Les xarxes neuronals és un sistema de ML (Machine Learning o Aprenentatge Automàtic) que és basa en la biologia, a la estructura neuronal dels animal. Aquestes xarxes és representen en forma de graf, on els vèrtexs són les neurones encarregades de emmagatzemar la informació i on les arestes són les connexions (amb un número del -1 a l’1 associat per cadascuna) encarregades de transformar aquesta informació i transmetre-la a les neurones connectades. La primera columna, vermella amb títol de *Input* són les neurones on col·locarem la informació necessària per fer funcionar la xarxa neuronal; la segona columna, blava amb títol *Hidden,* són les neurones encarregades d’emmagatzemar la informació que encara no s’ha acabat de processar (només per problemes que són lineals aquesta capa és prescindible), i la tercera columna, verda amb títol Output, són les neurones encarregades de proporcionar les dades de sortida. Per saber el valor d’una neurona s’ha de sumar les multiplicacions entre les neurones estrictament precedents i la seva connexió [(en cas d’un problema no lineal els resultats de les neurones ocultes l’haurem de passar per una funció d’activació, com pot ser sigmoide). En el cas del *Flappy Bird només* la neurona de sortida s’haurà de passar per una funció d’activació, en aquest cas anomenada sigmoide, que acotarà el resultat entre un valor del 0 i l’1, si el valor és superior a 0,7 el personatge saltarà.](mailto:https://github.com/CodeReclaimers/neat-python?subject=Neat%20Python)

Font: By Glosser.ca - Own work, Derivative of File:Artificial neural network.svg, CC BY-SA 3.0. <https://commons.wikimedia.org/w/index.php?curid=24913461>

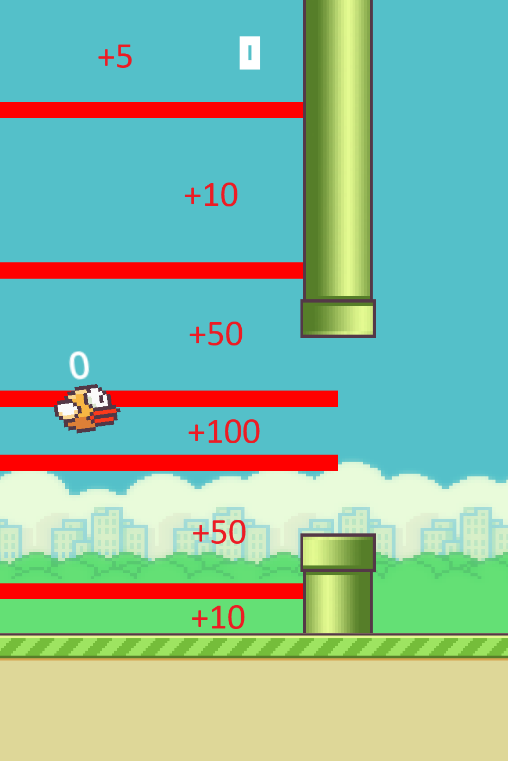
Per entrenar la xarxa neuronal utilitzem una versió d’un algorisme genètic, inspirat en la teoria Darwiniana, anomenat NEAT (Neuro Evolution of Augmenting Topologies). En primer lloc es crearà un nombre N (a definir) de xarxes neuronals, que anomenem de forma individual “agents” i de forma col·lectiva “població” i per cada població l'anomenarem segons el seu ordre d'aparició com a generació. La primera població que apareix serà la generació 0 (zero) i així successivament. Les connexions de cada agent de la generació 0 serà totalment aleatòria i l'estructura de la xarxa neuronal serà predeterminada una neurona per cada input exsistent, una neurona de sortida i connexions que connectin les neurones input amb la neurona output. Per la següent generació agafarem el 10% més 1 dels millors agents de la generació anterior, a través d’una puntuació anomenada *fitness* que segons cada tasca te els seus criteris de puntuació, i la resta de població restant es crearà a partir de "mutar" els agents escollits de la generació anterior o de barrejar dues xarxes neuronals que s'han escollit de la població anterior.

La barreja de dos agents, o *crossover (Crs)*, consisteix en comparar si totes dues tenen la mateixa estructura i en aquest cas es fan les mitjanes aritmètiques de cada connexió amb la seva respectiva per crear un agent nou amb la mateixa estructura i les mitjanes aritmètiques com els valors de les seves connexions.

Per la mutació d'un agent, el procediment és més complex, perquè pot mutar de diferents maneres (la manera s’escollirà de forma aleatòria), de forma no estructural (MNE) o de forma estructural (ME). La no estructural consisteix en canviar un valor d'una connexió per un altre valor aleatori. En canvi, les estructurals, de les que hi ha més d'un tipus, modifiquen el graf perquè poden crear una neurona nova o destruir-la i també poden crear una nova connexió, o destruir-la. Quan es crea una neurona nova, és necessari connectar-la amb la xarxa, aleshores es crea una aresta que té com a direcció aquesta nova neurona i té una altra connexió que parteix des d'aquest vèrtex fins un altre, i quan es destrueix una neurona també han de desaparèixer totes les connexions amb les quals interactuava directament.

Un cop s'ha creat la nova generació es repeteix el procés fins a aconseguir un agent que assoleix fer la tasca de forma correcta.

**FITNESS**

El fitness és un sistema de puntuació que s’utilitza per saber quina xarxa neuronal a realitzat millor la tasca. En el *Flappy Bird* s’entreguen 1000 punts per passar un obstacle, es penalitza amb -100 punts si mor per tocar el terra, segons com aprop vola l’ocell de la PYF, com és mostra a la imatge, es recompensa l’ocell segons la zona amb 100, 50, 10 o 5 punts per segon de més a prop a més llunya i per ser viu s’entrega 1 punt per segon. Després de que totes les xarxes neuronals morin o arribin a la puntuació màxima establerta per fer d’un joc infinit un finit, en aquest cas després de passar per 150 tubs, és farà una llista de la millor a la pitjor xarxa neuronal segons els criteris anteriors per poder escollir les millors per la següent generació.

INPUTS

Els inputs són aquelles dades que donem a la xarxa neuronal per que porti a terme la seva tasca correctament. En el Flappy Bird donarem a cada xarxa neuronal 4 possibles dades que anirem variant per poder fer l’anàlisi, aquestes són: la posició Y de l’ocell (PYO), la posició Y del centre del forat de l’obstacle (PYF), la posició X del centre del forat de l’obstacle (PXF).

Altres constants

Per la recollida de dades hi ha variables que canvien, que son els inputs, els tipus de mutació i la població per generació, en canvi n’hi ha d’altres que no variant. Per cada configuració és fa entre 20 i 30 partides, per cada partida hi ha 125 generacions i la puntuació màxima establerta en 150, es a dir que després d’haver passat per 150 tubs automàticament passarà a la següent generació.