

**Grado en Ingeniería Informática**

**Inteligencia Artificial**

**Curso 2019/2020**



**Universidad de Jaén**

**Guión 3**  
**Juegos con Inteligencia**  
**Computacional**

# 1. Juegos con sistemas inteligentes incorporados

La utilización de la Inteligencia Artificial en un videojuego hace referencia al conjunto de técnicas utilizadas en computadores y videojuegos para producir la ilusión de inteligencia en el comportamiento del propio juego.

Desde hace unos años se vienen incorporando sistemas inteligentes a los juegos y son muchos los que han dado una buena muestra de lo que una inteligencia artificial puede lograr, sin embargo, aún estamos mucho más cerca de la palabra "artificial" que de "inteligencia".

Las nuevas tecnologías y los dispositivos de mayor capacidad han permitido a los desarrolladores crear motores más poderosos para los personajes no jugadores y con esto Inteligencia Artificial mejorada. Sin embargo, el avance se ha dado más en forma horizontal que vertical, no es tanto que las inteligencias hayan mejorado sino que ahora se puede crear más cantidad de personajes simultáneamente y mayor diversidad de acciones para los mismos.

Por otro lado, hay que reseñar que la industria parece un poco reacia a generar sistemas tan inteligentes que sobrepasen al jugador, al menos en cuanto a desarrollo de enemigos se refiere.

Desde empresas como Gamasutra (uno de los portales para desarrolladores de videojuegos más importantes del mundo) indican que un juego que tenga una **inteligencia artificial muy alta**, puede ir en contra de la propia capacidad de entretenimiento del mismo. **La inteligencia artificial en los videojuegos ha madurado, pero no lo suficiente.** Esto respondería a que los jugadores son capaces de **resistir un cierto nivel de frustración**. Pero si perdieran todo el tiempo ante una inteligencia artificial imbatible, dejarían de jugar al título ya que la imposibilidad de avanzar termina por producir **falta de interés y aburrimiento**.

## 2. Flappy Bird

**Flappy Bird** es un juego desarrollado en 2013 y publicado por GEARS Studios. El jugador controla un pájaro intentando volar entre filas de árboles o tuberías sin poder tocarlas. La escena se va desplazando lateralmente y el juego termina cuando el pájaro se estrella contra uno de los árboles.



Pero la mejor forma de conocer su funcionamiento es jugando:

<https://flappybird.io/>

## 3. Ejercicio 1

Tienes que adjuntar una captura de pantalla de tu puntuación obtenida una vez transcurridos 5 minutos de juego, y debes contabilizar el número de partidas que has realizado.

Número de partidas	Puntuación máxima	Minuto en el que alcanzaste esta puntuación

## 4. Algoritmos evolutivos y redes neuronales

Una de las ramas que podemos encontrar dentro de la Inteligencia Artificial es la Inteligencia Computacional.

La Inteligencia Computacional se centra en el diseño de sistemas informáticos inteligentes que imitan la naturaleza y el razonamiento lingüístico humano para resolver problemas complejos. En este campo de investigación confluyen tres líneas de desarrollo de sistemas inteligentes que conforman el centro de la inteligencia computacional: la emulación del razonamiento lingüístico humano mediante la teoría de los conjuntos difusos, las redes neuronales que imitan al sistema nervioso, y los algoritmos de optimización bioinspirados (agrupados dentro del área de investigación denominada Metaheurísticas).

A continuación introducimos brevemente las redes neuronales y los algoritmos evolutivos.

### 4.1. Redes neuronales

Una red neuronal es un sistema inteligente para el aprendizaje a partir de una base de datos de ejemplos de un problema para obtener el modelo asociado a estos. Al igual que las neuronas biológicas, las neuronas artificiales se interconectan para formar redes de neuronas artificiales. Cada neurona artificial utiliza una función procesamiento que agrega la información de conexiones de entrada con otras neuronas artificiales, una función de activación y una función de transferencia para dar una salida de la neurona en sus conexiones de salida. Las redes neuronales son uno de los paradigmas de aprendizaje automático más importantes en la actualidad.

### 4.2. Algoritmos bioinspirados, algoritmos evolutivos

Los algoritmos bioinspirados son algoritmos capaces de proporcionar soluciones aceptablemente buenas (no necesariamente óptimas) en un tiempo razonable. A lo largo de la segunda mitad del siglo pasado, apareció un conjunto de técnicas bioinspiradas en la evolución y en la inteligencia colectiva que han tenido repercusiones trascendentes en el campo de la optimización como los algoritmos evolutivos, los algoritmos de optimización basados en colonias de hormigas y los algoritmos de optimización basados en nubes de partículas.

Los algoritmos evolutivos son enfoques basados en los procesos naturales de evolución y genética. Estos algoritmos mantienen una población de cromosomas (que representan soluciones candidatas al problema) que evolucionan mediante tres procesos: mecanismo de selección, operadores de cruce y mutación y estrategia de reemplazo.

## 5. Flappy bird con Inteligencia Computacional

El juego con el que vamos a trabajar en esta práctica es un juego inteligente basado en el clásico Flappy Bird que incorpora algoritmos de Inteligencia Computacional para resolver el problema.

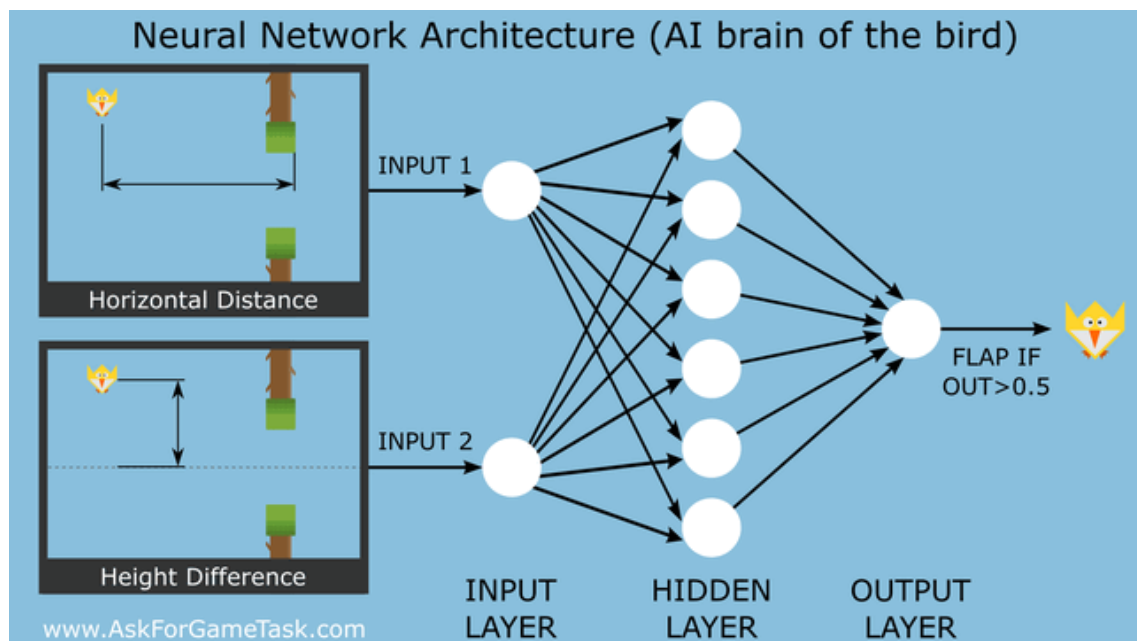
- ¿Cómo se incorpora la IA en el juego?

Mediante una co-evolución entre algoritmos evolutivos y redes neuronales.

Inicialmente se crea una población con 50 pájaros que sería la población del algoritmo genético. Cada individuo del algoritmo genético es un pájaro.

- ¿Qué es realmente cada individuo o pájaro?

Cada individuo codifica una red neuronal con 3 capas de neuronas, tal y como se muestra en la imagen.



- ¿Y qué son las entradas?

La primera entrada es la distancia horizontal entre el pájaro y la posición del hueco por donde tiene que pasar.

La segunda entrada de la red neuronal es la altura entre el pájaro y la posición del objeto por donde debe pasar.

- ¿Y la salida?

Una vez se pasa la información de entrada a la red neuronal, la red entrena con ellas y devuelve una salida en el intervalo  $[0, 1]$  de forma que el pájaro volaría si el valor está por encima de 0.5 y se quedaría quieto en otro caso.

- ¿Y cómo evoluciona la población del algoritmo genético?

La población evoluciona mediante un esquema generacional elitista, esto quiere decir, que los dos mejores individuos se pasan a la población de la siguiente partida, y el resto de individuos se crean mediante operadores de cruce y mutación como si de una especie animal se tratase.

- ¿Cuándo se activa la red neuronal?

Cada vez que un pájaro pasa un árbol o realiza un movimiento de vuelo o no vuelo necesita calcular nuevamente su posición respecto al siguiente hueco para calcular de nuevo el movimiento a realizar.

## 6. Ejercicio propuesto

### 6.1. Ejercicio

Para este ejercicio vamos a realizar un estudio sobre el juego en sí y después vamos a modificar operadores del algoritmo genético para conseguir mejorar la evolución del juego, es decir conseguir el mejor pájaro lo antes posible.

En un primer apartado deberás completar la siguiente tabla con 10 ejecuciones, una ejecución se considerará satisfactoria si el pájaro supera 25 generaciones o una puntuación superior a 10.000. En una última fila deberás realizar la media aritmética de los valores obtenidos en las 10 partidas.

Partida – Juego original	Generación	Puntuación
1		
2		
3		
...		
10		
MEDIA ARITMÉTICA		

## 6.2. Ejercicio

Debes realizar modificaciones sobre ciertos operadores del algoritmo genético y realizar un nuevo estudio experimental pero esta vez con 5 partidas para cada cambio.

Todos los cambios se deben realizar sobre el fichero `Neuroevolution.js`

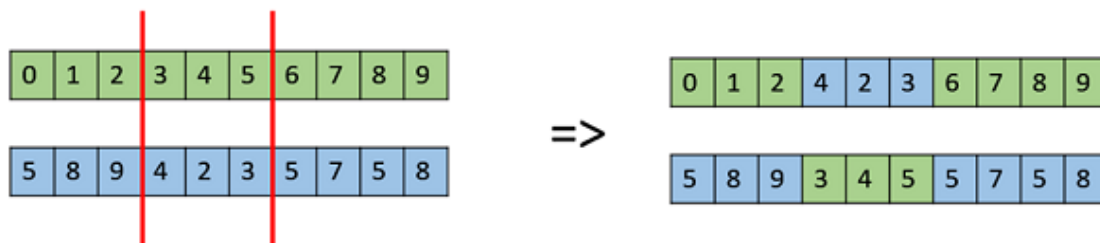
### Operador de cruce (Genetic crossover)

```

324  /**
325  * Breed to genomes to produce offspring(s).
326  *
327  * @param {g1} Genome 1.
328  * @param {g2} Genome 2.
329  * @param {nbChilds} Number of offspring (children).
330  */
331  Generation.prototype.breed = function (g1, g2, nbChilds) {
332    var datas = [];
333    for (var nb = 0; nb < nbChilds; nb++) {
334      // Deep clone of genome 1.
335      var data = JSON.parse(JSON.stringify(g1));
336      for (var i in g2.network.weights) {
337        // Genetic crossover
338        // 0.5 is the crossover factor.
339        // FIXME Really should be a predefined constant.
340        if (Math.random() <= 0.5) {
341          data.network.weights[i] = g2.network.weights[i];
342        }
343      }
344
345      // Perform mutation on some weights.
346      for (var i in data.network.weights) {
347        if (Math.random() <= self.options.mutationRate) {
348          data.network.weights[i] += Math.random() *
349            self.options.mutationRange *
350            2 -
351            self.options.mutationRange;
352        }
353      }
354      datas.push(data);
355    }
356
357    return datas;
358  }

```

El cruce implementado por defecto es una recombinación y tú debes implementar el cruce en dos puntos. El funcionamiento del cruce en dos puntos es como se muestra gráficamente en la imagen



Aspectos a tener en cuenta:

- El punto A y el punto B deben ser diferentes.
- El punto de la izquierda no puede ser igual a 0, ni el punto de la derecha tampoco puede ser igual al tamaño del individuo.

## Operador de mutación (Mutation operator)

```

324  /**
325   * Breed to genomes to produce offspring(s).
326   *
327   * @param {g1} Genome 1.
328   * @param {g2} Genome 2.
329   * @param {nbChilds} Number of offspring (children).
330   */
331  Generation.prototype.breed = function (g1, g2, nbChilds) {
332    var datas = [];
333    for (var nb = 0; nb < nbChilds; nb++) {
334      // Deep clone of genome 1.
335      var data = JSON.parse(JSON.stringify(g1));
336      for (var i in g2.network.weights) {
337        // Genetic crossover
338        // 0.5 is the crossover factor.
339        // FIXME Really should be a predefined constant.
340        if (Math.random() <= 0.5) {
341          data.network.weights[i] = g2.network.weights[i];
342        }
343      }
344
345      // Perform mutation on some weights.
346      for (var i in data.network.weights) {
347        if (Math.random() <= self.options.mutationRate) {
348          data.network.weights[i] += Math.random() *
349            self.options.mutationRange *
350            2 -
351            self.options.mutationRange;
352        }
353      }
354      datas.push(data);
355    }
356    return datas;
357  }
358 }

```

En este caso es un operador matemático que emplea un aleatorio entre el intervalo  $[0, 1]$  y un valor denominado *mutationRange*. Sin embargo, el nuevo operador que tienes que implementar debe ser una operación matemática totalmente distinta a la anterior cuyo valor final esté dentro del intervalo  $[0, 1]$  y debes emplear una función exponencial, logarítmica o trigonométrica. Tienes total libertad para implementarlo como consideres oportuno.

Tendrás que repetir el proceso del ejercicio anterior para las distintas combinaciones posibles y montar una tabla para cada uno de ellos:

- A. Nuevo cruce y mutación por defecto.
- B. Cruce por defecto y mutación nueva.
- C. Cruce y mutación nueva.

Partida – Juego A	Generación	Puntuación
1		
...		
5		
MEDIA ARITMÉTICA		

Nota: Debe haber una tabla de estas para cada juego: A, B y C



### 6.3. Ejercicio

**Deberás mostrar una tabla con cuatro filas y los resultados de las medias aritméticas de las tablas anteriores, y hacer un análisis (QUÉ FUNCIONA MEJOR Y POR QUÉ) de los resultados obtenidos con respecto a los resultados que conseguiste en las partidas que tú jugaste.**

Partida	Generación	Puntuación
MEDIA ORIGINAL		
MEDIA A		
MEDIA B		
MEDIA C		

## 7. Normativa

Este guión de prácticas no tendrá defensa.

El documento PDF debe incorporar los códigos implementados de cruce y mutación.

Lee atentamente la normativa de entrega de prácticas:

- Solo se admitirá el formato **PDF**. No se corregirán guiones en cualquier otro formato al indicado.
- El documento incluirá una portada con:
  - o Identificación de los dos alumnos (Nombre, apellidos y DNI).
  - o Identificación del guión.
- El nombre del fichero tendrá el siguiente formato “Ape11-Ape12-Ape21-Ape22-GuionX.pdf” donde
  - o Ape11 es el primer apellido del primer alumno.
  - o Ape12 es el segundo apellido del primer alumno.
  - o Ape21 es el primer apellido del segundo alumno.
  - o Ape22 es el segundo apellido del segundo alumno.
  - o X es el número del guión.

## 8. Entrega y evaluación

Este guión tiene un valor de 1 punto.

La entrega se llevará a cabo a través de la actividad correspondiente en ILIAS.  
El plazo de entrega termina el 2 de abril de 2020.