Universidade: presente!

UFRGS PROPESQ

XXXI SIC

Salão UFRGS 2019 — Conhecimento o Formação o Inovação

21 . 25 . OUTUBRO . CAMPUS DO VALE

Detecção de Deadlocks em Sokoban usando Redes Neurais

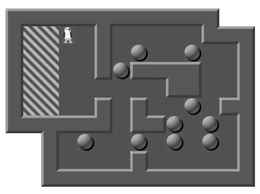
Mateus Davi Simon

Introdução

O objetivo da área de planejamento automatizado é alcançar a capacidade humana em solução de problemas. Uma *tarefa de planejamento* é uma tupla $\Pi = \langle \mathcal{V}, s_0, \mathcal{A}, S^* \rangle$, onde $\mathcal{V}$ são as variáveis de estado, s_0 é um estado inicial, $\mathcal{A}$ é um conjunto de ações e S^* é um conjunto de estado objetivos. Uma tarefa Π induz um *espaço de estados*: um grafo direcionado de todos os estados gerados por atribuições de valores as variáveis. Existe um arco nesse grafo de s para s' sse existe uma ação $a \in \mathcal{A}$ que quando aplicada ao estado s gera o estado s'. Uma *solução* para uma tarefa Π (um estado s) é uma sequência de ações que quando aplicada ao estado inicial s_0 (ao estado s) gera um estado objetivo $s^* \in S^*$. As abordagens mais efetivas para solução de tarefas de planejamento usam uma busca no espaço de estados guiada por uma função heurística h. Os estados de uma tarefa Π são classificados como *alive* se possuem solução e *dead* caso contrário. O problema de detecção de estados *dead* é essencial para a solução de tarefas de planejamento e os métodos mais efetivos usam as funções heurísticas de abstração *pattern databases* h^{PDBk}. Heurísticas h^{PDBk} representam perfeitamente a informação de subproblemas limitados por um subconjunto de $k = |V|$ variáveis $V \subseteq \mathcal{V}$. Assim, heurísticas h^{PDBk} detectam estados *dead* causados pela interação de no máximo k variáveis. Propomos um modelo de classificação baseado em aprendizado para detecção estado *dead* causados pela interação de mais do que k variáveis. Para o treinamento do modelo, é necessário um conjunto de treinamento com rótulos *alive* ou *dead*. O ponto crucial desse trabalho é determinar como gerar o conjunto de treinamento que maximiza o desempenho de modelo de classificação durante a busca.

Sokoban

Utilizamos o domínio *Sokoban* como caso teste para nossa abordagem. Sokoban é um domínio PSPACE-completo e uma das tarefas de planejamento mais desafiadoras para planejadores automatizados.



Uma instância de Sokoban.

- **Movimento:** o *homem* pode andar pelo labirinto e empurrar uma pedra para uma posição livre adjacente.
- **Solução:** uma sequência de ações que coloca cada pedra em uma posição objetivo distinta.
- **Objetivo:** encontrar uma solução com o mínimo número de movimento de pedras.

O modelo de rede neural utilizado foi *feed forward* com quatro camadas de 256 neurônios. Nos testes, utilizamos o A* com o *EMM* como heurística e podando estados classificados como *deadlock* pela rede.

Geração de Deadlocks

Propomos duas abordagens para geração de estados *dead*:

- **Random Samples**: Gerar estados aleatórios e verificando usando h^{PDBk} se o estado gerado é *dead*.
- **Random Walk**: Para cada estado *alive* gerado realizamos uma busca local até um estado *dead* ser encontrado, novamente fazemos essa verificação com h^{PDBk}. O objetivo é gerar estados *dead* mais similares aos estados gerados durante a busca para resolver a tarefa Π.

Método	Verd. Negativo	Falso Positivo	Falso Negativo	Verd. Positivo
Random Samples	5178.97	0.09	18397.91	172946.68
Random Walk	5178.97	0.09	8078.50	183266.09

Matriz de confusão para detecção de deadlocks usando os dois métodos de geração de deadlocks. Positivo significa que a rede classificou os estado como deadlock

Método	Solucionadas	Custo	Gerado	Expandido	Tempo
Random Samples	33	29.69	21423.47	15806.97	40.03
Random Walk	32	29.69	12011.22	8525.50	22.34

Resultados da busca usando os dois métodos. Foram usadas 33 instancias no teste, nas colunas Custo, Gerado, Expandido e Tempo é a média dos valores resolvidos por todos métodos.

Geração de Estados Alive

Para geração dos estados *alive* propomos quatro métodos baseados em busca *backward* a partir do conjunto de estados objetivo S^* – estados gerados dessa forma possuem garantidamente solução:

- **Breadth-First Search** (BFS): gera estados estados próximos ao conjunto de estados objetivo S^*.
- **A***: usa como estado objetivo o estado inicial s_0. Gera estados que seriam gerados durante a busca para resolver a tarefa Π.
- **Greedy Best-First Search** (GBFS): maximiza valor-h em relação ao conjunto S^*. Gera estados distantes do conjunto de estados objetivo S^*.
- **Reverse GBFS** (rGBFS): minimiza o valor-h em relação ao estado inicial s_0. Gera estados distantes de S^* e próximos ao estado inicial s_0.

Método	Verd. Negativo	Falso Positivo	Falso Negativo	Verd. Positivo
ASTAR	15454.33	2994.67	95096.89	644136.11
BFS	7518.78	10930.22	56436.78	682796.22
GBFS	12705.22	5743.78	63310.89	675922.11
RGBFS	11753.78	6695.22	52546.78	686686.22

Matriz de confusão para detecção de deadlocks usando os quatro métodos de geração de alive. Positivo significa que a rede classificou os estado como deadlock

Método	Solucionadas	Custo	Gerado	Expandido	Tempo
ASTAR	7	38.17	52047.83	30975.83	108.67
GBFS	7	38.17	26214.67	16422.33	54.70
RGBFS	8	38.17	30793.67	19260.00	60.12

Resultados da busca usando os três métodos. Foram usadas 9 instancias no teste, nas colunas Custo, Gerado, Expandido e Tempo é a média dos valores resolvidos por todos métodos. O BFS não foi incluído pois não resolveu nenhuma instancia.

Restrição de Recursos

Para melhor entender o comportamento do método em problemas reais – onde o espaço costumam ser muito grandes e apenas uma pequena parte pode ser gerada para o treinamento – testamos o algoritmos restringindo o número de estados a uma porcentagem do número de estados alive da instância.

Porcentagem	Verd. Negativo	Falso Positivo	Falso Negativo	Verd. Positivo
5.00%	64879	101162	81840	6571258
10.00%	93674	72367	98111	6554987
20.00%	94021	72020	56858	6596240
50.00%	164531	1510	178007	6475091
100.00%	166041	0	273761	6379337

Matriz de confusão para detecção de deadlocks usando os diferentes porcentagem do número total de estados alive. Positivo significa que a rede classificou o estado como deadlock.

Conclusão

Neste trabalho mostramos o impacto que diferentes algoritmos de geração de estados *alive* e *dead* têm no treinamento de uma rede neural de classificação. Destacamos que a geração de estados similares aos vistos na busca levam a um melhor desempenho da rede.

- Investigar métodos para minimizar falsos positivos: mesmo tendo um número muito pequeno de falsos positivos, uma única classificação errada é suficiente para impedir que uma solução seja encontrada, por isso é crucial minimizar esse tipo de erro.
- Desenvolver algoritmos que levem em conta a imprecisão da rede: alternativamente ou em complemento ao item acima, poderia ser desenvolvido um algoritmo utilize a informação da rede mas sem comprometer a busca por falsos positivos.
- Utilizar outros modelos de redes neurais.