

Pong Game with Neural Networks (A.I.)

1.0 About

This project is to create a simple pong game played using a neural network. In this game user can play pong with the neural network (AI) which plays as the opponent. This contains two programmings one to train the neural network and other is for the play with it.

2.0 Implementation

Here I have used a neural network with five input neurons, one output neuron and one hidden layer of ten neurons. The inputs are the x and y coordination and x and y velocities of the ball and the current location of the paddle. And the network outputs the distance of the paddle to be moved according to hit the ball.

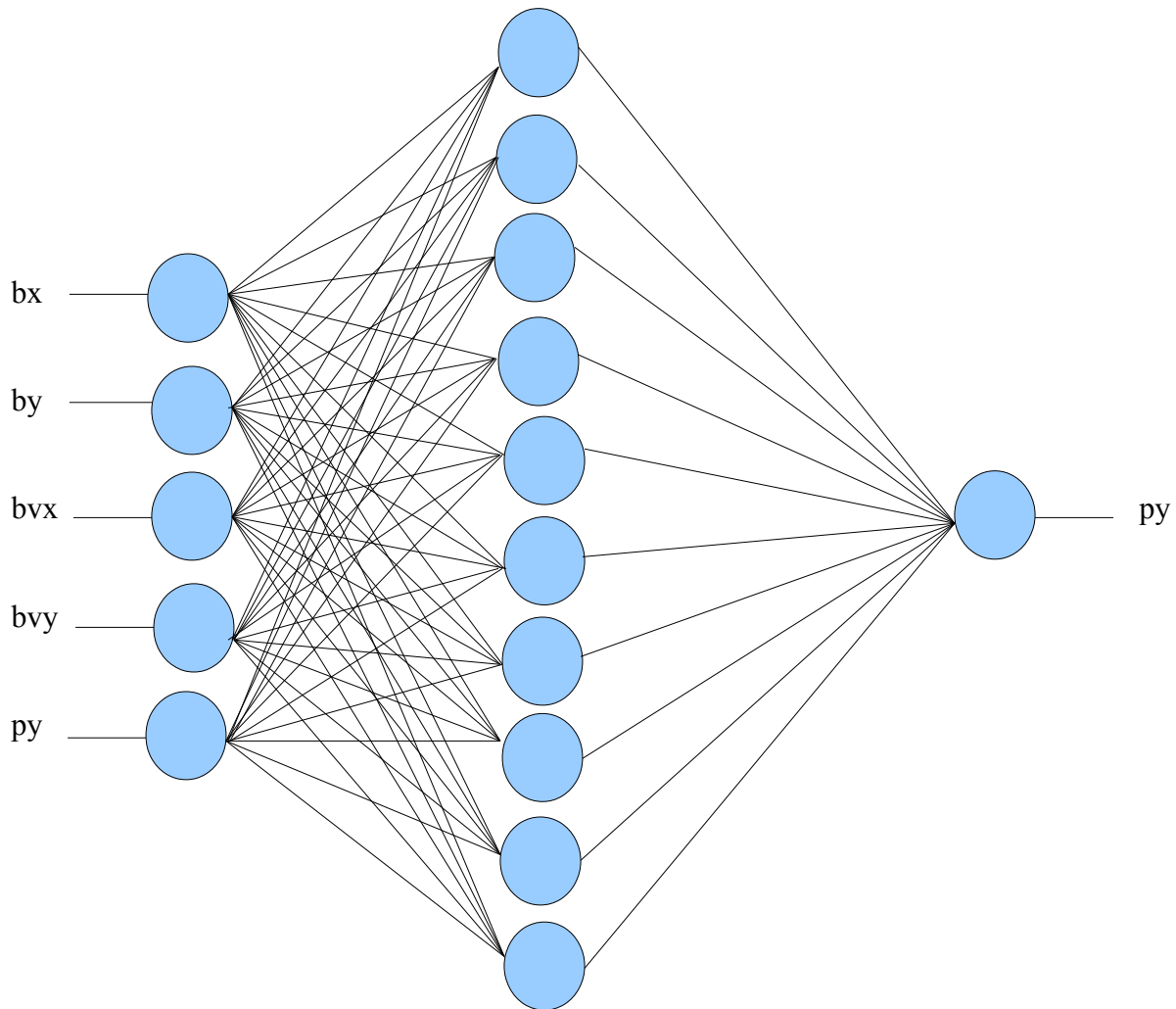


figure 2.0 : structure of the neural network

The network uses back propagation with 0.2 learning rate to learn. The first program (trainer) uses two networks and make them to play with each other to train them. In every miss the the error is fad to the network to tweak the weights. The training process can be done faster by reducing the time between two frames. But some times the training process stops because the ball and paddles are sometimes get stuck in a loop (ball ans paddles keep going in same path over and over again). So you have to keep resetting the ball (get the ball to the center and give it a random x and y velocities) while running in high speed.

When the training process is done (when the paddles seems to act with low miss count) the weights of the two neural networks are recorded in to two files, “p1.data” and “p2.data”. Then the second program uses “p1.data” file to set its weights. In the second program you can play with the AI using mouse.

3.0 Compiling and Running

You need 'openGL' and glut to compile . Install them by running,

```
sudo apt-get install freeglut3-dev  
sudo apt-get install libgl1-mesa-dev
```

on terminal. Then use make commend to compile.

Then to use this first you have to use trainer to train the network. Run it using terminal using,

```
./[executable]
```

command. The trainer has following keyboard commands.

- p - pause
- r - return the ball to the center and give a random x and y speeds
- z - increase the time between two frames
- x - decrease the time between two frames
- w - record the weights in to a file.
- esc - close

When the trainer is started the ball will be running slow. In this way it will take long time to train the

network. So the speed has to be increased by pressing 'x'. When it is running in a moment the changing of the miss count will be stopped. Then the 'r' must be pressed. Then it will be increasing again. We need to keep doing it until the miss count becomes more than '10,000'. Then the weights must be recorded using 'w'. Some trained weights are already in the given files ("p1.data" and "p2.data").

Then use the second program to play. It will read the "p1.data" and load the weights in to it. Commands of the game are,

- p - pause
- r - return the ball to the center and give a random x and y speeds and reset the misses and hits
- k - return the ball to the center and give a random x and y speeds without changing scores.
- z - increase the time between two frames
- x - decrease the time between two frames
- esc - close

And use mouse to move the right paddle. This program is also doing back propagation every time it misses the ball.

4.0 Screen Shots

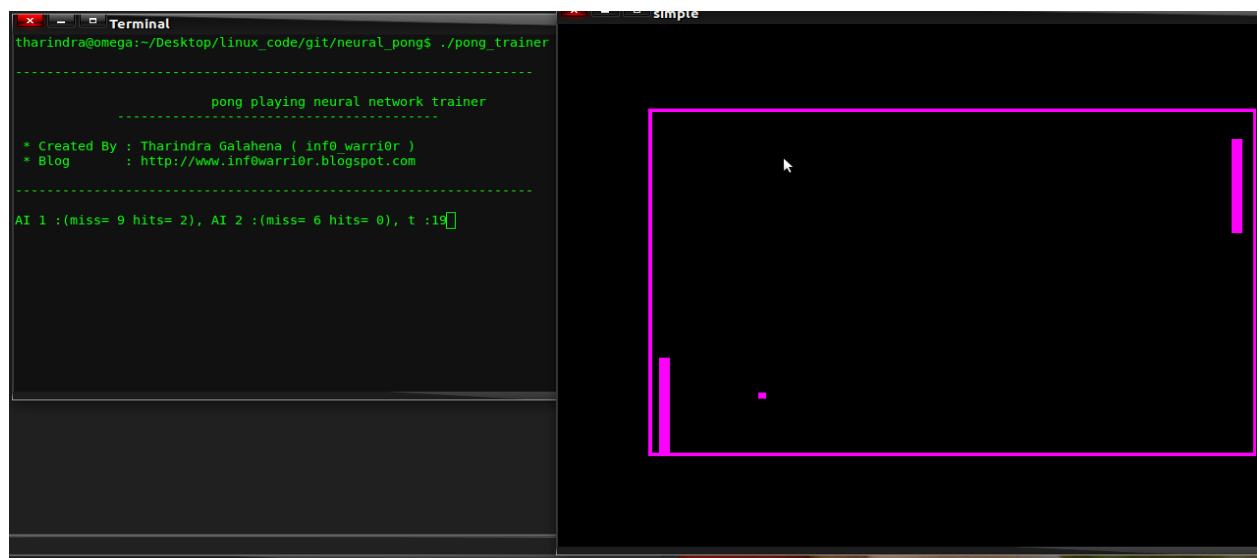


figure 4.0 : neural network trainer

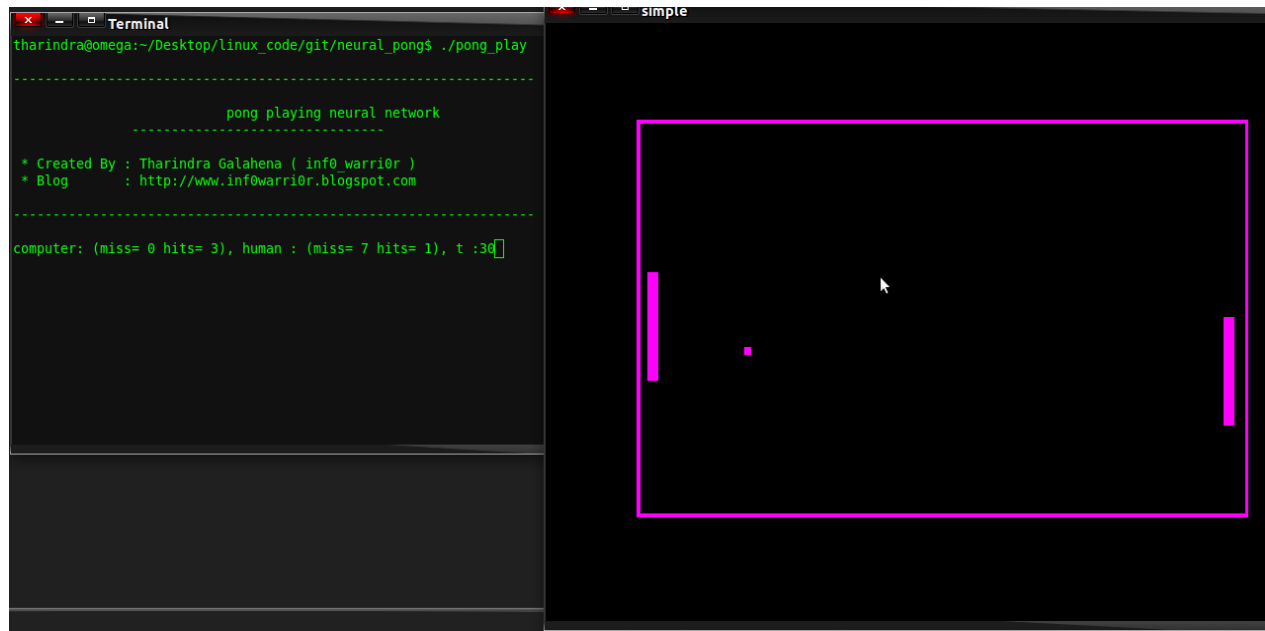


figure 4.1 : the actual pong game

5.0 More Info

- tcg.galahena@gmail.com
- <http://www.info0warri0r.blogspot.com>

6.0 License

Copyright 2012 Tharindra Galahena

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with This program. If not, see <http://www.gnu.org/licenses/>.