

Uma Inteligência Artificial para Jogar o Jogo Dino

Enzo B. Cussuol¹

Vitória, Espírito Santo

Abstract

Esse artigo apresenta uma Inteligência Artificial para jogar o jogo do Dino da Google. Para se obter essa IA, utilizou-se a metodologia de aprendizado por reforço, fazendo-se uso de uma rede neural como classificador e um algoritmo genético como meta heurística de aperfeiçoamento da rede neural. Os resultados foram orquestrados em tabelas e gráficos.

1. Introdução

Algoritmos classificadores relacionados à Inteligência Artificial [1] vem ganhando destaque na literatura e no mercado de trabalho nas últimas décadas devido ao grande potencial de realizar predições acuradas para certos tipos de problemas. Um subconjunto desses classificadores, as redes neurais [2], pode ser considerado o mais popular, visto que é amplamente utilizado não só para trabalhos acadêmicos, mas também por grandes empresas tais como Google e Facebook para realizar recomendações aos seus usuários.

Esses classificadores geralmente são submetidos à sequências de fases de treino e teste, nas quais os resultados obtidos pela fase de treino são aplicados à fase de teste, o que irá produzir um resultado que irá, teoricamente, ser melhorado à medida que as fases são iteradas. Contudo, esse cenário é impossível de ser implementado quando o problema não possui rótulos, uma vez que não existem fases de teste. Nesse caso, os classificadores podem ser implementados utilizando a metodologia de aprendizado por reforço [3] em conjunto com uma meta heurística [4].

Neste trabalho, foi implementada uma rede neural, a qual realizou o papel de classificador, e um algoritmo genético, o qual realizou o papel de meta heurística, para jogar o jogo Dino da Google.

O restante deste artigo está organizado como se segue: A Seção 2 descreve o classificador. A Seção 3 discute a meta heurística utilizada. A Seção 4 aborda os experimentos realizados e resultados obtidos. Finalmente, a Seção 5 apresenta as conclusões.

¹enzo.cussuol@edu.ufes.br

2. Descrição do Classificador

20 Como mencionado na Introdução, o classificador escolhido foi uma rede neural. A rede neural assumiu a responsabilidade de, a cada quadro do jogo, escolher uma entre três possíveis teclas a serem apertadas, o que implica na movimentação do dinossauro. Dito isso, é natural que a rede neural implementada possua 3 neurônios na camada de saída, que estão associados às teclas KEY_UP, KEY_DOWN e KEY_NO.

25 Quanto à camada de entrada, foram utilizados 4 neurônios, que representam as seguintes medidas de um quadro do jogo: distância do dinossauro ao obstáculo, comprimento do obstáculo, altura do obstáculo e velocidade do jogo. Inicialmente existia mais um neurônio que associava-se à altura do dinossauro, mas chegou-se a conclusão de que essa característica não estava sendo relevante nos resultados.

30 Com as camadas de entrada e saída construídas, resta definir as camadas ocultas. Por se tratar de um problema relativamente simples, foi utilizada apenas uma camada oculta. Com relação ao número de neurônios nessa camada, utilizou-se a regra de polegar que diz que o tamanho de uma camada interna deve ser igual a $2/3$ o tamanho da camada de entrada mais a de saída, logo, foram utilizados 5 neurônios.

35 Além disso, foi definido que os pesos assumiriam valores entre -1 e 1. Ademais, na camada oculta e na camada de saída foram aplicadas funções de transformação linear seguidas de uma função de ativação, a qual foi escolhida a função retificador, que faz com que neurônio seja ativado somente se a entrada for positiva.

Para implementar a rede neural na prática foi utilizada a biblioteca pytorch [5], que facilita a
40 criação da rede, a definição dos pesos, entre outras funcionalidades.

3. Descrição da Meta Heurística

Assim como discutido na Introdução, a meta heurística escolhida foi o algoritmo genético. Esse algoritmo ficou responsável por buscar o melhor conjunto de pesos da rede neural. Para isso, foi definido que cada indivíduo, ou cromossomo, ou estado, representaria um conjunto de pesos da rede.

45 Com relação à cada etapa do algoritmo genético, primeiro temos a avaliação da população, que consiste em jogar o jogo para cada indivíduo, isto é, definir os pesos da rede neural como sendo os pesos que o indivíduo representa, e receber uma pontuação de retorno. Indivíduos mais aptos são aqueles que obtiverem maior pontuação. Nessa parte, foi adotada a política de jogar o jogo três vezes por indivíduo, a fim de evitar situações de sorte ou azar.

50 Depois foi implementado o elitismo, que simplesmente escolhe os melhores indivíduos e os garante na próxima geração. Posteriormente, a seleção é feita baseando-se no método da roleta com pesos,

onde indivíduos mais aptos possuem mais chances de serem sorteados. Em seguida, realizam-se os cruzamentos, nos quais é definido um ponto de corte e são gerados dois filhos que são combinações do pai e da mãe cortados nesse ponto.

55 Por fim, ocorre a mutação, que escolhe uma certa quantidade de indivíduos para realizar uma troca no valor de seus genes. Foi definido uma taxa de alteração em 10% dos genes de um dado cromossomo. O novo valor do gene é somado à um número aleatório que varia entre -1 e 1, que é a faixa de possíveis valores para um peso da rede neural. Se a soma resultar em um valor maior que 1, o novo valor será -1 mais o valor excedente e, se resultar em um valor menor que -1, o novo valor será 1 menos o valor
60 excedente.

Com relação aos parâmetros do algoritmo, vários valores foram testados, mas os que obtiveram melhores resultados foram: 90% como taxa de cruzamento, 10% como taxa de mutação e 3 indivíduos a serem selecionados no elitismo. O número de indivíduos por população ficou em 100, uma vez que esse foi o valor limite para o qual a execução do jogo em paralelo não apresentava problemas de atraso
65 e travamento.

4. Resultados

Para se obter os resultados, treinou-se o dinossauro, isto é, executou-se a meta heurística por 8 horas. Após o término das 8 horas, foi obtido o estado ótimo e o valor ótimo. O estado ótimo é a melhor configuração de pesos para a rede neural que a meta heurística foi capaz de encontrar. Já o
70 valor ótimo é o melhor resultado obtido por um indivíduo durante a execução da meta heurística.

O valor ótimo foi 2280 e o estado ótimo (configuração ótima dos pesos) pode ser encontrado no arquivo *best_results.txt* no diretório do código-fonte, uma vez que o inserir aqui excederia o máximo de páginas permitidas.

Após a obtenção desses valores, foi realizada a validação que consistiu em jogar o jogo 30 vezes
75 com o estado ótimo. Essa validação retornou 30 resultados, os quais encontram-se na Tabela 1, que os compara com os resultados obtidos pelo professor. Na tabela da direita, a penúltima linha é a média e a última linha é o desvio padrão dos resultados, destacados em negrito.

Podemos organizar esses resultados em um gráfico boxplot, o qual encontra-se na Figura 1.

Além disso, foi elaborada a Tabela 2, a qual mostra os resultados dos testes de hipótese entre os
80 resultados do professor e meus resultados. Na matriz triangular superior estão os resultados do teste t pareado e na matriz triangular inferior estão os resultados do teste não paramétrico de Wilcoxon [6]. Os valores em negrito aceitaram a hipótese nula para um nível de significância de 95%, isto é, para aquele par de resultados, pode-se dizer que eles apresentaram resultados significativamente diferentes.

Resultados do Professor	Meus Resultados	Resultados do Professor	Meus Resultados
1214.0	2100.0	1359.5	2208.5
759.5	2249.25	1000.25	2010.75
1164.25	1932.25	1284.5	1932.5
977.25	1869.75	1350.0	2244.0
1201.0	1819.0	751.0	1596.75
930.0	2389.75	1418.75	2301.75
1427.75	1591.5	1276.5	1799.5
799.5	1925.5	1645.75	2263.75
1006.25	1985.5	860.0	1401.0
783.5	2067.25	745.5	1627.0
728.5	2205.0	1426.25	2321.25
419.25	1669.0	783.5	1824.0
1389.5	1727.75	1149.75	2056.0
730.0	1948.0	1482.25	1876.0
1306.25	1440.25	1068.18	1951.28
675.5	2156.0	304.04	262.81

Table 1: Comparação dos 30 resultados da validação

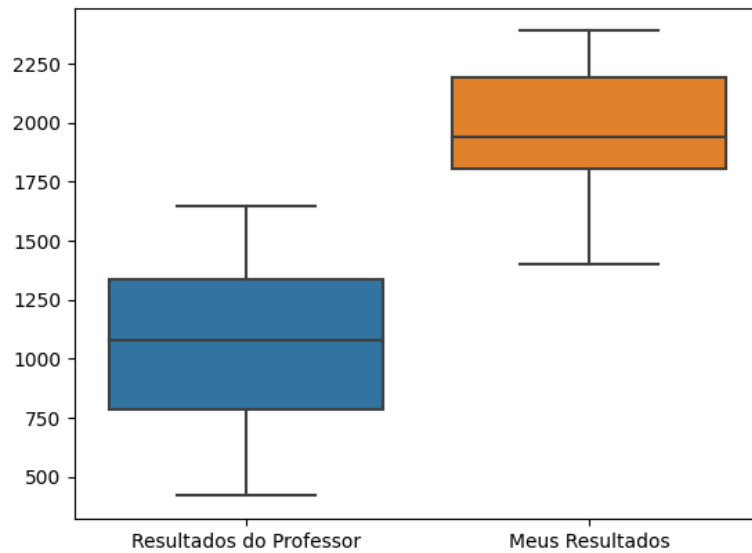


Figure 1: Comparação boxplot entre os resultados do professor e meus resultados

Resultados do Professor	0.0000000000
0.0000017333	Meus Resultados

Table 2: Tabela pareada dos resultados dos testes de hipótese entre os resultados

5. Conclusões

5.1. *Análise geral dos resultados*

Como discutido na Seção 4, os resultados obtidos foram significativamente melhores que os resultados divulgados pelo professor, chegando a uma média de quase 2000 pontos, que é uma pontuação relativamente boa, mas não ótima.

5.2. *Contribuições do Trabalho*

Esse trabalho apresentou uma Inteligência Artificial baseada em aprendizado por reforço que é utilizada para jogar o jogo Dino. Os resultados obtidos foram melhores aos obtidos pelo classificador do professor, que não possuía um critério de aprendizado.

5.3. *Melhorias e trabalhos futuros*

Como trabalhos futuros, convém a análise do desempenho da IA quando submetida à mais horas de treino, assim como quando os valores dos parâmetros da meta heurística são modificados. O presente trabalho apresentou o melhor resultado que o autor foi capaz de encontrar, contudo, isso não significa que ele é o melhor possível.

References

- [1] M. Majnik, Z. Bosnić, Roc analysis of classifiers in machine learning: A survey, Intelligent data analysis 17 (3) (2013) 531–558.
- [2] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, H. Arshad, State-of-the-art in artificial neural network applications: A survey, Heliyon 4 (11) (2018) e00938.
- [3] L. P. Kaelbling, M. L. Littman, A. W. Moore, Reinforcement learning: A survey, Journal of artificial intelligence research 4 (1996) 237–285.
- [4] I. H. Osman, J. P. Kelly, Meta-heuristics: an overview, Meta-heuristics (1996) 1–21.
- [5] V. Subramanian, Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch, Packt Publishing Ltd, 2018.
- [6] R. F. Woolson, Wilcoxon signed-rank test, Wiley encyclopedia of clinical trials (2007) 1–3.