Solving Zhed Using Reinforcement Learning

Daniel Brandão  
*Departamento de Engenharia Informática*  
*Faculdade de Engenharia da Universidade do Porto* Porto, Portugal   
[up201705812@fe.up.pt](mailto:up...@fe.up.pt)

Gaspar Pinheiro  
*Departamento de Engenharia Informática*  
*Faculdade de Engenharia da Universidade do Porto* Porto, Portugal

[up201704700@fe.up.pt](mailto:up...@fe.up.pt)

Pedro Moás  
*Departamento de Engenharia Informática*  
*Faculdade de Engenharia da Universidade do Porto* Porto, Portugal   
[up201705208@fe.up.pt](mailto:up201705208@fe.up.pt)

*Abstract*— Este artigo contém a descrição de um trabalho no âmbito da unidade curricular de Inteligência Artificial do Mestrado Integrado em Engenharia Informática e Computação na FEUP. O objetivo deste é treinar um modelo de IA para que seja capaz de resolver vários níveis do puzzle Zhed, utilizando algoritmos de Aprendizagem por Reforço. Inicialmente será formulado o problema, seguindo-se uma descrição dos algoritmos utilizados, concluindo-se com a análise de resultados experimentais, comparando-os entre si, tendo em conta fatores como a recompensa obtida, a entropia, percentagem de vitórias e *hit*.

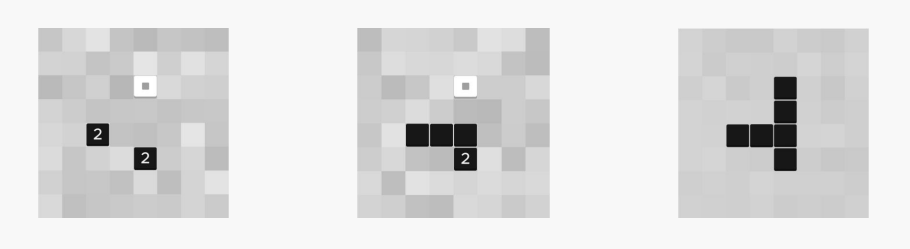
Palavras-chave— Inteligência Artificial, Machine Learning, Aprendizagem por Reforço, ML-Agents, Proximal Policy Optimization, PPO, Soft Actor Critic, SAC

# Introdução

A Aprendizagem por Reforço é uma área de Aprendizagem Computacional (Machine Learning), focada na forma como um agente toma ações num ambiente, procurando maximizar uma recompensa cumulativa que é atribuída a cada ação efetuada. Neste trabalho, foram aplicados algoritmos desta área, com o objetivo de treinar um modelo de Inteligência Artificial que seja capaz de resolver vários níveis do puzzle Zhed.

# Descrição do problema

O puzzle Zhed é um jogo de tabuleiro em que para passar cada nível é necessário que uma célula expandida atinja uma célula objetivo. Cada célula pode ser expandida numa de quatro direções, podendo sobrepôr outras células. Cada célula expande n células na direção escolhida, decrementando n em um por cada célula vazia atingida. Quando uma célula expandida sobrepôe uma outra não vazia, o valor de n não é decrementado, aumentando em um o valor total de células atingiveis pela célula numerada.



1. Exemplo de um nível Zhed.

# Formulação do problema

Para cada nível do puzzle, as posições das células numeradas, da célula objetivo e o tamanho do tabuleiro variam. Assim, usamos uma matriz (lista de listas) M, de dimensões NxN, sendo N então usado para descrever o tamanho do tabuleiro. Os valores guardados na matriz foram escolhidos de forma a facilitar a manipulação do estado atual da célula, podendo cada uma ter o seu valor, *val*, igual a:

* Um número positivo, representando uma peça com valor *val*, o tamanho expandível da célula.
* 0, representado uma célula vazia.
* -1, representando uma céula expandida.
* -2, representando uma peça de destino.
* -3, representando uma peça de destino alcançada.

Com isto, temos que o estado inicial do tabuleiro consiste numa matriz em que:

* Existe pelo menos uma célula com *val* = -2;
* Existe pelo menos uma célula com *val* > 0;
* Não existe nenhuma célula com *val* = -1 ou *val* = -3.

Por outro lado, o estado final do tabuleiro é representado por uma matrix em que:

* Existe pelo menos uma célula com val = -3.

# Trabalho relacionado

No sentido de pesquisar sobre trabalho relacionado com o nosso tema, procuramos projetos que tinham por objetivo aplicar algoritmos de Aprendizagem por Reforço em puzzles como o 2048 [2] ou Tetris [4], [5], assim como outros puzzles [1], [3].

# Ferramentas

Para implementar o jogo, utilizou-se a plataforma de criação de videojogos Unity3D, com o plugin ML-Agents (versão 1.0.2), que nos permitiu aplicar os algoritmos de Aprendizagem por Reforço.

TODO: moas poe print

# Algoritmos

Esta secção detalha os algoritmos utilizados no Zhed Solver.

## Algoritmos de aprendizagem por reforço

Neste trabalho, foram implementados os algoritmos disponíveis no ML-agents: Proximal Policy Optimization (PPO) e Soft Actor Critic (SAC). Estes diferem entre si na medida em que o algoritmo SAC é um algoritmo off-policy, sendo que, neste, o computador poderá aprender com as experiências obtidas em qualquer episódio anterior. Em ambos os casos, o programa tenta calcular qual a melhor ação a tomar (*policy*) para um determinado estado observado, obtendo feedback das suas ações através de uma recompensa.

Outro aspeto no qual os algoritmos PPO e SAC se diferenciam é nos hiperparâmetros disponíveis, sendo possível variá-los de modo a obter melhores (ou piores) resultados. Os mais analisados para o algoritmo PPO foram: λ (lambda), que pode ser visto como o quão o agente valoriza futuras recompensas (visão a longo prazo), ε (epsilon), que influencia o quão rapidamente a *policy* pode evolui, e β (beta), que determina a força da entropia durante o treino. Para o algoritmo SAC, os parâmetros analisados foram: init\_entcoef, um regularizador da entropia no início do treino, e τ (tau), que corresponde à magnitude da alteração de Q durante a atualização do modelo SAC.

## Abordagem **Wild**

Aplicando os algoritmos ao problema descrito, foram implementadas, essencialmente, duas abordagens. Na primeira, o agente faz uma única observação ao meio, que é a matriz de tabuleiro inteira, tendo de escolher os seguintes valores para a sua ação:

* Coordenadas onde jogar
* Direção da jogada (Cima, baixo, esquerda, direita)

Tanto o espaço de observação como o espaço de ações é, então, discreto. O agente é recompensado se a jogada for válida, mas punido ligeiramente (cerca de 20 vezes menos, em valor absoluto) se não o for, recebendo uma recompensa um pouco maior se o tabuleiro resultante ficar mais próximo de estar resolvido. Finalmente, vitórias e derrotas resultam, do mesmo modo, em recompensas positivas e negativas, respetivamente.

## Abordagem **Regular**

Na segunda abordagem, o agente efetua as seguintes observações do meio:

* Peças disponíveis para jogar (coordenadas e valor)
* Coordenadas da peça de destino

O agente tem de escolher os seguintes valores para a sua ação:

* Peça a jogar
* Direção da jogada (Cima, baixo, esquerda, direita)

Tal como na abordagem anterior, trata-se, portanto, de um espaço discreto. Neste caso, a recompensa dá muito mais ênfase à qualidade das jogadas, beneficiando aquelas que mais aparentam levar a uma solução. Para isso, foi utilizada uma função de avaliação do tabuleiro que soma a média dos valores de extensão de cada peça, penalizando fortemente os casos em que não há peças alinhadas com a peça de destino. Do mesmo modo, também estão definidas recompensas para vitórias e derrotas.

No geral, o processo de aprendizagem variará bastante, pois, enquanto na abordagem *Wild* os agentes estarão inicialmente a tentar descobrir quais são as coordenadas certas, e só depois resolver o puzzle, na abordagem *Regular* o foco estará sempre em resolver o puzzle baseado nas peças que tem à disposição. Em ambas as abordagens, os episódios terminam quando o tabuleiro chega a um estado de derrota ou vitória.

## Some Common Mistakes

* The word “data” is plural, not singular.
* The subscript for the permeability of vacuum **0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter “o”.
* In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
* A graph within a graph is an “inset”, not an “insert”. The word alternatively is preferred to the word “alternately” (unless you really mean something that alternates).
* Do not use the word “essentially” to mean “approximately” or “effectively”.
* In your paper title, if the words “that uses” can accurately replace the word “using”, capitalize the “u”; if not, keep using lower-cased.
* Be aware of the different meanings of the homophones “affect” and “effect”, “complement” and “compliment”, “discreet” and “discrete”, “principal” and “principle”.
* Do not confuse “imply” and “infer”.
* The prefix “non” is not a word; it should be joined to the word it modifies, usually without a hyphen.
* There is no period after the “et” in the Latin abbreviation “et al.”.
* The abbreviation “i.e.” means “that is”, and the abbreviation “e.g.” means “for example”.

An excellent style manual for science writers is [7].

# Using the Template

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

## Authors and Affiliations

**The template is designed for, but not limited to, six authors.** A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### For papers with more than six authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

### For papers with less than six authors: To change the default, adjust the template as follows.

#### Selection: Highlight all author and affiliation lines.

#### Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

#### Deletion: Delete the author and affiliation lines for the extra authors.

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Figures and Tables

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)
2. Example of a figure caption. (*figure caption*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

1. H. Oonishi, and H. Iima, “Improving Generalization Ability in a Puzzle Game Using Reinforcement Learning”, IEEE’s 2017 Conference on Computational Intelligence in Games. Available: http://www.cig2017.com/wp-content/uploads/2017/08/paper\_71.pdf [Accessed: May 24th, 2020]
2. J. Amar, and A. Dedieu, “Deep Reinforcement Learning for 2048”, Massachusetts Institute of Technology. Available: http://www.mit.edu/~amarj/files/2048.pdf [Accessed: May 24th, 2020]
3. D. Budakova, and V. Vasilev, “Applying Reinforcement learning to find the logic puzzles solution”, Technical University of Sofia, Plovdiv Branch. Available: https://journals.tu-plovdiv.bg/index.php/journal/article/view/16/14 [Accessed: May 24th, 2020]
4. M. Stevens, and S. Pradhan, “Playing Tetris with Deep Reinforcement Learning”, Stanford University. Available: http://cs231n.stanford.edu/reports/2016/pdfs/121\_Report.pdf [Accessed: May 24th, 2020]
5. N. Faria, “A deep reinforcement learning bot that plays tetris”, https://github.com/nuno-faria/tetris-a