Redes Neurais - perceptron

Diego Feijó da Silva <diegofeijo.silva@gmail.com>

Tales Viegas <talesbv@gravatai.ulbra.tche.br > Professor

Universidade Luterana do Brasil (Ulbra) – Curso de Ciência da Computação – Campus Gravataí

Av. Itacolomy, 3.600 – Bairro São Vicente – CEP 94170-240 – Gravataí - RS

18 de abril de 2014

Resumo

Este artigo descreve a montagem de uma rede neural do tipo perceptron que realiza o reconhecimento das letras de a até y. Após o reconhecimento das letras o sistema permite que se entre com uma palavra no padrão estabelecido e a rede identifica letra a letra.

Palavras-chave: Inteligência Artificial; perceptron;redes neurais.

Abstract

Title: Neural Networks - Perceptron

This article describes the assembly of a perceptron neural network that performs the recognition of letters a through y. After recognition of the letters the system allows you to enter a word in the established pattern and the network identifies letter by letter.

Keywords: ARTIFICIAL INTELLIGENCE; PERCEPTRON;NEURAL NETWORKS.

1. INTRODUÇÃO

O coletor de lixo é um agente reativo na qual tem como objetivo percorrer um ambiente definido através de uma matriz de 10x10 procurando lixos secos e orgânicos. O ambiente é gerado de forma dinâmica sorteando as células conforme a quantidade de lixeiras definida nos parâmetros.

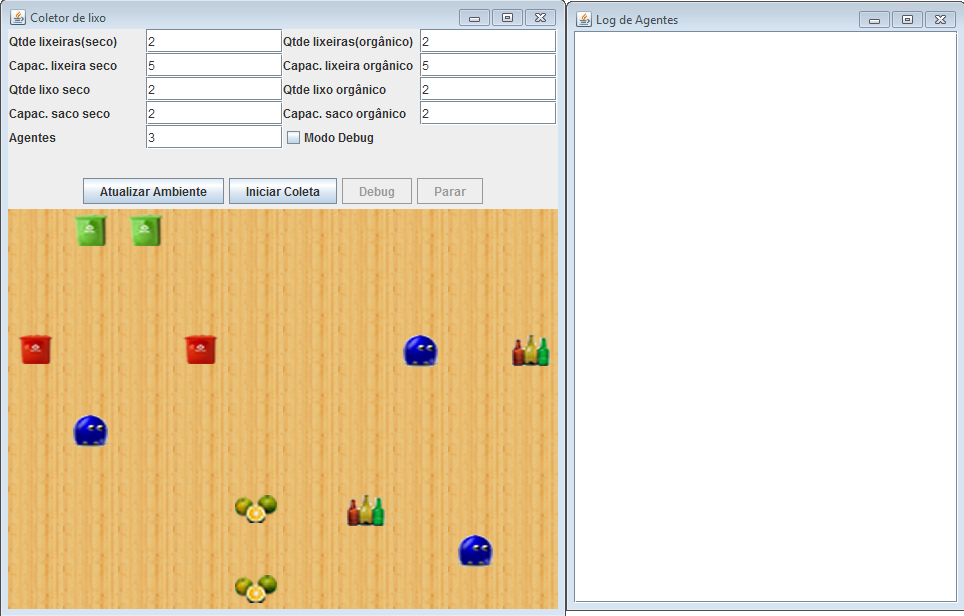
Após encontrar um determinado tipo de lixo (orgânico ou seco) ele coloca em sacos conforme seu tipo e após encher este saco ele o leva até uma lixeira específica mais próxima. O agente sabe onde fica a posição de cada lixeira, essa é a única informação que ele dispõe do ambiente.

O agente tem um campo de visão de duas células para reconhecer os lixos, ele se move uma célula de cada vez (cima, baixo, esquerda ou direita) caso o agente não encontre nenhum lixo após 3 ciclos, então este agente passa a andar em uma direção continua, sorteada aleatoriamente, até encontrar lixo, chegar no final da matriz, bater em uma lixeira ou bater em outro agente.

Os itens a seguir são configuráveis na inicialização do ambiente: quantidade de agentes, quantidade de lixo seco, quantidade de lixo orgânico, quantidade de lixeiras e capacidade das lixeiras, sendo distribuídos pela matriz aleatoriamente, de modo que o agente não fique com os movimentos limitados.

Os agentes irão esvaziar os sacos de lixos quando se posicionar na célula ao lado da lixeira e é permitido que o agente passe por cima do lixo caso o saco de lixo esteja cheio. Não é permitido que o agente passe por cima de outro agente ou outra lixeira. Cada célula do ambiente só pode ser ocupada por um único objeto (agente, lixeira ou lixo).

A execução só acaba quando os agentes recolhem todos os lixos para as suas respectivas lixeiras. A figura 1 mostra um exemplo de uma matriz 10x10 preenchida com suas lixeiras, lixos e agentes.



**Figura1**

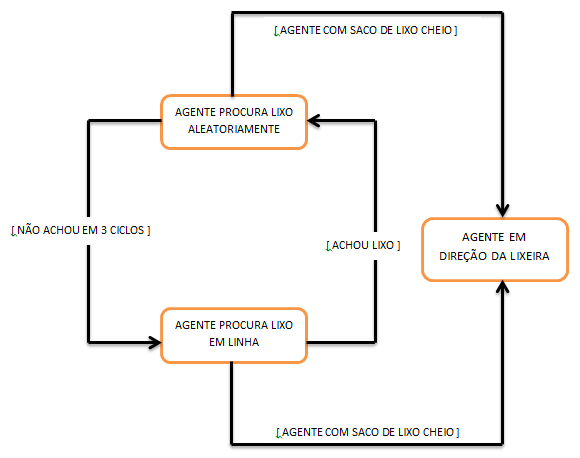
1. Fundamentos

Um agente reativo é designado para uma tarefa especifica. Como não possui nenhum histórico das ações passadas, ele interage com o ambiente conforme percorre o mesmo em busca de seu objetivo.

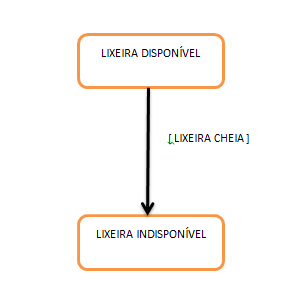
Agentes reativos podem ir desde jogos em que o jogador levanta e se move pelo cenário até sensores de presença que iluminam somente ambientes onde há pessoas e de acordo a atividade que elas estão desempenhando de modo a economizar energia (RUSSEL e NORVIG, 2009, p. 961).

1. **Modelagem**

A figura 2 apresenta a modelagem dos estados do agente



**Figura 2- estados do agente**

A figura 3 apresenta a modelagem dos estados da lixeira.

**Figura 3 – Estados da lixeira**

1. Implementação

O agente coletor de lixo foi implementado usando a linguagem orientada a objetos Java, os componentes da biblioteca SWING foram utilizados para exibição visual dos objetos. Para gerar a matriz foi necessário criar uma classe que herda de JLABEL. Essa implementação foi necessária para que pudéssemos utilizar os ícones conforme a Figura 1.

A classe foi chamada de MyLabel.java e possui o atributo (Flag) onde definimos os seguintes parâmetros:

* + - * + **LO:** Lixeira de lixo orgânico
      * **LS:** Lixeira de lixo seco
      * **S:** Lixo seco
      * **O:** lixo orgânico
      * **A:** Agente

A cada ciclo, que é medido em um segundo, o agente coletor de lixo verifica duas células na horizontal e vertical, não havendo lixo é sorteada uma direção aleatória para o deslocamento do agente, após 3 ciclos sem achar lixo é sorteada uma direção fixa (linha ou coluna) para deslocamento do agente. Caso o agente reciclador esteja andando em direção fixa ou indo em direção da lixeira e esbarre em algum obstáculo, como por exemplo, outro agente ou uma lixeira, ele volta a andar aleatoriamente por três ciclos de modo a não ficar preso nos obstáculos.

Caso o agente reciclador esteja andando em direção fixa ou indo em direção da lixeira e esbarre em algum obstáculo, como por exemplo, outro agente ou uma lixeira, ele anda aleatoriamente por três ciclos de modo a não ficar preso nos obstáculos.

Problemas encontrados ao longo da implementação:

**Matriz dinâmica:** Para que a matriz que representa o ambiente possa ser redimensionada em tempo de execução, foi usada uma variável local que cria uma matriz estática cada vez que o usuário solicita a atualização do ambiente e então passa essa referência para uma matriz global.

**Passar por cima do lixo**: Foi necessário criar uma variável global para controlar a passagem do coletor por cima do lixo quando este está com o saco cheio.

1. Resultados

Após algumas execuções é possível chegar a algumas conclusões sobre a característica do trabalho proposto. São elas:

Obtém-se melhor resultado utilizando mais agentes Coletores, pois melhora o desempenho do sistema, porém deve-se ter o cuidado para não lotar o ambiente de agentes, pois os mesmos não terão espaço suficiente para se movimentar entre as células da matriz.

A posição inicial dos objetos pode interferir no desempenho final.

1. Conclusão

Este artigo apresentou agente reativo coletor de lixo, desenvolvido como primeiro trabalho para a disciplina de inteligência artificial.

1. **Referência**

RUSSELL, Stuart J.; NORVIG Peter. **Artificial Intelligence: A Modern Approach**, third edition, 2009.