SOUTHWESTERN UNIVERSITY

Evolving Custom Convolutional Neural Network Architectures in SZ-Tetris

Devon Fulcher

Introduction

Neural networks are a leading technology in approximating optimization problems. However, their efficacy relies on highly designed architectures. Various custom architectures are tested in the domain of SZ-Tetris.

SZ-Tetris

SZ-Tetris is a harder variant of Tetris where only the S and Z blocks are present.

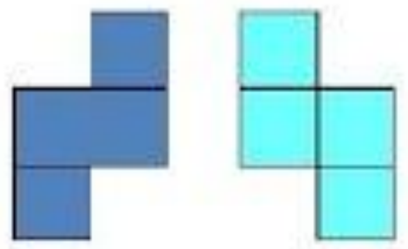

Figure 1: S and Z blocks

Convolutional Neural Networks

Convolutional neural networks are used to encode the locality of data. They are utilized on data where the spatial organization of the input is relevant. Previous work [1] has shown that they are more effective than fully connected layers in Tetris because the relative location of blocks is relevant for skilled play. In Tetris, convolution is implemented by connecting adjacent layers via a receptive field, a local area of connectivity.

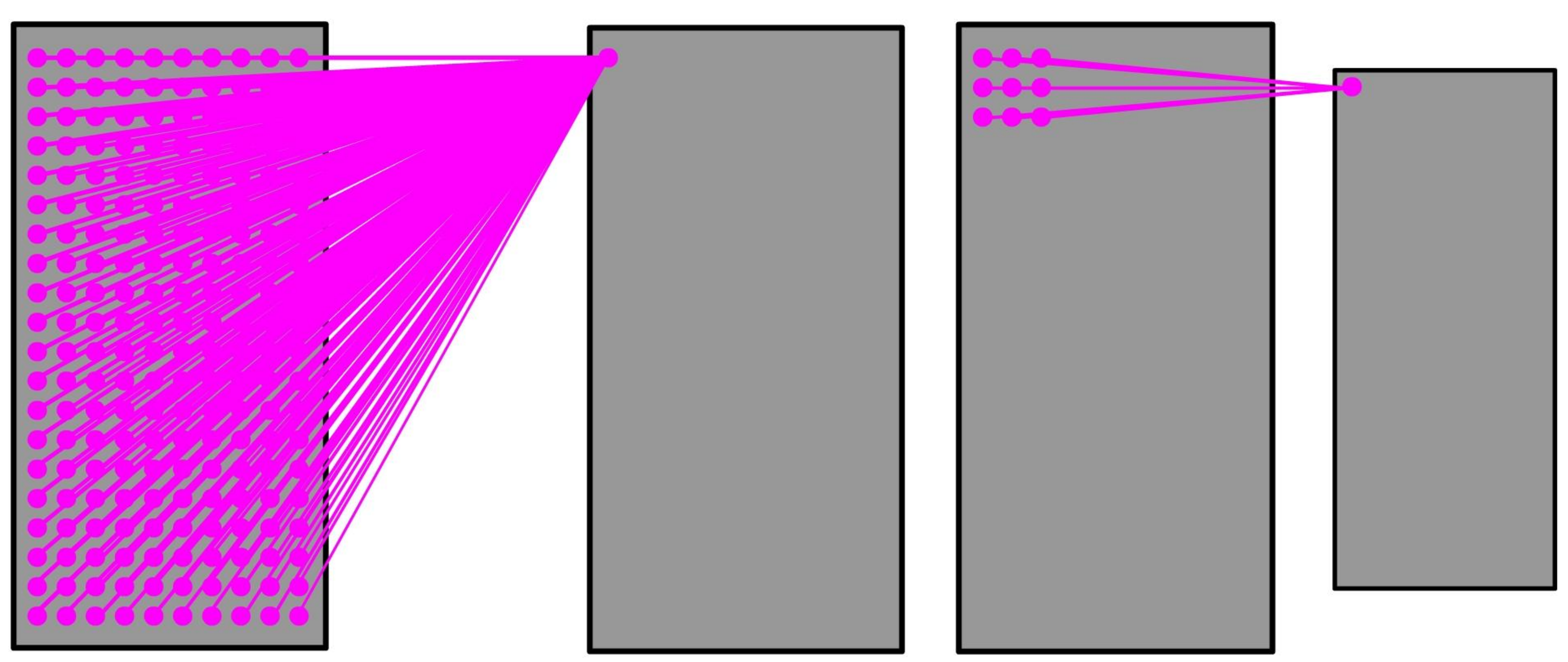

Figure 2: A densely connected layer (left) and a convolutionally connected layer with a 3x3 receptive field (right).

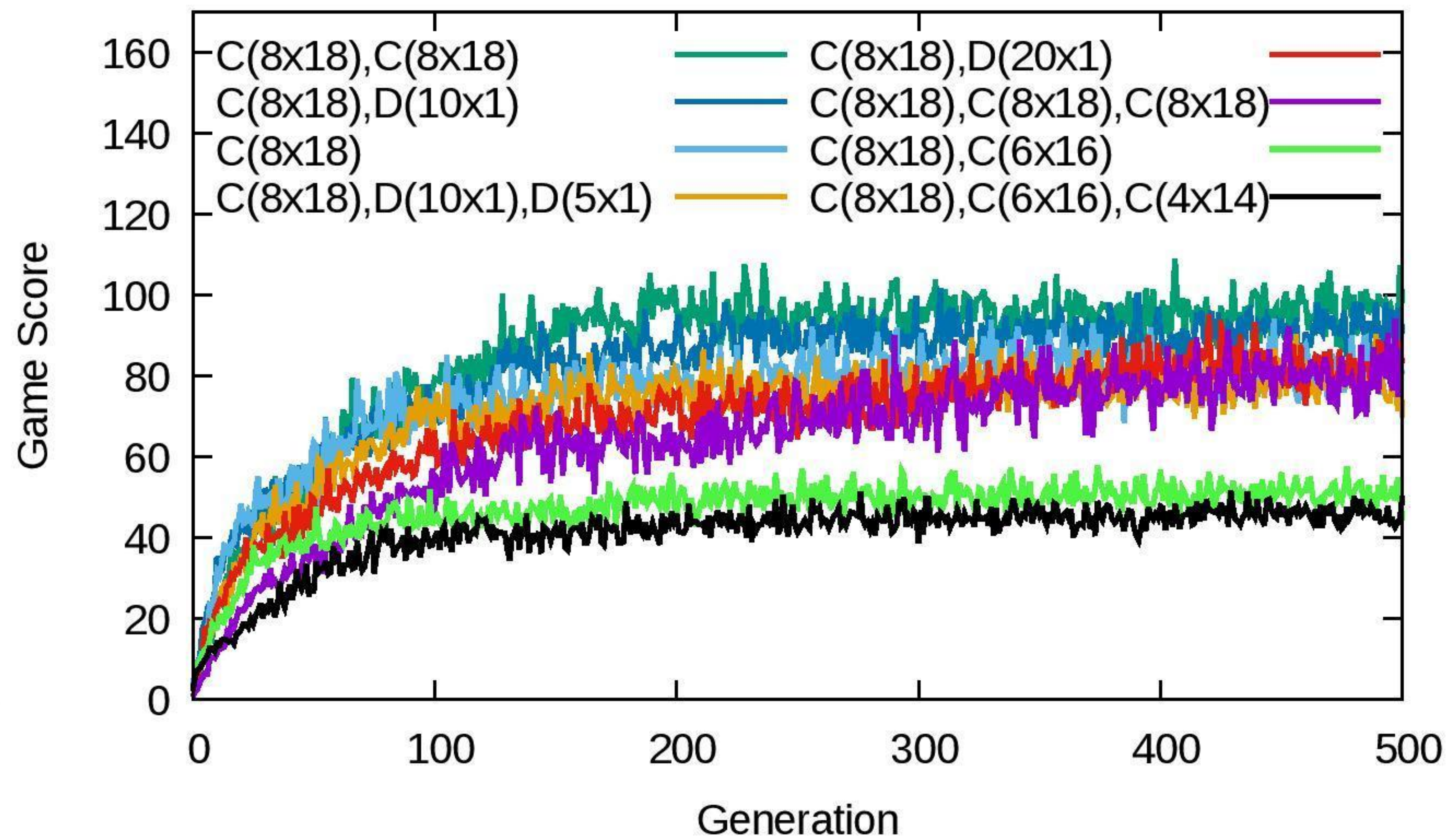

Figure 3: Median scores in SZ-Tetris across 10 runs. C(WxH) represents a convolutional layer and D(WxH) represents a densely connected layer with width W and height H. With the addition of an identically sized convolutional layer and additional trainable weights C(8x18),C(8x18) had the best performance. C(8x18),D(10x1) was a close second by down sampling the dimensionality of trainable weights.

HyperNEAT

Variants of the HyperNEAT [2] algorithm with convolution are tested. In HyperNEAT, the edge weights of a large neural network are determined by a smaller compositional pattern producing network (CPPN) that is evolved with the NEAT algorithm. The NEAT algorithm models evolution through natural selection to favor successful networks and random mutation to create new networks from successful networks. CPPNs use the positions of neurons in a network to generate weighted links between neurons that trasmit information. This indirect encoding emulates the generation of complex organisms from compact DNA.

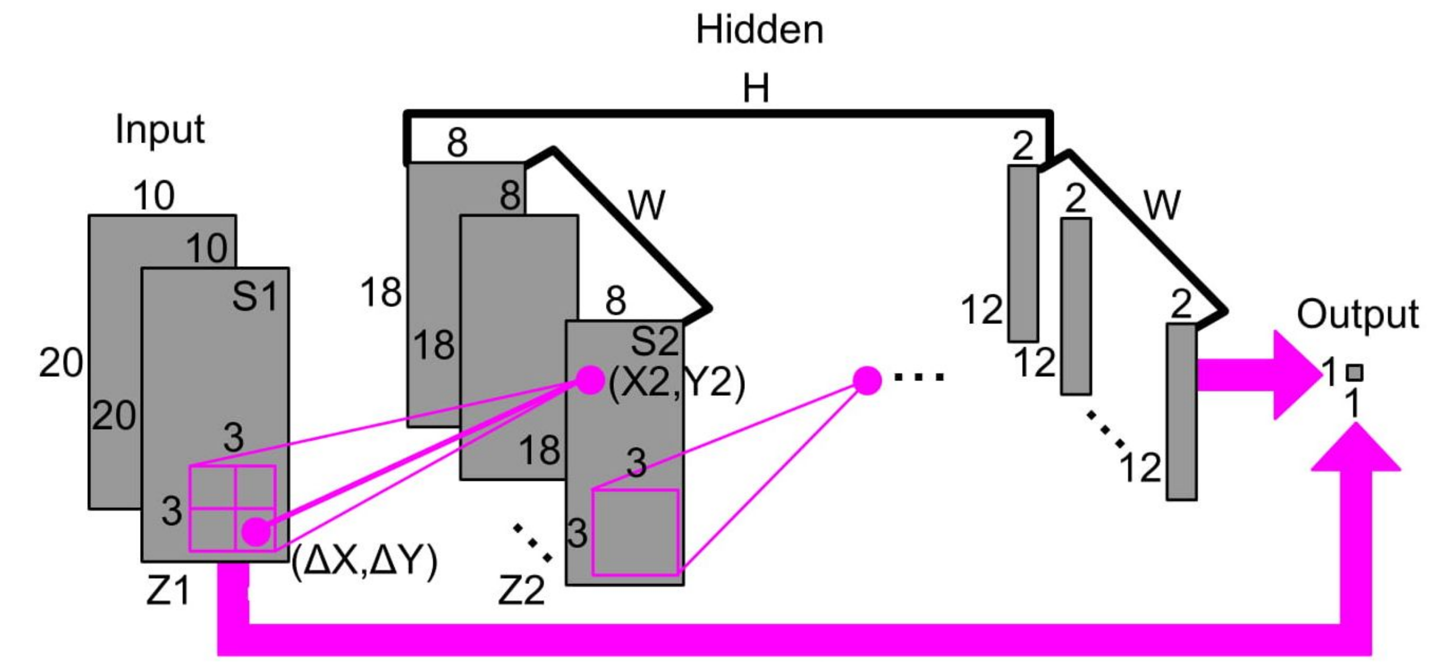

Figure 4: A HyperNEAT architecture with convolution

Custom Architectures

Scores were improved in SZ-Tetris with the addition of layers with reduced dimensionality and layers with equivalent structure as prior layers but with additional evolvable edge weights. This addition of evolvable weights shows that a layer with receptive field size 1x1 can be effective.

Conclusion

This work has shown that effective deeper architectures are possible in HyperNEAT with careful design considerations. Even deeper neural networks are typically more successful than shallow ones but designing effective deep HyperNEAT architectures is very laborious and successful architectures beyond three layers could not be successfully evolved. Achieving very deep performant networks with HyperNEAT is a goal of future work.

References

[1] Jacob Schrum. 2018. Evolving Indirectly Encoded Convolutional Neural Networks to Play Tetris With Low-Level Features. In Genetic and Evolutionary Computation Conference, July 15–19, 2018.

[2] Kenneth O. Stanley, David D'Ambrosio, Jason Gauci. 2009. A Hypercube-Based Indirect Encoding for Evolving Large-Scale Neural Networks. *Artificial Life,* 2009.