



Automating Pong with an Artificial Neural Network

By: Nick Bonavia & Matt Liddy

Goal

- Train a Neural Network
Approach to play pong
- Play against the Collective
Learning System from the
previous project



What is a Neural Network?

- *“A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.” - MIT*
- In our case, we need to train our network to play pong.



Approach

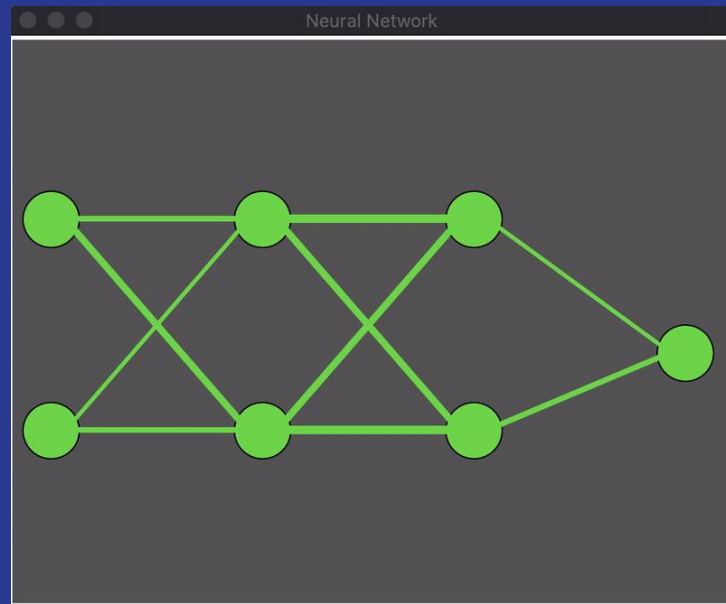
1. Used Python with numpy for training NN
2. Generated Training set
 - a. Input: Ball (X,Y) coordinates
 - b. Output: AI (Y) coordinate
 - c. Successful block if AI matches Y coordinate of Ball
 - d. Normalized training set in range (0.0, 1.0)
3. Implemented Forward-Prop in Godot
 - a. Used trained weights from Python program
 - b. Implemented our own basic Linear Algebra library to simplify Forward-Prop to one line of code



Training

- Learning Rate $\alpha = 0.01$
- Topology:
 - 2 Hidden Layers
- Weights Randomly Initialized
- Backpropagation
 - $l3_delta = l3_error * \text{Sigmoid}'(l3)$
 - $l2_error = l3_delta \cdot \text{syn2}^T$
 - $l2_delta = l2_error * \text{Sigmoid}'(l2)$
 - $l1_error = l2_delta \cdot \text{syn}^T$
 - $l1_delta = l1_error * \text{Sigmoid}'(l1)$
- Weight Updates
 - $\text{Syn}_n = \text{layer}_n^T \cdot (\text{layerDelta}_{n+1} * \alpha)$

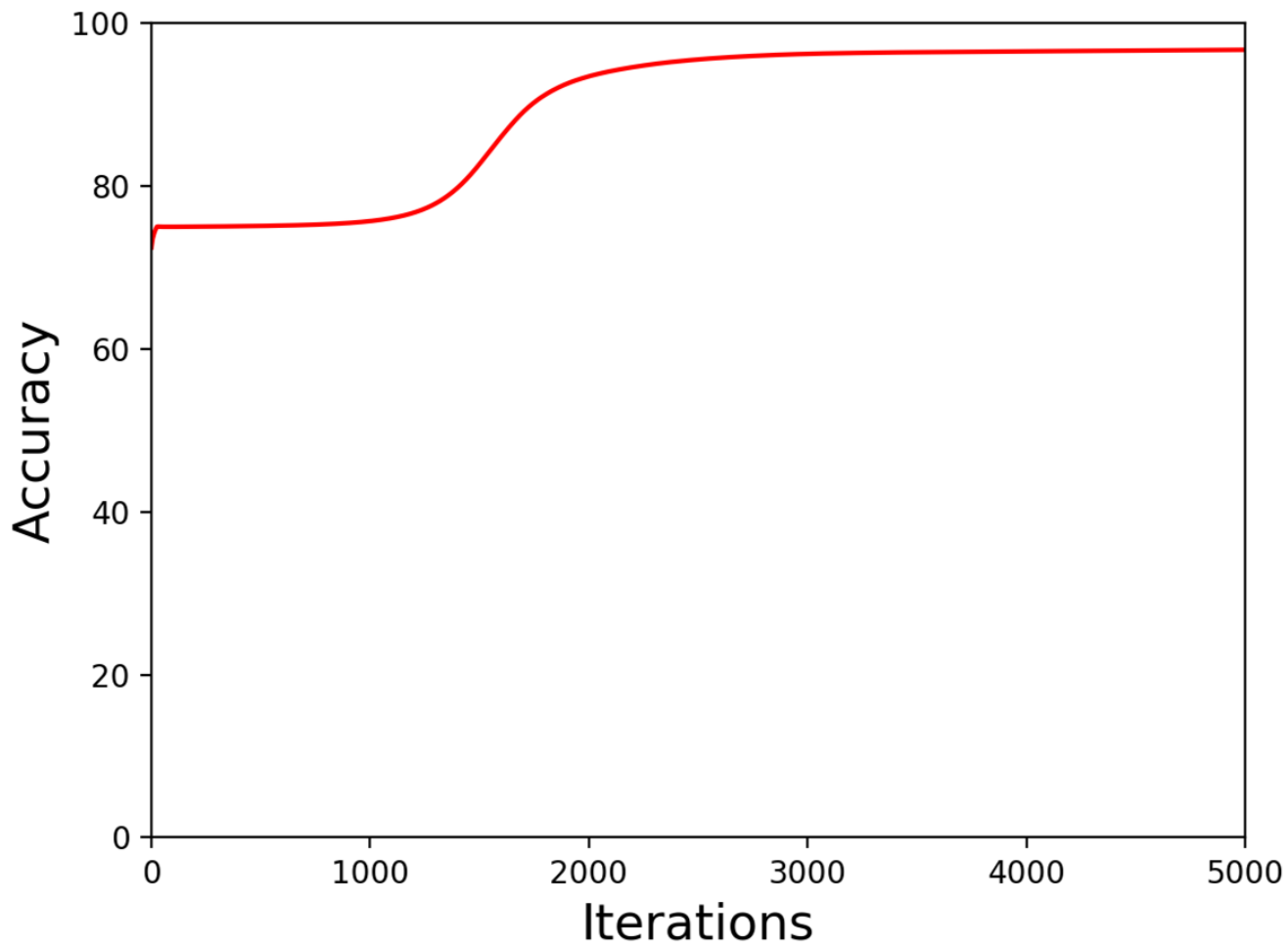
Training Network Topology



Line thickness represents weight

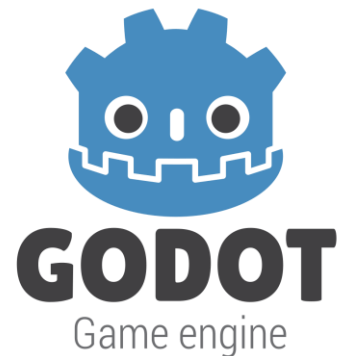
Results

- 99%+ Accuracy
- 100,000 iterations
- 5 min of training time



Technology

- Open source game engine
- Easily create 2D and 3D games
- Supported Languages
 - GDScript
 - C++
 - C#
 - VisualScript
- Multiplatform
 - Windows, Mac, *nix
 - IOS, Android
- Stable and well supported
 - Initial Release January 2014
 - Received \$250k Epic grant in February 2020



Live Demo

