Neural Network

Dawit Abay & Alif Dastegir , Artificial Intelligence , Dr. Yanqing Zhang

Code server: http://codd.cs.gsu.edu/~Dabay1//Neuralnetworks/index.html

Abstract

This paper introduces the Neuroevolutionary of Augmenting Topologies method for evolving artificial neural networks to play game 2020. Over the course of several hundred generations, the network evolves and learns how to play game.

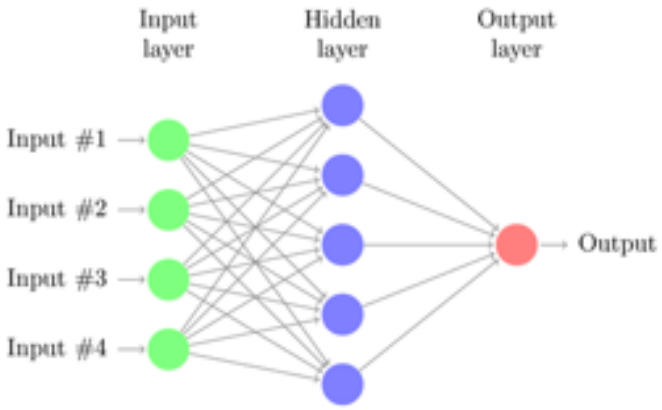
I. INTRODUCTION

What are Neural networks are a set of algorithms, modeled loosely after the human brain, that is designed to recognize patterns? They interpret sensory data through a kind of machine perception, labeling or clustering raw input. Neural networks are a technique for building computer programs that is loosely based on the way we think the human brain works. You start with a collection of software neurons. Each neuron in a brain has a bunch of inputs and one output. When enough electrical signals arrive at the inputs, the neuron will activate and send a signal out of the output. The reason I picked Neural networks is to show how to build a game using a Neural network. Artificial neural networks or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. I come up with an idea to implement a neural network program that learns how to play Flappy Bird game.

II. RELATED PUBLICATION

The neural network needs some data to learn on. Input data is very important part of machine learning. If you have a huge amount of data, you can achieve great results even if an architecture of your network is not good. That's why companies like Google are trying to gain all information they can get from their users of course not because they have bad architectures but because really big data is precious. So, we need to generate some data. You can sit and play as many games as you can, but it is always good when you can generate data automatically from scratch or modifying data that you have. In our case it is easy to create data just randomly choosing direction and observing if the bird is still alive after the turn.

.

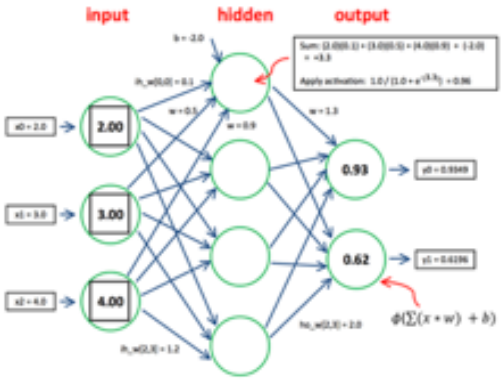


For Example, 1: Neural Network

when I was doing my researching, I found this article Neural Network Plays Flappy Bird written by Woojae Kim on Apr 9, 2017. His in his articles he says “When the program runs the first time, it does nothing because it has no idea about the game. However, a few hours of training, it learns how to play the game.” The idea was using a Neural Network to build a game and see how many generations it takes to master the game. Example 1 on top shows us how new Neural Network in he’s an explanation “Every neural network has one input layer, one output layer, and one or more hidden layers. Each circle represents a neuron and a neuron has a connection to every neuron in the next layer”. We see this kind of discussion in our life when we have to choose between iPhone or android way-out the options in our minds.

III. THE NEW METHOD & IMPLEMENTATION

Architecture of neural network is choosing the right architecture, or your neural network is always hard. You can choose number of neurons in layers, number of layers and types of neurons. It always depends on task that you trying to solve. It’s better to try different variations and choose the one that fits more than others AI solves problems in a fundamentally different way to regular programming. Normally you code a program by writing specific instructions for every situation that your program might encounter.



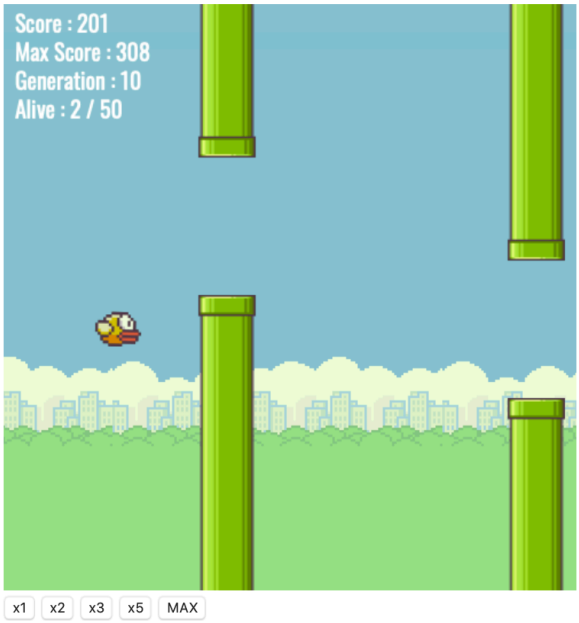
For Example, 2: Neural Network implementation

“Each connection has weight value and each neuron has a bias value. For example, the below diagram shows what’s happening in the neural network. When a neural network calculates the output, it does a lot of math with weight and bias. You can easily expect that if I change one of the weight or bias value, the final output will also be changed. In other words, training a neural network means finding and adjusting weight and bias values

that give the best output that we want." The idea is how to predict where the object is by using the total distance covered by a bird and its current distance to the closest gap.

IV. Simulation & Analysis Results

Ok, so now we have a brain. Next we need to teach it how to play. Teaching the network how to play Flappy Bird is really a case of searching for the best configuration of weights for its neural network. There are a lot of ways that you could configure this neural network. Some of those ways would result in a plane that flies really well, and others would result in a plane that crashes immediately. To find the configuration that flies best, we're going to use a search and optimization technique called a genetic algorithm. It's inspired by the natural processes that drive biological evolution and although it sounds complicated, the principles behind it are actually quite simple. First, we create 250 neural networks all with random configurations. This is called initialization. Then we try them out. Let's put each of our 250 randomly configured neural networks in a plane and see how they fly. Ok, that didn't go very well. But if you watch closely, you'll see that three of them actually flew past a few obstacles before they crashed. So now we apply some natural selection. It's survival of the fittest, so we measure which of the planes did best and use them create the next generation of 250 new neural networks. When we create the 250 new neural networks, we'll also add a little bit of randomness. This is called mutation and it helps us search for even better configurations to the next generations.



For Example, 3: Simulation Run

V. Helpful Hints

We have built a simulation that shows our theories out using Neural network and Machine Learning Algorithm. To test this kind of system have used a JS and html code to run the theories out and to if it would work. Here is the link (http://codd.cs.gsu.edu/~Dabay1//Neuralnetworks/index.html) for you to see how it looks and work .We created a link for you if you want to see the codes (https://github.com/DawitAbay/AI-Neutral-Network-).

VI. Conclusion

In conclusion my result I have programmed a code using a Neural network and Machine Learning Algorithm by predicting how many generations does it take to get it perfectly right every time. In

our case it took 8 generations get it prefect. Our plan for future work is test Neural network and Machine Learning Algorithm in different environment to see how it would adapt or react to it environment. In the future we can test this in a real-world situation see how the Neural network and Machine Learning Algorithm will react. Even know Google, Facebooks and Amazon are building this kind of AI machine to leaner what we like and dislike.

REFERENCES

[1] S. D. Webb, S. Soh, "Cheating in networked computer games: a review", *DIMEA '07: Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts*, pp. 105-112, 2007.

[2] J. Y. Wang, Y. B. Lin, "Implementation and Comparison the Dynamic Pathfinding Algorithm and Two Modified A* Pathfinding Algorithms in a Car Racing Game", *International Journal of Computer Theory and Engineering*, vol. 4, no. 4, pp. 551-555, August 2012.

[3] J. Togelius N. Shaker S. Karakovskiy N.G. Yannakakis "The Mario AI Championship 2009-2012" AI Magazine vol. 34 no. 3 pp. 89-92 2013.

[4] J. Togelius S. Risi "Neuroevolution in Games: State of the Art and Open Challenges" arXiv:1410.7326.

[5] Kenneth O. Stanley Miikkulaien Risto Evolving Neural Networks through Augmenting Topologies MIT Press 2002.

[6] Game 2048 [online] Available: https://en.wikipedia.org/wiki/2048_(video_game).

[7] ANJI library [online] Available: http://anji.sourceforge.net/.