Neuroevolution Basics

- A single chromosome encodes a full neural network.
- Each gene, a single bit (or a real number), maps to a connection weight in the neural network.

Neuroevolution Basics: Operations

- Cross-over.
- Mutation.
Neuroevolution Basics: Cross-Over in Detail

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

Why Neuroevolution?
- Neural nets powerful in many statistical domains
  - E.g. control, pattern recognition, prediction, decision making
- Good supervised training algorithms exist
  - Learn a nonlinear function that matches the examples
- What if correct outputs are not known?

Sequential Decision Tasks
- POMDP: Sequence of decisions creates a sequence of states
- No targets: Performance evaluated after several decisions
- Many important real-world domains:
  - Robot/vehicle/traffic control
  - Computer/manufacturing/process optimization
  - Game playing

Forming Decision Strategies
- Traditionally designed by hand
  - Too complex: Hard to anticipate all scenarios
  - Too inflexible: Cannot adapt on-line
- Need to discover through exploration
  - Based on sparse reinforcement
  - Associate actions with outcomes
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 disambiguated through memory
  - Recurrency in neural networks

How well does it work?

<table>
<thead>
<tr>
<th>Poles</th>
<th>Method</th>
<th>Evals</th>
<th>Succ.</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>VAPS</td>
<td>500,000</td>
<td>0%</td>
</tr>
<tr>
<td>SARSA</td>
<td>13,562</td>
<td>59%</td>
<td></td>
</tr>
<tr>
<td>Q-MLP</td>
<td>11,331</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NE</td>
<td>589</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two</td>
<td>NE</td>
<td>24,543</td>
<td></td>
</tr>
</tbody>
</table>

- Difficult RL benchmark: Non-Markov Pole Balancing
- NE 2 orders of magnitude faster than standard RL
- NE can solve harder problems

Role of Neuroevolution

- Powerful method for sequential decision tasks
- Optimizing existing tasks
- Discovering novel solutions
- Making new applications possible
- Also may be useful in supervised tasks
- Especially when network topology important
- Unique model of biological adaptation and development
Outline

• Basic neuroevolution techniques
• Advanced techniques
  – E.g. combining learning and evolution
• Extensions to applications
• Application examples
  – Control, Robotics, Artificial Life, Games

Neuroevolution Decision Strategies

• Input variables describe the state
• Output variables describe actions
• Network between input and output
  – Hidden nodes
  – Weighted connections
• Execution:
  – Numerical activation of input
  – Nonlinear weighted sums
• Performs a nonlinear mapping
  – Memory in recurrent connections
• Connection weights and structure evolved

Conventional Neuroevolution (CNE)

• Evolving connection weights in a population of networks
• Chromosomes are strings of weights (bits or real)
  – E.g. 10010110101100101111001
  – Usually fully connected, fixed topology
  – Initially random

Conventional Neuroevolution (2)

• Each NN evaluated in the task
  – Good NN reproduce through crossover, mutation
  – Bad thrown away
  – Over time, NNs evolve that solve the task
• Natural mapping between genotype and phenotype
• GA and NN are a good match!
Problems with CNE

- 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 Neurons

- Evolving individual neurons to cooperate in networks \(^1,2,24\)
  (Agogino GECCO’05)

- E.g. Enforced Sub-Populations (ESP \(?\))
  - Each (hidden) neuron in a separate subpopulation
  - Fully connected; weights of each neuron evolved
  - Populations learn compatible subtasks

Advanced NE 2: Evolutionary Strategies

- Evolving complete networks with ES \(^{15}\)
- 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 Topologies

- Optimizing connection weights and network topology
- E.g. Neuroevolution of Augmenting Topologies (NEAT)
- Based on Complexification
  - Mutations to add nodes and connections
  - Elaborates on earlier behaviors

How Can Crossover be Implemented?

- Problem: Structures do not match
- Solution: Utilize historical markings

<table>
<thead>
<tr>
<th>Genome (Genotype)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>Genes</td>
<td>Node</td>
<td>Genes</td>
<td>Node</td>
</tr>
<tr>
<td>Node 1</td>
<td>Sensor</td>
<td>Node 2</td>
<td>Sensor</td>
<td>Node 3</td>
</tr>
<tr>
<td>In 1</td>
<td>Out 4</td>
<td>Weight of 0.7</td>
<td>Enabled</td>
<td>Innov 1</td>
</tr>
<tr>
<td>In 2</td>
<td>Out 4</td>
<td>Weight of -0.5</td>
<td>DISABLED</td>
<td>Innov 2</td>
</tr>
<tr>
<td>In 3</td>
<td>Out 4</td>
<td>Weight of 0.5</td>
<td>Enabled</td>
<td>Innov 3</td>
</tr>
<tr>
<td>In 5</td>
<td>In 1 In 4</td>
<td>Out 4</td>
<td>Weight of 0.4</td>
<td>Enabled</td>
</tr>
<tr>
<td>In 6</td>
<td>In 1 In 4</td>
<td>Out 5</td>
<td>Weight of 0.6</td>
<td>Enabled</td>
</tr>
<tr>
<td>Node 5</td>
<td>Hidden</td>
<td>Node 6</td>
<td>Hidden</td>
<td>Node 11</td>
</tr>
</tbody>
</table>

How Can We Search in Large Spaces?

- Need to optimize not just weights but also topologies
  - Hidden nodes, connections, input features
  - Solution: Start with minimal structure and complexify

How can Innovation Survive?

- Problem: Innovations have initially low fitness
- Solution: Speciate the population
  - Innovations have time to optimize
  - Mitigates competing conventions
  - Promotes diversity
Further NE Techniques

- Incremental evolution$^{13,33,39}$
- Utilizing population culture$^{2,18}$
- Evolving ensembles of NNs$^{16,23,36}$
  (Pardoe GECCO’05)
- Evolving neural modules$^{25}$
- Evolving transfer functions and learning rules$^{4,26}$
- Combining learning and evolution

Extending NE to Applications

- Evolving composite decision makers$^{36}$
- Evolving teams of agents$^{3,28,41}$
- Utilizing coevolution$^{30}$
- Real-time neuroevolution$^{28}$
- Combining human knowledge with evolution$^{8}$

Applications to Control

- Pole-balancing benchmark
  - Originates from the 1960s
  - Original 1-pole version too easy
  - Several extensions: acrobat, jointed, 2-pole, particle chasing$^{23}$
- Good surrogate for other control tasks
  - Vehicles and other physical devices
  - Process control$^{34}$

Competitive Coevolution

- Evolution requires an opponent to beat
- Such opponents are not always available
- Co-evolve two populations to outdo each other
- How to maintain an arms race?
Competitive Coevolution with NEAT

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

Robot Duel Domain

- Two Khepera-like robots forage, pursue, evade
  - Collect food to gain energy
  - Win by crashing to a weaker robot

Early Strategies

- Crash when higher energy
- Collect food by accident
- DEMO

Mature Strategies

- Collect food to gain energy
- Avoid moving to lose energy
- Standoff: Difficult to predict outcome
- DEMO
Sophisticated Strategy

- “Fake” a move up, force away from last piece
- Win by making a dash to last piece
- Complexification $\rightarrow$ arms race
- DEMO

Discovering Novel Strategies in Othello

- Players take turns placing pieces
- Each move must flank opponent’s piece
- Surrounded pieces are flipped
- Player with most pieces wins

Applications to Games

- Good research platform
  - Controlled domains, clear performance, safe
  - Economically important; training games possible
- Board games: beyond limits of search
  - Evaluation functions in checkers, chess $^{5,9,10}$
  - Filtering information in go, othello $^{20,31}$

Strategies in Othello

- Positional
  - Number of pieces and their positions
  - Typical novice strategy
- Mobility
  - Number of available moves: force a bad move
  - Much more powerful, but counterintuitive
  - Discovered in 1970’s in Japan
Evolving Against a Random Player

- Network sees the board, suggests moves by ranking
- Networks maximize piece counts throughout the game
- A positional strategy emerges
- Achieved 97% winning percentage

Example game

- Black's positions strong, but mobility weak
- White (the network) moves to f2
- Black's available moves b2, g2, and g7 each will surrender a corner
- The network wins by forcing a bad move

Discovering Novel Strategies

- Neuroevolution discovered a strategy novel to us
- "Evolution works by tinkering"
  - So does neuroevolution
  - Initial disadvantage turns into novel advantage
Future Challenge: Utilizing Knowledge

- Given a problem, NE discovers a solution by exploring
  - Sometimes you already know (roughly) what works
  - Sometimes random initial behavior is not acceptable

- How can domain knowledge be utilized?
  - By incorporating rules (Yong GECCO’05)
  - By learning from examples

Numerous Other Applications

- Creating art, music
- Theorem proving
- Time-series prediction
- Computer system optimization
- Manufacturing optimization
- Process control optimization
- Etc.

Conclusion

- NE is a powerful technology for sequential decision tasks
  - Evolutionary computation and neural nets are a good match
  - Lends itself to many extensions
  - Powerful in applications

- Easy to adapt to applications
  - Control, robotics, optimization
  - Artificial life, biology
  - Gaming: entertainment, training

- Lots of future work opportunities
  - Theory not well developed
  - Indirect encodings
  - Learning and evolution
  - Knowledge and interaction

References


