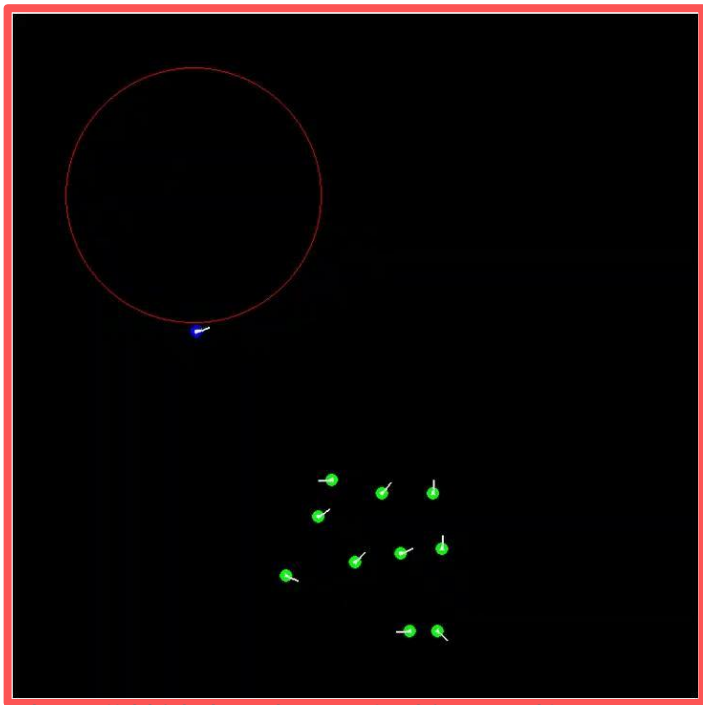

Coevolución de sistema rebaño-pastor

Trabajo final - Inteligencia Computacional 2025

Giuliana Cagnola - Franco Hergenreder - Lucas Saurin

Motivación



<https://abhishek-sankar.com/multi-agent-rl/>

Muchos rebaños de ovejas son criados por pastoreo, que consiste en dejarlas desplazarse libremente y consumir vegetación silvestre. Aquí entra en acción el perro pastor, cuyo objetivo es el de proteger el rebaño de posibles amenazas, y guiar y controlar su movimiento durante los desplazamientos. El objetivo de este proyecto es el entrenamiento del agente pastor con la finalidad explorar estrategias que podrían usarse para la evacuación de multitudes

Pipeline de Entrenamiento: Algoritmo Genético



Pipeline de Ejecución: Red Neuronal



Comportamiento del rebaño: Modelo de Strombon

Repulsión local + repulsión del pastor +
cohesión al centro de gravedad + ruido

$$\mathbf{v}_i^{t+1} = h \mathbf{v}_i^t + \sum_{j \in \mathcal{N}_i} \mathbf{f}_{ij}^{(aa)} + \mathbf{f}_i^{(s)} + \alpha(\mathbf{c}_i - \mathbf{s}_i) + \boldsymbol{\eta}$$
$$\mathbf{s}_i^{t+1} = \mathbf{s}_i^t + \Delta t \mathbf{v}_i^{t+1}$$

Comportamiento del pastor: NN de Napolitano

Red Neuronal de Napolitano:

- 14 entradas: la posición del target, las posiciones de las 5 ovejas más cercanas, la posición del centroide de ovejas. Todo relativo a sí mismo y normalizado por la diagonal.
 - 256 neuronas en la capa de entrada - activación ReLu
 - 128 neuronas en la capa de salida - activación ReLu
 - 2 salidas: vector dirección (luego se normaliza)
-

Función de Aptitud

$$F(g_i) = w_c C + w_g G + w_i I + w_d D + w_f F$$

- C: Cohesión del rebaño
 - G: Progreso hacia el target
 - I: Proporción de ovejas dentro del target
 - D: Conducción
 - F: Finalización
-

Implementación

Estructura del proyecto (implementación modular)

- *Simulación*
 - *Modelos*
 - *Entrenamiento*
 - *Scripts*
-

Implementación

1. Simulación

Este módulo implementa el motor del entorno. Contiene:

- La representación del mundo
- El control de estados globales como:
 - detección de finalización,
 - cálculo de métricas (centroide, cohesión, distancias).
- Una clase que integra el motor de simulación y la interfaz con el visualizador.

En esta capa no se definen comportamientos inteligentes: solo se aplican las reglas físicas y de interacción.

Implementación

2. Modelos

Incluye las implementaciones de los agentes del sistema:

- **Ovejas:** Basadas en el modelo heurístico de Strömbom, que combina repulsión local, repulsión del pastor, cohesión y ruido estocástico.
- **Pastor:** Manejado por
 - Una versión manual (“follow mouse”)
 - Red neuronal del pastor (ShepherdNN) basada en la arquitectura descrita por Napolitano.

Este módulo encapsula los comportamientos individuales de los agentes.

Implementación

3. Entrenamiento

Implementación completa del Algoritmo Genético:

- Representación del individuo como genoma binario
- **Proceso evolutivo:**
 - selección por ventana con elitismo
 - cruza simple
 - mutación
- **Decodificación de pesos** para la red neuronal.
- **Función de aptitud:** cohesión, progreso, conducción, proporción dentro del objetivo y finalización.

Este módulo produce los pesos optimizados que usará el pastor en la simulación.

Implementación

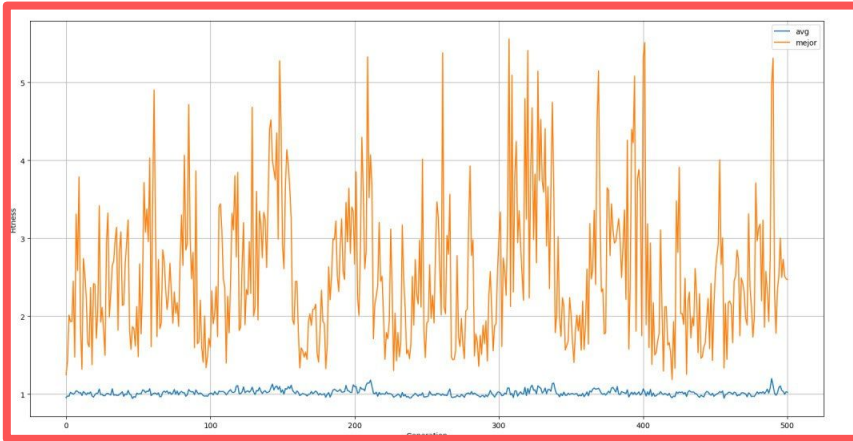
4. Scripts

Funcionan como wrappers para la ejecución:

- `run_sim`: ejecuta una simulación interactiva, permite seleccionar un modelo entrenado y visualiza el comportamiento.
- `run_train`: inicia el proceso de entrenamiento evolutivo completo, utilizando los parámetros declarados en la configuración.

No contienen lógica compleja: solo cargan configuraciones, seleccionan modelos y lanzan los módulos principales.

Resultados: Fitness



- Comportamiento errático del fitness
- Aptitud promedio = 1 → La mayoría de los pastores fallan
- El buen comportamiento existe, pero no logra transmitirse
- No hay convergencia hacia una solución óptima

Resultados: Métricas por generación

Generación	Cohesión	To goal	Driving	Fitness total
2	0.86	0.14	0.94	1.97
636	0.96	0.85	1.64	3.47
1924	0.98	2.52	1.63	5.13

Conclusión

- Se implementó un sistema de pastoreo multiagente, combinando un modelo heurístico para el rebaño y algoritmo genético más red neuronal para el pastor
 - El pastor logró aprender comportamientos emergentes simples
 - La función de aptitud pondera muchos términos, pero se podrían implementar mejoras a futuro
 - El modelo no generalizó de manera consistente ni convergió a una solución óptima
 - Se necesita explorar más la función de aptitud para concluir si el enfoque evolutivo es bueno para la resolución del problema.
-

Muchas Gracias!
