Final Project Report

170050106 - Shashij Gupta

170050109 - Rachit Bansal

170050110 - Manthan Jindal

170100078 - Tarush Agarwal

Flappy Bird Using Artificial Intelligence

Flappy Birds using genetics Algorithm and Neural Network:

The Technique implemented was a neural network trained based on genetics algorithm

Neural Network:

We are using a neural network with 2 fully connected layers, having sigmoid activation function for the hidden layer. The architecture of our neural network is as follow:

- layer 1 - this is the input layer comprised of 3 nodes decribed below
- layer 2 - this is the hidden layer of 7 nodes
- layer 3 - this is the output layer with only 1 node as required

Inputs are :

Height of the bird, distance from next pipe (obstacle), pipe height of the next pipe (gap between pipes is fixed) on a scale of 0-1(since maximum value of each of these is fixed)

Hyperparameters for the network:

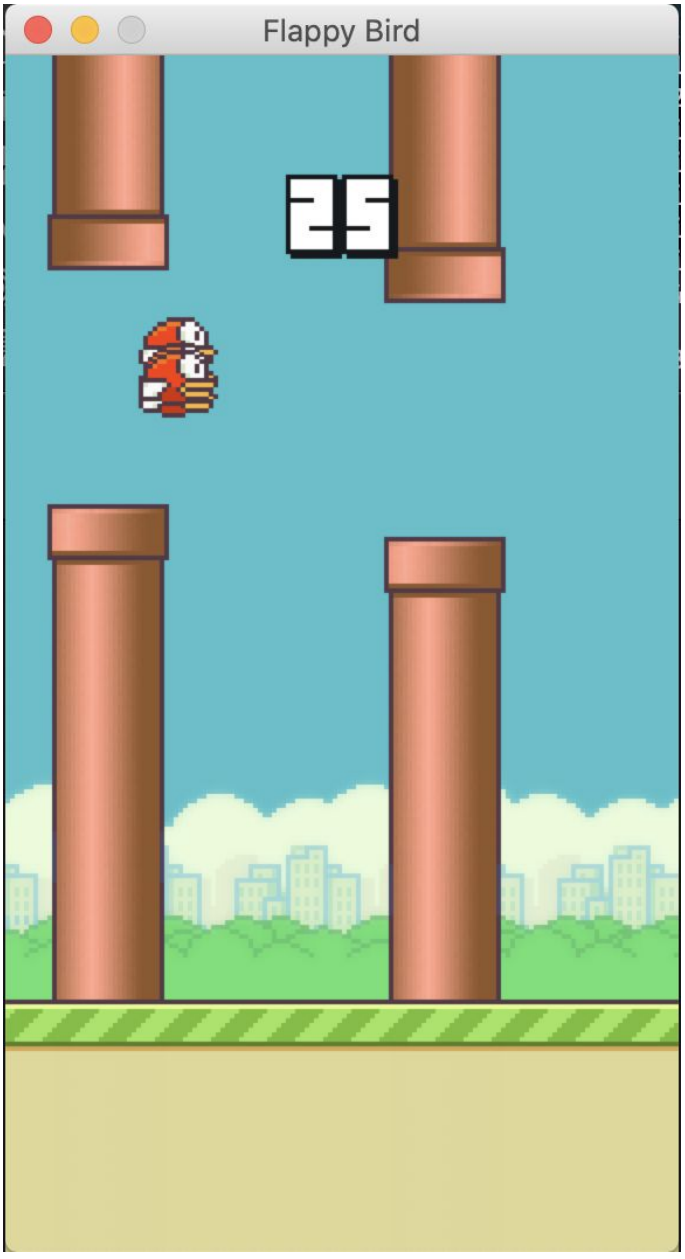
We took inspiration from the neural network code of one of our labs.
Weights are initialised randomly based Gloroth Uniform Initialisation

W = -sqrt(6/(number of in nodes + no of out nodes))

This is how they are initialised in the keras library
Biases are initialised to zero but using gaussian initialisation also works fine and no change was observed
The input was limited to 3 nodes as the decision was based primarily on these three features. A few other combinations were tried like taking just

the difference in height between pipe and bird but this was observed to be optimal.
The hidden layer had 7 nodes concluded mainly by trial and error.
The output is one node corresponding to 1 if the decision is to jump and 0 if the decision is to not jump.

Other Techniques tried :
Other prediction algorithms (logistic regression) were not found satisfactory.
Gaussian Initialisation for weights but the results were not found satisfactory.



Genetics Algorithm:

The Neural Network is trained using genetics algorithm. The inspiration was drawn from the internet and primary reason to implement was to learn something new. Training through practical implementation for such a problem is found to be very tiresome as we either need to use already existing gameplay but convert it into the desired form for our model or we can train it by manually playing the game. Both these methods fail when compared to the genetics algorithm which learns by itself and hence it is just a matter of time before we have a well-trained model which can play the game very well.

The Algorithm is used to train the model by itself. For every run, we use multiple agents (birds in our case) and a reward (stored in variable fitness) is associated with the run of each such agent. After every run, the weight vectors for each agent is updated according to this algorithm :

1) top 40% agents (based on fitness score) get the same weight vector

2) 50% of this top 40% is then randomly picked and mutated slightly

3) The next 10% of the total agents come by the crossover of top 10% agents

4) Now from the top 40% agents we again pick random agent pairs followed by crossover to generate the remaining 30% of updated agents.

All the above-updated weight vectors are mutated with a mutation rate of 15%.

The reward (fitness) is a measure of how much time an agent ran proportional to the distance travelled by a bird and how many pipes it crossed successfully(+1 for every frame passed and +25 for every pipe crossed)

Limitations of Genetics Algorithm:

- This evolution algorithm has a "big" stochastic component and this means that you need to find a statistical convergent solution with many simulations.
- Also, the solution depends on the total number of models used which must be optimal
- The algorithm varies widely with changes in update rule of agents and also on techniques of cross over and mutation rate.
- The Algorithm peaks after a certain point of time in terms of performance.

Another algorithm given a thought was Q-learning. But the limitation is that the algorithm takes time to train.
We also studied regarding deep Q-learning but again the game of flappy birds does not require such complex networks.
Hence, we find A simple neural network trained using genetics algorithm to be optimal.