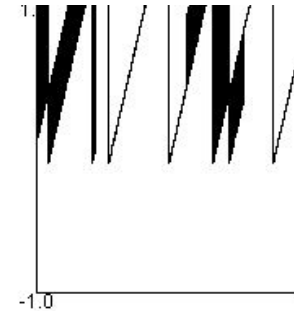
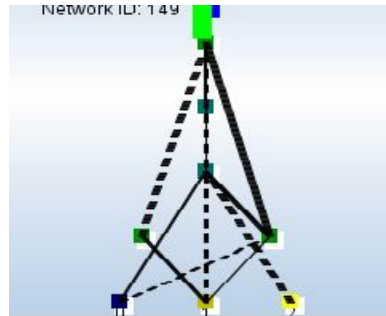
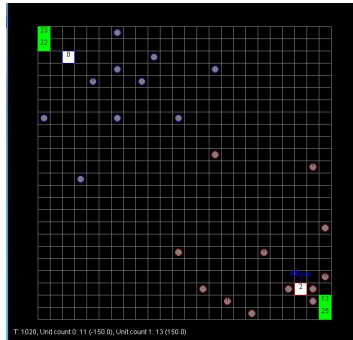
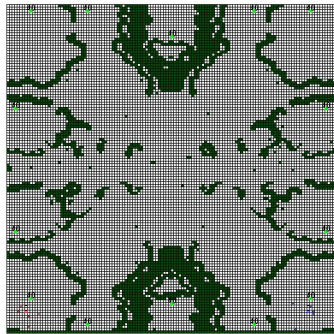


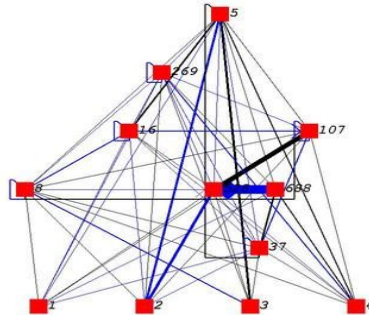
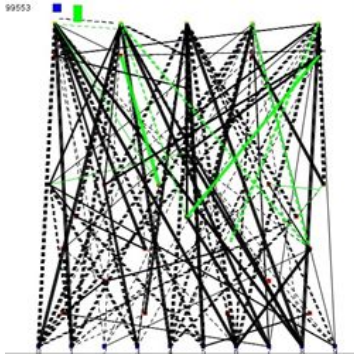
Evolutionary Computation for Creativity and Intelligence

By Darwin Johnson, Alice Quintanilla, and Isabel Tweraser



Introduction to NEAT

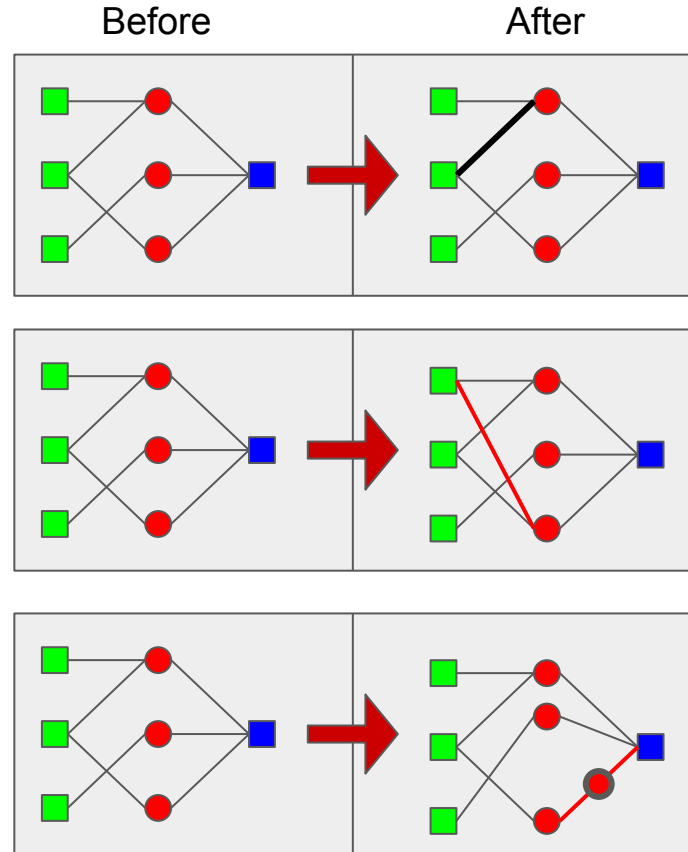
- Stands for NeuroEvolution of Augmenting Topologies (NEAT)
- Evolves an Artificial Neural Network of nodes (simple artificial “brain”)
 - Generates population of ANNs to solve a problem (usually ineffective at first)
 - Best performing ANNs continue to the next generation and produce children
 - Mutates the ANNs to change the behavior
 - Alters the weight of an existing link between two nodes
 - Creates a new link between nodes
 - Creates new nodes along existing links



Mutations

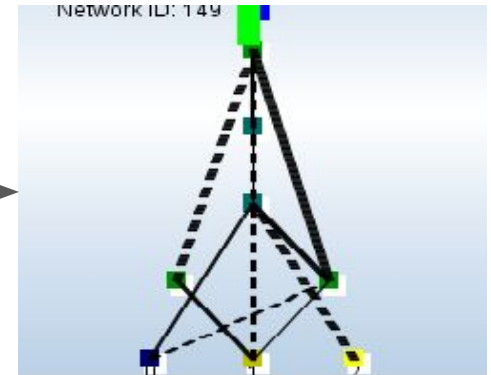
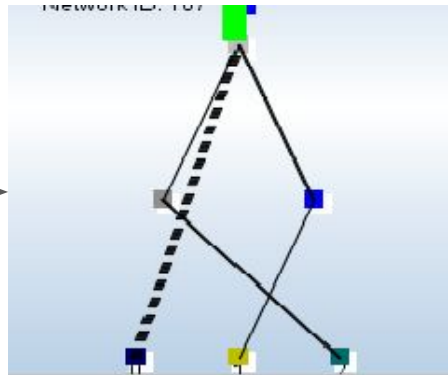
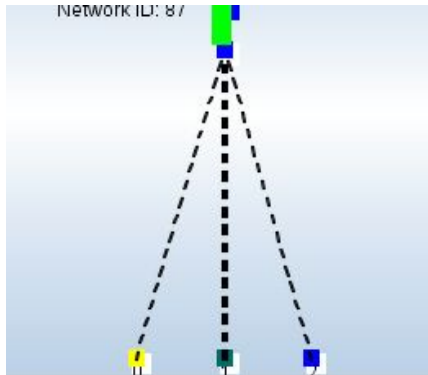
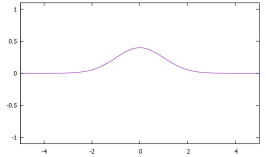
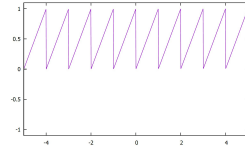
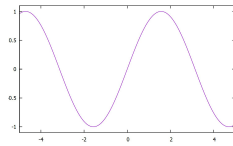
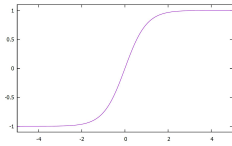
- Alter Link Weight
 - Alters the influence of an existing link
 - May increase or decrease the weight
- Add Link
 - Adds a link between two existing nodes
 - Creates influence from one node to another
- Add Node
 - Creates a new node along an existing link
 - Adds new influence to the final output
 - Creates new location for links to connect to

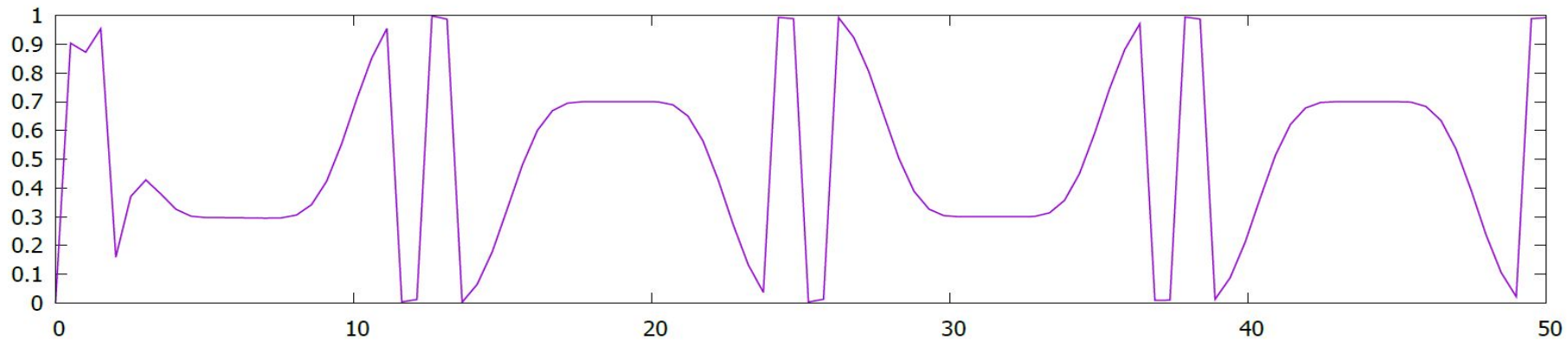
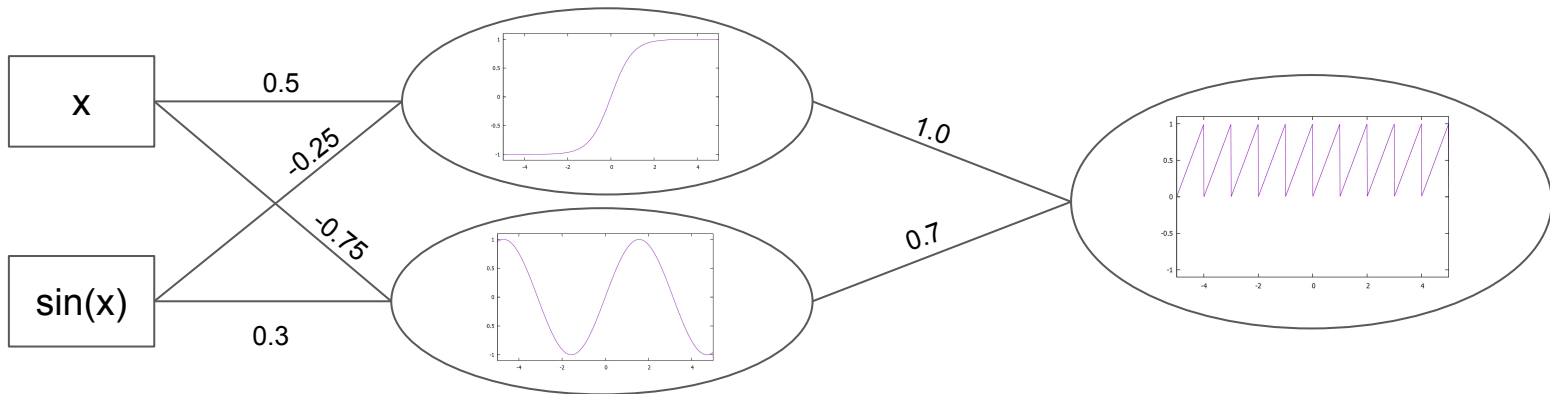
Child ANNs

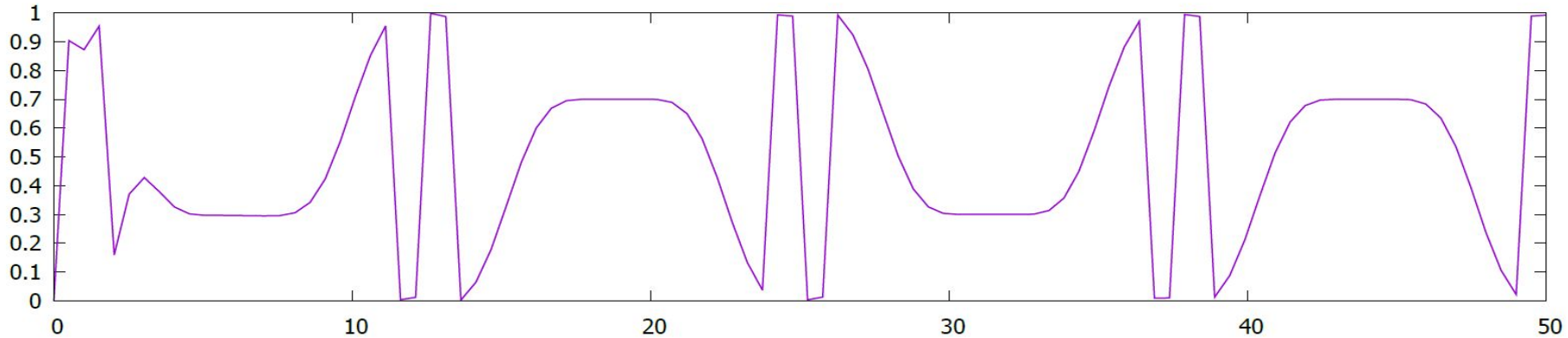
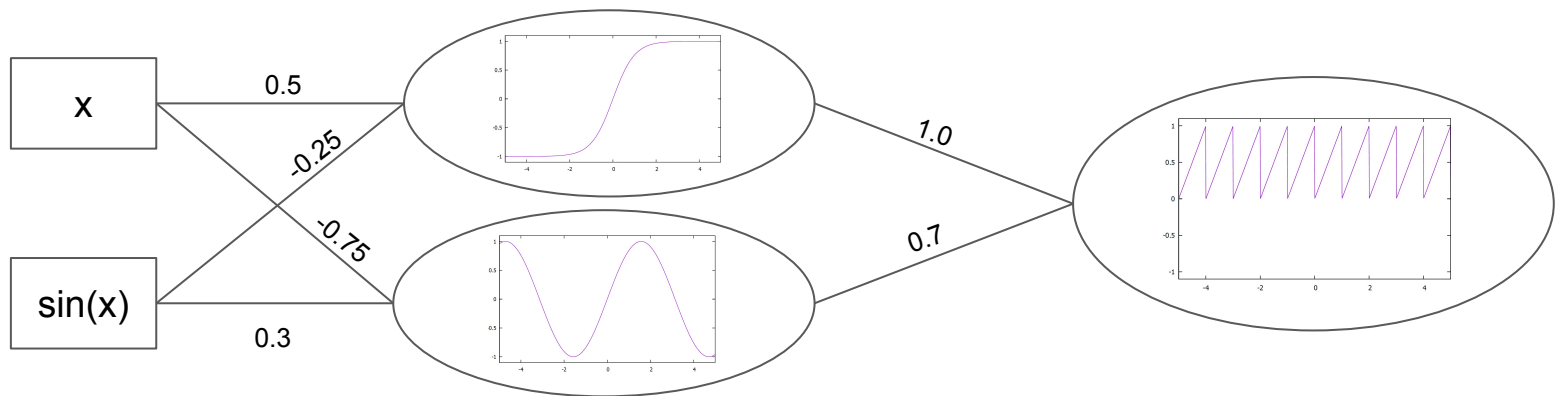


Compositional Pattern Producing Networks (CPPNS)

- Variant of ANN with variety of activation functions in its nodes
- Activation functions create patterns reminiscent of features of natural life:
 - Repetition
 - Symmetry
 - Variation
- Can be used to generate interesting images and sounds



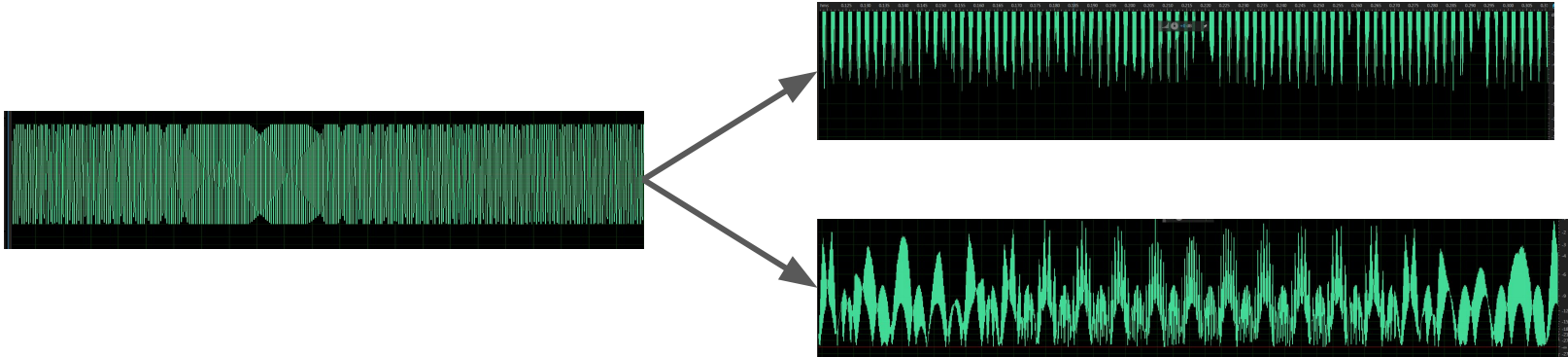




```
saw(1*tanh(x*0.5+(-0.75)*sin(x)) + (0.7*sin(x*(-0.25)+0.3*sin(x)))
```

Evolving Music and Sounds

- Use a CPPN to generate an amplitude wave
- Can be displayed and played as a sound wave
- Two extensions:
 - **Breedsizer** - Evolves sounds that can be played with different frequencies/notes*
 - **Remixbreeder** - Takes in a song and outputs a remixed version

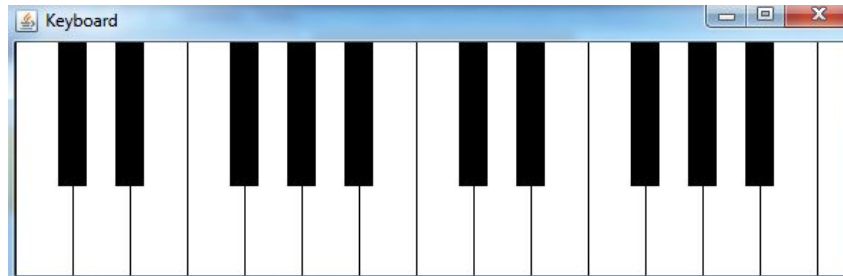
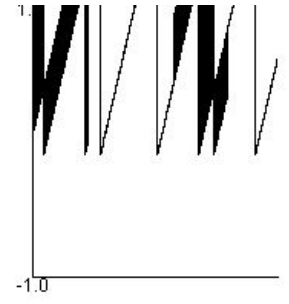
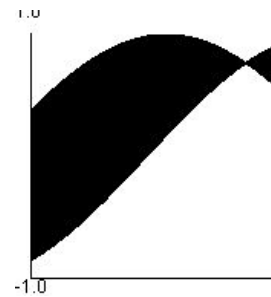
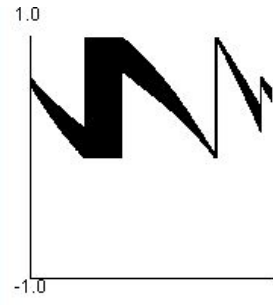
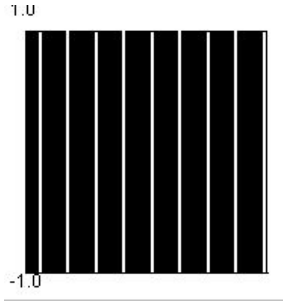


Breedsizer Interface



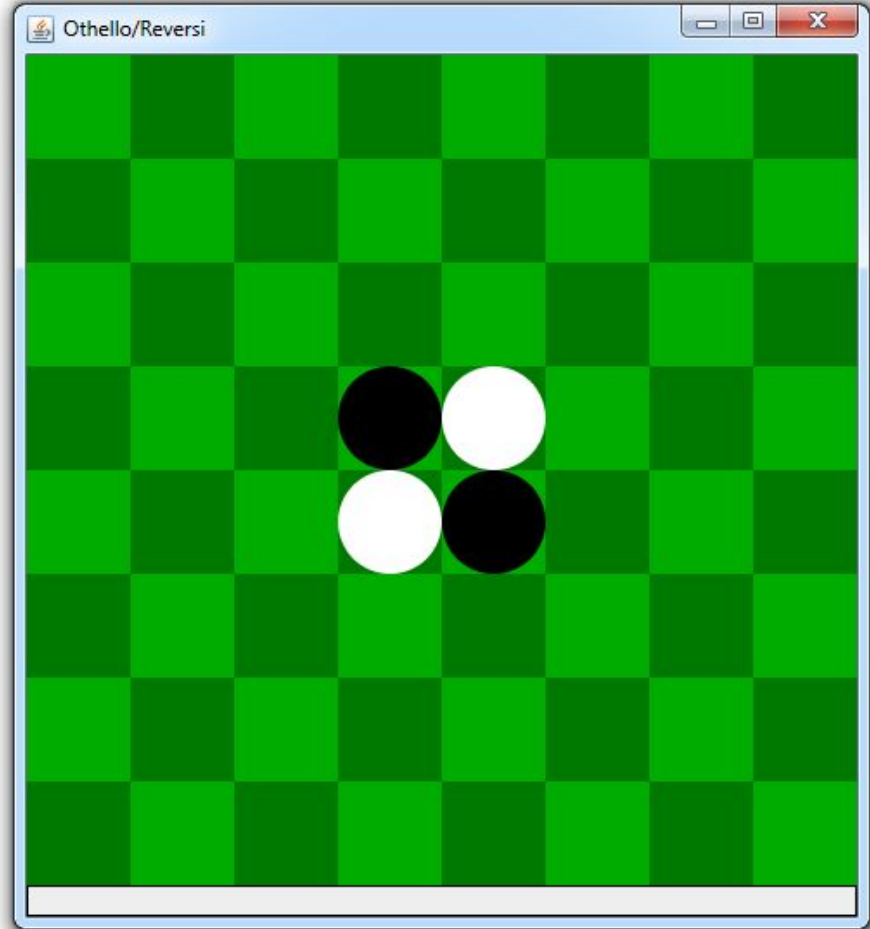
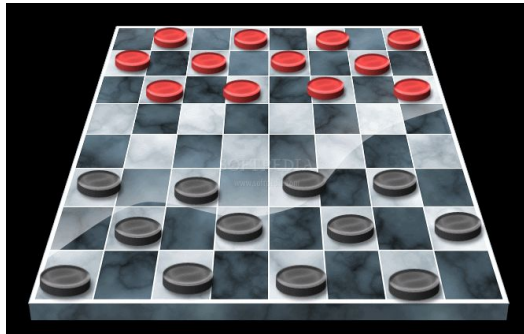
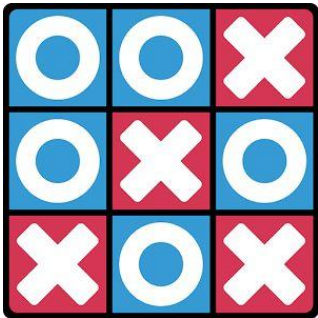
The interface control bar includes several icons for navigation and actions: a tree diagram, a circular arrow, a network diagram, a right-pointing arrow, a floppy disk, a circular arrow with a plus sign, and a power button. Below these icons are two sliders: the first is labeled 'Fewer Mutations' and 'More Mutations', and the second is labeled 'Shorter clip' and 'Longer clip'. To the right of the sliders are three checked checkboxes: 'Time', 'Sine_of_time', and 'bias'. A 'PlayWithMIDI' button is located on the far right.

half_linear absolute_value sawtooth sine gaussian sigmoid tanh ID full_approximate approximate stretched_tanh ReLU softplus leaky_ReLU full_sawtooth triangle_wave square_wave



Board Games

- Common tests for Artificial Intelligence
- Tic-Tac-Toe, Checkers, Othello, and more
- Several Opponent choices to create an Agent:
 - Static Opponent
 - Co-Evolution
- Evolve board evaluation functions
 - Board state evaluated by ANN
 - Move with highest output selected

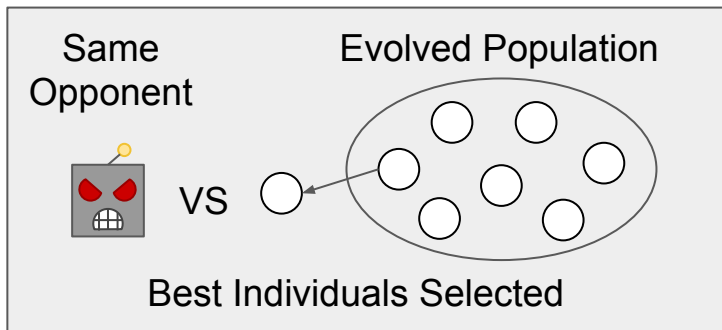


Board Game Opponents

Static Opponent

Agent is evolved against a non-evolving agent.

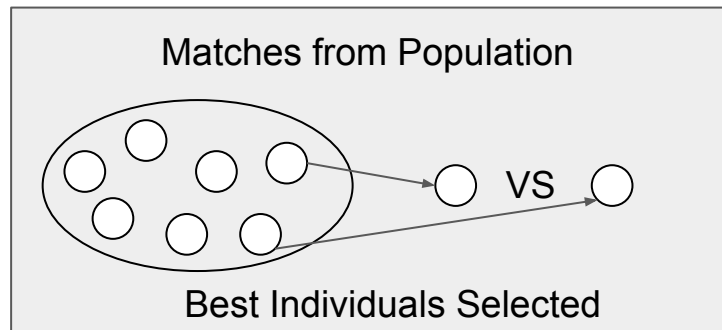
- Used as a Benchmark
 - Easier to compare against
 - Can be considered a “goal” to reach
- Agents evolve to beat this specific Opponent
 - May not be able to beat other opponents
 - Not necessarily “good” agents



Co-Evolution

Agents are evolved against each other.

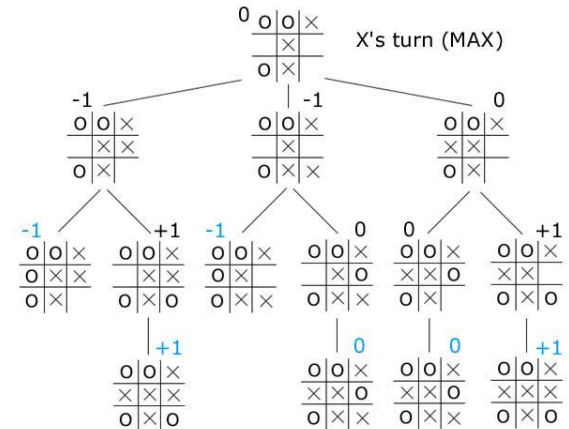
- Agents evolve as a group
 - Fitness depends on other agents in population
 - Should learn general intelligent behavior
- More difficult to benchmark
 - Unable to have a consistent opponent
 - Emergence of unusual weaknesses possible



Evaluation of Game States

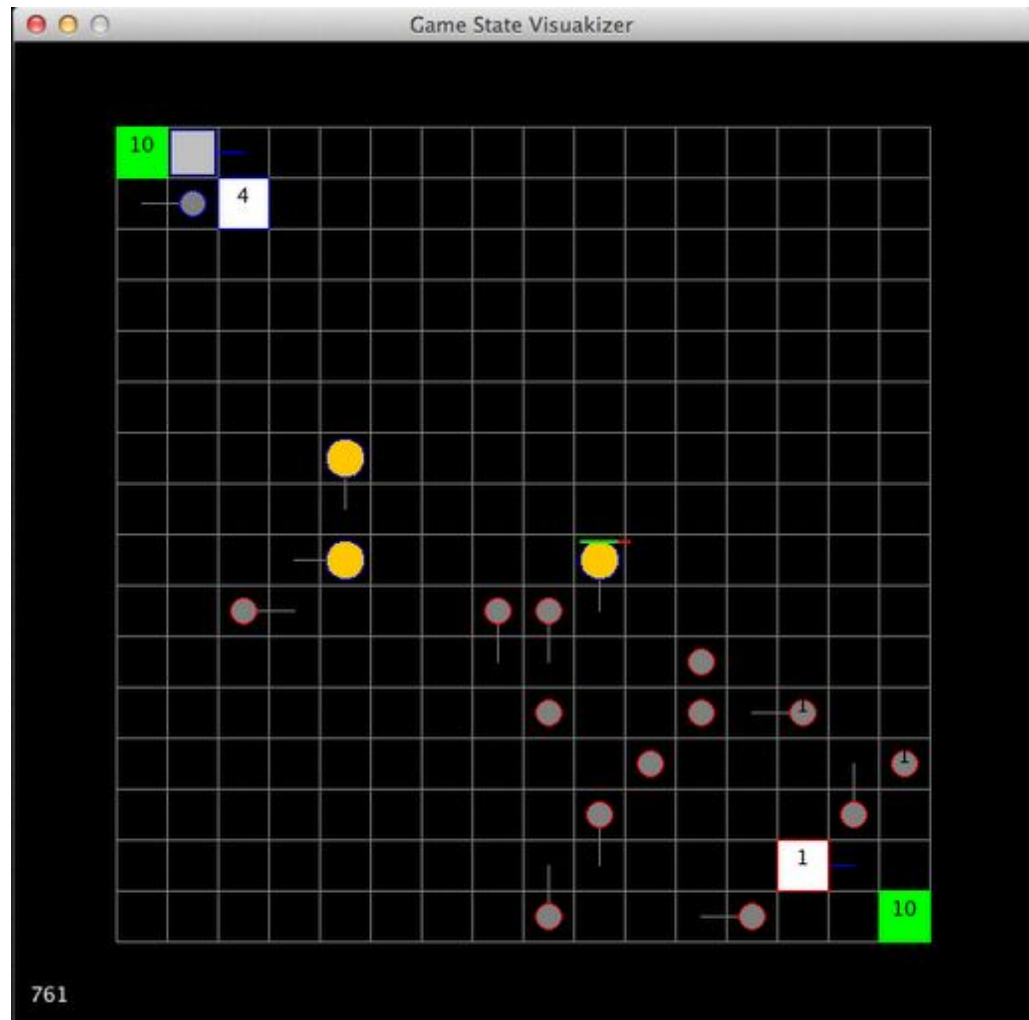
Game Trees - A series of branching game states

- Created from all possible sequences of moves in a board game
- Evolved ANNs evaluate move sequences to determine the best current action
- Tree-Search several board states ahead: focus on long-term outcomes
- Searches a limited number of states due to time limit
- Several tree search algorithms exist
 - Monte-Carlo
 - Minimax
 - Minimax with alpha-beta pruning



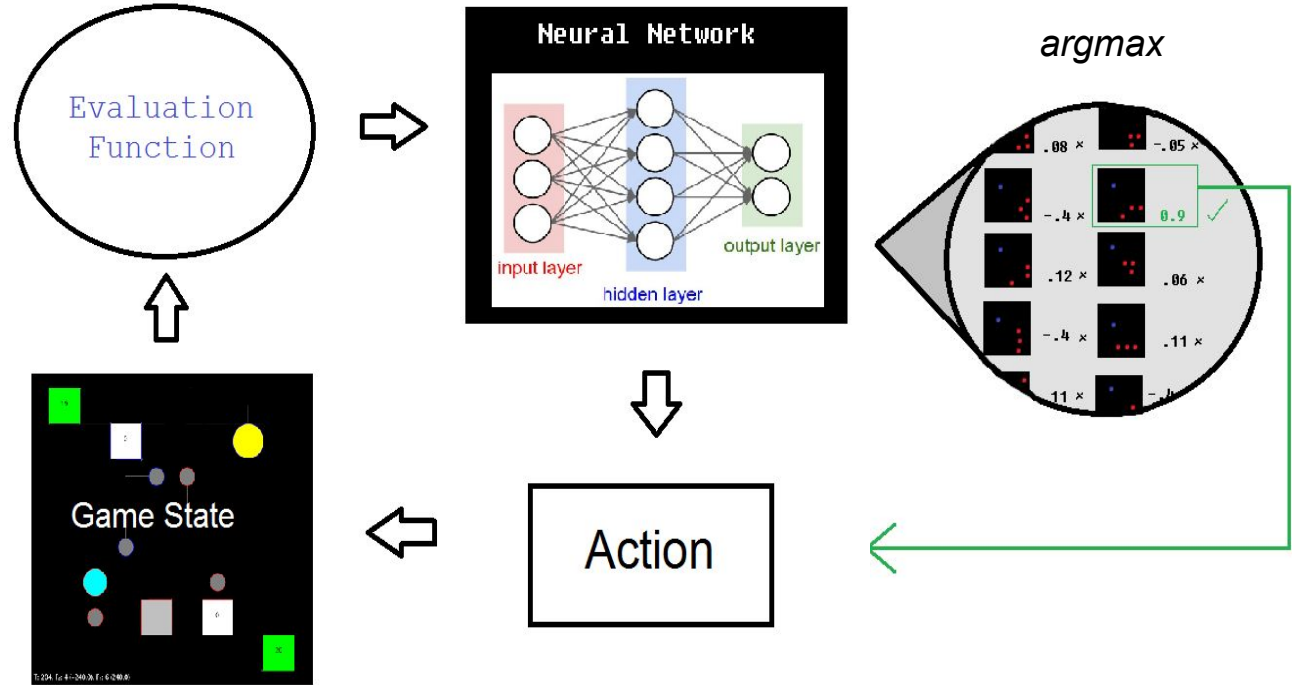
Applying tree-search: MicroRTS

- RTS : Real-time strategy
 - Players act simultaneously
 - Actions cost time
 - Large branching factor
- MicroRTS
 - Much simpler than real RTS
 - Developed as AI benchmark
 - Generic unit classes
 - Forward simulation
 - Know all possible future states
 - Tree-search
 - Adjustable size
 - International AI competition
- Using NN to evaluate game states



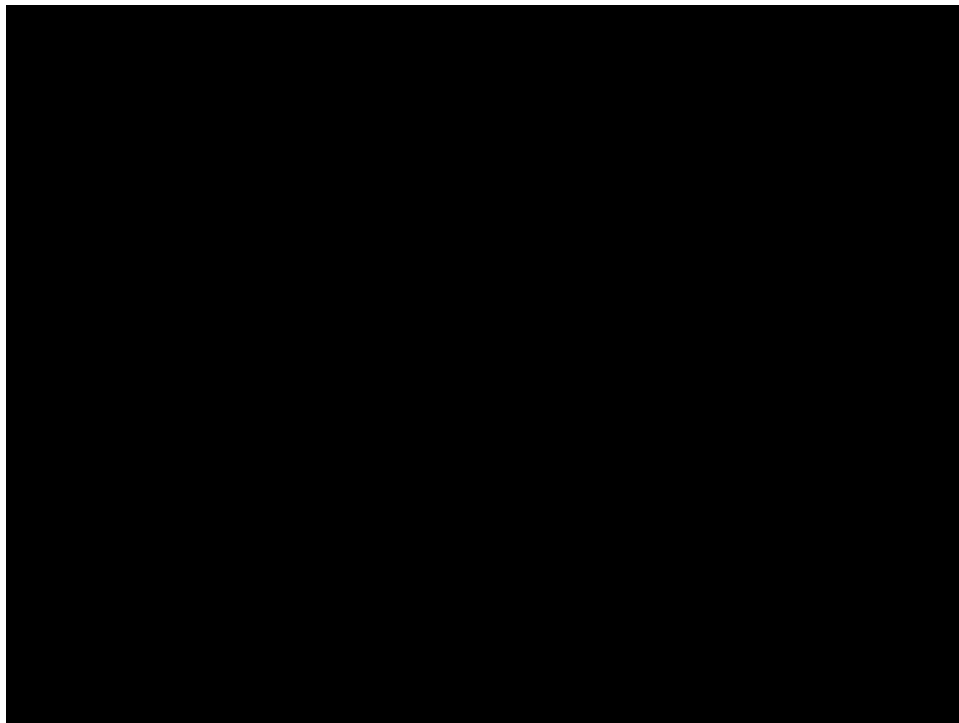
What does it mean to “evaluate a state” in this domain?

- Units' locations
- How many of each unit type
- Available resources
- Remaining base health
- Etc...



Evolved Agent in action!

- Blue player is evolved NN
 - Evolved over night
 - Unsuitable for larger maps
 - Video shows its best match
 - Performance from 21 gens
- Red player is a simple AI
 - Random behavior
 - Biased towards performing a predetermined list of actions
 - Not particularly hard to beat
- Future work
 - Coevolution
 - Beating harder opponents
 - Evolve for longer



Acknowledgements

Southwestern University

Dr. Jacob Schrum

Past SCOPE students: Gabriela Gonzalez, Alex Rollins, & Lauren Gillespie

Santiago Ontañón (MicroRTS developer)

Questions?