How to train your A.I. Dragon

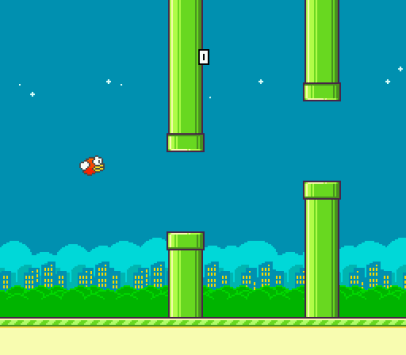


A Neural Network Implementation with training via the backpropagation algorithm using Tensorflow in Python.

**The game environment:**

We will use the Flappy Bird game as an example.

What are the important features?



**Rules**:

Flappy must avoid the pipes

Flappy must stay on the screen (can’t go off the top or bottom)

The player presses the space bar to make Flappy flap his wings (go up)

If the space bar is not pressed, Flappy falls due to gravity.

What are the important environment attributes/conditions that makes the player press the space bar?

* The position of the gap in the upcoming pipe
  + X and Y of bottom of upper pipe
  + X and Y of top of lower pipe
* The position of Flappy
  + X and Y of Flappy

So, within each game loop our “AI Flappy” must decide whether to flap or not.

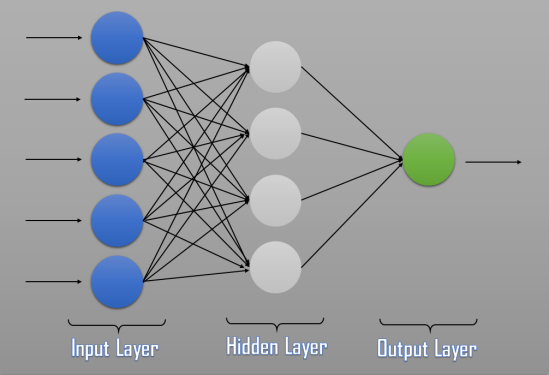
So Flappy must have a Brain (Artificial Neural Network) which must decide whether to flap or not to flap.

So what should the architecture of our Brain (ANN) look like?

A 3 layer Feed Forward Neural Network.

* First our Input Layer has 5 neurons
* Second or hidden layer has 4 neurons
* Third or output layer has 1 neuron.

If the output from the last neuron is of a value greater than 0.5, Flappy jumps/flaps/moves up, and if less than 0.5, it goes down (due to gravity).



3-Layer Neural Network

The 5 inputs to the Neural Network are:

1. X – Coordinate of the front – most “Pillar”
2. Y – Coordinate of the lower part of the Upper “Pillar”
3. Y – Coordinate of the upper part of the Lower “Pillar”
4. X – Coordinate of the Bird
5. y – Coordinate of the Bird

**Activation Functions**:

For the first Layer of the Neural Network, the activation function is ReLU (Rectified Linear Unit)

ReLu(x) = max(0, x)

For the output layer activation function we will use Sigmoid.

Another possibility would be the Softmax function, a wonderful activation function that turns numbers into probabilities that sum to one (normalisation).

Now we will be able to perform a Feedforward though the network.

So, we need:

* **A Brain**
  + What attributes should it have?
    - A neural network structure (layers and neurons)
  + What methods should it have?
    - Feedforward
    - ReLu
    - Softmax (I don’t actually use this though)
    - Sigmoid
* Individual **Birds**
  + What attributes should a Bird have?
    - Its own Brain (ANN)
  + What methods should it have?
    - Update
    - Think/Decide

In the Game loop we call the Update() method on each member (Bird) in the population. This in turn will call the Brird’s Think() method which executes the FeedForward() routine in the Brain.

**Training.**

We will use Backpropagation to find the best weights for the Brain’s neural network. Backpropagation is a supervised learning process, therefor we will need training data.

1. We will generate training data by playing the game and capturing the necessary state and user controls on each loop into a file “Training\_data.csv”.
   1. In Game.h
      1. set player = true
      2. set ai = false
      3. set birdCount = 1
      4. During game play press “C” to start capturing data
2. The program outputs the decision as either “glide” or “flap” to the file. Since Tensorflow does not handle text, we need to find and replace the word glide with a 0 (zero) and flap with a 1. This way our model will learn to output a 0 or 1 which we will then interpret as glide or flap. Once you understand this it would make sense to change the C++ code to output ‘0’ for ‘glide’ and ‘1’ for ‘flap’ and save us this renaming step!
3. We will then use this training data to train an Artificial Neural Network (ANN) using Tensoflow in Python. The ANN is created and trained in the Python script below. **You need to have Python and the Tensorflow library installed** (I used python -m pip install tensorflow). The resulting neural network weights are then exported to a file “flappy\_model.h5”.
   1. Source: **keras\_flappy-bird\_network.py**
4. We then convert the model to a txt file “flappy\_tf\_weights.csv”
   1. Source: **convert\_flappy\_model\_weights\_to\_csv.py**
5. These weights will then be read into our Flappybird’s Brain(s) which then takes control of the bird.
   1. Make sure flappy\_tf\_weights.csv is copied to the project folder
   2. In Game.h
      1. set player = false
      2. set ai = true
      3. set birdCount = 1 (you can also set this to a bigger number e.g. 10 and all birds will have the same brain but start off in slightly different locations and then converge onto the same spot. It’s fun, so try it 😊)