

Redes Neurais Profundas como Funções Heurísticas para Algoritmos de Pathfinding

Jairo F. Gez, Claiton H. C. Neisse, Juliano L. Soares, Luis A. L. Silva

Curso de Graduação em Ciência da Computação
Universidade Federal de Santa Maria (UFSM) – Santa Maria, RS – Brasil

{jfggez, chneisse, jlsoares, luisalvaro}@inf.ufsm.br

Resumo. Algoritmos de pathfinding e redes neurais profundas têm ganho destaque na área de Inteligência Artificial (IA). O problema é que funções heurísticas tradicionalmente usadas por vários algoritmos de pathfinding não consideram as características dos mapas virtuais em que o caminho é computado. Para abordar esse problema, a proposta descrita neste artigo visa investigar como usar redes neurais profundas na construção de funções heurísticas para algoritmos de pathfinding hierárquicos, permitindo otimizar a busca de caminhos em mapas de grandes dimensões e diferentes naturezas.

1. Introdução

Redes Neurais Profundas (Deep Neural Networks - DNN) [Goodfellow 2016] são baseadas em modelos matemáticos que adquirem, armazenam e utilizam conhecimento através da experiência. Tais DNN vêm sendo exploradas na otimização de algoritmos de busca em diferentes aplicações [Ariki *et al.* 2019], [Agostinelli *et al.* 2019], [Jindal *et al.* 2017], [Li *et al.* 2016] e [Doebber 2019]. Entre outras técnicas, essa otimização tem sido voltada para a construção de funções heurísticas ajustadas para problemas de aplicação, permitindo obter melhores estimativas de custo para alcançar um estado objetivo. Neste contexto de pesquisa, [Doebber 2019] descreve como implementar, treinar e testar DNN no desenvolvimento de algoritmos de pathfinding [Algfoor *et al.* 2015]. Embora esse trabalho tenha apresentado resultados promissores, um maior conjunto de testes das técnicas de pathfinding e DNN exploradas ainda precisa ser desenvolvido. Em particular, existe a necessidade de propor e testar novas técnicas hierárquicas de pathfinding onde as funções heurísticas usadas pelos algoritmos de busca são computadas por DNN, permitindo otimizar a busca de caminhos em mapas virtuais contendo grandes espaços de busca. A proposta descrita neste trabalho, portanto, envolve explorar algoritmos hierárquicos de pathfinding a partir do emprego de DNN como suas funções heurísticas, visando realizar análises estatísticas de resultados experimentais atualmente sendo computados em diferentes tipos de mapas.

2. Revisão da Literatura

Algoritmos de pathfinding buscam, geralmente, o menor caminho trafegável deste uma posição inicial (nodo inicial) até uma posição final (nodo objetivo) usando a representação do mapa virtual. Os elementos principais da busca de caminhos são um grafo de representação do espaço de estados, um algoritmo de busca e uma função heurística para orientar a pesquisa. Funções heurísticas amplamente usadas em algoritmos de pathfinding [Algfoor *et al.* 2015], tais como distância Euclidiana e distância de Manhattan, mostram-se cada vez menos eficientes à medida que o tamanho do espaço de busca e o número de obstáculos nos terrenos virtuais usados aumentam. Neste contexto,

em [Takahashi *et al.* 2019], uma alternativa ao uso de funções heurísticas tradicionais é abordada ao propor a utilização da rede neural U-Net. Essa DNN extrai informações de imagens de mapas coletadas para o aprendizado de heurísticas utilizadas no planejamento de rotas de robôs móveis. Em [Ariki *et al.* 2019], redes neurais convolucionais são empregadas para o aprendizado de heurísticas usadas por planejadores de caminhos. A entrada da rede é uma matriz com as posições dos obstáculos e a posição objetivo. A saída da rede é uma imagem que representa o mapa heurístico, contendo todos os valores heurísticos a partir do ponto objetivo. Esse mapa heurístico é então utilizado pelo planejador. DeepCubeA [Agostinelli *et al.* 2019] combina abordagens por reforço profundo, por reforço clássico e outros métodos para treinar uma DNN para aproximar uma função heurística para estimar o custo para atingir a meta. Após o treinamento, a rede neural é capaz de gerar heurísticas para problemas com espaços de busca extremamente elevados, tal como na solução do problema do cubo mágico, por exemplo. Combinando previsões do tempo de viagem com a hora do dia da viagem, ST-NN (Rede Neural Espacial-Temporal) proposto por [Jindal *et al.* 2017]) estima distâncias entre coordenadas GPS de origem e destino. Nesse modelo, coordenadas brutas do GPS são combinadas com as informações do horário atual, possibilitando prever uma distância de viagem sem construir uma rota entre dois pontos. ANN* [Li *et al.*, 2016] é um algoritmo de busca heurística em grids baseado em DNN. O funcionamento desse algoritmo é semelhante ao A* tradicional, embora empregue uma heurística modificada gerada a partir de uma rede neural de regressão. Embora as arquiteturas de DNN utilizadas nesses trabalhos variem, eles propõem e analisam o emprego de DNN treinadas a partir de dados dos problemas alvo na redução do número de nodos expandidos durante a execução do processo de busca.

Baseado nos trabalhos descritos acima, a implementação e teste de uma DNN do tipo “feedforward” no desenvolvimento de novos algoritmos de pathfinding é detalhada em [Doebber 2019]. Embora não explorado em [Doebber 2019], o emprego de algoritmos de pathfinding hierárquicos (HPA*) é proposto em [Boeta *et al.* 2004]. Ainda sem considerar o emprego de DNN, estes algoritmos hierárquicos realizam um pré-processamento do mapa virtual, dividindo-o em clusters. A partir do conjunto de nodos que conectam as bordas desses clusters, uma estrutura abstrata de grafo permite orientar a busca hierárquica de caminhos em mapas de grandes dimensões. Em muitos sentidos, o presente trabalho pretende integrar, expandir e aprofundar as propostas apresentadas em [Doebber 2019] e [Boeta *et al.* 2004], detalhando como empregar uma abordagem hierárquica de pathfinding pelo emprego de DNN.

3. Metodologia

Tal como descrito em [Doebber 2019], resultados de pathfinding baseados no uso de DNN (Fig. 1) foram apresentados e comparados, os quais foram computados em mapas de labirintos (Fig. 2 A, para um labirinto de 100x100 nodos). Para desenvolver um maior conjunto de testes dos algoritmos de pathfinding e DNN propostos naquele trabalho, esta pesquisa deve explorar mapas virtuais com naturezas distintas (Fig. 2), todos com tamanho de 404x404 nodos (espaços de busca de 163.216 nodos): mapa virtual gerado a partir da discretização de uma imagem aérea da cidade de Berlim (Fig. 2 B), que tem aproximadamente 120.000 nodos livres para computação de caminhos, mapa virtual de uma floresta usada em um jogo virtual (Fig. 2 C), com aproximadamente 65.000 nodos livres e, finalmente, mapa virtual descrevendo o interior de um prédio (Fig. 2 D), também usado em um jogo digital, com aproximadamente 70.000 nodos livres. Para tratar os

grandes espaços de busca destes mapas, este trabalho deve explorar uma abordagem hierárquica de busca de caminhos. Esta abordagem divide o mapa virtual original em clusters, os quais são previamente analisados e armazenados em memória. Posteriormente, caminhos computados nesses clusters são unidos, em um processo de refinamento dos caminhos encontrados no primeiro passo de busca em cada um dos clusters envolvidos. Apesar de retornar caminhos sub-ótimos (1% maiores que os menores caminhos), a principal vantagem desse processo hierárquico de pathfinding é otimizar a computação de rotas em mapas extremamente grandes.

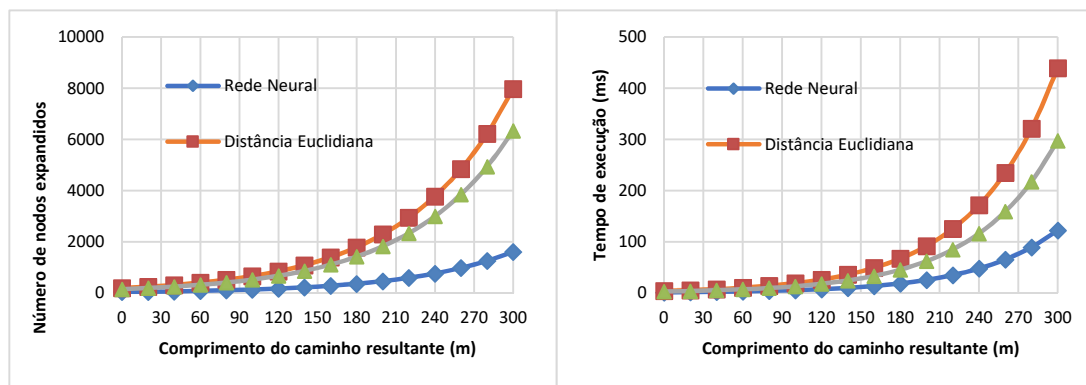


Figura 1 – Resultados de pathfinding com diferentes heurísticas [Doebber 2019].

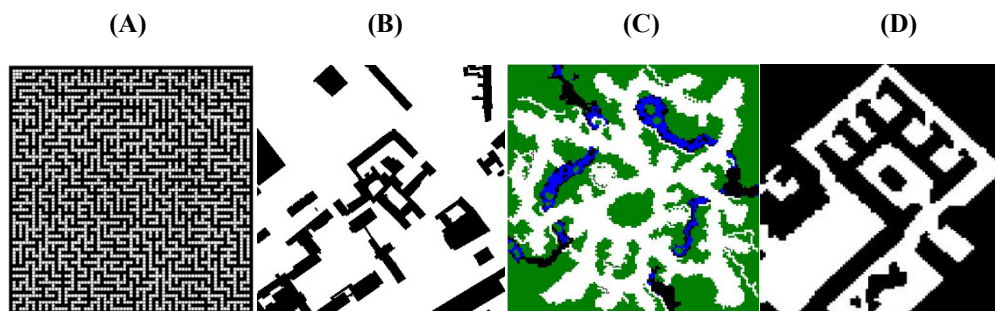


Figura 2 - Mapas virtuais a serem explorados neste trabalho.

Para gerar o dataset de treinamento das DNN que devem ser usadas como funções heurísticas para o algoritmo hierárquico de pathfinding, o algoritmo de Dijkstra deve ser usado. Para isso, a posição objetivo para este algoritmo não será determinada, permitindo gerar um grafo contendo caminhos com custos mínimos entre todos os nodos do mapa virtual (Minimum Spanning Tree). Tais caminhos devem então ser usados no treinamento das DNN, com 70% do total dos dados para treinamento e os 30% restantes para validação dos resultados. Existem desafios de treinamento das DNN que devem ser abordados neste trabalho. Por exemplo, dado um mapa virtual de (256×256) nodos, este mapa pode conter $\sim 45k$ nodos abertos (não-bloqueados) para computações de rotas. No pré-processamento deste mapa, os valores de distância associados aos menores caminhos entre pontos A e B no mapa devem ser usados como dados de treinamento da DNN. Logo, o tamanho do dataset de treinamento deve ser de $\sim 45k^2$ ($\sim 2.0 \times 10^9$ entradas). Portanto, o desafio é encontrar uma arquitetura de DNN que permita tornar viável o treinamento de um dataset com essas dimensões, como também determinar como explorar a integração de múltiplas DNN treinadas com dados oriundos de cada um dos clusters que o mapa virtual original deve ser dividido, assim como executado no processo hierárquico de pathfinding.

Funções de custo em algoritmos heurísticos de pathfinding são descritas como $f(n) = g(n) + h(n)$, onde $f(n)$ retorna o custo total do nodo (n), $g(n)$ retorna o custo do caminho até o nodo (n) e $h(n)$ retorna uma estimativa heurística de custo do nodo (n) até o nodo objetivo. Neste trabalho, a utilização das saídas das DNN treinadas devem ser exploradas de formas distintas nas computações de custo usadas pelo algoritmo hierárquico de pathfinding: (i) $f(n) = g(n) + h(n)$ onde $h(n) = DNN(n)$ e (ii) $f(n) = g(n) + h(n)$ onde $h(n) = DNN(n) * h(n)$. Essas alternativas indicam que os resultados da DNN devem ser analisados como os próprios valores heurísticos ou que os resultados da DNN podem ser multiplicados por valores heurísticos obtidos a partir do uso de funções heurísticas tradicionais. Por fim, resultados experimentais devem ser descritos e analisados via modelos estatísticos, permitindo comparar os algoritmos de pathfinding e DNN implementados.

4. Considerações Finais

Por não considerarem as características de diferentes mapas virtuais, algoritmos de busca baseados em funções heurísticas tradicionais tendem a perder eficiência conforme o tamanho dos mapas crescem. Para atacar esse problema, o presente trabalho apresenta uma proposta de implementação e experimentação de DNN como função heurística de algoritmos hierárquicos de pathfinding, utilizando sua capacidade de reconhecimento de padrões para aprimorar essas estimativas. O objetivo é verificar se tais funções heurísticas geradas a partir dos resultados das DNN permitem ou não obter melhores estimativas na determinação de rotas de menor custo em diferentes tipos de mapas.

Referências

- Agostinelli, F., McAleer, S. e Shmakov, A. (2019) "Solving the Rubik's cube with deep reinforcement learning and search." *Nature Machine Intelligence*, v. 1, n. 8, p. 356-363.
- Algfoor A., Sunar M. S. e Kolivand H. (2015) "A comprehensive study on pathfinding techniques for robotics and video games," *Int. Journal of Computer Games Technology*.
- Ariki, Y. e Narihira T. (2019) "Fully Convolutional Search Heuristic Learning for Rapid Path Planners," *arXiv preprint arXiv:1908.03343*.
- Botea, A., Muller, M. e Schaeffer, J. (2004) "Near optimal hierarchical path-finding". *Journal of game development*, v. 1, n. 1, p. 7-28.
- Doebber, Daniel M. (2019) "Uso de redes neurais profundas para o aprendizado de funções heurísticas para algoritmos de busca de caminhos." Trabalho de Conclusão do Curso em Ciência da Computação - Universidade Federal de Santa Maria, Rio Grande do Sul.
- Goodfellow I., Bengio Y. e Courville A. (2016) "Deep learning". MIT press.
- Jindal, I. Qin, T e Xueue, Q. (2017) "A unified neural network approach for estimating travel time and distance for a taxi trip". *arXiv preprint arXiv:1710.04350*.
- Li, G., Wang, H., Wang, N. e Yeung Y. (2016) "ANN: a heuristic search algorithm based on artificial neural networks". *Proceedings of the 2016 International Conference on Intelligent Information Processing*. ACM. p.51.
- Takahashi T., Sun H., Tian T. e Wang Y. (2019) "Learning Heuristic Functions for Mobile Robot Path Planning Using Deep Neural Networks". *Proceedings of the International Conference on Automated Planning and Scheduling*, 2019. p.764-772.