Southwestern University

Comparing Direct and Indirect Encodings Using Both Raw and Hand-Designed Features in Tetris

Lauren Gillespie and Gabriela Gonzalez

Abstract

Intelligent agents have a wide range of applications in robotics, video games, and computer simulations. However, fully general agents should function with as little human guidance as possible. Specifically, agents should learn from large collections of raw state variables instead of small collections of hand-designed features. Learning from raw state variables is difficult, but can be easier when agents are aware of the geometry of the input space. Indirect encodings allow agents to take advantage of the geometry of the task, and scale up to large input spaces. This research demonstrates the relative benefits of a direct and indirect encoding using raw or hand-designed features in Tetris, a challenging video game. Specifically, the direct encoding NEAT is compared against the indirect encoding HyperNEAT. Both algorithms create neural networks to play the game, but HyperNEAT makes better use of raw screen inputs, due to its ability to generate large networks that take advantage of the domain's geometry. However, hand-designed features lead to higher scores with both algorithms. HyperNEAT makes better use of hand-designed features early in evolution, but NEAT eventually produces the best overall champions. Since each method succeeds in different circumstances, approaches combining the strengths of both should be explored.

Raw Features

- **Direct Encoding: NeuroEvolution of Augmenting Topologies (NEAT [3])**
  - NEAT is a direct encoding: evolved networks are board state evaluators
  - Direct encoding means networks struggle with many raw, unprocessed inputs
- **Indirect Encoding: Hypercube-Based NEAT (HyperNEAT [2])**
  - HyperNEAT is an indirect encoding: evolved networks construct board state evaluators that are applied to the game (Fig. 1)
  - Indirect encoding allows these agents to be geometrically aware of the domain
  - This geometric awareness allows HyperNEAT-evolved networks to make sense of raw features and perform well using them [4]

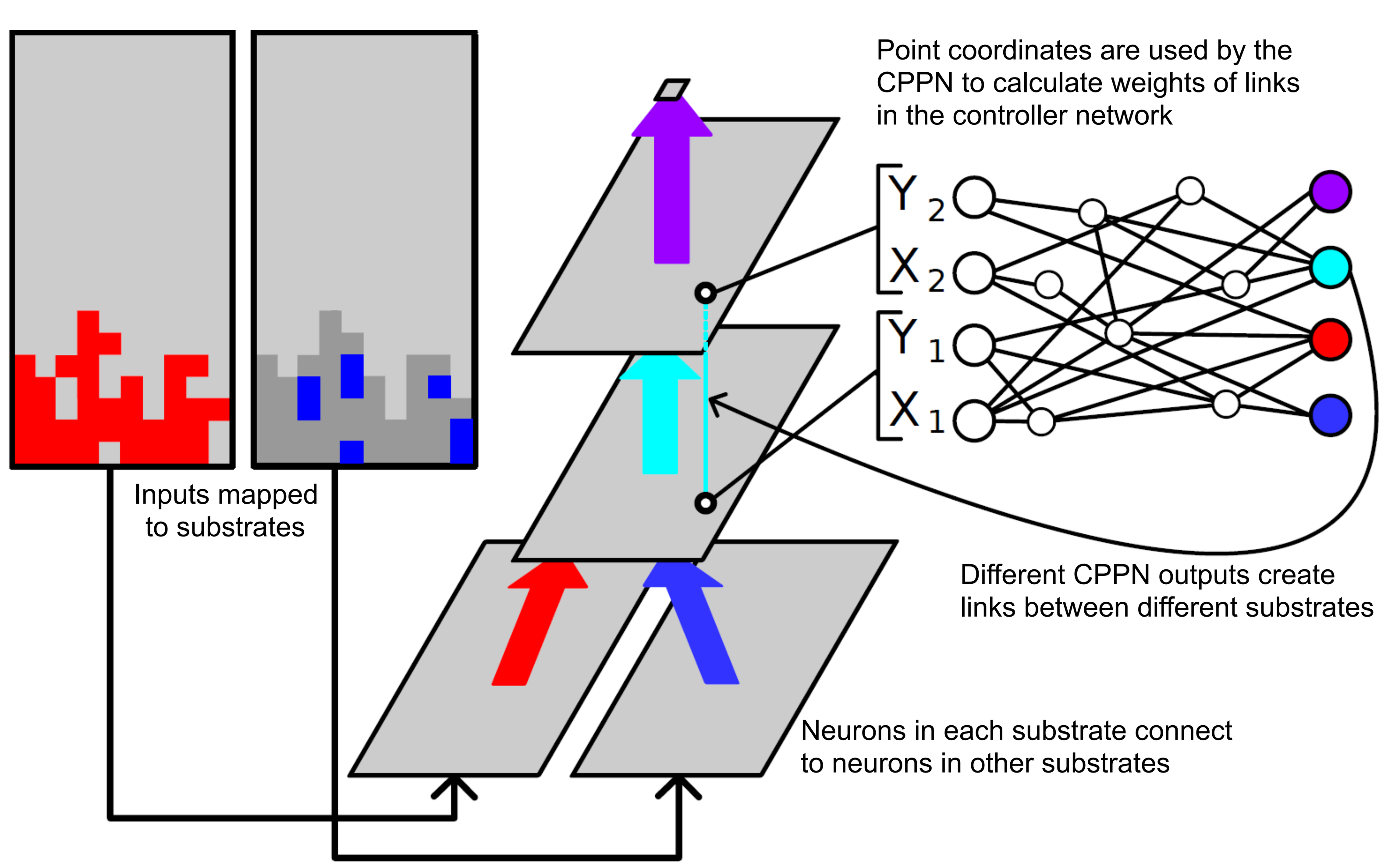

Figure 1. HyperNEAT-Encoded Networks Using Raw Screen Inputs

Hand-Designed Features

- **NEAT and Hand-Designed Features**
  - Unlike with raw features, NEAT performs well with hand-designed features
- **HyperNEAT and Hand-Designed Features (Fig. 2)**
  - HyperNEAT also performs well with hand-designed features
  - Geometric awareness is less important

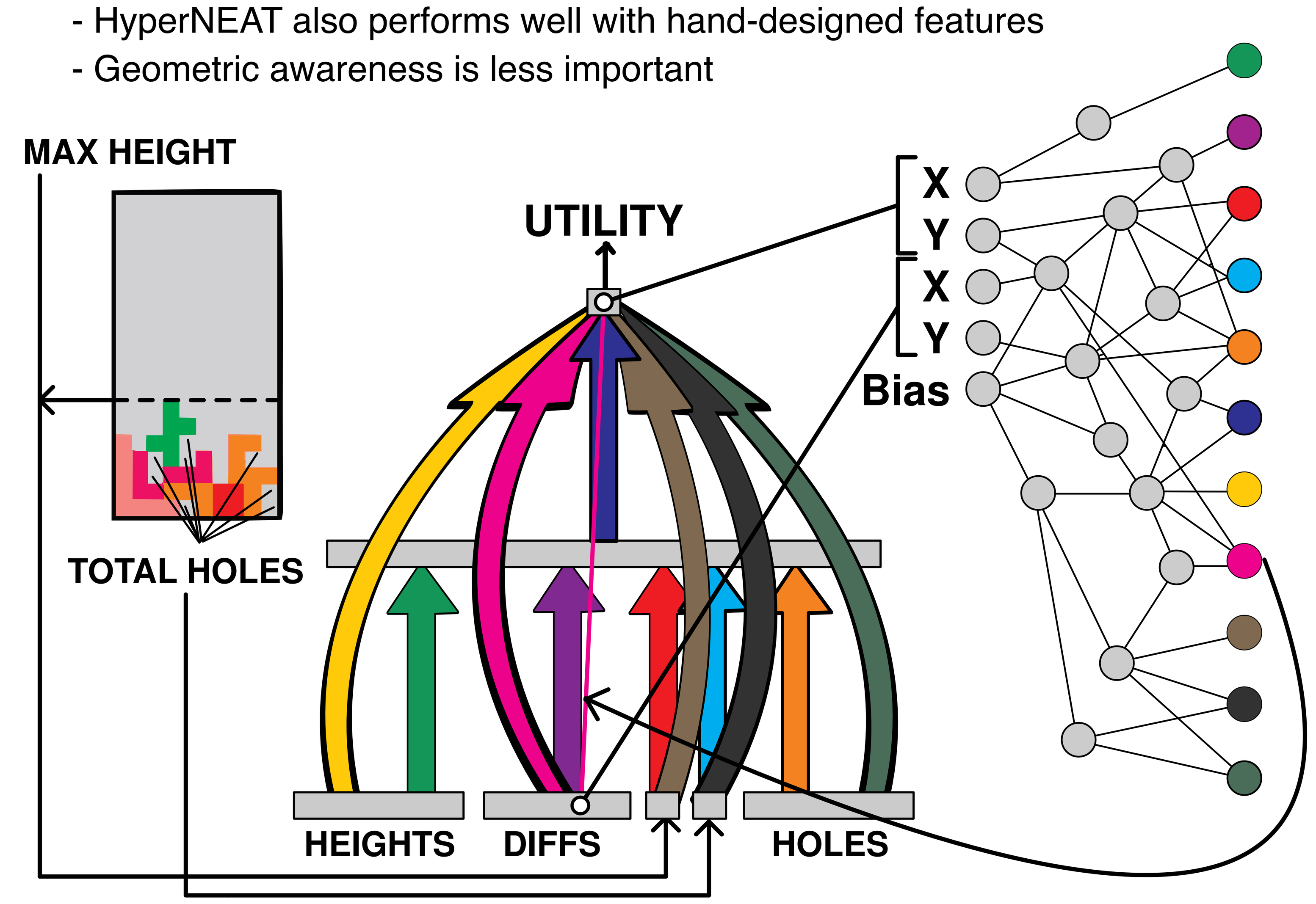

Figure 2. HyperNEAT-Encoded Networks Using Hand-Designed Features

Experiments

- Network state evaluators query the after-state of each possible piece placement
- Networks output a single utility score between 0 and 1 for the given piece position
- Agent chooses the piece placement with the highest utility score
- Raw inputs are fed to the networks as two board states:
  - One board for the block positions with a 1 for a block and 0 for no block
  - The other board represents hole positions as -1 and all other positions as 0
- Hand-designed inputs are 17 different features consisting of the heights of each column, the height of the tallest column, the total number of holes, the difference in height between each column and the number of holes per column [1]
- Experiments consisted of 30 runs of 500 generations using both encodings and both types of features with populations of 50 parents and 50 children per generation.
- Each agent was evaluated 3 times due to noisiness in the domain
- The fitness scores were calculated by taking the average score across the 3 runs along with the average time the agent survived (multiobjective optimization)

References

[1] C. Thiery, and B. Scherrer. Building Controllers for Tetris. In International Computer Games Association 32(1), 2009.

[2] K. O. Stanley, D. D'Ambrosio, and J. Gauci. A Hypercube-Based Indirect Encoding For Evolving Large-Scale Neural Networks. In Artificial Life Journal 15(2) MIT Press, 2009.

[3] K. O. Stanley and R. Miikkulainen. Evolving Neural Networks through Augmenting Topologies. In Evolutionary Computation Journal 10(2) MIT Press, 2002.

[4] M. Hausknecht, P. Khandelwal, R. Miikkulainen, and P. Stone. HyperNEAT-GGP: A HyperNEAT-based Atari General Game Player. In Genetic and Evolutionary Computation Conference. 217–224.

Results/Discussion

- **Raw Features**
  - HyperNEAT significantly outperformed NEAT (Fig. 3)
  - HyperNEAT networks performed decently using raw features
  - Median game score of over 150
  - NEAT unable to learn any intelligent behavior: game score always below 10

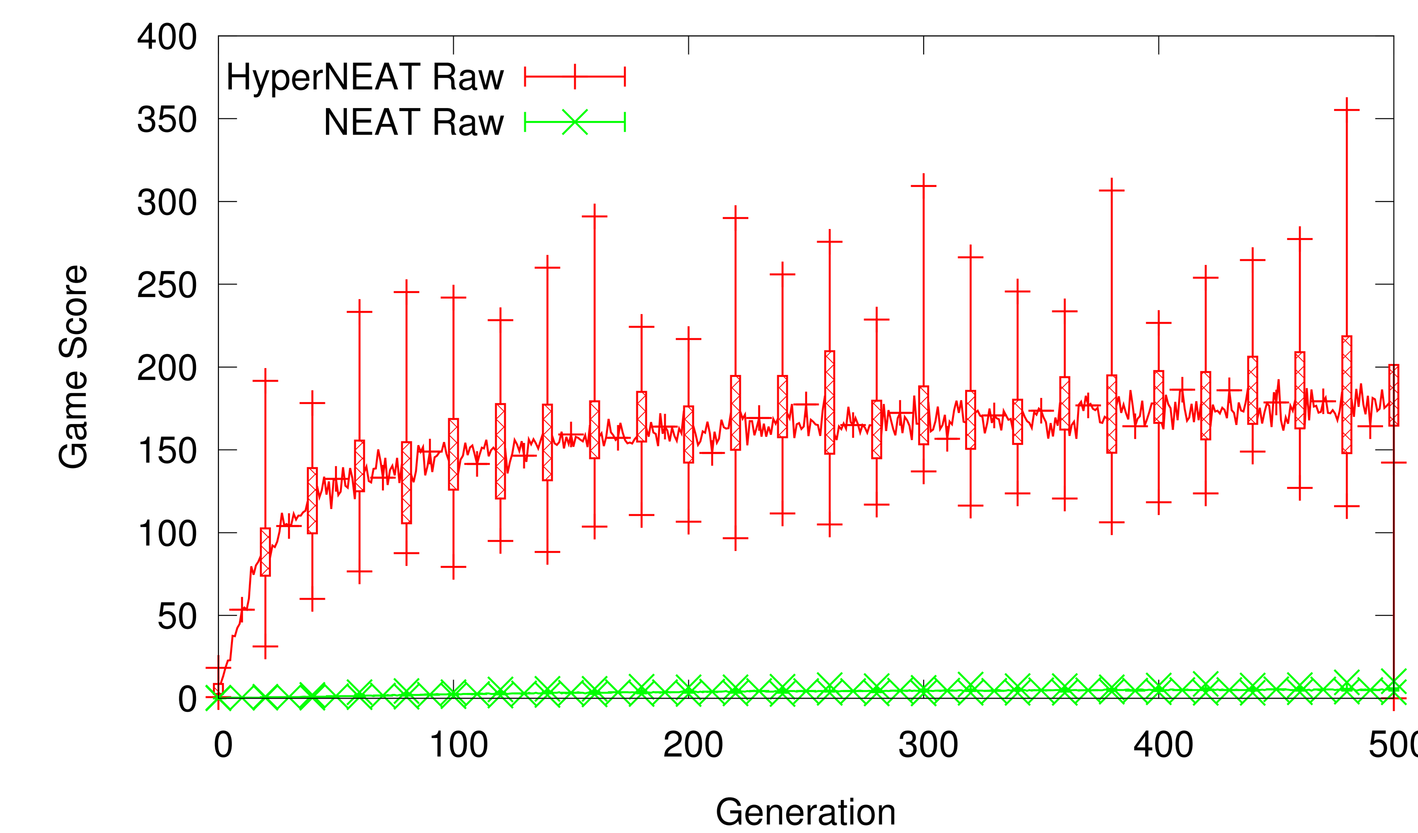

Figure 3. Comparison of game score of champion agent from each generation using both HyperNEAT and NEAT with raw features. Results presented as periodic box plots.

- **Hand-Designed Features**
  - NEAT performed as well as HyperNEAT (Fig. 4)
  - For the first 300 generations, HyperNEAT outperformed NEAT
  - HyperNEAT performance leveled off while NEAT continued to improve
  - NEAT's median score actually surpasses HyperNEAT's at generation 500

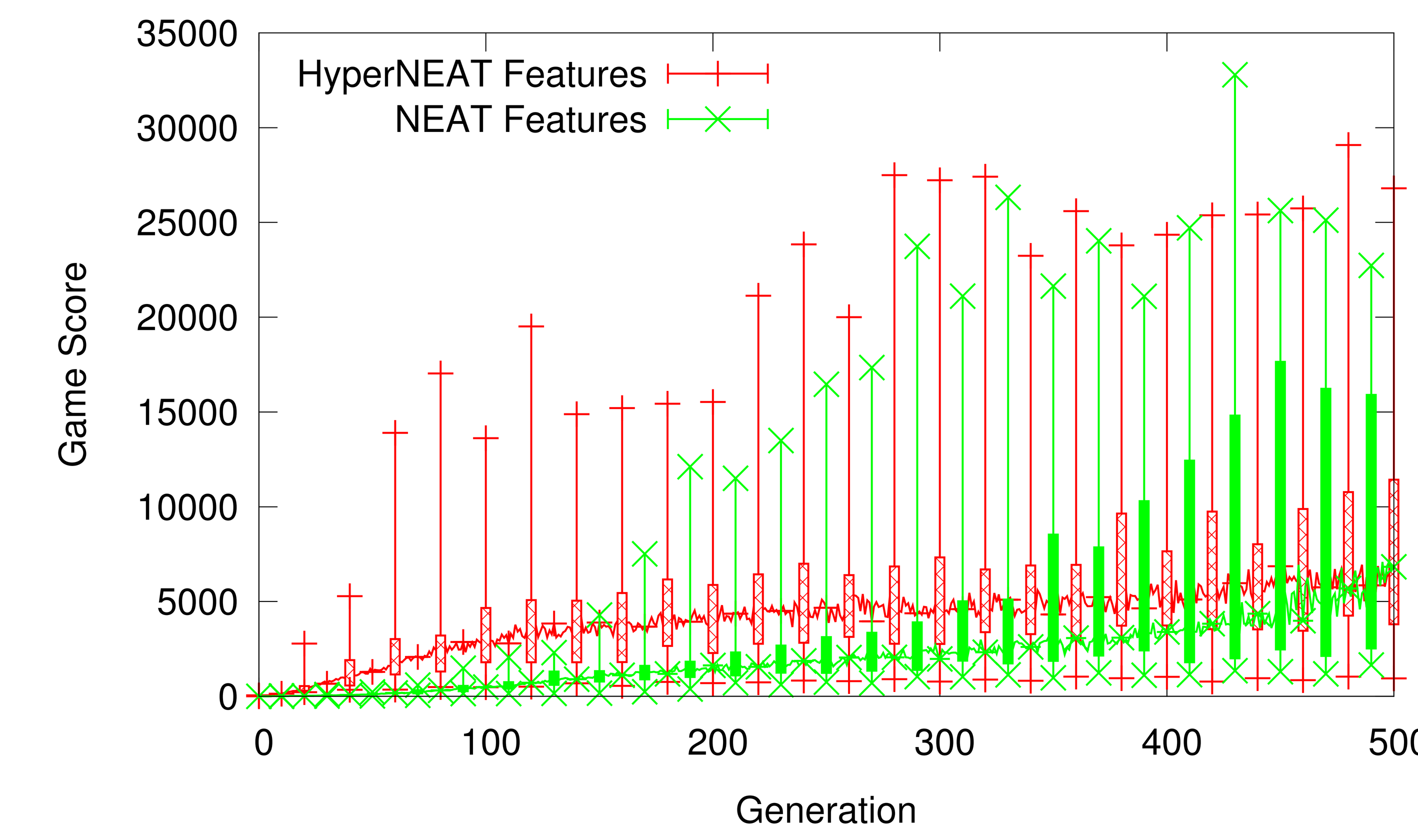

Figure 4. Comparison of game score of champion agent from each generation using both HyperNEAT and NEAT with hand-designed features. Results presented as periodic box plots.

- The geometric awareness of HyperNEAT-encoded networks gave them the ability to process raw features and outperform NEAT-evolved networks
- However, with hand-designed features geometric awareness was not as crucial and therefore NEAT and HyperNEAT networks performed similarly
- Videos of champion agents: southwestern.edu/~schrum2/SCOPE/tetris.html

hhmi Howard Hughes Medical Institute