Exploring Hyperdimensional Computing Recall of Reactive Behavior through 2D Obstacle Map

Youbin Kim, Alisha Menon, Braeden Benedict

Berkeley | EECS
Electrical Engineering and Computer Sciences

VS265 - Fall 2020

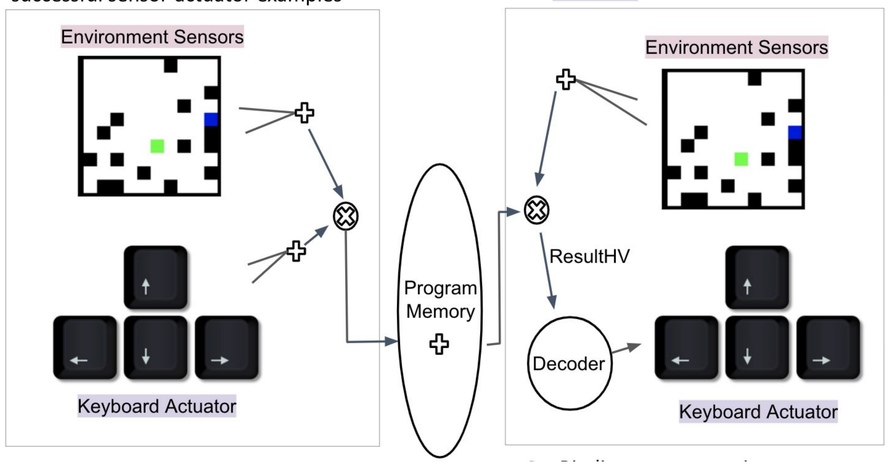
Overview/Background

This project focuses on exploring the idea of Hyperdimensional Computing (HDC) through the lens of recalling reactive behavior with the specific focus on navigating a 2D map that contains obstacles. HDC is based on the idea that human brains do not perform inference tasks using scalar arithmetic, but rather manipulate large patterns of neural activity. These same principles can be and have been previously preliminarily leveraged for recall of reactive behavior with limited results [1]. We hope this project will showcase the potential of HD recall and provide insight into adapting the algorithm to more complex real-world problems.

HDC Recall Of Reactive Behavior

Training: Form an associative memory from successful sensor-actuator examples

Query: Use conditions to retrieve the actuators

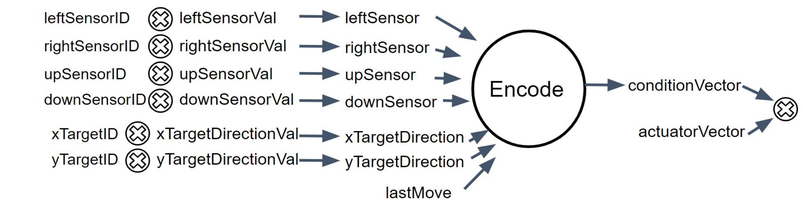


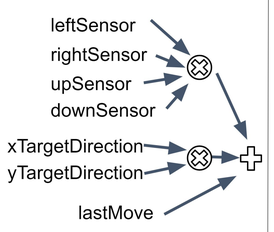
: Bundling creates a vector similar to the inputs

: Binding creates a unique vector different from the inputs

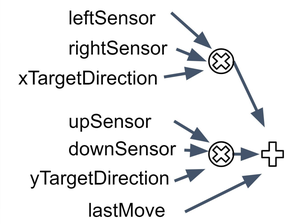
Using the bundling operation and the binding operation, the various sensor and actuator modality signals can be fused to create a set of vectors over the course of the training period that are bundled together and stored in the "program memory" as a single hypervector.

Sensor encoding schemes

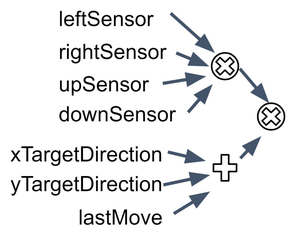




Method 0: Sensor modalities



Method 1: Directional representations



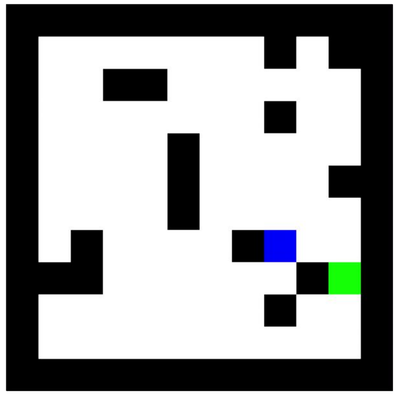
Method 2: Constraints vs goals

During training, condition vectors are compared with a vector of previously added conditions. If dissimilar, the new condition is added to the stored condition vector and the new example is added to the program vector. This prevents training on repeated conditions.

During testing, the observed sensor values are used to unbind an actuator value from the program, which is selected probabilistically based on Hamming distance.

Training & Testing Environment

The game world is a 10X10 grid with randomly placed obstacles (black). The blue square represents the current position and the green is the goal location.



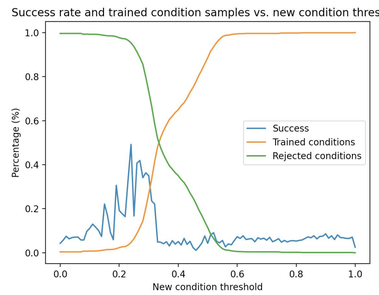
On the left, we show learned behavior using HDC recall in a new environment.

Training is performed on human game-play and then tested on randomly generated environments.

Analysis & Results

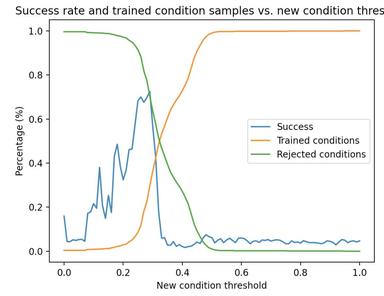
All three methods of encoding were analyzed for success rate across similarity thresholds

	Method 0	Method 1	Method 2
Success Rate	49%	70.4%	89%
Threshold	0.24	0.29	0.06



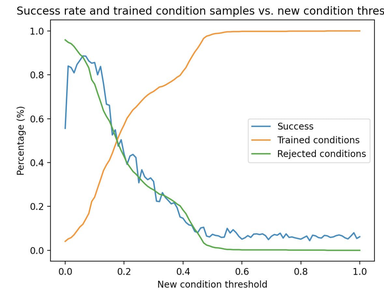
Method 0 worked well only for scenarios that had unobstructed paths to the goal

It demonstrated a peak success rate of 49%



Method 1 had improved ability to avoid obstacles by combining sensor and goal information for each direction

It demonstrated a peak success rate of 70.4%



Method 2 projected each input combination into a pseudo-orthogonal vector more specific to each of the sensors and included probabilistic actuators

It demonstrated a peak success rate of 89%

Threshold is a measure of similarity of vectors, thus too high would result in all 1001 vector being trained, too low would result in no vectors being trained. For the third method, a change in any sensor would result in a different pseudo-orthogonal vector, hence the lower optimal threshold to more strictly allow only very different vectors

[1] P. Neubert, S. Schubert, and P. Protzel, "Learning vector symbolic architectures for reactive robot behaviours," in Proceedings of the International Conference on Intelligent Robots and Systems (IROS'16) and the Workshop on Machine Learning Methods for High-Level Cognitive Capabilities in Robotics, 2016.