-MLP	11,331	
	NE	127	
Two	NE	3,416	

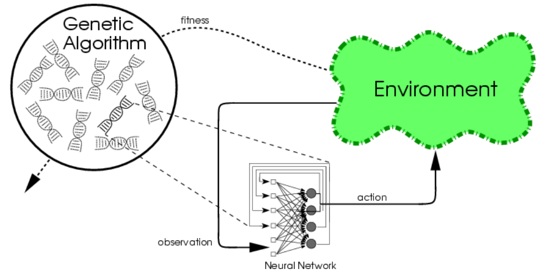
- Difficult RL benchmark: POMDP Pole Balancing
- NE 2-3 orders of magnitude faster than standard RL[22]
- NE can solve harder problems

Neuroevolution for POMDP



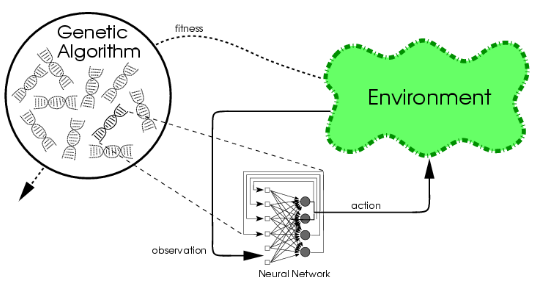
- Input variables describe the state observed through sensors
- Output variables describe actions
- Network between input and output:
  - Recurrent connections implement memory
  - Memory helps with POMDP

Basic Neuroevolution (1)



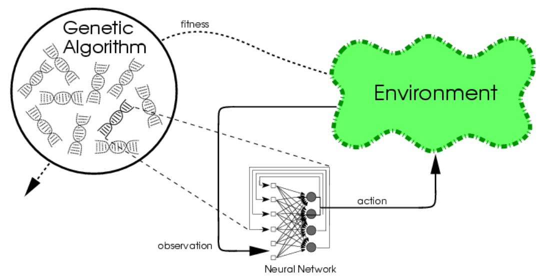
- Evolving connection weights in a population of networks [44,58,87,88]
- Chromosomes are strings of connection weights (bits or real)
  - E.g. 10010110101100101111001
  - Usually fully connected, fixed topology
  - Initially random

Basic Neuroevolution (2)



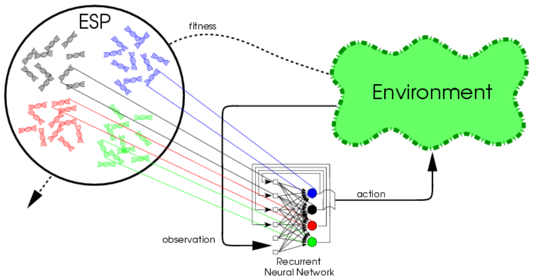
- Parallel search for a solution network
  - Each NN evaluated in the task
  - Good NN reproduce through crossover, mutation
  - Bad thrown away
- Natural mapping between genotype and phenotype
  - GA and NN are a good match!

Problems with Basic Neuroevolution



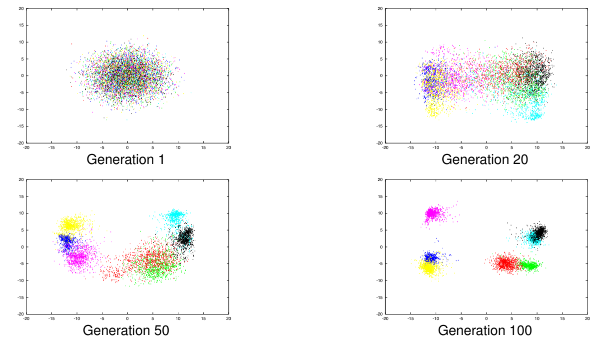
- Evolution converges the population (as usual with EAs)
  - Diversity is lost; progress stagnates
- Competing conventions
  - Different, incompatible encodings for the same solution
- Too many parameters to be optimized simultaneously
  - Thousands of weight values at once

Advanced NE 1: Evolving Partial Networks



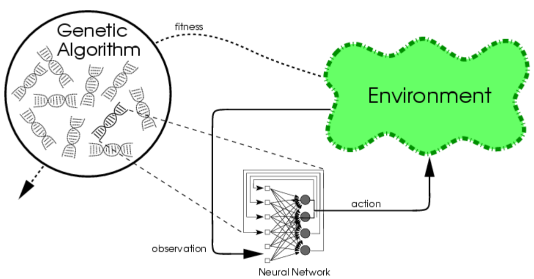
- Evolving individual neurons to cooperate in networks[1,45,51]
- E.g. Enforced Sub-Populations (ESP[19])
  - Each (hidden) neuron in a separate subpopulation
  - Fully connected; weights of each neuron evolved
  - Populations learn compatible subtasks
- Can be applied at the level of weights, and modules

Evolving Neurons with ESP



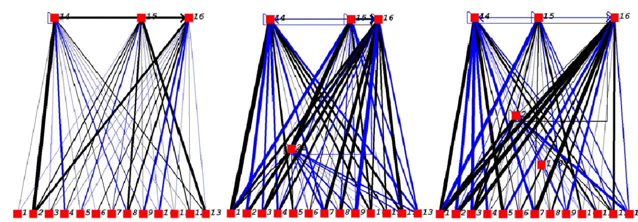
- Evolution encourages diversity automatically
  - Good networks require different kinds of neurons
- Evolution discourages competing conventions
  - Neurons optimized for compatible roles
- Large search space divided into subtasks
  - Optimize compatible neurons

Advanced NE 2: Evolutionary Strategies



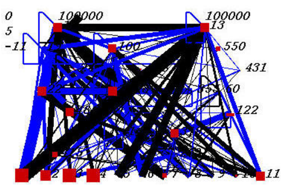
- Evolving complete networks with ES (CMA-ES[28])
- Small populations, no crossover
- Instead, intelligent mutations
  - Adapt covariance matrix of mutation distribution
  - Take into account correlations between weights
- Smaller space, less convergence, fewer conventions

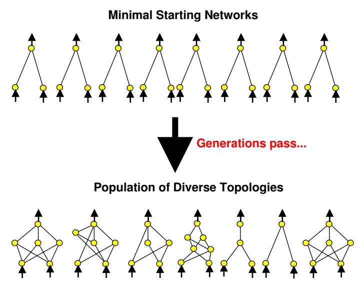
Advanced NE 3: Evolving Network Structure



- Optimizing connection weights and network topology[2,15,17,89]
- E.g. Neuroevolution of Augmenting Topologies (NEAT[66,69])
- Based on *Complexification*
- Of networks:
  - Mutations to add nodes and connections
- Of behavior:
  - Elaborates on earlier behaviors

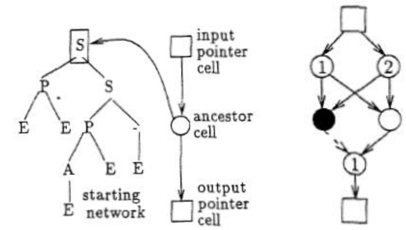
Why Complexification?





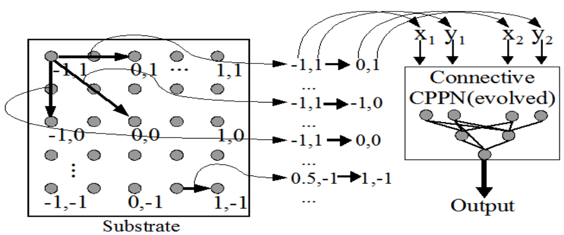
- Challenge with NE: Search space is very large
- Complexification keeps the search tractable
  - Start simple, add more sophistication
- Incremental construction of intelligent agents

Advanced NE 4: Indirect Encodings (1)

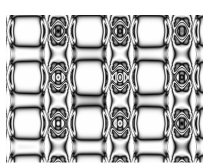


- Instructions for constructing the network evolved
  - Instead of specifying each unit and connection[2,15,43,64,89]
- E.g. Cellular Encoding (CE[24])
- Grammar tree describes construction
  - Sequential and parallel cell division
  - Changing thresholds, weights
  - A "developmental" process that results in a network

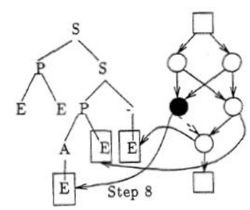
Indirect Encodings (2)

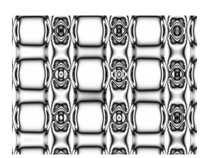


- Encode the networks as spatial patterns
- E.g. Hypercube-based NEAT (HyperNEAT[9])
- Evolve a neural network (CPPN) to generate spatial patterns
  - 2D CPPN: (x, y) input → grayscale output
  - 4D CPPN: (x_1, y_1, x_2, y_2) input → w output
  - Connectivity and weights can be evolved indirectly
  - Works with very large networks (millions of connections)



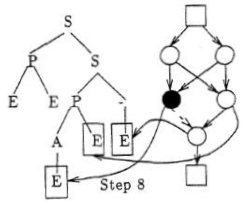
Properties of Indirect Encodings (1)

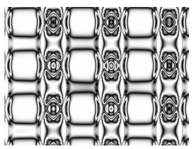




- Smaller search space
- Avoids competing conventions
- Describes classes of networks efficiently
- Modularity, reuse of structures
  - Recurrency symbol in CE: XOR → parity
  - Repetition with variation in CPPNs
  - Useful for evolving morphology

Properties of Indirect Encodings (2)





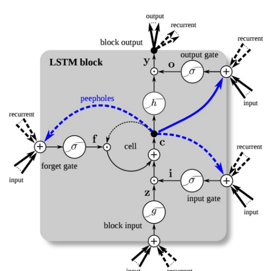
- Not fully explored (yet)
  - See e.g. CS track at GECCO
- Promising current work
  - More general L-systems; developmental codings; embryogeny[70]
  - Scaling up spatial coding[10,18]
  - Genetic Regulatory Networks[54]
  - Evolution of symmetries[80]

Further NE Techniques

- Incremental and multiobjective evolution[21,61,75,88]
- Utilizing population culture[5,40,72]
- Utilizing evaluation history[37]
- Evolving NN ensembles and modules[29,36,50,55,84]
- Evolving transfer functions and learning rules[7,56,71]
- Bilevel optimization of NE[35]

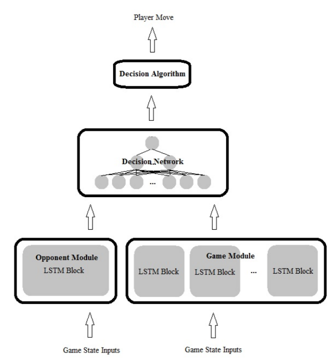
- Evolving LSTMs for strategic behavior
- Combining learning and evolution
- Evolving for novelty

Extending to LSTMs



- A re-discovered way to implement recurrency in NNs
- Allow integrating inputs over longer time scales
  - Recognize and implement strategic behavior?
- Can neuroevolution take advantage of LSTMS as well?

Adapting to Opponent Strategies in Poker (1)



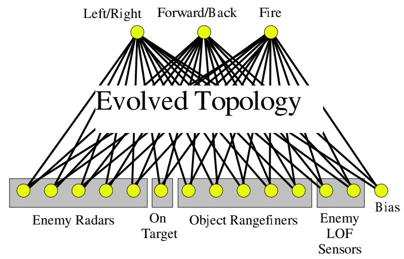
- Evolve weights of poker players[34]
  - 10-LSTM Game Module integrates over each game
  - A 1-LSTM Opponent Module integrates over each opponent
  - A fully connected Decision Network makes moves

Adapting to Opponent Strategies in Poker (2)

Opponent	Evolved LSTM	Slumbot
Scared Limper	999	792
Calling Machine	40368	2761
Hothead Maniac	36158	4988
Candid Statistician	9800	4512

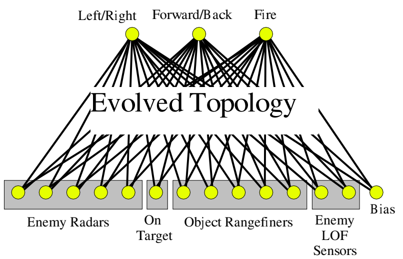
- Does not evolve a single strategy against all opponents
  - Changes the strategy according to games played
  - Better than Slumbot against these opponents (in mBB)
- Indeed LSTMs extend neuroevolution to strategic behavior

Combining Learning and Evolution



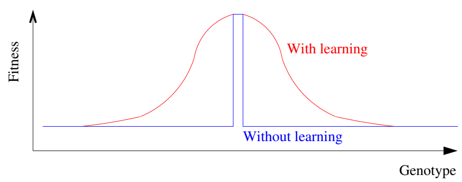
- Good learning algorithms exist for NN
  - Why not use them as well?
- Evolution provides structure and initial weights
- Fine tune the weights by learning

Lamarckian Evolution



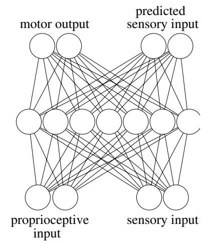
- Lamarckian evolution is possible[6,24]
  - Coding weight changes back to chromosome
- Difficult to make it work
  - Diversity reduced; progress stagnates

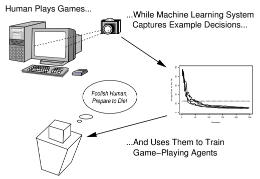
Baldwin Effect



- Learning can guide Darwinian evolution as well[4,24,25]
  - Makes fitness evaluations more accurate
- With learning, more likely to find the optimum if close
- Can select between good and bad individuals better
  - Lamarckian not necessary

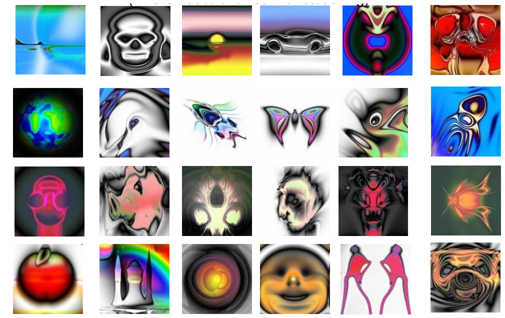
Where to Get Learning Targets?





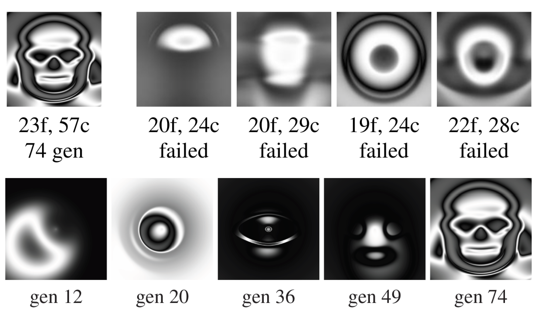
- From a related task[48]
  - Useful internal representations
- Evolve the targets[49]
  - Useful training situations
- From Q-learning equations[85]
  - When evolving a value function
- Utilize Hebbian learning[16,67,78]
  - Correlations of activity
- From the population[40,72]
  - Social learning
- From humans[6]
  - E.g. expert players, drivers

Evolving for Novelty



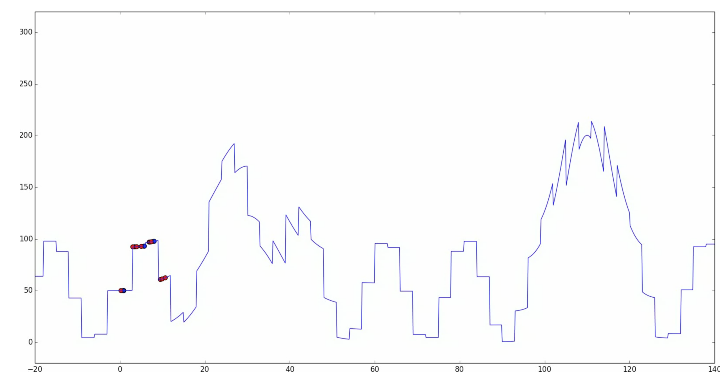
- Motivated by humans as fitness functions
- E.g. picbreeder.com, endlessforms.com[62]
  - CPPNs evolved; Human users select parents
- No specific goal
  - Interesting solutions preferred
  - Similar to biological evolution?

Novelty Search



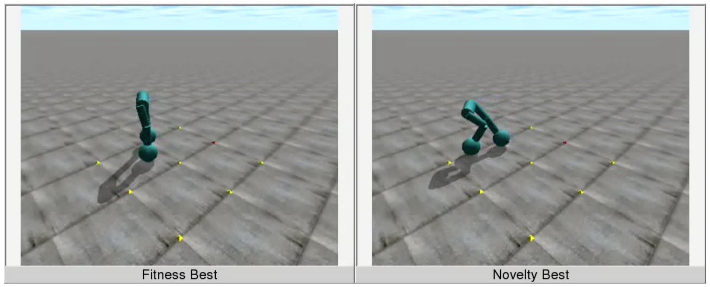
- Evolutionary algorithms maximize a performance objective
  - But sometimes hard to achieve it step-by-step
- Novelty search rewards candidates that are simply different[31,68]
  - Stepping stones for constructing complexity (Meyerson GECCO'17)[41,42]

Novelty Search Demo (1)



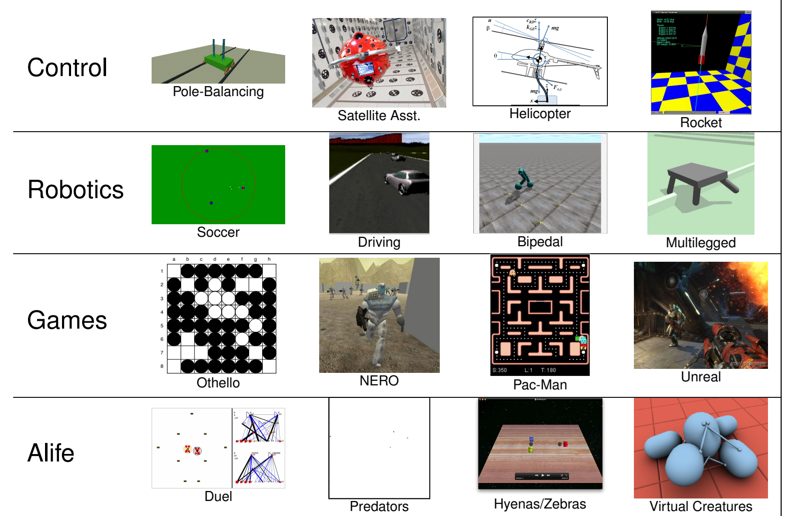
- 1D function to optimize; Fitness-based search would converge
- Novelty search finds stepping stones
- DEMO

Novelty Search Demo (2)

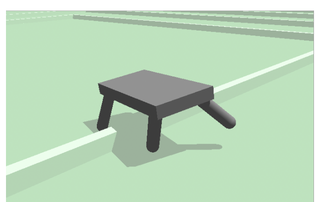


- Fitness-based evolution is rigid
  - Requires gradual progress
- Novelty-based evolution is more innovative, natural[31,68]
  - Allows building on stepping stones
  - As a secondary objective—or even the only one!
- DEMO

Neuroevolution Applications

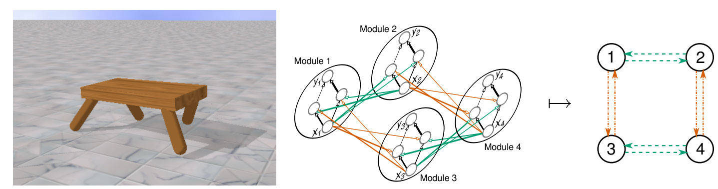


Robotics: Multilegged Walking



- Navigate rugged terrain better than wheeled robots
- Controller design is more challenging
  - Leg coordination, robustness, stability, fault-tolerance, ...
- Hand-design is generally difficult and brittle
- Large design space often makes evolution ineffective

ENSO: Symmetry Evolution Approach



- Symmetry evolution approach[77,79,80]
  - A neural network controls each leg
  - Connections between controllers evolved through symmetry breaking
  - Connections within individual controllers evolved through neuroevolution

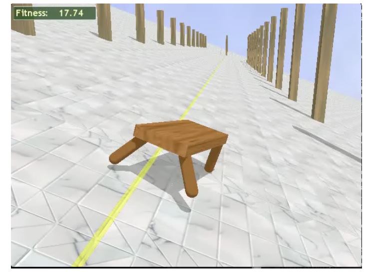
Versatile, Robust Gaits


Different gaits


Obstacle field

- Different gaits on flat ground
  - Pronk, pace, bound, trot
  - Changes gait to get over obstacles
- DEMO

Innovative, Effective Solutions

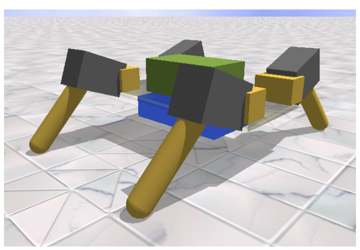


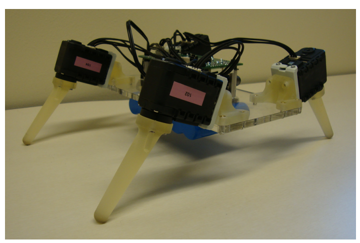
Evolved

Handcoded

- Asymmetric gait on inclines
  - One leg pushes up, others forward
  - Hard to design by hand
- DEMO

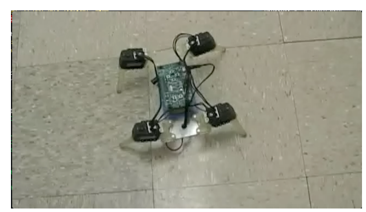
Transfer to a Physical Robot I

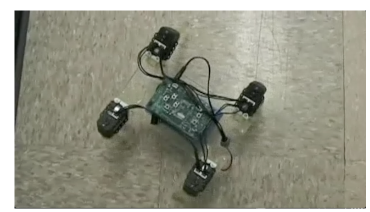

Simulated


Real

- Built at Hod Lipson's lab (Cornell U.)
  - Standard motors, battery, controller board
  - Custom 3D-printed legs, attachments
  - Simulation modified to match
- General, robust transfer[76]
  - Noise to actuators during simulation
  - Generalizes to different surfaces, motor speeds
- DEMO

Transfer to a Physical Robot II


Evolved


Handcoded

- Evolved a solution for three-legged walking!
- DEMO

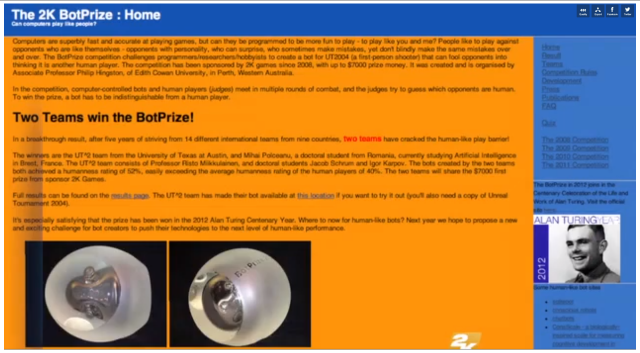
Games: Evolving Humanlike Behavior

- 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

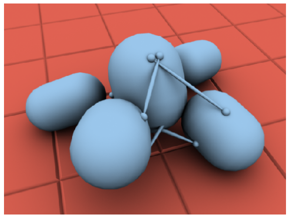
- Evolve effective fighting behavior
  - Human-like with resource limitations (speed, accuracy...)
- Also scripts & learning from humans (unstuck, wandering...)
- 2007-2011: bots 25-30% vs. humans 35-80% human
- 6/2012 best bot better than 50% of the humans
- 9/2012...?

Success!!!

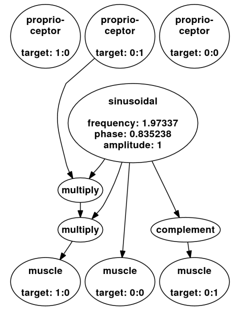


- In 2012, two teams reach the 50% mark!
- Fascinating challenges remain:
  - Judges can still differentiate in seconds
  - Judges lay cognitive, high-level traps
  - Team competition: collaboration as well
- DEMO

Alife: Evolved Virtual Creatures



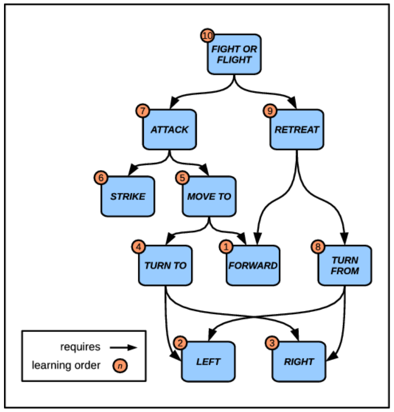
Body



Brain

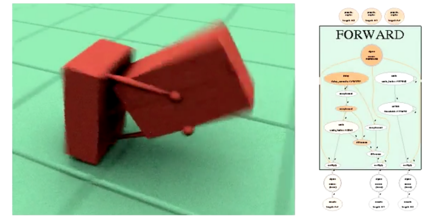
- Body-Brain Coevolution[32,33,65]
  - Body: Blocks, muscles, joints, sensors
  - Brain: A neural network (with general nodes)
  - Evolved together in a physical simulation
- Syllabus, Encapsulation, Pandemodium

Syllabus



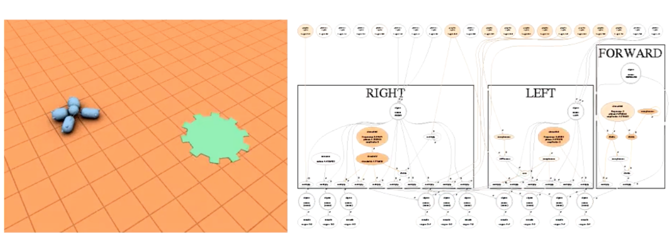
- Constructed by hand; body and brain evolved together

Encapsulation



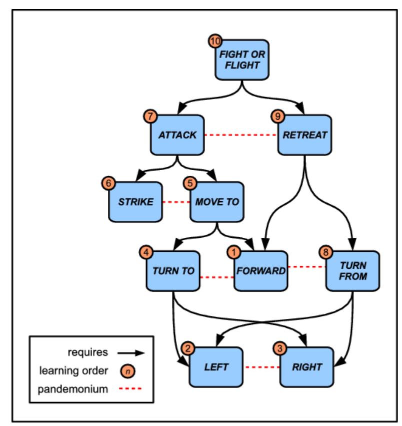
- Once evolved, a trigger node is added
- DEMO

Pandemonium



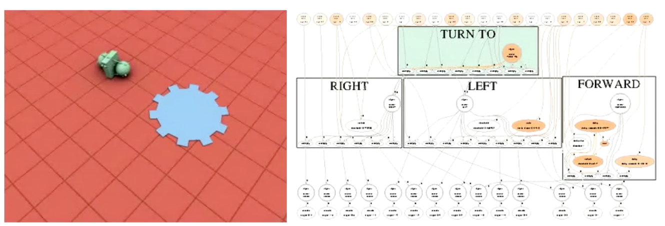
- Conflicting behaviors: Highest trigger wins
- DEMO

Evolving Fight-or-Flight Behavior



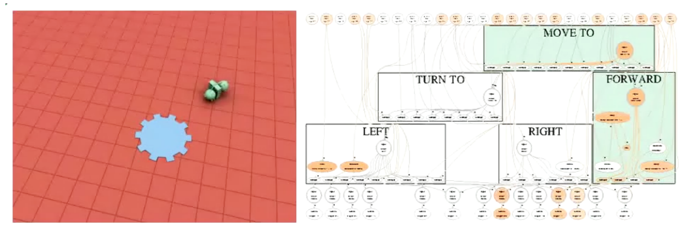
- Step-by-step construction of complex behavior
- Primitives and three levels of complexity
- DEMOS

Turn to Light



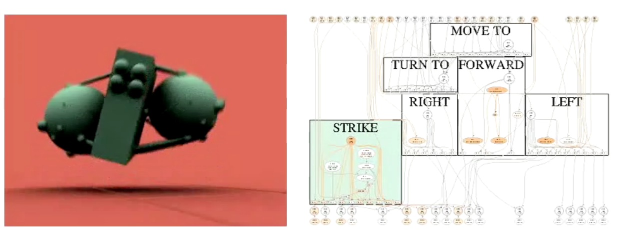
- First level of complexity
- Selecting between alternative primitives

Move to light



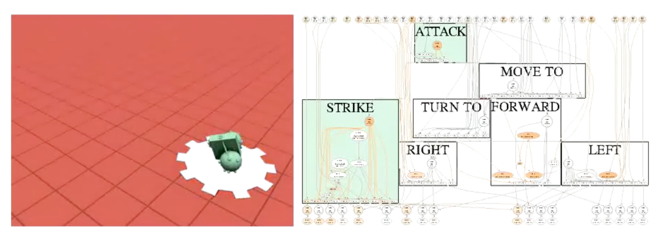
- First level of complexity (Sims 1994)
- Selecting between alternative primitives

Strike



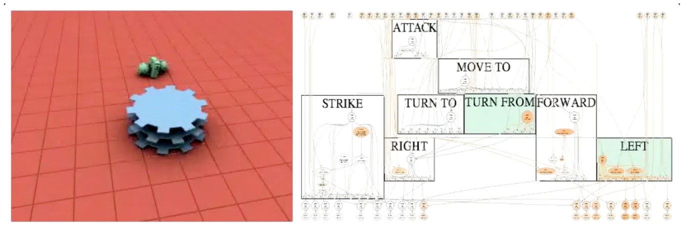
- Alternative behavior primitive

Attack



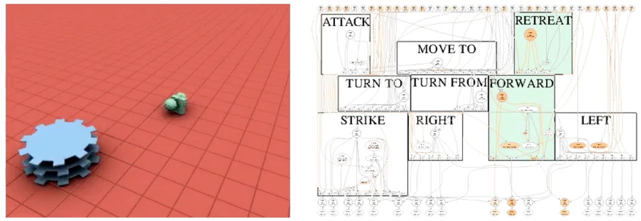
- Second level of complexity (beyond Sims and others)

Turn from Light



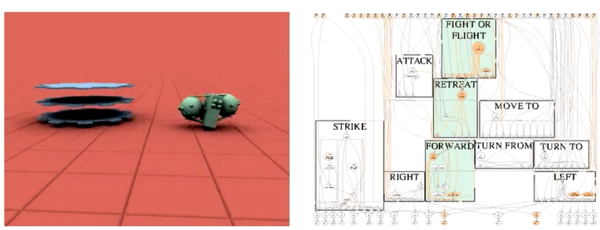
- Alternative first-level behavior

Retreat



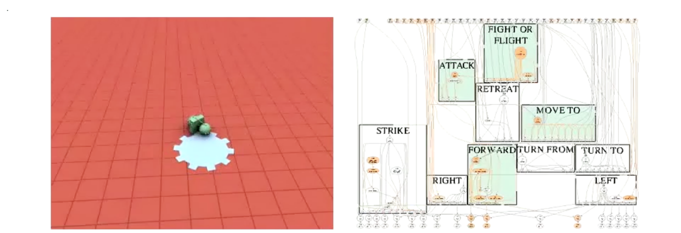
- Alternative second-level behavior

Fight or Flight



- Third level of complexity

Insight: Body/Brain Coevolution

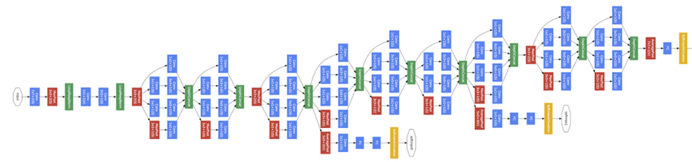


- Evolving body and brain together poses strong constraints
  - Behavior appears believable
  - Worked well also in BotPrize (Turing test for game bots)[60]
- Possible to construct innovative, situated behavior

Numerous Other Applications

- Creating art, music, dance...[8,12,27,63]
- Theorem proving[11]
- Time-series prediction[39]
- Computer system optimization[20]
- Manufacturing optimization[23]
- Process control optimization[81,82]
- Game strategy optimization[3]
- Measuring top quark mass[86]
- Etc.

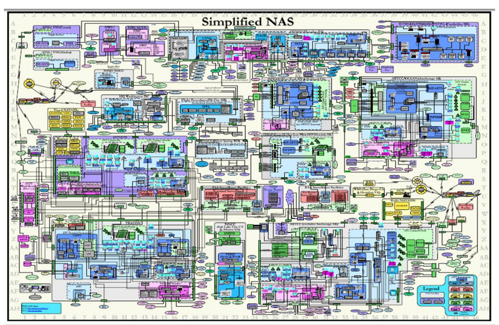
II. Optimization of DL Architectures



Szegedy et al. 2014

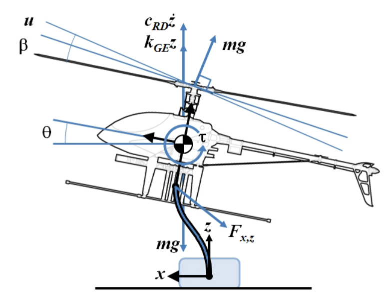
- Big Data and Big Compute available since 2000s
  - Machine learning systems have scaled up
- E.g. Deep Learning ideas existed since the 1990s
  - With million times more data & compute, they now work!
- A new problem: How to configure such systems?

Configuring Complex Systems



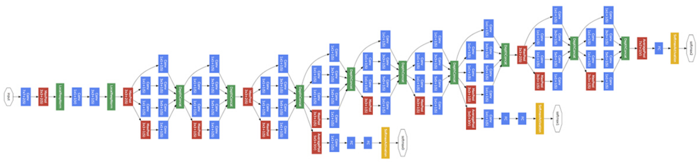
- A new general approach to engineering
  - Humans design just the framework
  - Machines optimize the details
- Programming by optimization[26]

E.g. Optimizing NE in Helicopter Hovering



- A challenging benchmark
  - RL, NE solutions exist
- Eight parameters optimized by hand[30]
  - Hard for a human designer to do more
- With EA, increased to 15
  - →Significantly better performance[35]

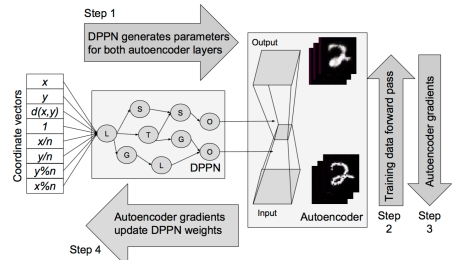
Evolving Deep Learning Architectures



Szegedy et al. 2014

- Different (complex) architectures for different tasks
  - Components matter—how to design them?
  - Architecture matters—how to compose it?
  - Hyperparameters matter—how to set them?
- Need to optimize architectures for each task

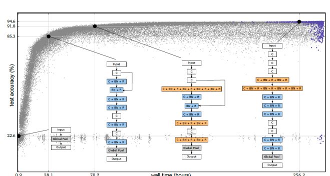
State of the Art in ENN/DL



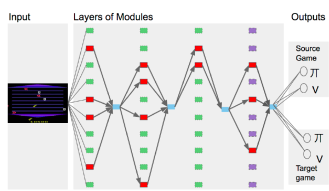
Fernando et al. 2016

- Partial optimization only, due to limited resources
  - Evolve DL hyperparameters[38]
  - Evolve a CPPN for weights; Lamarckian training[13]
  - Evolve weights with limited evaluation[47]
- Emerging area starting in 2016

State of the Art in ENN/DL (2017)



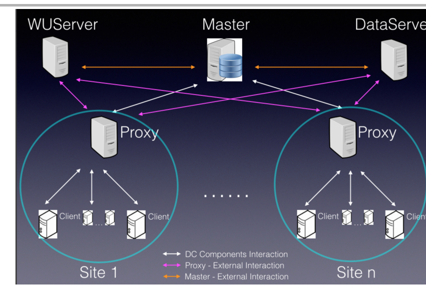
Esteban et al. 2017



Fernando et al. 2017

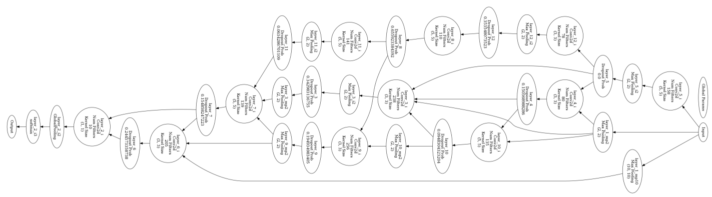
- PathNet (DeepMind)
  - Pathways across multiple supervised and RL tasks[14]
- Evolutionary Strategy (OpenAI)
  - Using ES instead of RL to construct networks for games[57]
- NEAT (Google Brain)
  - Evolution of deep networks on CIFAR-10 and CIFAR-100[53]

Computational Requirements



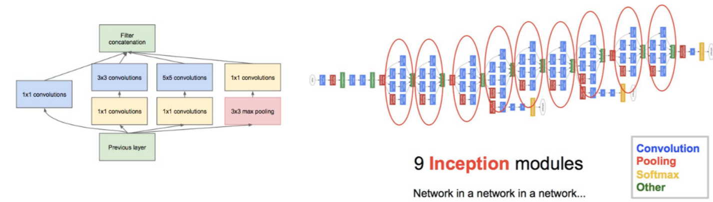
- Requires significant computational resources
  - Each DL network trains for 2 days on a GPU
- E.g. Sentient DarkCycle Distributed AI platform
  - Developed to harness idle cycles around the world
  - Includes 2M CPUs, 5K GPUs
  - In trading, 40 trillion candidates evaluated / year
  - Peak performance 9 Petaflops - #6 in the world

Initial Approach: NEAT



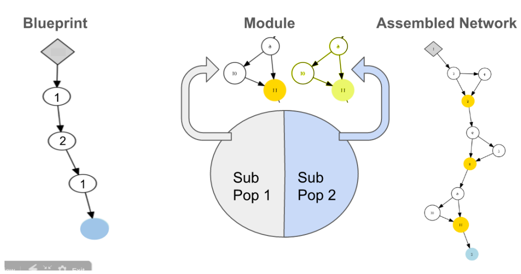
- Use NEAT to discover optimal network topology
  - Select components and connect them
- Also optimize hyperparameters
  - Sizes of layers, kernels, etc.
- Results in a complex network architectures
  - Tend to have less structure than best DL networks

Advanced Approach: Cooperative Coevolution



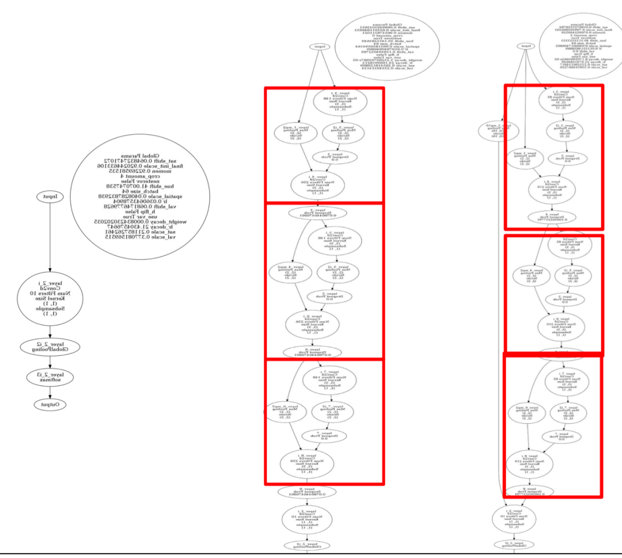
- Many of the best architectures are modular
  - E.g. Googlenet, residual networks...
  - Implements stepwise refinement?
- Does not emerge in NEAT by itself
- Solution: Evolve modules and blueprints
  - cf. ESP, bilevel evolution; Hierarchical SANE[46]

Cooperative Coevolution (2)

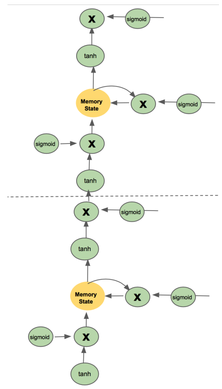


- Evolution at two levels
  - Module subpopulations optimize building blocks
  - Blueprint population optimizes their combinations
- Fitness of the complete network drives evolution
- Applies to both CNN (vision), LSTM (language) networks

Evaluation in CIFAR-10

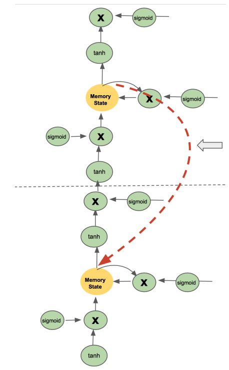


Evaluation in Language Modeling (1)



- Evolution of LSTM units with skip and gated connections
- At the blueprint level, combined into layers

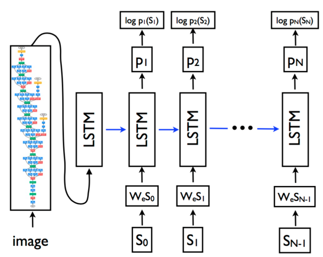
Evaluation in Language Modeling (2)



- Discovered a new LSTM unit with cell-to-cell connection
- In a 2-layer stacked LSTM, improves perplexity by 5%

Image Captioning Application

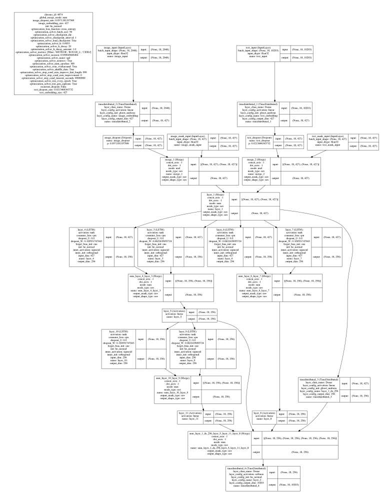
an airplane sitting in the middle of a city
an airplane on a bridge over a highway
a large jetliner taking off from an airport runway
very large jal jet taxiing on the runway at airport in urban environment
a jal airplane taxiing down a runway in a city



Vinyals et al. 2015

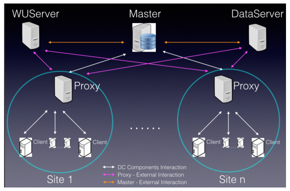
- Generating image captions for the blind
- Automatically on a magazine website
- Added 17,000 iconic image/caption pairs to MSCOCO
- Evolves elements from Show & Tell network[83]

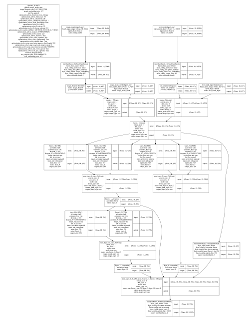
Evolved Image Captioning Network



- Complex network with repeated modules, a bypass pathway
- Improves 9% over Show and Tell baseline on MSCOCO
- Good on 50% of iconic, 20% of all images

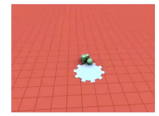
Future Work on ENN/DL

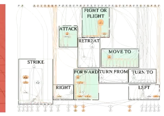


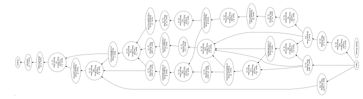


- Utilize HPC such as DarkCycle
- Extend the search space for DL
  - Evolve with more components: residuals, timing
- Utilize ensembles for LSTMs
  - Evolve diversity through novelty search
- A promising start on image captioning
- Automated design of DL for new applications

Conclusion







- Neuroevolution is a powerful approach for POMDPs
  - Discovers robust, believable behavior
  - Games, robotics, control, alife...
- Evolution makes more complex DL architectures possible
  - Structure, components, hyperparameters fit to the task
  - Vision, speech, language,...
  - Automatic design of learning machines

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)
- www.scholarpedia.org/article/Neuroevolution
  - A short summary of 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] 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] E. Bahceci and R. Miikkulainen, Evolving strategies for social innovation games, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2015)*, Madrid, Spain (July 2015).

[4] J. M. Baldwin, A new factor in evolution, *The American Naturalist*, 30:441–451, 536–553 (1896).

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

[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. [74], 81–90.

[8] C.-C. Chen and R. Miikkulainen, Creating melodies with evolving recurrent neural networks, in: *Proceedings of the INNS-IEEE International Joint Conference on Neural Networks*, 2241–2246, IEEE, Piscataway, NJ (2001).

[9] 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).

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

[11] N. S. Desai and R. Miikkulainen, Neuro-evolution and natural deduction, in: *Proceedings of The First IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, 64–69, IEEE, Piscataway, NJ (2000).

[12] G. Dubbin and K. O. Stanley, Learning to dance through interactive evolution, in: *Proceedings of the Eighth European Event on Evolutionary and Biologically Inspired Music, Sound, Art and Design*, Springer, Berlin (2010).

References II

[13] C. Fernando, D. Banarse, F. Besse, M. Jaderberg, D. Pfau, M. Reynolds, M. Lactot, and D. Wierstra, Convolution by evolution: Differentiable pattern producing networks, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2016)*, Denver, CO (2016).

[14] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Hay, A. A. Rusu, A. Pritzel, and D. Wierstra, Pathnet: Evolution channels gradient descent in super neural networks, *CoRR*, abs/1701.08734 (2017).

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

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

[17] 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).

[18] 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).

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

[20] F. Gomez, D. Burger, and R. Miikkulainen, A neuroevolution method for dynamic resource allocation on a chip multiprocessor, in: *Proceedings of the INNS-IEEE International Joint Conference on Neural Networks*, 2355–2361, IEEE, Piscataway, NJ (2001).

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

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

[23] B. Greer, H. Hakonen, R. Lahdelma, and R. Miikkulainen, Numerical optimization with neuroevolution, in: *Proceedings of the 2002 Congress on Evolutionary Computation*, 361–401, IEEE, Piscataway, NJ (2002).

References III

[24] 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).

[25] G. E. Hinton and S. J. Nowlan, How learning can guide evolution, *Complex Systems*, 1:495–502 (1987).

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

[27] A. K. Hoover, M. P. Rosario, and K. O. Stanley, Scaffolding for interactively evolving novel drum tracks for existing songs, in: *Proceedings of the Sixth European Workshop on Evolutionary and Biologically Inspired Music, Sound, Art and Design*, Springer, Berlin (2008).

[28] 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).

[29] 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).

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

[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).

[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. 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).

References IV

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

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

[38] I. Loshchilov and F. Hutter, CMA-ES for hyperparameter optimization of deep neural networks, *CoRR*, abs/1604.07269 (2016).

[39] J. R. McDonnell and D. Waagen, Evolving recurrent perceptrons for time-series modeling, *IEEE Transactions on Evolutionary Computation*, 5:24–38 (1994).

[40] 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.

[41] E. Meyerson, J. Lehman, and R. Miikkulainen, Learning behavior characterizations for novelty search, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2016)*, Denver, CO (2016).

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

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

[44] 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).

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

[46] D. E. Moriarty and R. Miikkulainen, Hierarchical evolution of neural networks, in: *Proceedings of the 1998 IEEE Conference on Evolutionary Computation (ICEC'98)*, 428–433, IEEE, Piscataway, NJ (1998).

[47] G. Morse and K. O. Stanley, Simple evolutionary optimization can rival stochastic gradient descent in neural networks, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2016)*, Denver, CO (2016).

References V

[48] S. Nolfi, J. L. Elman, and D. Parisi, Learning and evolution in neural networks, *Adaptive Behavior*, 2:5–28 (1994).

[49] 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).

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

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

[52] A. Rawal and R. Miikkulainen, Evolving deep lstm-based memory networks using an information maximization objective, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2016)*, Denver, CO (2016).

[53] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, Q. Le, and A. Kurakin, Large-scale evolution of image classifiers, *CoRR*, abs/1703.01041 (2017).

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

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

[56] 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).

[57] T. Salimans, J. Ho, X. Chen, and I. Sutskever, Evolution strategies as a scalable alternative to reinforcement learning, *CoRR*, abs/1703.03864 (2017).

[58] 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).

[59] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Networks*, 61:85–117 (2015).

References VI

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

[61] 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).

[62] 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).

[63] J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, and K. O. Stanley, Picbreeder: Evolving pictures collaboratively online, in: *Proceedings of Computer Human Interaction Conference*, ACM, New York (2008).

[64] 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).

[65] 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).

[66] 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).

[67] 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).

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

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

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

[71] 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).

References VII

[72] 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).

[73] M. Taylor, S. Whiteson, and P. Stone, Comparing evolutionary and temporal difference methods in a reinforcement learning domain, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2006).

[74] 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).

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

[76] V. Valsalam, J. Hiller, R. MacCurdy, H. Lipson, and R. Miikkulainen, Constructing controllers for physical multilegged robots using the enso neuroevolution approach, *Evolutionary Intelligence*, 14:303–331 (2013).

[77] V. Valsalam and R. Miikkulainen, Evolving symmetry for modular system design, *IEEE Transactions on Evolutionary Computation*, 15:368–386 (2011).

[78] V. K. Valsalam, J. A. Bednar, and R. Miikkulainen, Constructing good learners using evolved pattern generators, in: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2005*, H.-G. Beyer et al., eds., 11–18, New York: ACM (2005).

[79] V. K. Valsalam and R. Miikkulainen, Modular neuroevolution for multilegged locomotion, in: *Proceedings of the Genetic and Evolutionary Computation Conference GECCO 2008*, 265–272, ACM, New York, NY, USA (2008).

[80] 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).

[81] A. van Eck Conradie, R. Miikkulainen, and C. Aldrich, Adaptive control utilising neural swarming, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, W. B. Langdon, E. Cantú-Paz, K. E. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. K. Burke, and N. Jonoska, eds., San Francisco: Morgan Kaufmann (2002).

[82] A. van Eck Conradie, R. Miikkulainen, and C. Aldrich, Intelligent process control utilizing symbiotic memetic neuro-evolution, in: *Proceedings of the 2002 Congress on Evolutionary Computation* (2002).

References VIII

[83] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, Show and tell: A neural image caption generator, in: *Proc. of CVPR*, 3156–3164 (2015).

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

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

[86] S. Whiteson and D. Whiteson, Stochastic optimization for collision selection in high energy physics, in: *Proceedings of the Nineteenth Annual Innovative Applications of Artificial Intelligence Conference* (2007).

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

[88] A. P. Wieland, Evolving controls for unstable systems, in: Touretzky et al.[74], 91–102.

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