Evolution of Neural Networks

CSCE 644

Texas A&M CSE

Yoonsuck Choe

Part I&II largely based on Risto Miikkulainen's tutorial at the GECCO 2005.`http://www.cs.utexas.edu/users/risto`. Part III based on Ruppin [7] and Floreano et al. [2].

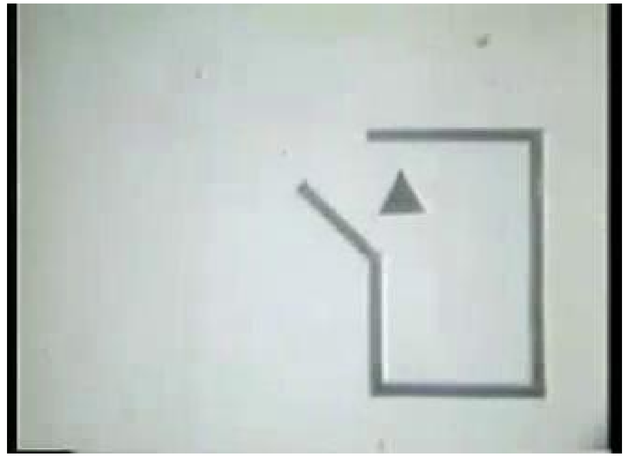
Why Do We Have the Brain?

- Survival and reproduction? Think again!

Tree (no Brain)

http://homepages.inf.ed.ac.uk/jbednar/

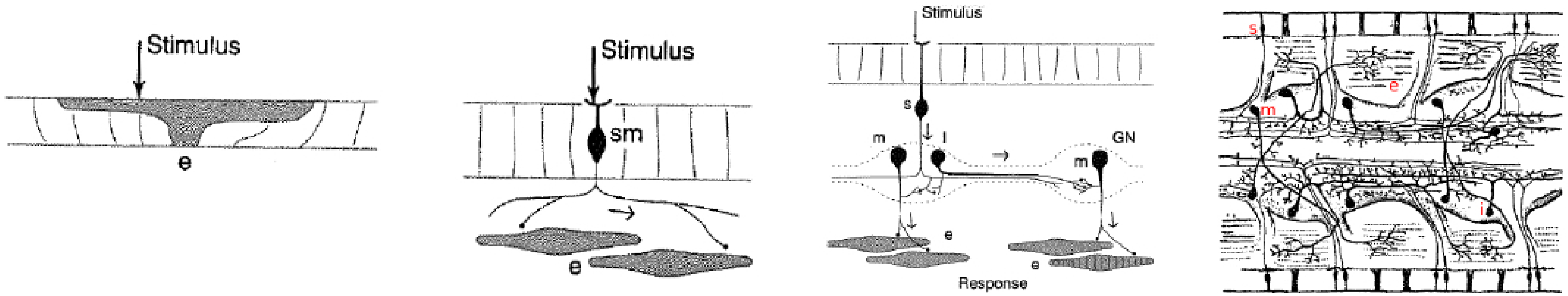
Brain: Source of Complex Behavior?



Heider and Simmel [3]

- Does this look realistic?
- Does it look intelligent?
- Would it need complex nervous system?

Evolution of Complex Behavior



(a) Single e (b) sm $\rightarrow$ e (c) s $\rightarrow$ m $\rightarrow$ e (d) Flatworm nerve net

Swanson [11]

- Evolution of the brain = evolution of the circuit, more so than the circuit elements.
- e: effector ($\sim$muscle), sm: sensorimotor, s: sensory, m: motor

Outline

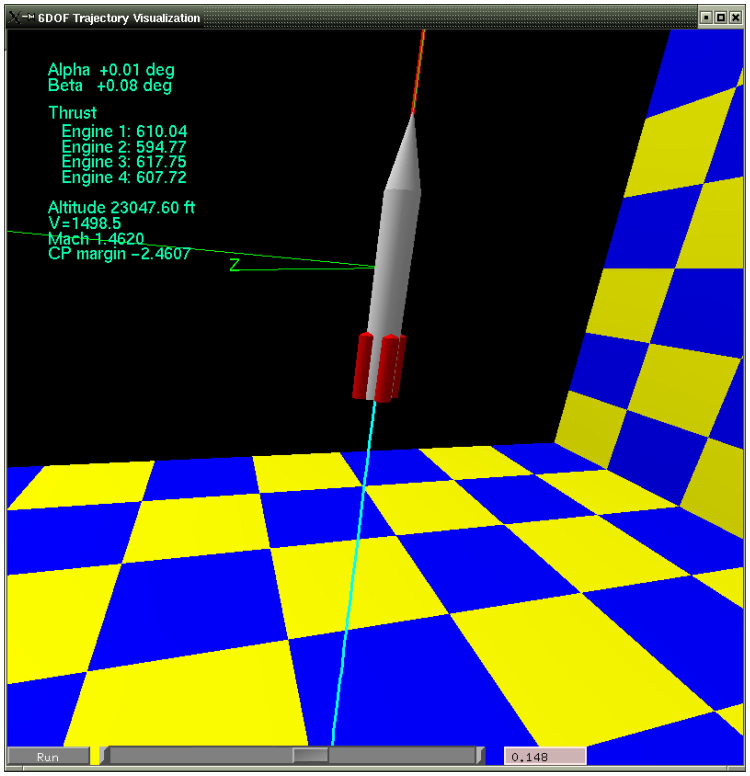
- Introduction to neuroevolution
- Evolving complex behavior through complexification and co-evolution (Stanley, Miikkulainen)
- Neuroevolution for neuroscience research
- Discussion

I. Intro to Neuroevolution

Neuroevolution of Complex Behavior

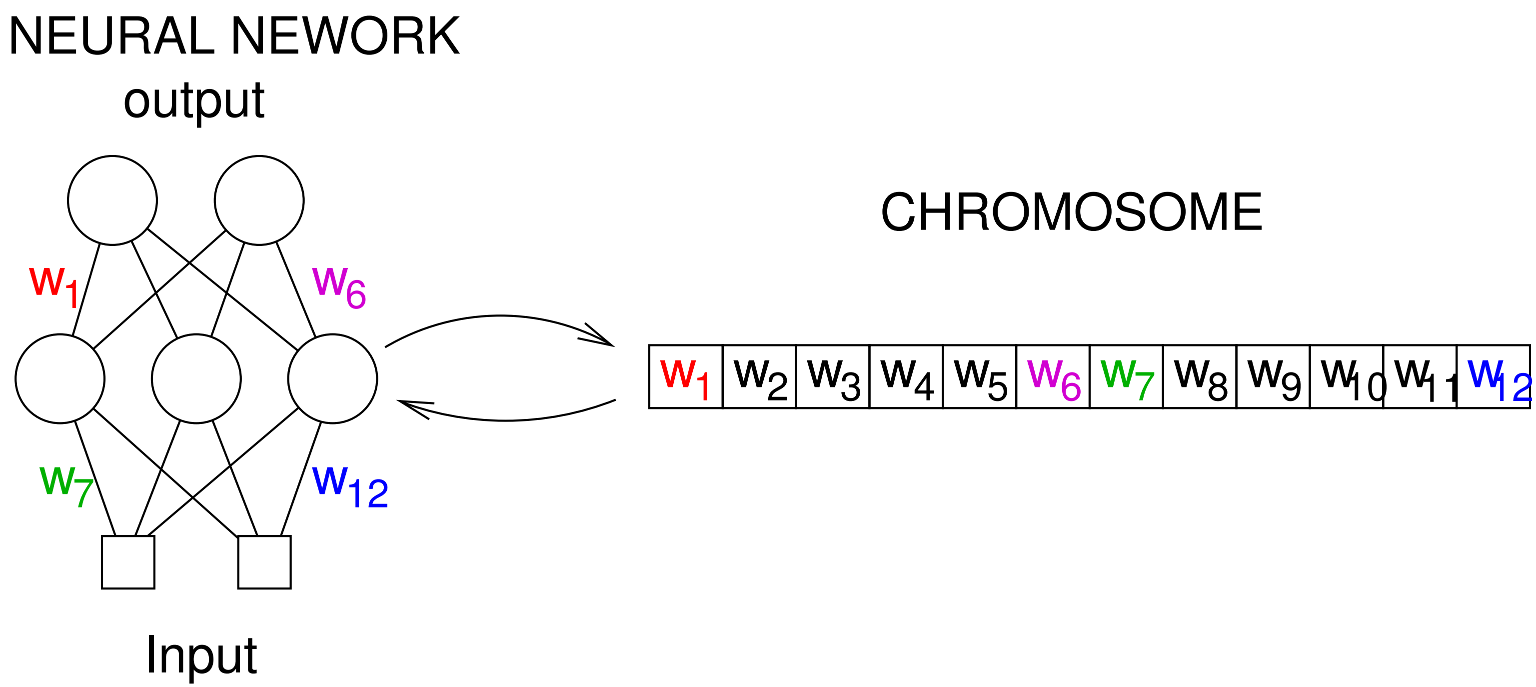
- Neuroevolution: Evolving artificial neural networks to control behavior of robots and agents.
- Main idea: Mimic the natural process of evolution that gave rise to the brain, the source of intelligence.
  - Population
  - Competition
  - Selection
  - Reproduction and mutation

Why Neuroevolution?



- Neural networks are effective but with limitations.
- Can solve tough, complex problems: fin-less rockets, robotic agents.

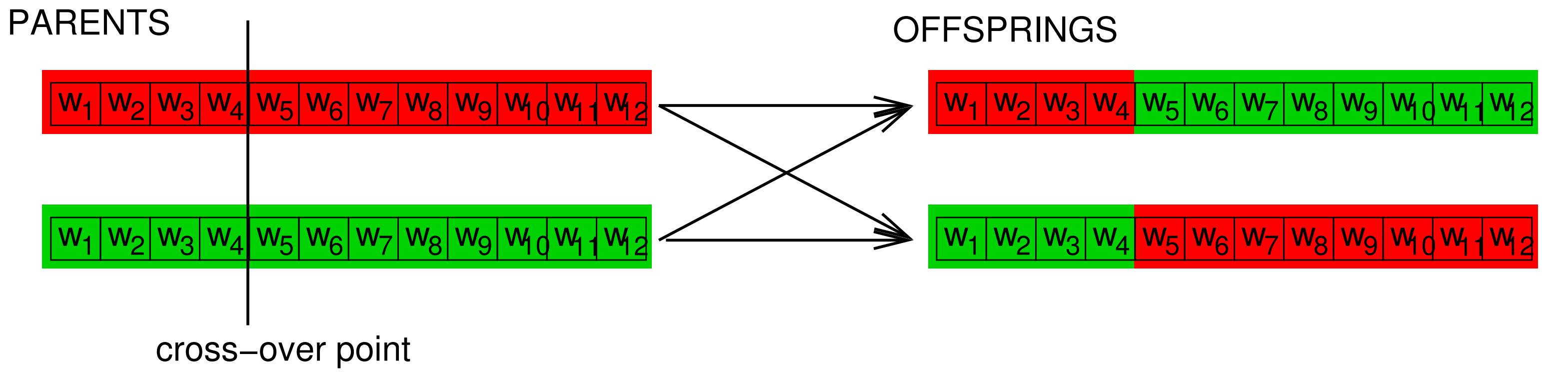
Neuroevolution Basics



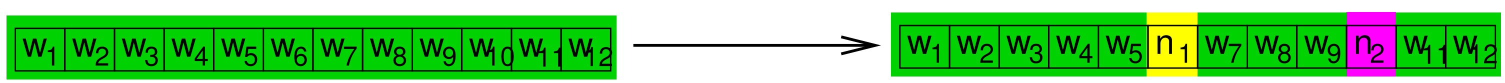
- A single chromosome encodes a full neural network.
  - Inputs hooked up to sensors, and outputs to actuators.
- Each gene, a single bit (or a real number), maps to a connection weight in the neural network.

Neuroevolution Basics: Operators

CROSS–OVER

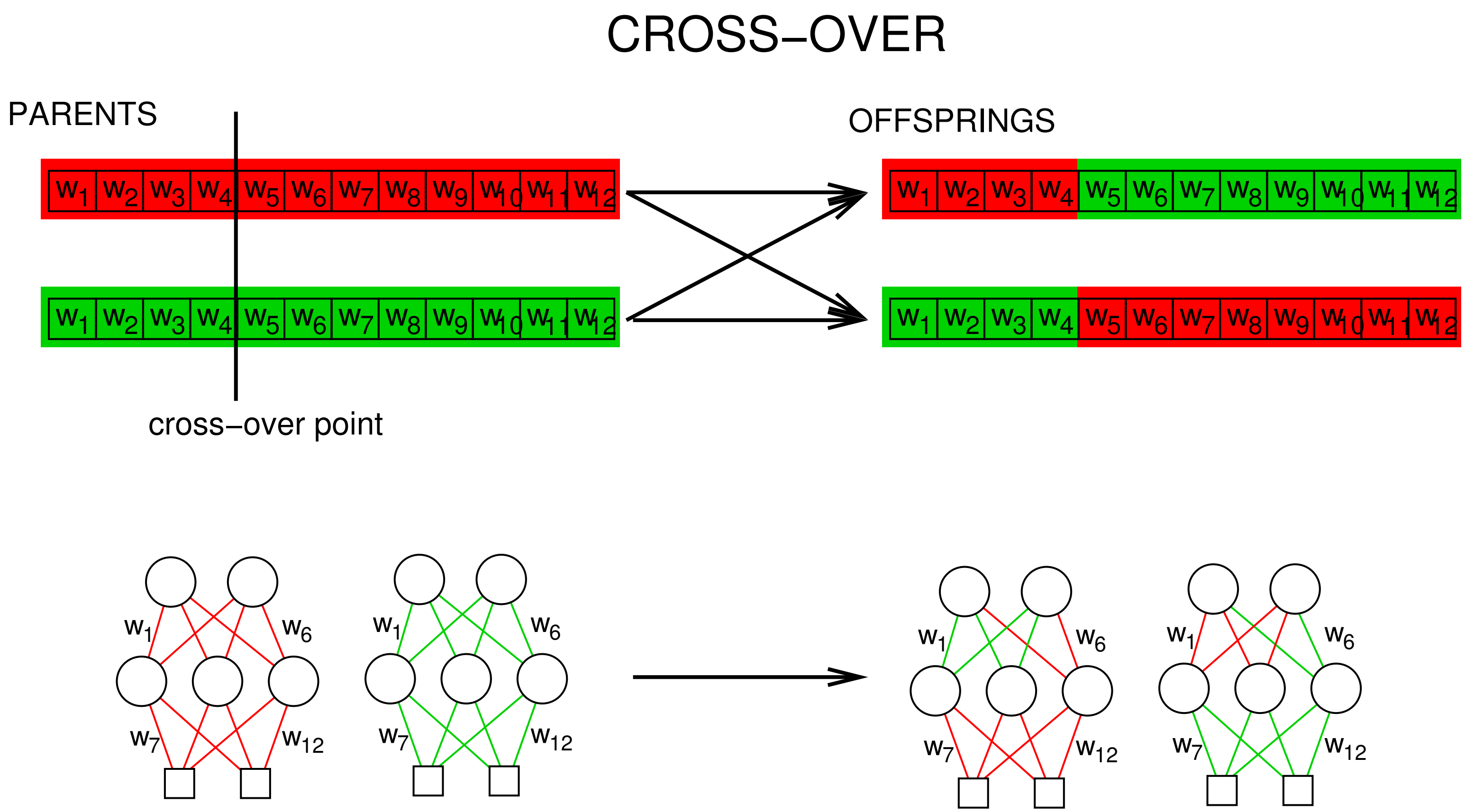


MUTATION



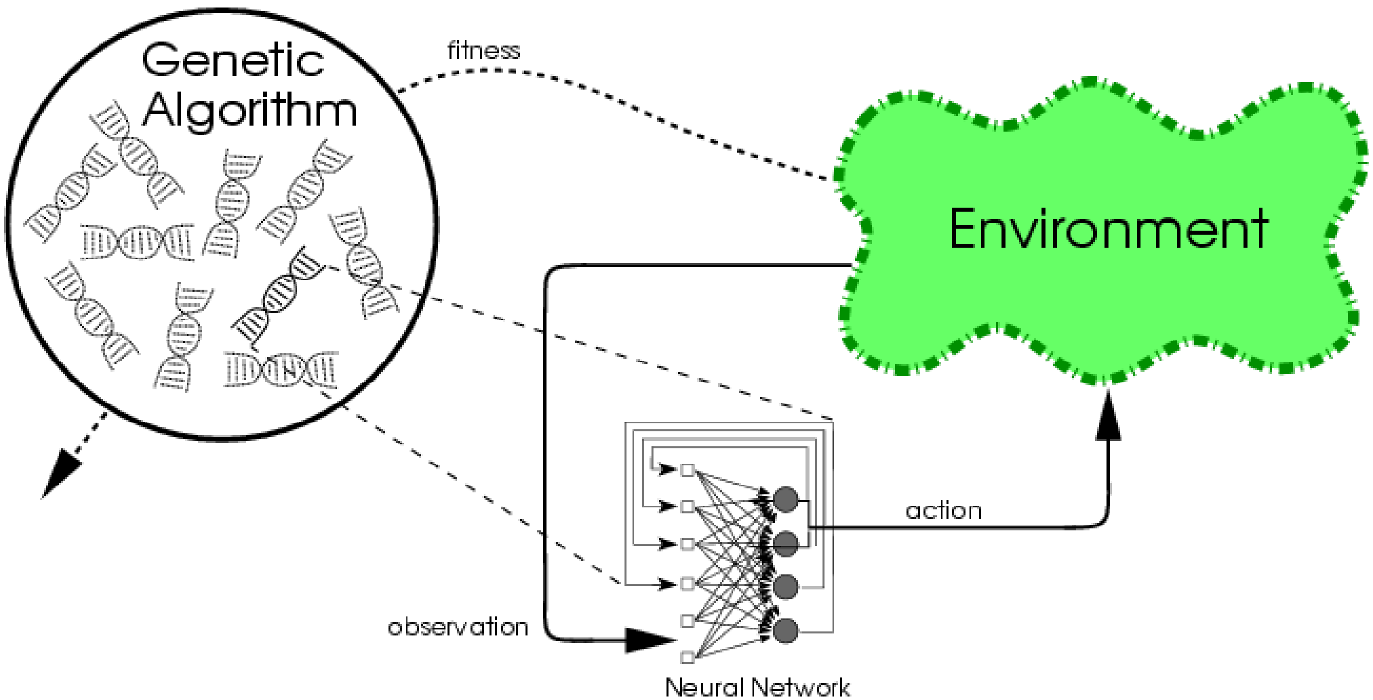
- Cross-over: Combine traits from both parents.
- Mutation: Introduce randomness (innovation).

Neuroevolution Basics: Cross-Over in Detail



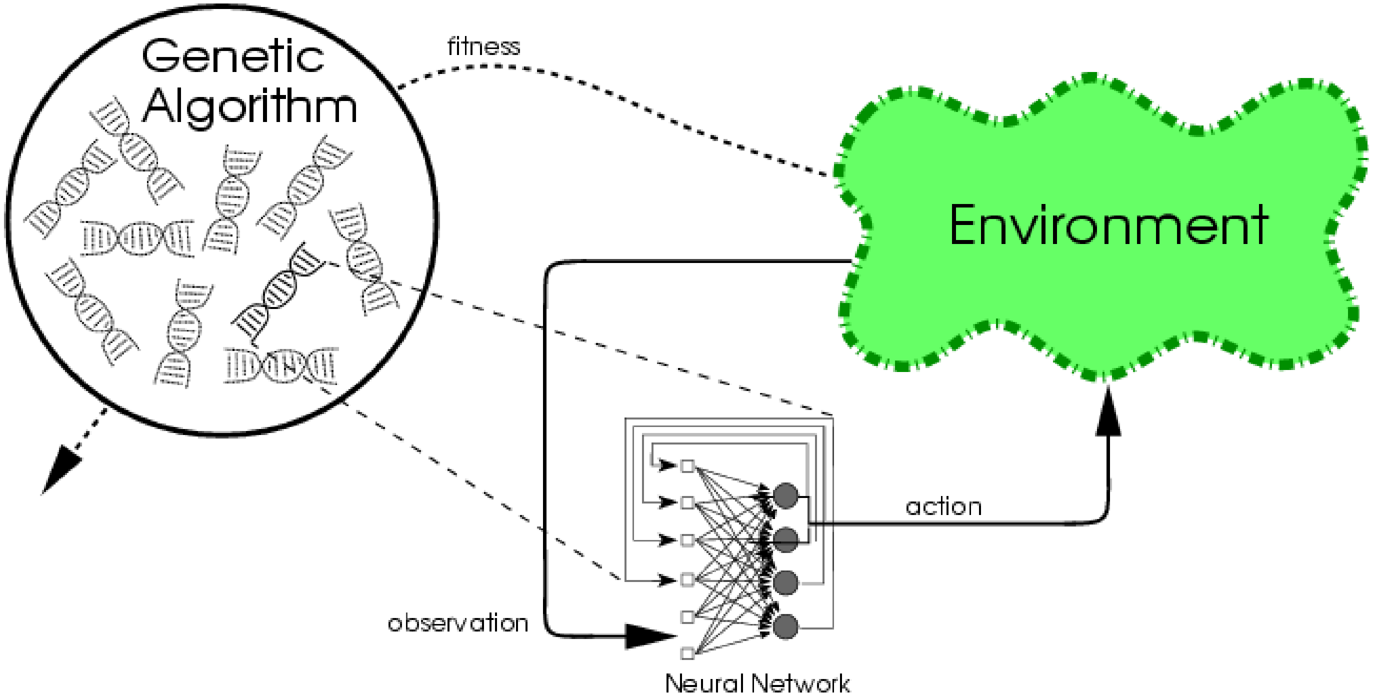
- Cross-over of two individuals produces two offsprings with a mixed heritage.

Conventional Neuroevolution (2)



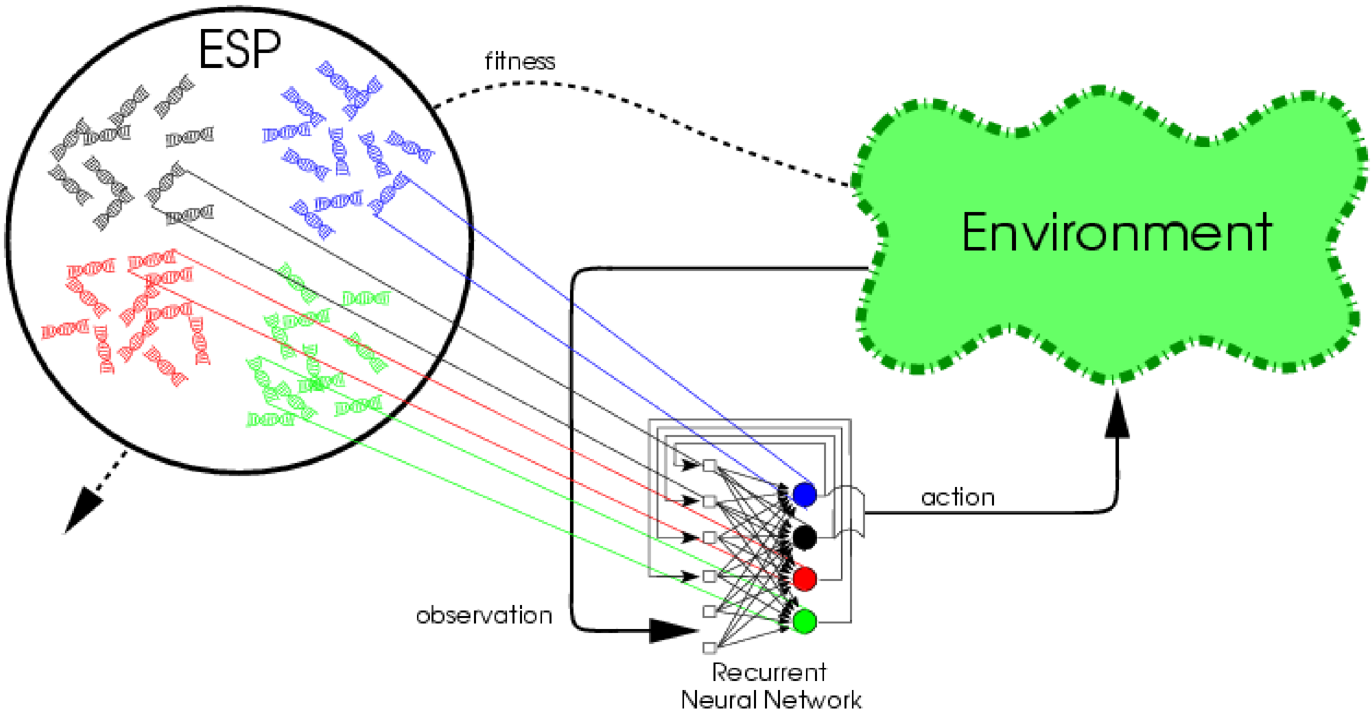
1. Fitness Evaluation: Construct NN with chromosome, put in the environment, observe outcome.
2. Selection: Choose best ones.
3. Reproduction: Mate the best ones and put back in the population.

Problems with CNE



- Evolution tends to converge to a small homogeneous population
  - 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

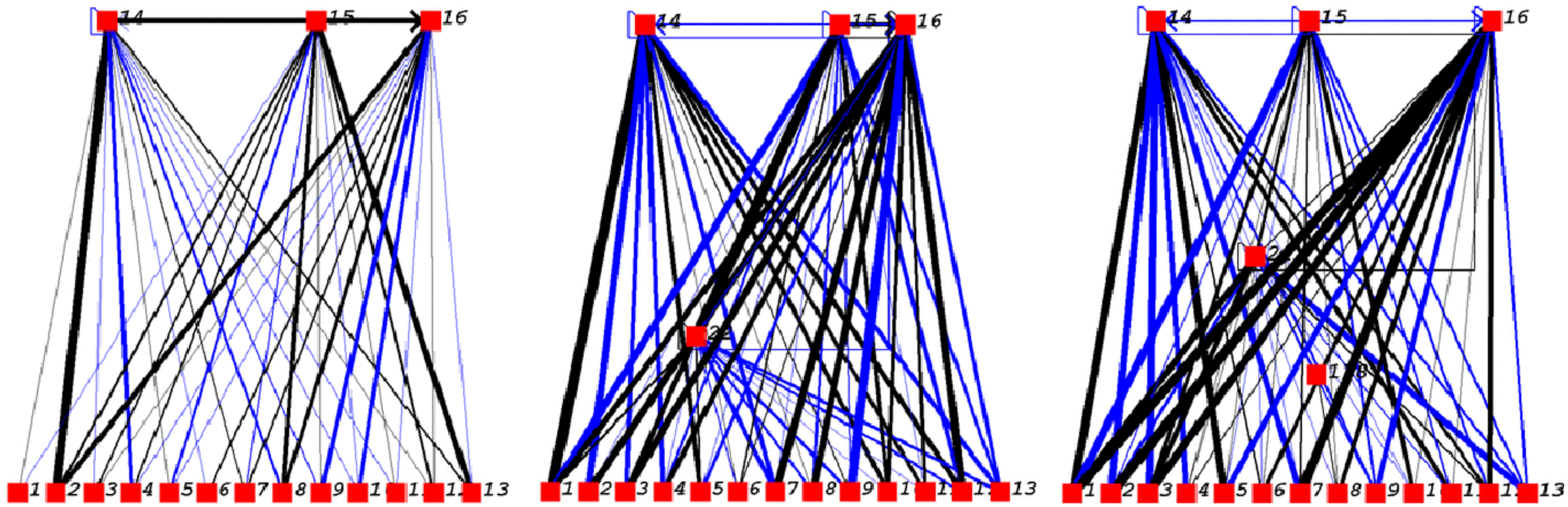
Advanced Neuroevol.: Evolving Neurons



- Evolving individual neurons: Chromosome = neuron. [1,5,6]
- Construct network with neurons, evaluate, reproduce, and repeat.
  – Network has fixed topology.
- Fitness of network determines that of participating neurons.
- Shown to improve diversity.

II. Evolving Complex Behavior: Co-Evolution & Topology Evolution[8,9]

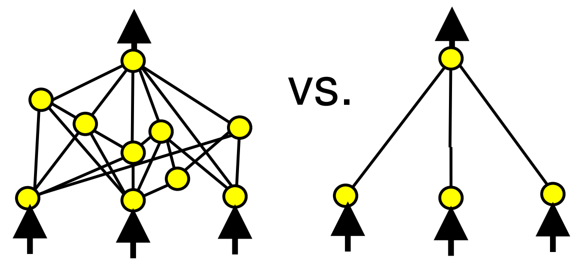
Evolving Topologies



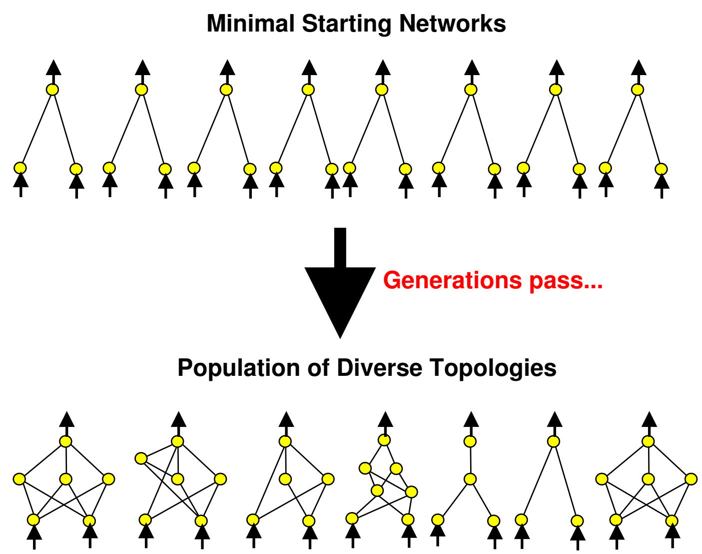
- Fixed topology has limitations.
- Idea: Evolve network topology, as well as connection weight.
- Neuroevolution of Augmenting Topologies (NEAT[8,9])
- Based on *Complexification*:
  - Network topology
  - Behavior

How Can We Complexify?

- Can optimize not just weights but also topologies



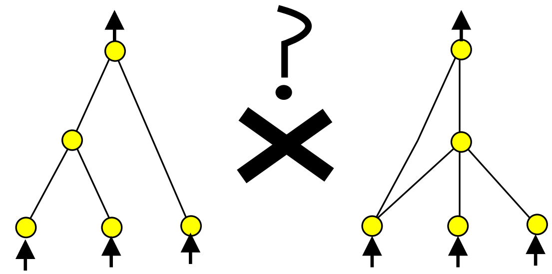
- Solution: Start with minimal structure and complexify [12]



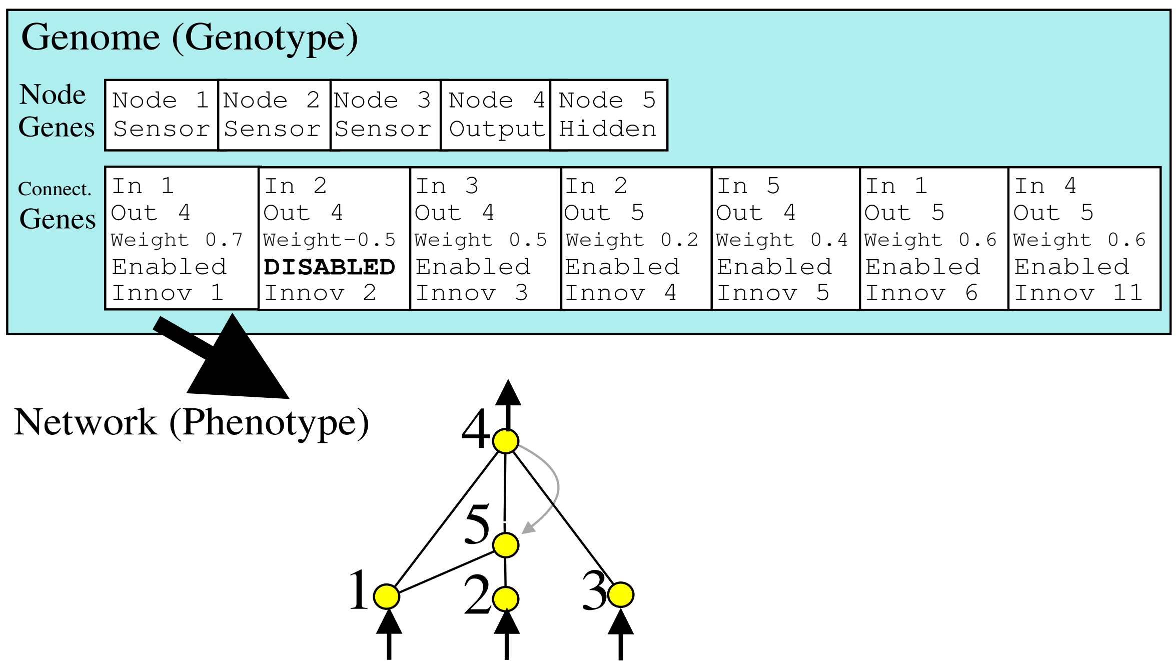
- Can search a very large space of configurations!

How Can Crossover be Implemented?

- Problem: Structures do not match

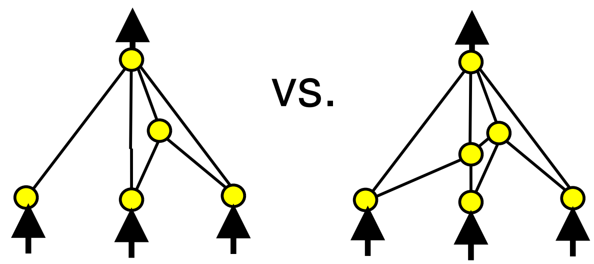


- Solution: Utilize historical markings



How can Innovation Survive?

- Problem: Innovations have initially low fitness

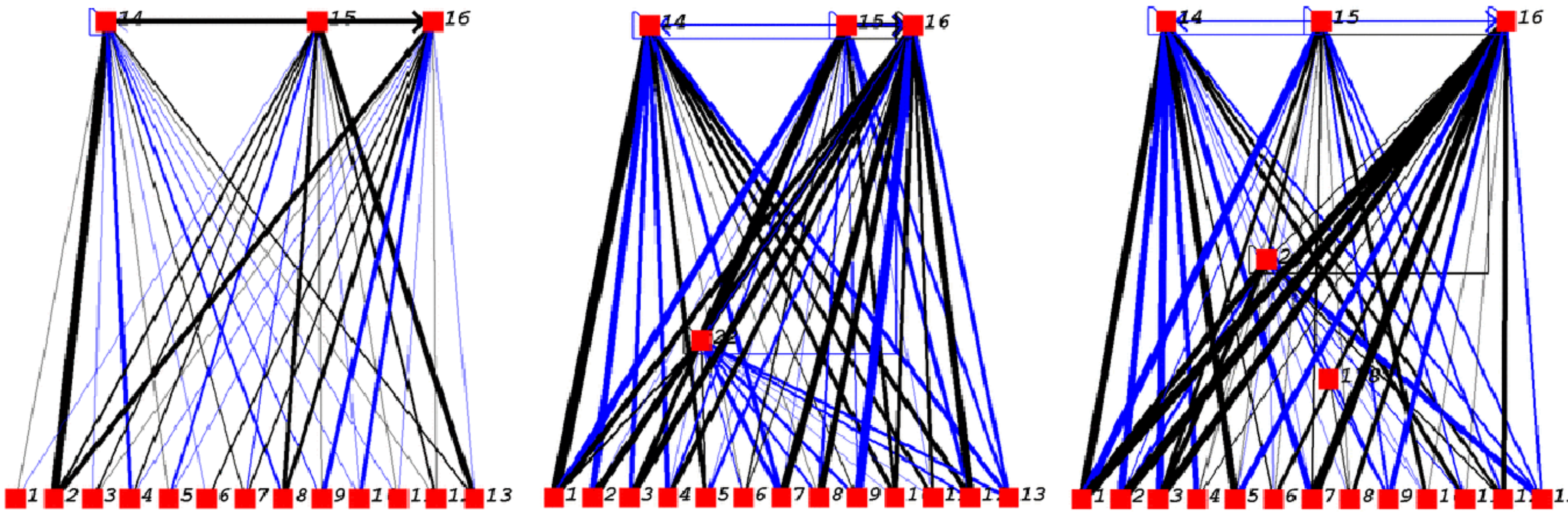


- Solution: Speciate the population
  - Innovations have time to optimize
  - Mitigates competing conventions
  - Promotes diversity

Competitive Coevolution

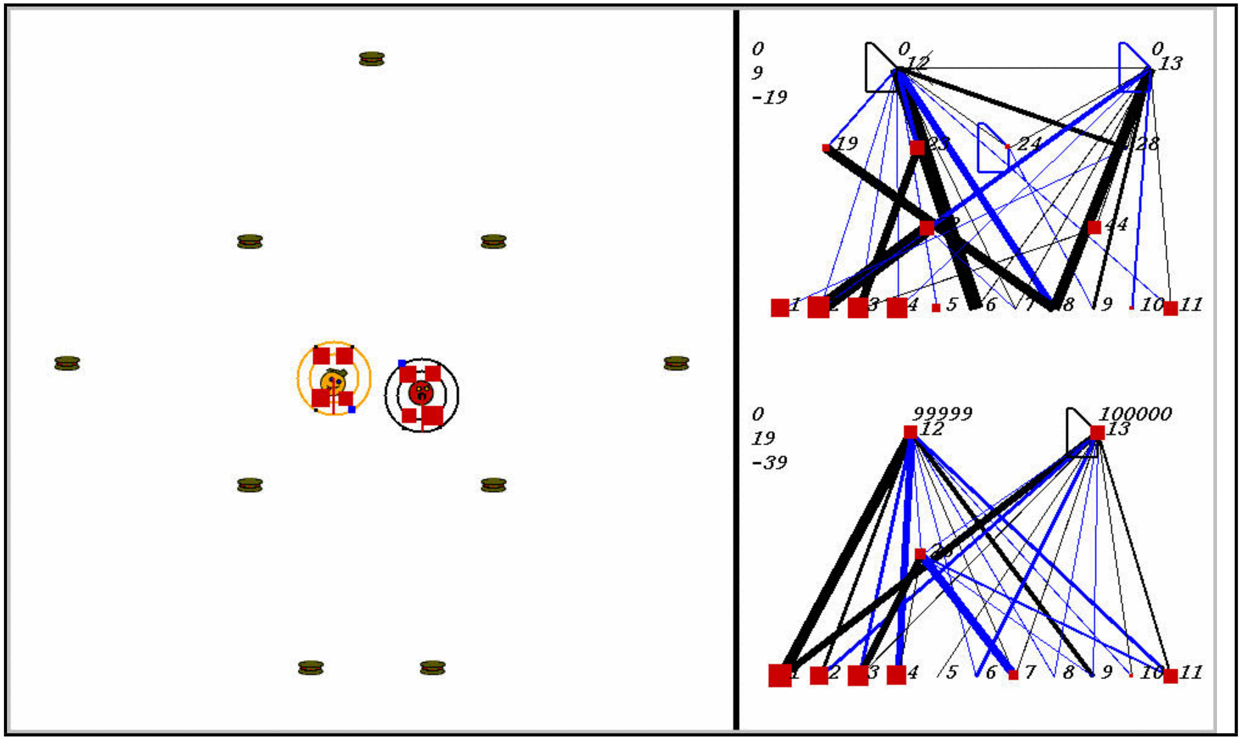
- Progress in evolution is based on competition.
- Better solutions emerge when given tougher opponents.
- Tough opponents do not exist from the beginning.
- Co-evolution solves this problem.
  - Start out with naive populations.
  - Make populations compete with each other.
  - Coevolutionary arms race (poison toxicity vs. tolerance).

Competitive Coevolution with NEAT



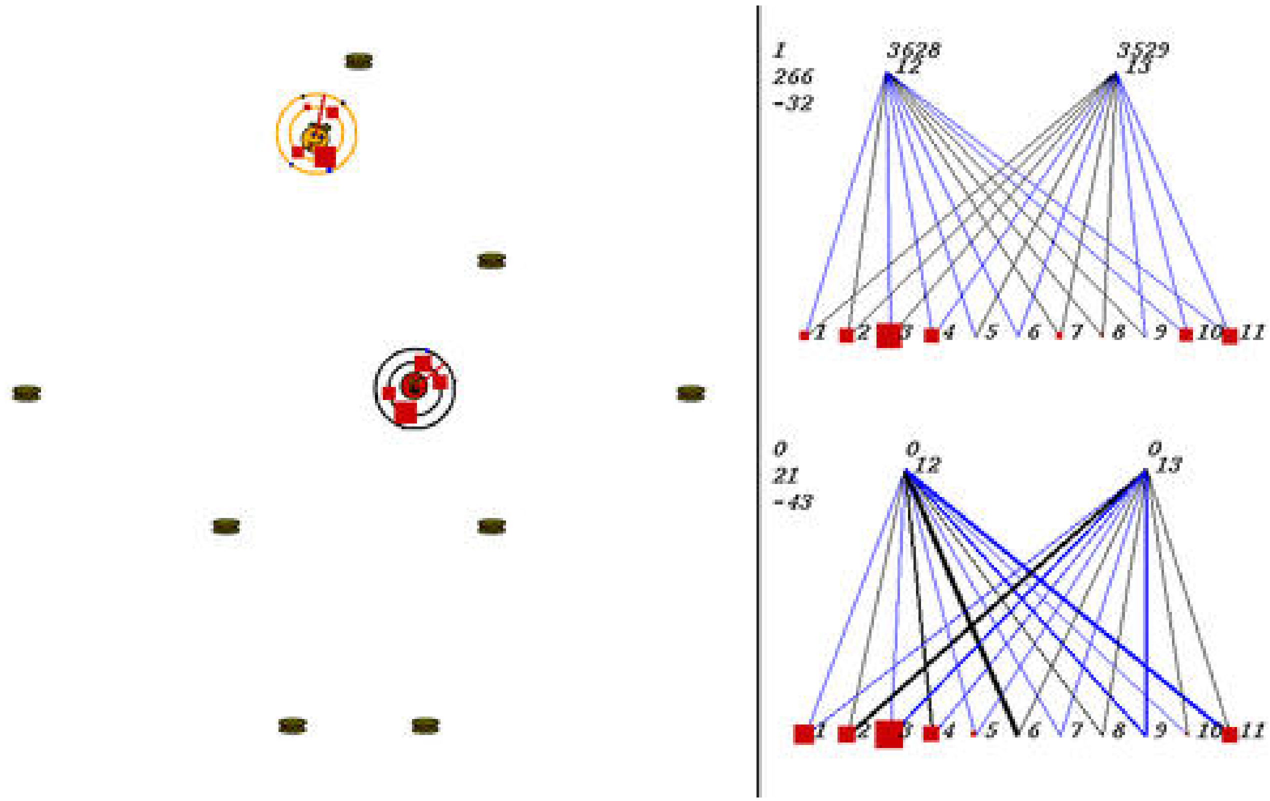
- Complexification elaborates on the solution
  - Adding more complexity to existing behaviors
- Can establish a coevolutionary arms race
  - Two populations continually outdo each other
  - Absolute progress, not just tricks

Coevolution Demo (by Ken Stanley)



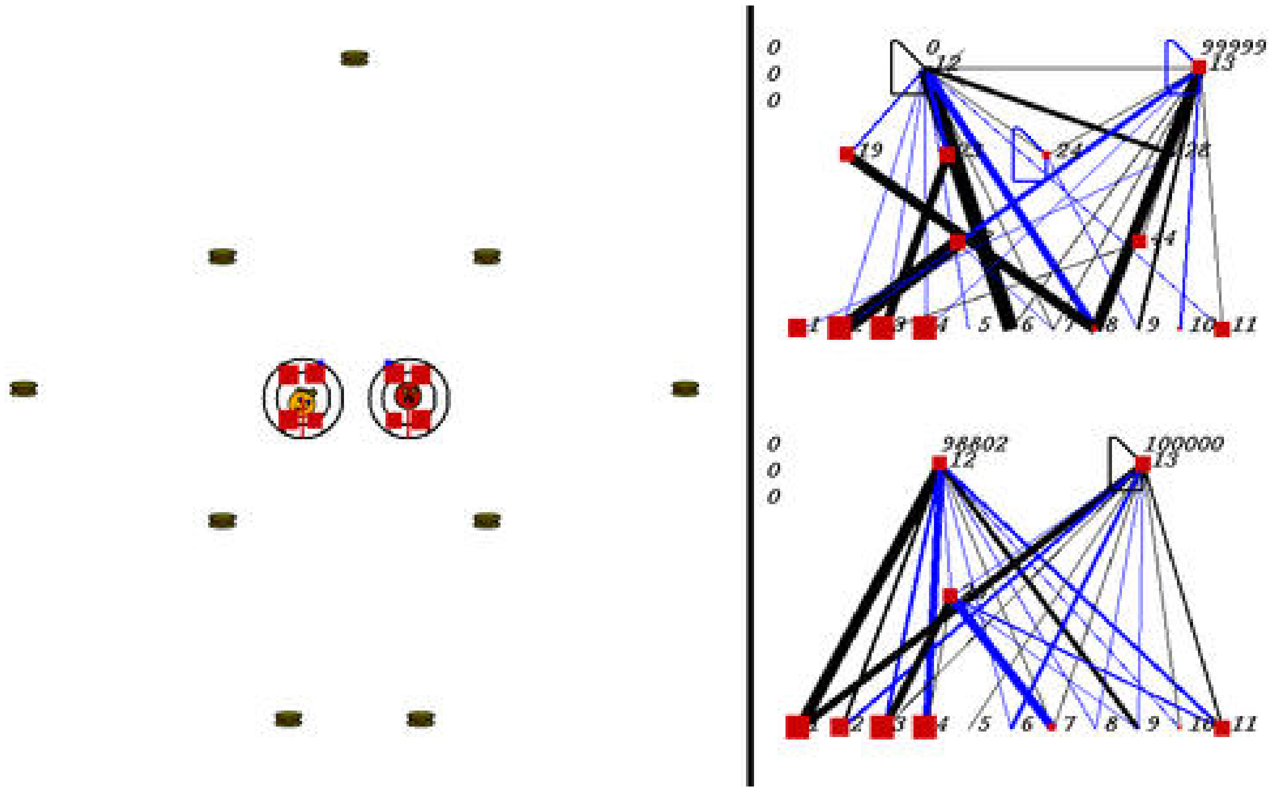
- Two robots pitted against each other[10]
  - Food sensor, Enemy sensor, Energy difference sensor, Wall sensor
  - Eat food to incr. Energy, Moving around decr. energy.

Early Poor Strategy



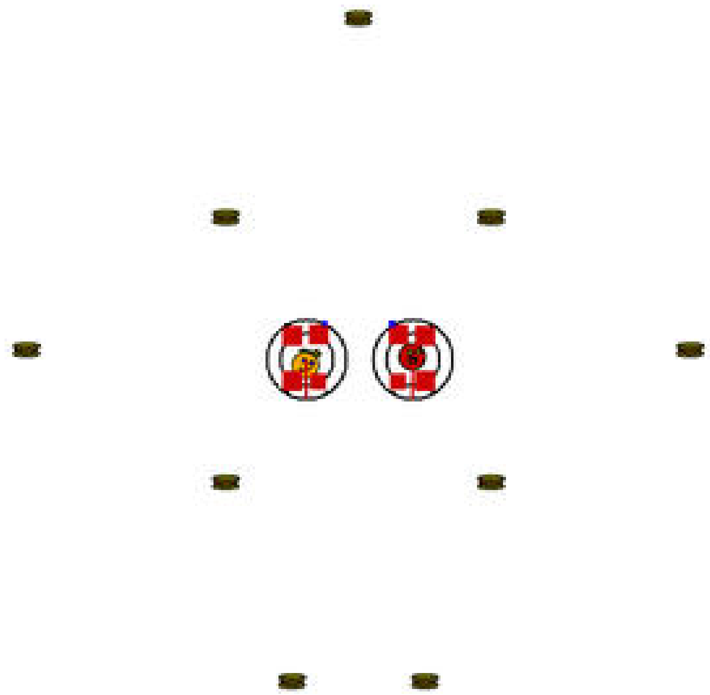
- Generation 1 and 3 champs.
- Very goal-directed: eat food, attack opponent

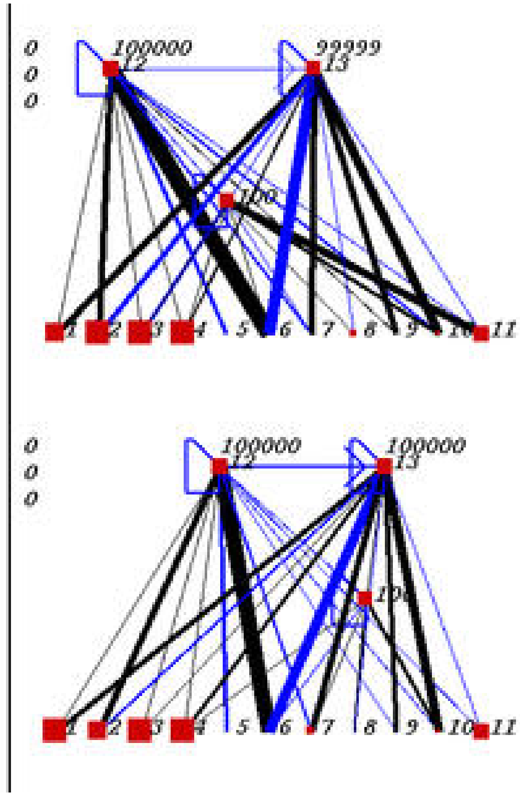
Later Poor Strategy



- Champs from two different population in gen 40.
- No food consumption (poor strategy).
- Waste energy while idly moving (teasing?).

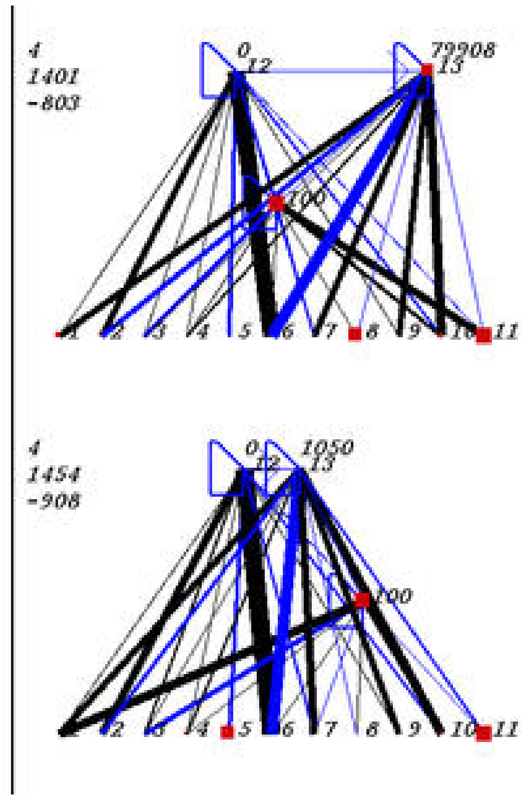
First Successful Strategy





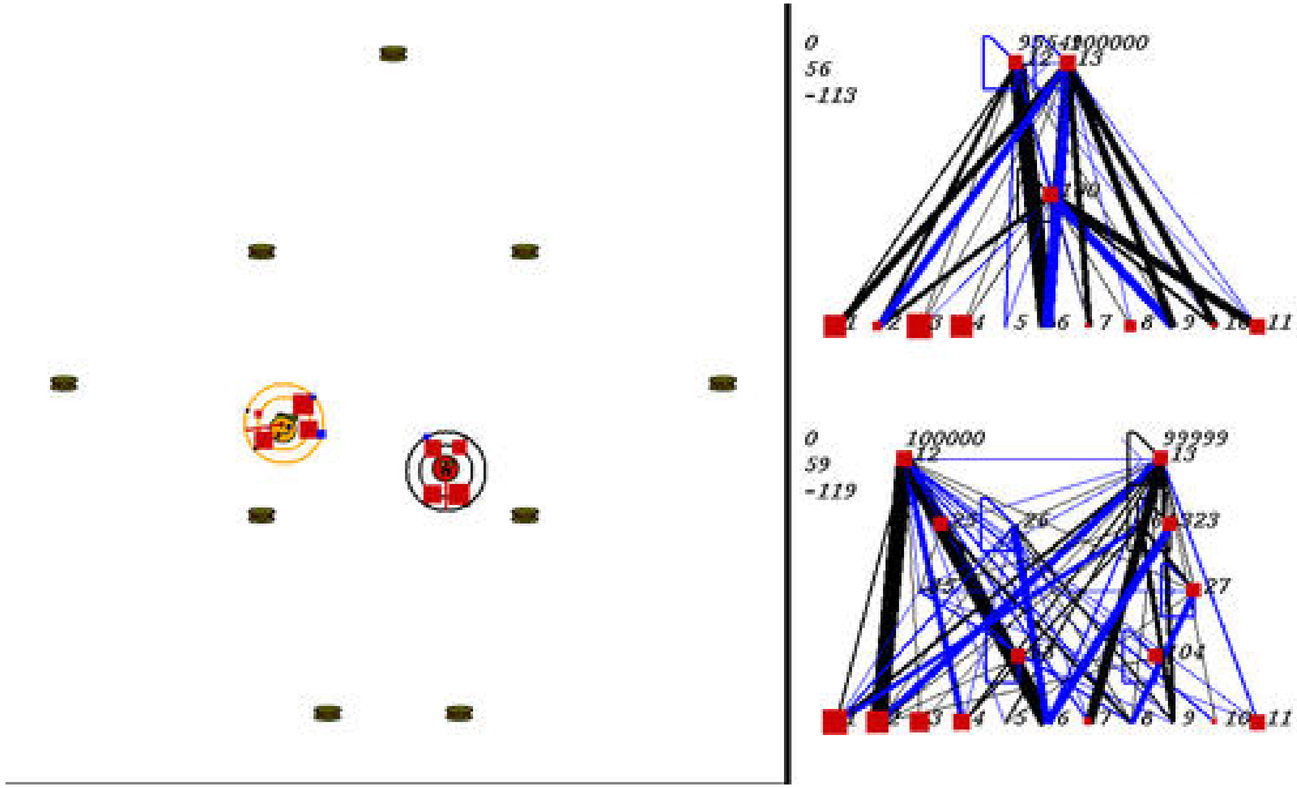
- Gen 80 champ vs. Gen 95 descendant
- Switching behavior between foraging, caution, predation; Final standoff.

Old West-Style Standoff



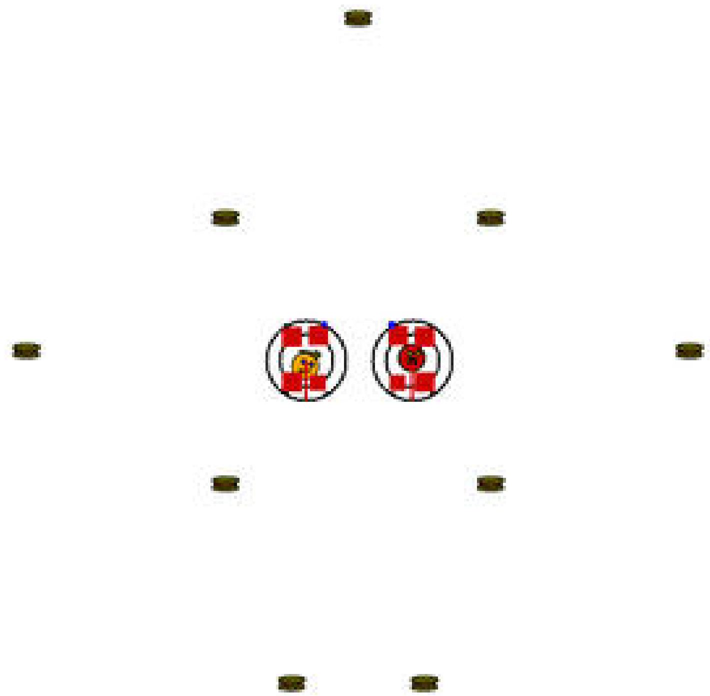
- Gen 95 vs. gen 90 champ.
- Extended standoff

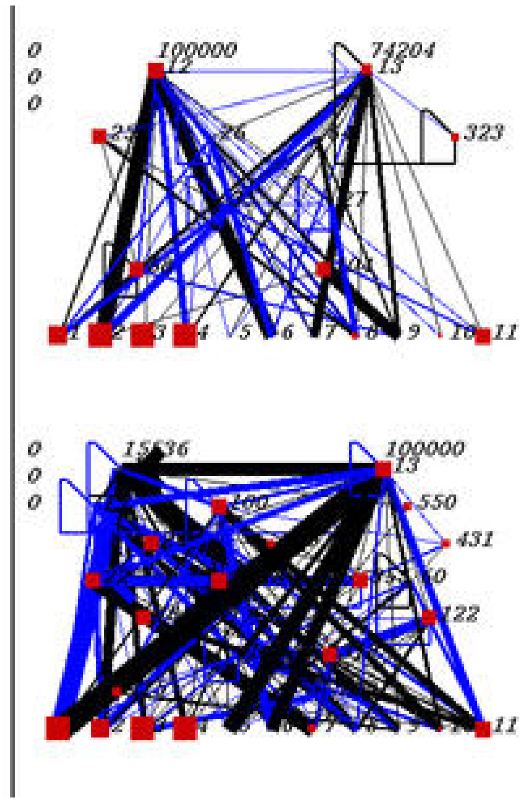
Later Dominant vs. Early Good Str.



- Gen 221 champ (later dominant strategy) vs. gen 130 champ (first good strategy).
- Caution when seeking food. Switching of strategy observed.

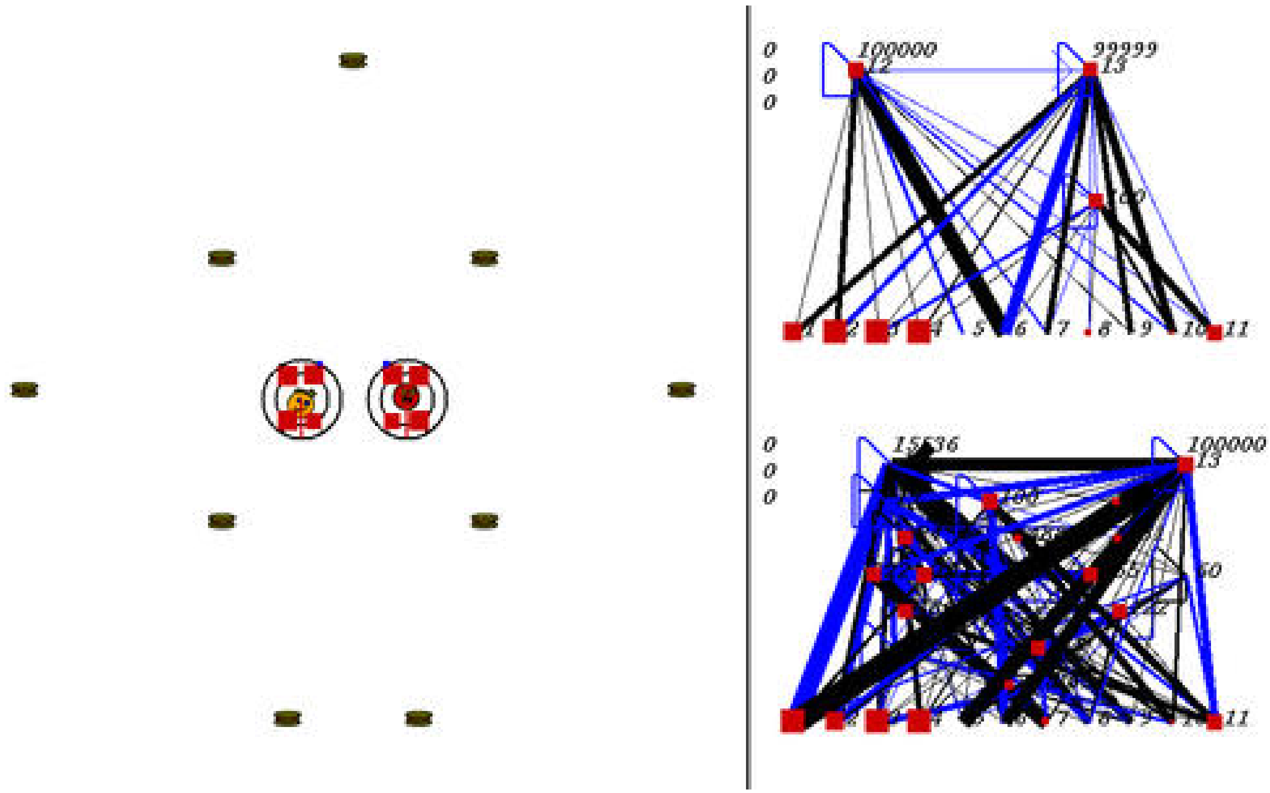
Highest- vs. Prior-Dominant Str.





- Gen 313 champ vs. gen 210 champ.
- Waiting until the moment is just right.
- Food nearby, enemy wasting energy, etc. all considered.

Highest Dominant vs. First Good Str.



- Gen 313 champ vs. gen 95 champ.
- Highest dominant is dominant over all past dominant.

Other Applications of NEAT

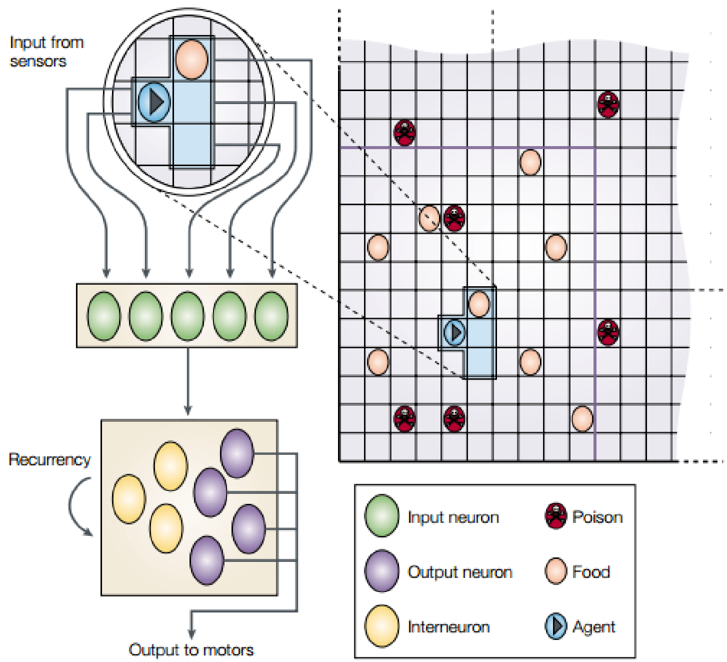
- NERO (NeuroEvolution of Robotic Operatives): Interactive neuroevolution for realtime strategy game-like environment (`http://nerogame.org`)
- Dancing, driving, generation of art, etc.
- See Ken Stanley's web page.

Summary (NEAT)

- Evolving neural network topologies helps evolve complex emergent behavior.
- Co-evolution ensures continuous progress.
- Diverse applications possible.

III. Neuroevolution for Neuroscience Research

Evolutionary Autonomous Agents for Neuroscience Research



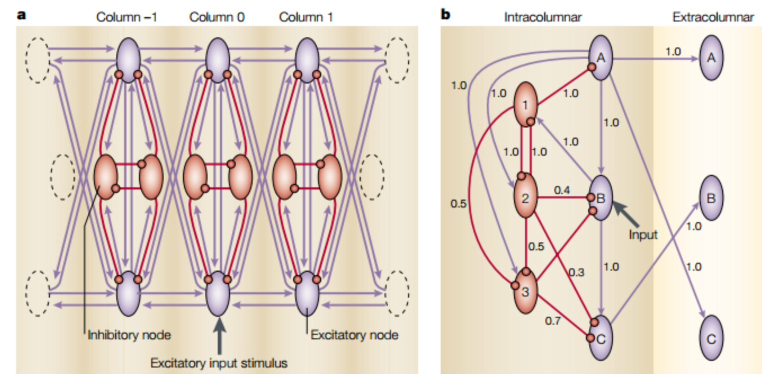


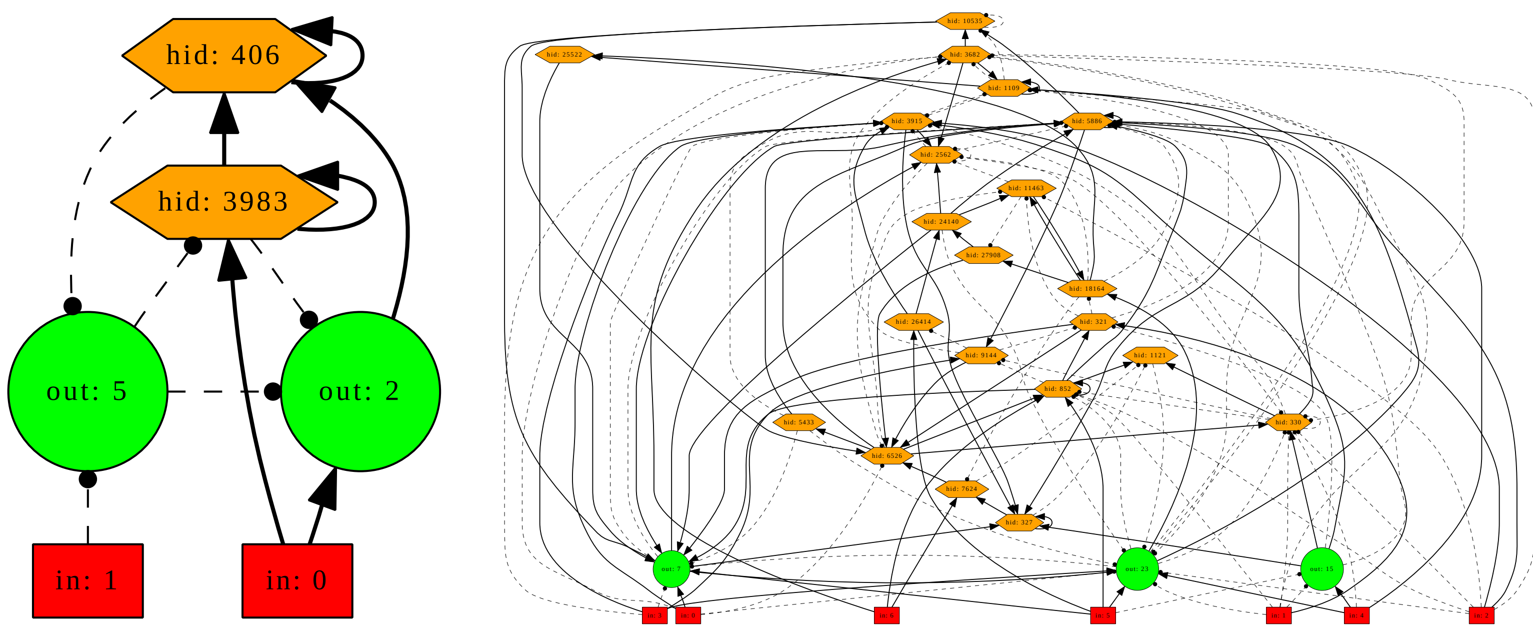
Figure 4 | **Evolution of cortical circuits. a** | An example initial cortical circuit before evolution with two excitatory and two inhibitory nodes per column. Each node represents a small population of similar neurons. Initial synaptic strengths were random, and different evolutionary runs started with different numbers of excitatory/inhibitory nodes. **b** | A diagram of an evolved cortical circuit that produces a 'Mexican hat pattern' of activity to a point stimulus in the absence of horizontal, intercolumnar inhibitory connections. Only the magnitudes of inhibitory weights are shown (red connections); they are multiplied by a negative gain constant when used. Connections with weights of <20% of the maximum were omitted. Connections for all columns are the same. Lateral inhibition arises because excitation of some excitatory neurons (nodes B and C) in a stimulated column causes other excitatory neurons (node A) in the same column (those sending lateral excitatory connections to other columns) to turn off, decreasing lateral excitation of adjacent columns and causing their mean activation levels to fall. Inhibitory neurons evolved not only to inhibit intracolumnar excitatory neurons, but also to inhibit each other in a highly specific pattern. Adapted from REF. 43 © 2001 IEEE Press.

Ruppin [7]

- Need to study not just the neural network but also the body of evoloved agents.
- Helps study a natural computational framework to study the interaction between learning and evolution.
- Address open questions, e.g., why the Mexican-hat type inhibition?

Evolutionary Autonomous Agents: Discussion

- Indirect genotype-to-phenotype encodings: More efficient and flexible to use grammar rewriting encodings instead of direct encoding of weights.
- Analysis of neural information processing in evolved brains: activity monitoring, leision studies, stimulation, etc.



Huang and Choe (Unpublished)

Active Vision and RF Development

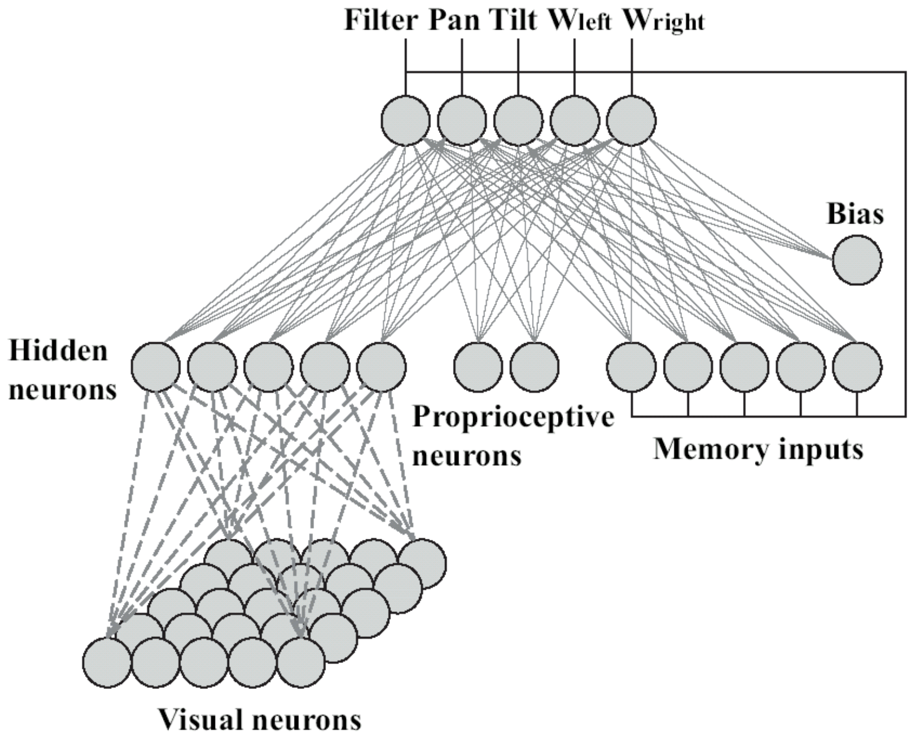
Floreano et al. [2]

- Mobile robot with pan/tilt camera.
- Neural controller mapping visual input to motor output.
- Combined GA and Hebbian learning (in visual system).
- Addition of Hebbian learning results in robust adaptive behavior.
- Active vision gives better RFs than from randomly sampled images.
- Interplay of active vision and RF formation amounts to selection and exploitation of a small and constant subset of visual features available to the robot.

Motivation

- "... it is generally accepted that animals and models of early-vision stages extract the dominant statistical features of their visual environment. Within that perspective, the visual system is a passive, albeit plastic, device shaped by the environment."
- "Active vision can simplify the computation involved in vision processing by selecting only characteristics of the visual scene that are relevant for the task to be solved, thus reducing the information load on the neural system."
- "... the interaction between receptive field formation and active vision has been largely neglected in the biological and computational literature."

Neural Architecture



- Network weights evolved using GA.
- 5×5 array of visual neurons, each with 48×48 receptive field.
- Visual-to-hidden RF weights learned through Hebbian learning.
- Sigmoidal units: hidden and output.

Two Visual Activation Modes

- Left: Input
- Middle: visual neuron's activity = mean of all pixels in RF.
- Right: visual neuron's activity = one pixel sample in RF.
- **Note**: visual neuron's RF is not adaptive; visual neuron to hidden neuron weights are adaptive.

Adaptation and Evolution

- Standard sigmoid of weighted sum.
- Weight adaptation follows $x_j \longrightarrow y_i$:

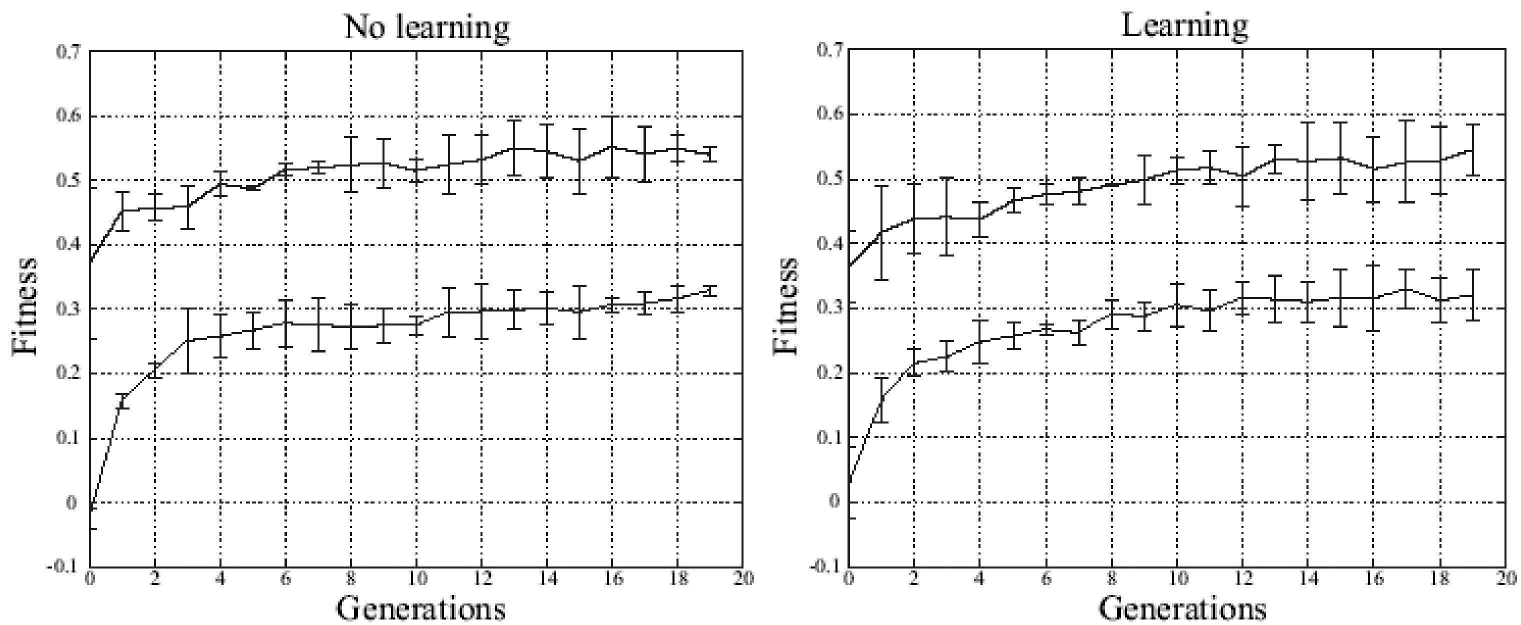
$$\Delta w_{ij} = y_i \left(x_j - \sum_{k=1}^{i} w_{kj} y_k \right).$$

- 100 genomes, evolved for 20 generations.
- Best 20% individuals reproduces. Crossover prob. 0.1 per pair and mutation prob. 0.01 per bit.
- Fitness: ability to move straight forward.

Operating Environment

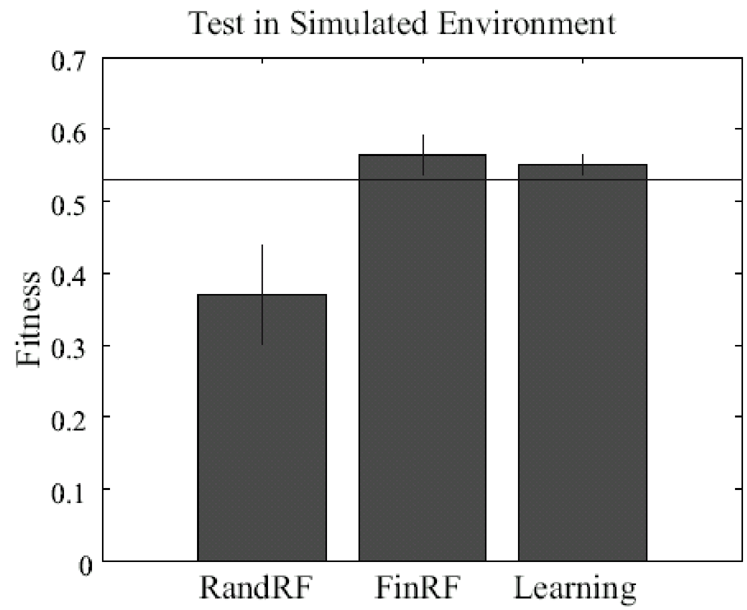
- Simulated or real.
- Trained in simulation (ontogenetic development), tested in real environment.
  - Simulation to real transition is not always easy.
  - Does learning make the transition easier?

Learning in Simulated Environment



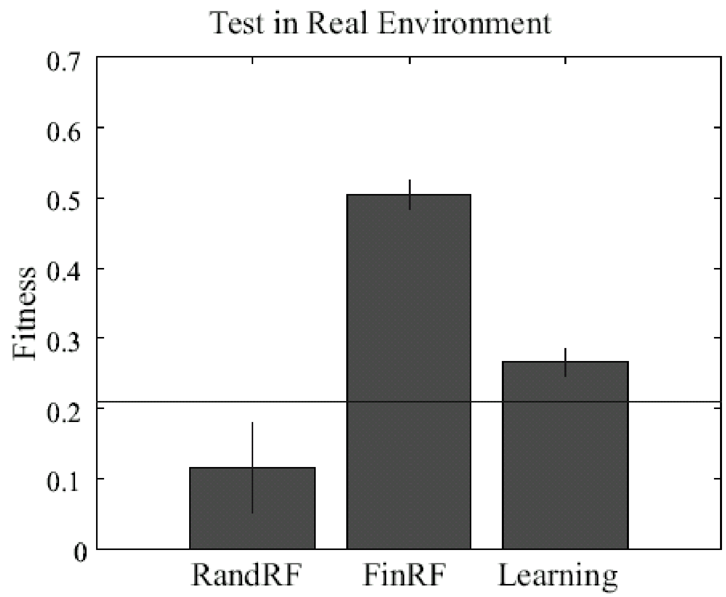
- No difference between learning vs. no learning (evolution only).

Learning vs. Fixed (Nearly Optimal) RF



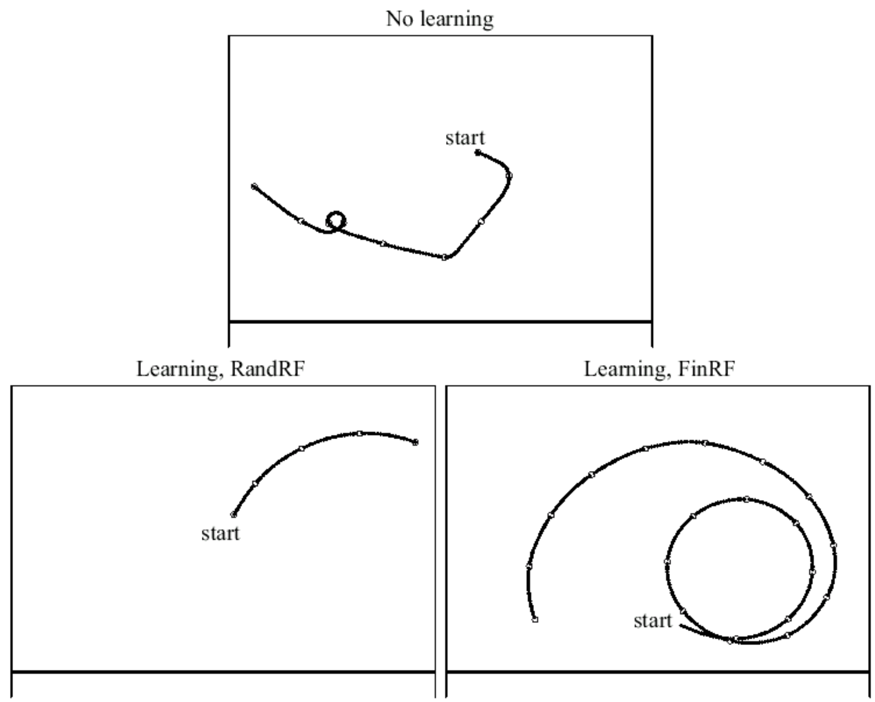
- RandRF: Random visual to hidden connections.
- FinRF: Visual to hidden weights fixed during lifetime (to well-formed RF)
- Learning: Initially random, learn weights during individual's life.
- No difference between FinRF and Learning conditions: no overhead for learning, in terms of fitness.

Simulation to Real Environment



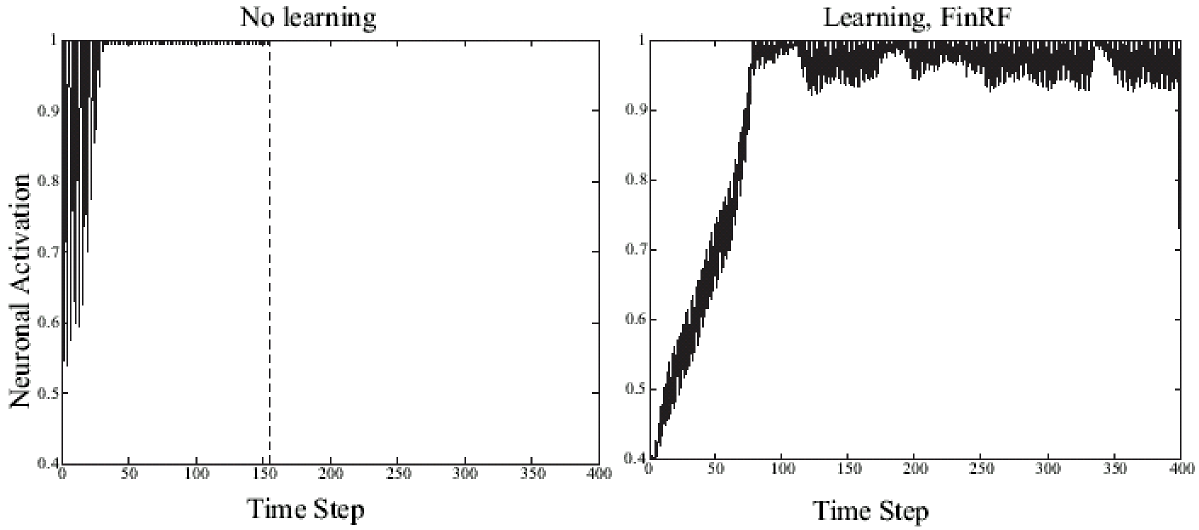
- RandRF: Random visual to hidden connections.
- FinRF: Visual to hidden weights fixed during lifetime (to well-formed RF in new env)
- Learning: Initially random, learn weights during individual's life.
- Overhead in learning, due to novelties in environment.

Behavior in Real Environment



- Evolution only: collision, tilt camera down
- Learning, initial: collison
- Learning, final: no collision, pan camera (possibly due to smaller number of features encoded)

Pan Angle of Learning vs. No Learning Agents

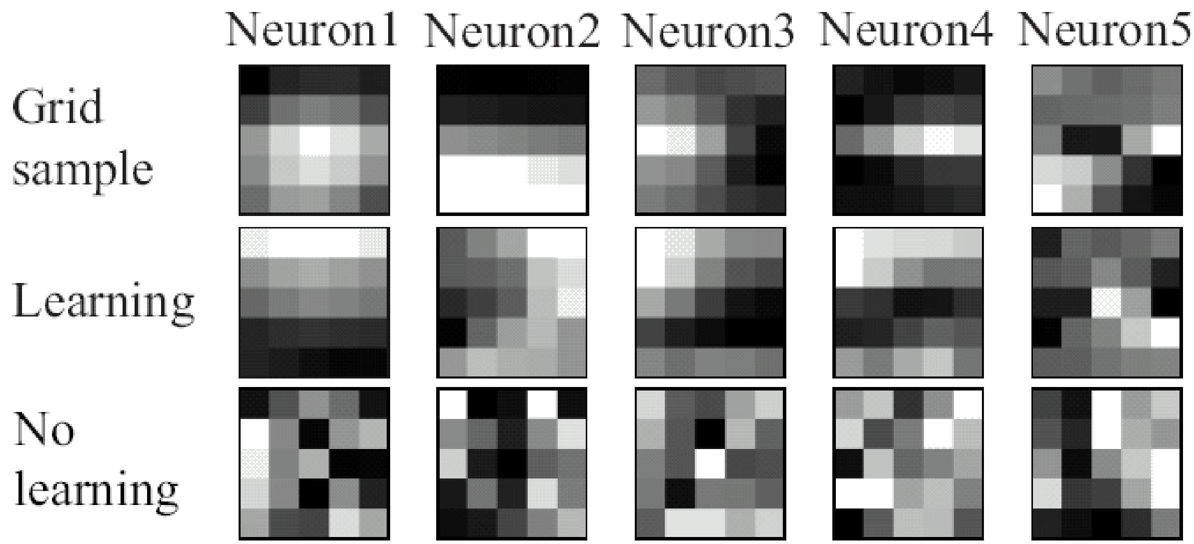


- Learning condition give more variability in panning angle.

RF Training

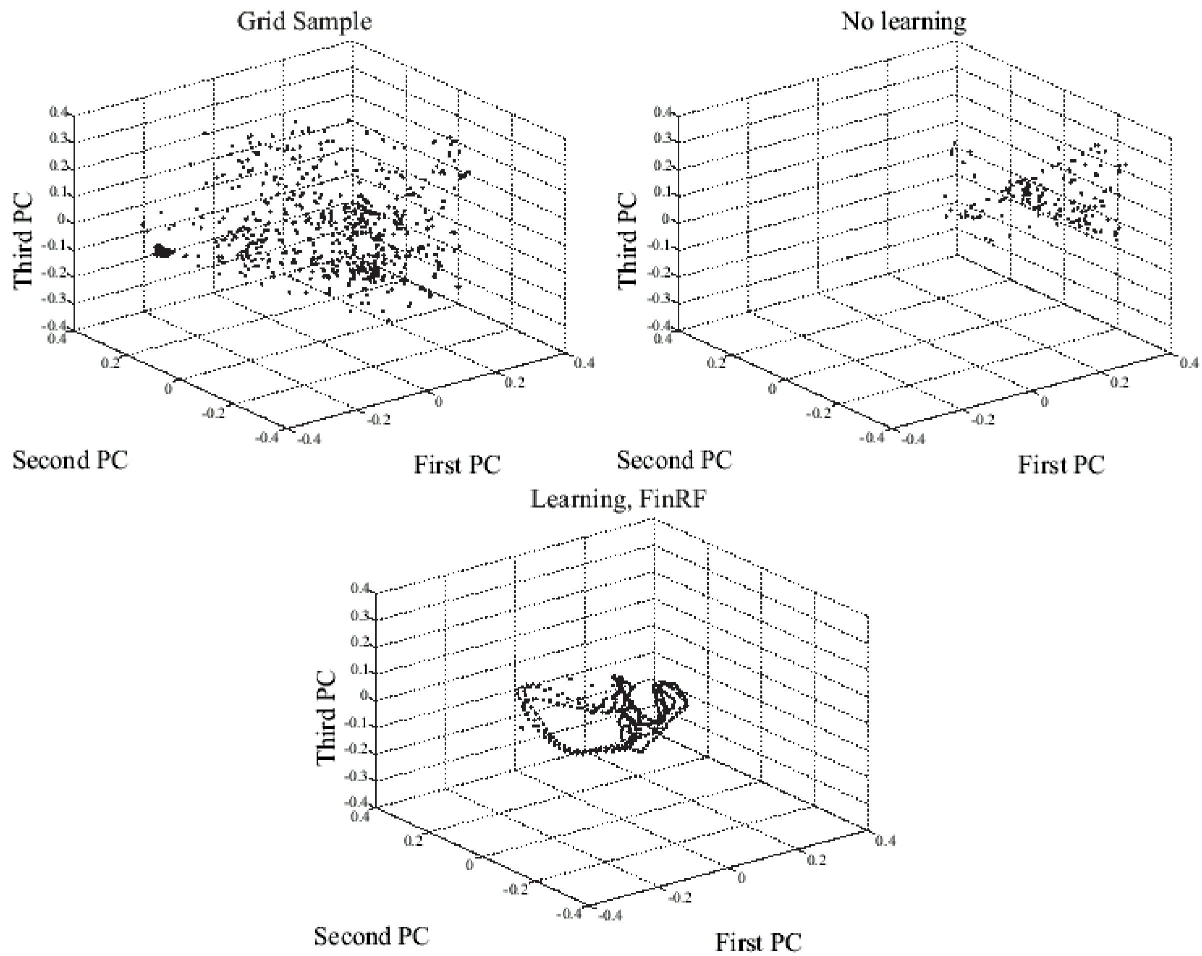
- Systematically sampled outdoor image samples for training RF, compared to Hebbian learning.
- Typical input scenes shown above.

RF Development



- Grid sample: RF learned off-line.
- Learning: RF learned on-line.
- No learning: Random RF of best no-learning case.
- "...robots self-select ... a significantly smaller and consistent subset of visual features ... for performing ..."

PCA of Observed Scenes



- Scenes gathered (PCA) in different types of agents.
- “...robots self-select ... a significantly smaller and consistent subset of visual features ... for performing ...”

Summary

- Behavior affects learning by selecting a subset of learning experiences that are functional to the survival task.
- Learning affects behavior by generating selection pressure for actions that actively search for situations that are learned.
- Learning contributes to the adaptive power of evolution by coping with change that occurs faster than the evolutionary timescale.

Discussion (YC)

- “Interplay of active vision and RF formation amounts to selection and exploitation of a small and constant subset of visual features available to the robot.”
  - That “constant subset” may not be that of “visual features” but of **motor primitives**.
- The fitness “ability to move straight forward” seems too arbitrary.
- The main conclusions are a bit too far reaching.

IV. Wrap Up

Discussion and Conclusion

- Neuroevolution can help us understand the evolution of complex behavior.
  - Need to study brain + body, in an evolutionary context.
  - Action plays a key role.
- Neuroevolution can be used as a novel tool in computational neuroscience.
- Analyzing evolved artificial neural networks is a challenge.

References

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

[2] Floreano, D., Suzuki, M., and Mattiussi, C. (2005). Active vision and receptive field development in evolutionary robots. *Evolutionary Computation*, 13:527–544.

[3] Heider, F., and Simmel, M. (1944). An experimental study of apparent behavior. *The American Journal of Psychology*, 57:243–259.

[4] Llinás, R. R. (2001). *I of the Vortex*. Cambridge, MA: MIT Press.

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

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

[7] Ruppin, E. (2002). Evolutionary autonomous agents: A neuroscience perspective. *Nature Reviews Neuroscience*.

[8] Stanley, K. O. (2004). *Efficient Evolution of Neural Networks Through Complexification*. PhD thesis, University of Texas at Austin.

[9] Stanley, K. O., and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127.

[10] Stanley, K. O., and Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100.

[11] Swanson, L. W. (2003). *Brain Architecture: Understanding the Basic Plan*. Oxford: Oxford University Press.

[12] Whiteson, S., Stone, P., Stanley, K. O., Miikkulainen, R., and Kohl, N. (2005). Automatic feature selection in neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*.