

Proyecto IA - Práctica 4: Redes Neuronales

ESNE, CURSO 4.3 18/19

Santiago Arribas Maroto

Luis Chamarro Alonso

Amadeo Delgado Casado

Rafael Enrique Gautier Barderas

Mayo 2019

1. Concepto del experimento

Este experimento se basa en la implementación de redes neuronales para el comportamiento de los animales, además de redes neuronales, hemos pensado que sería positivo unir tanto la práctica de algoritmos genéticos como esta, para crear un experimento lo más interesante posible.

Respecto al apartado de redes neuronales, especificar que su comportamiento se basa en el proceso de pensamiento que los animales deben realizar al detectar nuevos elementos en su campo de visión, múltiples elementos (Desarrollados más adelante) presentan el input de la red neuronal, siendo uno de ellos, el elemento que ha entrado en el campo de visión, con la información recibida en el input, el animal debe decidir si acercarse o alejarse del elemento encontrado, poniendo un ejemplo, si un conejo es perseguido por un animal y posee energía suficiente, al detectar una fruta nueva en su campo de visión, debería decidir continuar con la huida, aunque, al no estar predefinido, podríamos ver resultados más complejos.

En primera instancia, el comportamiento de las redes neuronales solamente se aplicará sobre los conejos en el experimento, sería fácilmente ampliable a los zorros, pero sería repetir el mismo proceso, puesto que no cada conejo posee una red independiente, sino que, debido a la combinación de redes neuronales y algoritmos genéticos, la esperanza de vida de un animal puede ser efímera para que una red pueda entrenar suficiente para ver resultados notables, de modo que, los conejos poseen una red neuronal compleja y conjunta, lo reconocido como "Mente colmena", ya que la red neuronal aprende de todos los sujetos, pero cada individuo puede ejecutar el proceso neuronal de manera independiente, y, aun muriendo un sujeto, el aprendizaje continúa.

Ya que somos conscientes de que puede ser difícil percibir el aprendizaje, por esto, hemos implementado una mecánica que permite que aun muriendo todos los animales, reaparecen 5 nuevos sujetos continuando con el aprendizaje de la red neuronal, de modo que, planeamos dejar el experimento activo por más de 12h para ver un escenario de largo aprendizaje y mostrar los resultados más visibles.

2. Capas de la red neuronal

La red neuronal posee tres capas, en primer lugar la capa de input, que recibe los parámetros: distancia a la comida, distancia al enemigo, distancia a otro conejo, diferencia de tamaño con el depredador, energía restante, ¿es la estación primavera?, ¿es la estación verano?, ¿es la estación otoño?, ¿es la estación invierno? La "Hidden Layer" contiene 6 nodos que resultan del cálculo de la raíz cuadrada del producto del número de inputs por el número de outputs. Los parámetros en la capa de salida resultan en: Acción (Acercarse / Alejarse), Tipo de elemento al que hacer la acción (Conejo, Depredador, Comida).

a. Capa de Input

En la capa de input planteamos todos los elementos necesarios para que el aprendizaje fuera efectivo, a continuación explicaremos el motivo de cada input elegido. Hay un total de 8 parámetros

Distancia a la comida: Se calcula la distancia entre el conejo en cuestión y la comida que acabe de entrar en el radio de detección, para que pueda valorar si le conviene ir a por ello o no.

Distancia al enemigo: Al igual que con la comida, permite al sujeto evaluar si es óptimo continuar con su objetivo actual o huir en caso en el cual la distancia sea demasiado lenta

Distancia a otro conejo: Calcula la distancia a otro conejo, para tener en cuenta también si sería óptimo acercarse al otro conejo y reproducirse si se pudiera.

Energía Restante: Para saber la necesidad imperativa de una acción sobre otra hay que tener en cuenta la energía restante hasta la muerte del conejo.

Estación del año: 4 inputs de 0/1 que indican si es invierno, primavera, verano o invierno. La estación permite al conejo saber si se podrá reproducir con otro conejo e incluso le permitirá poder gestionar sus acciones según la estación, ya que la comida variará en su cantidad según la estación del año en la que se encuentre.

b. Capa oculta ("Hidden Layer")

El número de nodos en la capa oculta se han calculado a partir de la raíz cuadrada del producto del número de inputs por el número de outputs, resultando en 6 nodos diferentes

c. Capa de output

La capa de output devuelve la acción que debe realizar el conejo, si acercarse o alejarse del elemento en sí, así como el elemento al que se acerca o aleja (3 nodos diferentes: Conejo, depredador, comida).

3. Entrenamiento de la red

La red neuronal posee un entrenamiento inicial parametrizado, del cual parte la ejecución, además se encuentra un entrenamiento durante la ejecución para poder optimizar al máximo el comportamiento de los animales, el entrenamiento se realiza cuando se cumplen las siguientes propiedades: Si se pierde energía, Si han muerto presas, Si la energía ha descendido del (80%, 60%, 40% o 20%), si se ha arriesgado demasiado yendo a por comida obviando a los depredadores o si se ha reproducido o tiene intención de ello.

4. Clases implementadas

Layer.cs

Clase que posee la implementación de las capas de la red, dichas capas poseen cierta cantidad de nodos y los pesos, además ejecuta el aprendizaje y busca los pesos más ajustados

NeuralNetwork.cs

Codificación de la red neuronal global, clase de tipo Singleton que posee los métodos de interacción en la red neuronal, tales como operaciones de capa, getters por input / output, setter de input, funciones para la propagación, "feed" de la red neuronal o cálculo de error en la red neuronal. Según la codificación de esta clase, vale como "template" genérico para cualquier aplicación de red neuronal

Según esto, el constructor de la red neuronal permite inicializar las capas necesarias con el número de nodos deseado, así como asignar pesos aleatorios a cada nodo de cada una de las capas inicializadas.

NeuralNetworkRabbit.cs

Particularización de la red neuronal al caso específico de los conejos, posee las propiedades de peso específicas y el aprendizaje para presas

UINeuralNetwork.cs

Clase destinada a representar visualmente la actividad de la red neuronal, tanto en los nodos en sí como en las conexiones entre estos

5. Conclusiones

Las conclusiones del experimento, muestran ciertas dificultades para percibir un comportamiento lógico a corto plazo, puesto que los animales necesitan al menos 30min para empezar a ver un comportamiento "inteligente" a pesar de esto, siguen existiendo comportamientos erráticos, no es hasta más de 2h de entrenamiento que el comportamiento es se percibe lógico, a pesar de esto, puede difuminarse la percepción debido al algoritmo genético, puesto que por muy inteligente que sea el comportamiento de un conejo, si sus genomas no son válidos para el entorno y no ha sido capaz de adaptarse al entorno, morirá por ser demasiado lento, pequeño, o débil.

6. Bibliografía e información adicional para el desarrollo

Complejo ejemplo explicativo sobre el reconocimiento de dígitos escritos a mano

<http://neuralnetworksanddeeplearning.com/chap1.html>

Ejemplos de implementación del algoritmo y uso en situaciones comerciales

<https://skymind.ai/wiki/neural-network>

Página web (de aspecto antiguo), con interesantes conceptos, aplicaciones y explicaciones

https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

Experimento conceptual sobre redes neuronales en navegador, muy visual y entretenido

<https://playground.tensorflow.org>