Arthur de Senna Rocha

Desenvolvimento de uma Inteligência Artificial para aprender a jogar jogos em Allegro

Brasil

13 de outubro de 2020

Arthur de Senna Rocha

Desenvolvimento de uma Inteligência Artificial para aprender a jogar jogos em Allegro

Trabalho de Conclusão de Curso I apresentado como requisito parcial à obtenção de título de bacharel em Engenharia de Sistemas pela Escola de Engenharia da Universidade Federal de Minas Gerais.

Universidade Federal de Minas Gerais – UFMG

Escola de Engenharia

Engenharia de Sistemas

Orientador: Pedro Olmo Stancioli Vaz De Melo

Brasil

13 de outubro de 2020

Histórico de Revisões

Versão (xx.yy)	Data (dd/mm/yyyy)	Autor	Descrição
1.0	14/11/2019	Arthur de Senna Rocha	Texto inicial da monografia TCC I
1.1	04/12/2019	Arthur de Senna Rocha	Texto final da monografia TCC I
2.0	03/04/2020	Arthur de Senna Rocha	Adiciona Visão Geral TCC II
2.1	21/08/2020	Arthur de Senna Rocha	Atualiza Cronograma (ERE)

Resumo

O uso de inteligência artificial (IA) e de algoritmos de *machine learning* possibilita que máquinas aprendam com experiências, se ajustem à novas entradas de dados e performem tarefas como seres humanos. Com essas tecnologias, os computadores podem ser treinados para cumprir tarefas específicas ao processar grandes quantidades de dados e reconhecer padrões nesses dados. O presente trabalho se propõe a desenvolver uma IA capaz de aprender a jogar diferentes jogos, desde que se tenha acesso ao código fonte e feito em Allegro. Para isso, será implementado um algoritmo de *Deep Reinforcement Learning*, abordagem que consiste em fornecer ao sistema parâmetros relacionados ao seu estado e uma recompensa positiva ou negativa com base em suas ações. Nenhuma regra sobre o jogo é dada e, inicialmente, a IA não tem informações sobre o que precisa fazer. A única informação passada para a IA são os comandos básicos do jogo. O objetivo do sistema é descobrir e elaborar uma estratégia para maximizar a pontuação - ou a recompensa. Diferente de muitas IAs que focam na solução de um único problema, a proposta deste projeto é elaborar uma IA que seja genérica e capaz solucionar e elaborar estratégias para uma variedade de situações diferentes.

Palavras-chave: deep learning, allegro, inteligência artificial, jogos digitais, machine learning.

Abstract

The use of Artificial intelligence (AI) and machine learning algorithms enables computers to learn from experience, adjust to new data inputs, and perform tasks as human beings. With these technologies, computers can be trained to perform specific tasks by processing large amounts of data and recognizing patterns in that data. The present work aims to develop an AI capable of learning how to play different games, provided that it has access to the source code and the game is made in Allegro. For this, a Deep Reinforcement Learning algorithm will be implemented, which provides the system with parameters related to its state and a positive or negative reward based on its actions. No rules about the game are given and initially, the AI has no information on what it needs to do. The only information passed to AI is the basic commands of the game. The goal of the system is to discover and devise a strategy to maximize its score - or the reward. Unlike many AIs that focus on solving a single problem, the purpose of this project is to design a generic AI that can solve and develop a strategy for a variety of different situations.

Keywords: deep learning, allegro, artificial intelligence, video games, machine learning.

Lista de ilustrações

Lista de abreviaturas e siglas

ALE	*Allegro Learning Enviroment*
ANNs	*Artificial Neural Networks*
ASR	*Automatic Speech Recognition*
DL	*Deep Learning*
DQN	*Deep Q-network*
GMM	*Gaussian Mixture Model*
HMM	*Hidden Markov Model*
IA	Inteligência Artificial
ML	*Machine Learning*
NPC	*Non-player Character*
PCG	*Procedural Content Generation*
PNL	Processamento de Linguagem Natural
RL	*Reinforcement Learning*

Sumário

1 Introdução

A inteligência artificial (IA) vem ganhando manchetes no mundo todo, sendo anunciada tanto como uma salvação econômica quanto como precursora de desintegração social (ROBU et al., 2019). Quando computadores programáveis foram concebidos pela primeira vez, as pessoas se perguntavam se essas máquinas poderiam se tornar inteligentes, mais de cem anos antes de uma ser construída (MENABREA et al., 1843). Hoje, a inteligência artificial é um campo com inúmeras aplicações práticas e tópicos de pesquisa ativos. Buscamos softwares inteligentes para automatizar o trabalho de rotina, entender a fala ou as imagens, fazer diagnósticos em medicina e apoiar a pesquisa científica (GOODFELLOW; BENGIO; COURVILLE, 2016).

A IA adiciona inteligência a produtos existentes. Na maioria dos casos, a inteligência artificial não é vendida como uma aplicação individual. Pelo contrário, produtos já existentes são aprimorados com funcionalidades de IA, de maneira parecida como a Siri foi adicionada aos produtos da *Apple*. Automação, plataformas de conversa, robôs e aparelhos inteligentes podem ser combinados com grandes quantidades de dados para aprimorar diversas tecnologias para casa e escritório, de inteligência em segurança à análise de investimentos.

A maioria dos exemplos de IA sobre os quais se ouve falar hoje – de computadores mestres em xadrez a carros autônomos – dependem de *deep learning* e processamento de linguagem natural (PNL) (RODRIGUES, 2017). Treinar um agente para superar os jogadores humanos e otimizar sua performance pode nos ensinar como otimizar diferentes processos em uma grande variedade de situações. Foi o que o *DeepMind* do Google fez com seu popular *AlphaGo* e seu sucessor *AlphaZero*, vencendo os campeões mundiais em Go, xadrez e shogi, e obtendo resultados de performance nunca antes vistos.

1.1 Motivação

Técnicas de aprendizado de máquina e algoritmos de *deep learning* (DL) têm consistentemente melhorado a capacidade de um computador de fornecer reconhecimento de padrões e previsões cada vez mais precisas. Além disso, sistemas de DL são consistentemente aplicados com sucesso a conjuntos de aplicações cada vez mais amplos.

Ao mesmo tempo em que a escala e a precisão das redes neurais aumentaram, a complexidade das tarefas que podem ser resolvidas também cresceu significativamente. Uma conquista importante de sistemas de DL é a sua extensão ao domínio da aprendizagem por reforço ou *reinforcement learning* (RL) (SUTTON; BARTO, 2018). No contexto do

aprendizado por reforço, um agente autônomo deve aprender a executar uma tarefa por tentativa e erro, sem nenhuma orientação do operador humano.

Além do valor para pesquisa em múltiplas áreas da ciência, muitas dessas aplicações de aprendizado de máquina e *deep learning* são altamente lucrativas. O aprendizado de máquina hoje é usado por muitas empresas de tecnologia, incluindo *Google, Microsoft, Facebook*, IBM, *Baidu, Apple, Adobe, Netflix*.

Diante à crescente presença de sistemas que utilizam técnicas de *deep learning* no dia-a-dia, nota-se o grande potencial do investimento em pesquisa, modelagem de novos problemas e estudo de técnicas de aprendizado de máquina. Uma interessante aplicação desses sistemas está na área de jogos digitais. A indústria de videogames tem testemunhado um enorme crescimento, graças, em boa parte, ao incrível aumento no poder da computação em termos de representações visuais. Seja no controle de personagens não-jogadores (NPC), ou para a geração de conteúdo processual (PCG), são inúmeras as potenciais aplicações dessas técnicas em jogos digitais. O potencial dessas ferramentas de obter uma vantagem competitiva no mercado, ou simplesmente fornecer uma melhor experiência para o usuário é, no mínimo, instigante. Nesse contexto, a modelagem de novos problemas, implementação de soluções utilizando técnicas de *deep learning* e investimento na área, torna-se uma relevante contribuição para o estado da arte.

1.2 Objetivos

O presente trabalho tem como objetivo geral propor o desenvolvimento de uma IA capaz de aprender a jogar diferentes jogos, desde que se tenha acesso ao código fonte e feito em Allegro. Para isso, será implementado um algoritmo utilizando *Deep Reinforcement Learning* (DRL), abordagem que consiste em fornecer ao sistema parâmetros relacionados ao seu estado e uma recompensa positiva ou negativa com base em suas ações. Nenhuma regra sobre o jogo é dada e, inicialmente, a IA não tem informações sobre o que precisa fazer. A única informação passada para a IA são os comandos básicos do jogo. O objetivo do sistema é descobrir e elaborar uma estratégia para maximizar a pontuação - ou a recompensa.

Os objetivos mais específicos deste trabalho são:

1. Revisão da literatura do problema;

2. Descrição e modelagem do problema;

3. Proposta de critérios adicionais que possibilitem estimar outras características das possíveis soluções do projeto, tais como performance, confiabilidade, entre outras;

4. Modelagem de um ou mais jogos que atendam aos requisitos, para validação do sistema;

5. Proposta de um algoritmo de *deep learning* para a solução do problema.

Vale apena ressaltar que a ideia de se implementar um sistema que possa ser adaptado para uma grande variedade de cenários ou jogos, sugere uma ferramenta que possa ser aplicada não só na indústria de videogames, mas em diversas áreas da ciência. Uma situação ou problema do mundo real poderia, por exemplo, ser modelada na forma de um jogo. Nesse caso, a ferramenta utilizada poderia ser aplicada para maximizar sua pontuação. Essa pontuação, por sua vez, seria modelada dentro do jogo de forma a se aproximar do resultado ideal. Dessa forma, o sistema seria capaz de desenvolver estratégias para solucionar problemas e qualquer área da ciência.

Perante o exposto, a implementação de algoritmos que utilizam o aprendizado de máquina de forma a serem aplicados em diferentes cenários, apresenta um potencial de propor novas estratégias e otimizar sistemas já existentes, melhorar a qualidade do produto final e a experiência do usuário, além de proporcionar uma vantagem competitiva no mercado.

1.3 Descrição do problema

O campo da inteligência artificial é capaz de solucionar, com certa facilidade, problemas que são intelectualmente muito difíceis para os serem humanos, mas relativamente diretos para os computadores - problemas que podem ser descritos por uma lista de regras formais e matemáticas. Tarefas abstratas e formais que estão entre os empreendimentos mentais mais difíceis para um ser humano estão entre os mais fáceis para um computador.

Ironicamente, o grande desafio à inteligência artificial provou estar em resolver tarefas fáceis de executar para um ser humano. Problemas que parecem automáticos, que resolvemos intuitivamente, como reconhecer palavras faladas ou rostos em imagens. Os computadores há muito conseguem derrotar até o melhor jogador de xadrez humano (HSU, 2002), mas apenas recentemente começaram a alcançar algumas das habilidades dos seres humanos comuns, como reconhecer objetos ou fala.

A vida cotidiana de uma pessoa requer uma imensa quantidade de conhecimento sobre o mundo. A grande quantidade de informação desses cenários torna inviável a codificação de todas as regras do sistema e, por isso, o computador tem uma grande dificuldade para solucionar esses problemas. Além disso, grande parte desse conhecimento é subjetivo e intuitivo e, portanto, difícil de articular de maneira formal. Os computadores precisam capturar esse mesmo conhecimento para se comportarem de maneira inteligente.

Um dos principais desafios da inteligência artificial é como obter esse conhecimento informal em um computador.

As dificuldades enfrentadas por sistemas que dependem de conhecimento codificado sugerem que os sistemas de IA necessitam da capacidade de adquirir seu próprio conhecimento, extraindo padrões de dados brutos. Esse recurso é conhecido como aprendizado de máquina ou *machine learning* (ML). A introdução do aprendizado de máquina permitiu que os computadores resolvessem problemas que envolvem o conhecimento sobre o mundo real e tomassem decisões mais subjetivas.

O problema proposto nesse trabalho é o de implementar uma IA que, utilizando algoritmos de *deep reinforcement learning*, seja capaz de aprender e desenvolver estratégias para jogar diferentes jogos digitais. Os requisitos do sistema podem ser resumidos pelos seguintes critérios:

1. O sistema receberá, inicialmente, somente os comandos básicos do jogo. Nenhuma regra sobre o jogo é dada e, inicialmente, o agente não tem nenhuma informação sobre o que precisa fazer;

2. O agente deve ser capaz de elaborar uma estratégia para maximizar sua pontuação e que alcance resultados consideravelmente superiores aos de uma abordagem aleatória e próximos aos de um agente humano;

3. O sistema deverá ser capaz de lidar com cenários aleatórios, onde os obstáculos mudam a cada partida, e não aleatórios, onde os obstáculos são "fixos" e a dificuldade varia de acordo com o progresso no jogo;

4. O sistema deve ser generalizado para que possa ser aplicado à diferentes cenários e treinado para jogar diferentes jogos digitais.

De modo a garantir a factibilidade da implementação do sistema, algumas restrições devem ser acatadas. Por exemplo, além de haver a necessidade de se conhecer os comandos básicos do jogo, o sistema precisa ser capaz de obter informações atualizadas sobre o estado do jogo em que se encontra. No caso deste trabalho, foram definidas as seguintes restrições:

1. O sistema deve ter acesso ao código fonte do jogo no qual será aplicado;

2. O sistema deverá ter acesso à pontuação do jogo;

3. O jogo deverá ter sido implementado em *Allegro*;

4. O jogo deve ser 2D para garantir a viabilidade da implementação do sistema.

O acesso ao código fonte nos permite ter conhecimento dos comandos básicos do jogo, enquanto a biblioteca *Allegro* fornece rotinas de baixo nível comumente necessárias na programação de jogos (HARGREAVES, 1990). Essas rotinas, por serem fáceis de manipular, auxiliarão na implementação de um sistema de aprendizado.

O desafio nesse projeto é criar e treinar uma rede neural convolucional capaz de aprender políticas através de pixels brutos em ambientes complexos por meio de um algoritmo de *deep reinforcement learning*. O objetivo principal é implementar um agente que seja capaz de aprender a jogar o maior número de jogos possíveis sem conhecimento prévio do ambiente. Em outras palavras, o sistema deverá ser genérico e o agente não receberá nenhuma informação prévia sobre um jogo específico.

1.4 Revisão da literatura

Apesar de se falar sobre *deep learning* como uma emocionante nova tecnologia, este tem uma história longa e rica, mas apresentando diversos nomes, os quais refletem diferentes pontos de vista filosóficos. Em termos gerais, ocorreram três ondas de desenvolvimento com níveis de popularidade variados: DL conhecido como *cybernetics* nas décadas de 1940 a 1960, DL conhecido como *connectionism* entre as décadas de 1980 e 1990 e o ressurgimento atual sob o nome de aprendizado profundo ou *deep learning* a partir de 2006 (GOODFELLOW; BENGIO; COURVILLE, 2016).

Alguns dos primeiros algoritmos de aprendizado que são reconhecidos hoje pretendiam ser modelos computacionais de aprendizado biológico, isto é, modelos de como o aprendizado acontece ou pode acontecer no cérebro. Como resultado, um dos nomes que o DL passou é o de *artificial neural networks* (ANNs). No entanto, o termo moderno "*deep learning*" vai além da perspectiva neurocientífica da atual geração de modelos de aprendizado de máquina. Ele apela a um princípio mais geral de aprendizado de vários níveis de composição, que podem ser aplicados em estruturas de aprendizado de máquina que não são necessariamente inspiradas em neurônios.

Uma das muitas contribuições do DL está no reconhecimento de fala (Nassif et al., 2019). Até recentemente, os de reconhecimento automático de fala (ASR) combinavam principalmente modelos ocultos de Markov (HMMs) e modelos de mistura gaussianos (GMM). Com a introdução de redes neurais e, posteriormente, modelos de DL cada vez maiores e mais profundos e conjuntos de dados muito maiores, a precisão do reconhecimento foi dramaticamente aprimorada usando redes neurais para, eventualmente, substituir GMMs na tarefa de associar recursos acústicos a fonemas (GOODFELLOW; BENGIO; COURVILLE, 2016).

O *deep learning* também contribuiu para outras ciências. As redes convolucionais modernas para reconhecimento de objetos e visão computacional fornecem um modelo de

processamento visual com diversas aplicações na medicina (YEUNG et al., 2019; AFRAZ DANIEL L.K. YAMINS, 2014). O *deep learning* também fornece ferramentas úteis para processar grandes quantidades de dados e fazer previsões úteis em campos científicos. Ele tem sido usado com sucesso para prever como as moléculas irão interagir, a fim de ajudar as empresas farmacêuticas a projetar novos medicamentos (DAHL; JAITLY; SALAKHUTDINOV, 2014), a procurar partículas subatômicas (BALDI; SADOWSKI; WHITESON, 2014), e para o processamento de linguagem natural (YOUNG et al., 2018). Espera-se que o DL apareça em cada vez mais campos científicos no futuro.

Pesquisas recentes em IA deram origem a técnicas poderosas para o *deep reinforcement learning*. Na combinação de aprendizado de representação com comportamento orientado por recompensas, o DRL parece ter um interesse inerente à psicologia e neurociência. Um argumento contra essa abordagem foi o de que os procedimentos de aprendizado por DRL exigem grandes quantidades de dados de treinamento, sugerindo que esses algoritmos podem diferir fundamentalmente daqueles subjacentes ao aprendizado humano. Embora essa preocupação se aplique à onda inicial de técnicas de RL profunda, o trabalho subsequente de IA estabeleceu métodos que permitem que os sistemas de RL profunda aprendam mais rápida e eficientemente (BOTVINICK et al., 2019).

A IA em jogos digitais possui algumas peculiaridades (YANNAKAKIS, 2012; MILLINGTON; FUNGE, 2009), que a distinguem da IA clássica, especialmente porque, em muitos casos, ela deve lidar com aplicativos em tempo real e não necessariamente precisa otimizar resultados. Ela pode ser explorada para muitos propósitos, que podem ser coletados em três macro-categorias principais: ajudar na jogabilidade, melhorar a imersão do jogador no mundo do jogo (também simular a psicologia dos agentes que representam os personagens que não jogam - NPCs) e apoiar o trabalho de designers de jogos e níveis (Piergigli et al., 2019). Entre as técnicas de IA mais difusas, podemos contar aquelas usadas para gerar procedimentalmente conteúdos (Karavolos; Liapis; Yannakakis, 2018; RIPAMONTI et al., 2017) e aquelas destinadas a apoiar o sistema de tomada de decisão dos agentes artificiais (Ripamonti et al., 2017).

O aprendizado de máquina e as redes neurais são aplicadas aos jogos há muito tempo, mas seu uso recentemente conheceu um interesse renovado e aborda uma ampla variedade de tópicos. No entanto, o uso dessas técnicas para treinar agentes em ambientes complexos, com várias ações simultâneas possíveis é um resultado bastante desafiador a ser alcançado (Piergigli et al., 2019).

O *DeepMind* do Google desenvolveu o *Deep Q-network* (DQN), uma arquitetura de rede neural, que demonstrou ser capaz de aprender políticas de controle no nível humano em vários jogos diferentes do Atari 2600 (MNIH et al., 2015). Os DQNs aprendem a estimar os valores Q (função de valor da ação do estado) de selecionar cada ação do estado atual do jogo. Como a função de valor da ação do estado é uma representação suficiente

da política do agente, um jogo pode ser jogado selecionando a ação com o valor Q máximo em cada etapa do tempo. Dessa forma, aprendendo políticas de pixels em tela bruta a ações, essas redes têm demonstrado desempenho avançado em vários jogos do Atari 2600. Vale ressaltar que a mesma rede pode ser usada em várias tarefas sem nenhuma alteração e que o aprendizado é de ponta a ponta, dos valores brutos dos pixels aos valores Q, sem a necessidade de intervenção humana. Os DQNs também foram estendidos para obter melhor desempenho em jogos ainda mais complexos (DWIBEDI; VEMULA, 2016).

Um dos feitos mais notáveis nesse contexto, realizado também pelo *DeepMind*, é o da implementação da IA conhecida como *AlphStar*. Essa inteligência artificial alcançou uma classificação de grande mestre depois de ter sido lançada nos servidores europeus do jogo *StarCraft II*, ficando entre os $0,15\%$ dos 90.000 jogadores da região. O domínio do *StarCraft* emergiu como um importante desafio para a pesquisa em inteligência artificial, devido ao seu status icônico e duradouro entre os mais difíceis e-sports profissionais e sua relevância para o mundo real em termos de complexidade bruta e desafios multi-agente. Ao longo de uma década e inúmeras competições, os agentes mais fortes simplificaram aspectos importantes do jogo, utilizaram capacidades sobre-humanas ou empregaram subsistemas artesanais. Apesar dessas vantagens, nenhum agente anterior chegou perto de igualar a habilidade geral dos melhores jogadores de *StarCraft* (VINYALS et al., 2019). Tudo isso torna os resultados obtidos pelo AlphaStar ainda mais impressionantes: a IA foi classificada no nível grande mestre e acima de $99,8\%$ dos jogadores humanos classificados oficialmente.

1.5 Organização do trabalho

Este trabalho está estruturado em cinco capítulos. O **Capítulo 1** consiste em uma breve introdução ao tema do projeto e uma análise da literatura do problema. O **Capítulo 2** apresenta uma contextualização do problema nos âmbitos social, ambiental e econômico. O **Capítulo 3** discorre a abordagem proposta para o problema, assim como sua respectiva modelagem matemática. O **Capítulo 4** encerra o trabalho com as conclusões e apresenta as propostas de continuidade para o Trabalho de Conclusão de Curso II.

2 Contextualização em Humanidades

Nos últimos anos, houve um progresso significativo na solução de problemas desafiadores em diversos campos, utilizando algoritmos de *deep reinforcement learning*. Como consequência, o RL experimentou um crescimento dramático na atenção e no interesse da comunidade científica.

Do ponto de vista econômico, são inúmeros os usos de *deep learning* no mercado. Desde ferramentas que melhoram a precisão dos sensores de precipitação por satélite e concentrando-se na redução do viés e dos alarmes falsos (TAO et al., 2016), à agentes que permitem que diferentes dispositivos eletrônicos interpretem dados de multimídia não estruturados e reajam de maneira inteligente aos eventos do usuário e do ambiente (Tang et al., 2017), o DL tem se tornado cada vez mais presente e essencial para a sociedade. Grandes setores e empresas na área da tecnologia não existiriam sem o uso dessas ferramentas.

Em relação aos impactos sociais do DL podemos mencionar a sua utilização para estimar as características socioeconômicas de regiões de 200 cidades dos Estados Unidos usando 50 milhões de imagens de cenas de rua reunidas com carros do *Google Street View* (GEBRU et al., 2017). O DL também teve impactos em diversas áreas da ciência, desde pesquisa em física de partículas (BALDI; SADOWSKI; WHITESON, 2014), à medicina (Nassif et al., 2019).

É interessante ressaltar que apesar de todos os benefícios oferecidos pela IA, alguns indivíduos notáveis como o famoso físico Stephen Hawking, e o líder da Tesla e da SpaceX Elon Musk, sugerem que a IA pode ser potencialmente muito perigosa. De fato, existem muitos aplicativos de IA que tornam nossa vida cotidiana mais conveniente e eficiente. São os aplicativos de IA que desempenham um papel crítico para garantir a segurança que Musk, Hawking e outros estavam preocupados quando proclamaram sua hesitação sobre a tecnologia. Por exemplo, se a IA for responsável por garantir a operação de nossa rede elétrica, de uma usina nuclear ou outro sistema de alto risco, e a IA for invadida ou tiver seus objetivos desalinhados com os nossos, isso poderá resultar em danos enormes (MARR, 2018).

Apesar de todo o medo ao redor dessa nova tecnologia, muitos argumentam que os mesmos são exagerados e que os benefícios oferecidos são muito maiores que os potenciais riscos, desde que sejam gerenciados adequadamente. O crescimento de pesquisas em DRL revelam seu grande potencial e benefícios para a sociedade. Reproduzir e comparar os trabalhos existentes existente e julgar com precisão as melhorias oferecidas por novos métodos é vital para sustentar esse progresso.

No contexto de jogos digitais, treinar um agente para superar os jogadores humanos e otimizar sua pontuação pode nos ensinar como otimizar processos diferentes em uma variedade de subcampos diferentes e intrigantes (COMI, 2018). Os impactos econômicos e sociais que essas técnicas podem oferecer são diversos.

Situações do mundo real são muitas vezes complexas e apresentam problemas com um número muito grande de variáveis. Para tais problemas, encontrar a melhor solução pode ser um desafio muito grande para algoritmos de otimização tradicionais. Uma vez que se tenha um sistema capaz de aprender e elaborar estratégias para diferentes cenários, é fácil modelar problemas que possam ser resolvidos pelo mesmo. No caso, uma IA que possa aprender a jogar e a otimizar estratégias para maximizar a pontuação de um jogo, pode ser aplicada em um jogo que simule uma situação real e encontrar a melhor resposta ou solução para um dado problema.

Imagine, por exemplo, um jogo que simule o trânsito em uma cidade, e a pontuação desse jogo é calculada de acordo com a elaboração das rotas de ônibus, as quais devem alcançar o maior número de áreas da cidade e minimizar o tempo de cada trajeto. A IA proposta seria, idealmente, capaz de encontrar a melhor organização possível dessas rotas. Na área da biomedicina e química, poderíamos modelar um jogo que simule o comportamento de uma célula cancerígena, e a IA teria o objetivo de encontrar o tratamento mais efetivo para a doença. Um jogo que simule condições extremas de temperaturas, ambiente e terreno, poderia ser aplicado ao sistema e a IA poderia propor a modelagem das máquinas que iriam se adaptar melhor à dadas condições. Essas máquinas, por sua vez, poderiam ser utilizadas em diversas expedições espaciais ou de alta profundidade, por exemplo. O sistema proposto, portanto, poderia idealmente ser aplicado para quaisquer cenários ou jogos, os quais podem ser modelados para serem mais ou menos complexos, de forma a melhor atender a necessidade do usuário.

Em resumo, o DL já é utilizado com sucesso em diversas áreas da ciência, otimizando e solucionando diferentes problemas. Ao propor um sistema que seja flexível e capaz de se adaptar à diferentes cenários, seria capaz de unificar muitas dessas ferramentas em uma única. A mesma ferramenta poderia ser aplicada nas diferentes situações mencionadas anteriormente e propor soluções para inúmeros problemas, melhorar produtos já existentes e otimizar processos no mundo real.

3 Abordagem Proposta

No contexto de jogos digitais, treinar um agente para superar os jogadores humanos e otimizar sua pontuação pode nos ensinar como otimizar processos diferentes em uma variedade de subcampos intrigantes (COMI, 2018). Uma solução proposta na literatura, obtendo ótimos resultados, e que tem como objetivo treinar um computador pra aprender e desenvolver estratégias para jogar diferentes jogos, é o *deep reinforcement learning* (DRL).

No presente trabalho é proposto a implementação de uma inteligência artificial que, utilizando um algoritmo de *deep reinforcement learning*, seja capaz de aprender a jogar diferentes jogos e desenvolver estratégias para maximizar sua pontuação.

Diante das peculiaridades e restrições do problema discutidos em **1.3**, a biblioteca *Allegro* foi escolhida como a base para a implementação dos jogos que serão apresentados ao sistema. O *Allegro* é uma biblioteca multiplataforma destinada principalmente a jogos de vídeo e programação multimídia. A biblioteca fornece rotinas de baixo nível comumente necessárias na programação de jogos, como a criação de janelas, aceitação de entrada do usuário, carregamento de dados, desenho de imagens, reprodução de sons etc (HARGREAVES, 1990).

A IA será treinada apartir de capturas de tela em diferentes estado do jogo, e da pontuação obtida. Esses dados serão obtidos a partir de um "*Allegro Learning Enviroment*" (ALE), o qual consiste de uma ferramenta para o desenvolvimento de inteligência artificial em jogos implementados em *Allegro*. Seu objetivo é oferecer uma plataforma que facilite o desenvolvimento de algoritmos de ML para jogos em *Allegro*, o que irá auxiliar a implementação do sistema.

A **Figura 1** mostra a arquitetura da abordagem proposta. Inicialmente, a ALE irá extrair os comandos básicos do jogo para que o agente tenha conhecimento das limitações físicas do ambiente no qual ele será inserido. Uma vez que o treinamento seja iniciado, a ALE será responsável por obter as capturas de tela que conterão informações sobre o estado atual do jogo, assim como a pontuação obtida pelo agente. Com esses dados, o agente deverá elaborar uma política de decisão para tomar uma ação em cada estado. A ação tomada pelo agente será passada para o jogo, que irá atualizar o seu estado de acordo. Esse ciclo continua até o jogo ser finalizado (seja pelo sucesso ou falha do agente), e uma pontuação final ser obtida. O treinamento do agente consiste na repetição desse processo de modo que a IA, utilizando técnicas de *reinforcement learning*, seja capaz de elaborar uma estratégia para maximizar a pontuação final.

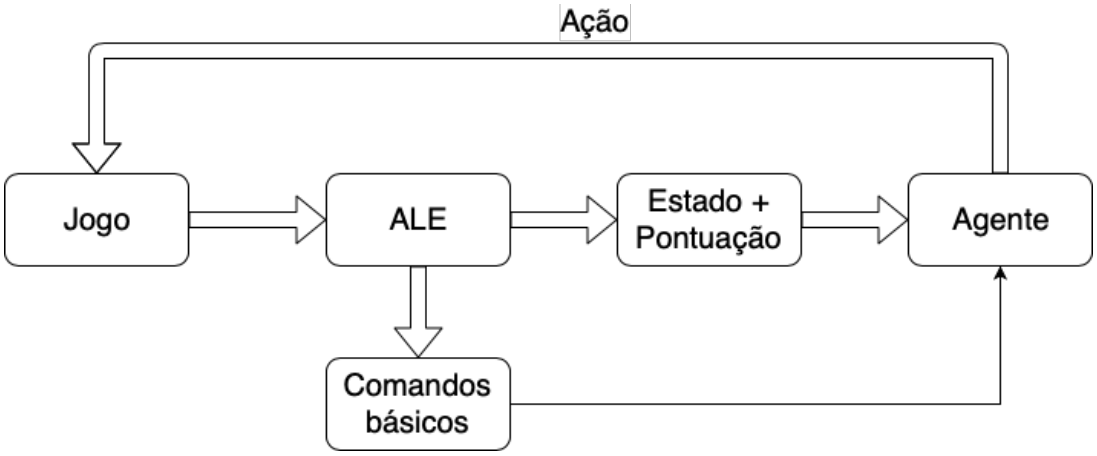

Figura 1: Arquitetura da abordagem proposta. A partir do código fonte do jogo, a ALE extrai os comandos básicos do jogo para que o agente tenha conhecimento de suas limitações físicas. Durante o processo de treinamento, para cada estado do jogo, a ALE passa as informações sobre o estado atual do jogo e a pontuação obtida até então. Com essas informações, o agente toma uma ação que irá influenciar o próximo estado do jogo

3.1 *Deep Learning*

O *deep learning* (DL) é uma área do aprendizado de máquina que propõe que os computadores aprendam com a experiência, se ajustem à novas entradas de dados e compreendam o mundo em termos de hierarquia de conceitos, sendo cada conceito definido por sua relação com conceitos mais simples. Ao reunir conhecimento a partir da experiência, essa abordagem evita a necessidade dos operadores humanos de especificar formalmente todo o conhecimento que o computador precisa. Além disso, a hierarquia de conceitos permite que o computador aprenda conceitos complexos, construindo-os a partir de conceitos mais simples. O *deep learning* apresenta grande poder e flexibilidade a nos permitir o treinamento de computadores para cumprir tarefas específicas ao processar grandes quantidades de dados e reconhecer padrões nesses dados.

A **Figura 2** mostra como um sitema de *deep learning* representa o conceito de imagem de uma pessoa combinando conceitos mais simples, como cantos e contornos, que por sua vez são definidos em termos de arestas.

O mapeamento de funções de um conjunto de pixels para uma identidade de objeto é uma tarefa complicada. O algoritmo de *deep learning* resolve essa dificuldade dividindo o mapeamento complicado desejado em séries de mapeamentos simples aninhados, cada um deles descrito por uma camada diferente do modelo. A entrada é apresentada na camada visível, em seguida, uma série de camadas ocultas extrai recursos cada vez mais abstratos da imagem. A camada de saída obtém a identidade de objeto abstrata a partir dos conceitos obtidos pelas camadas ocultas.

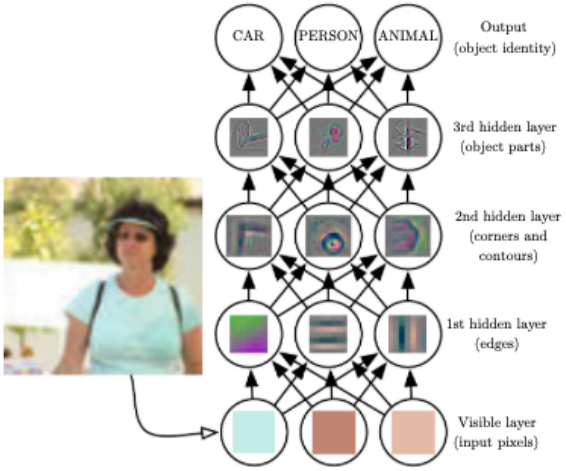

Figura 2: Ilustração de um modelo de aprendizado profundo. Cada camada é capaz de identificar dados de complexidade crescente a partir dos pixels pasados para a camada de entrada. Imagem retirada de (GOODFELLOW; BENGIO; COURVILLE, 2016)

O reconhecimento de imagens a partir da extração de padrões de pixels brutos, será crucial para o agente no processo de análise do estado atual do jogo. Como mencionado anteriormente, as informações do ambiente e estado atual do jogo serão extraídas a partir de capturas de tela em cada estado. A partir dessas capturas de tela, a IA deverá reconhecer padrões e identificar obstáculos, caminhos disponíveis e objetivos a serem alcançados dentro do jogo. A utilização de capturas de tela permite ao agente se adaptar à cenários onde os obstáculos e caminhos disponíveis são elaborados de forma aleatória, e sua disposição se altera a cada iteração do jogo.

3.2 *Reinforcement Learning*

O aprendizado por reforço ou *reinforcement learning* (RL) é uma abordagem computacional para entender e automatizar o aprendizado direcionado a objetivos e a tomada de decisões. O aprendizado por reforço distingue-se de outras abordagens computacionais por sua ênfase na aprendizagem de um agente apartir da interação direta com seu ambiente, sem exigir supervisão exemplar ou modelos completos do ambiente (SUTTON; BARTO, 2018).

Em algoritmos de *reinforcement learning*, o agente não é informado sobre quais ações executar, mas, em vez disso, deve descobrir quais ações geram mais recompensa, através de tentativa e erro. Em alguns casos mais interessantes, as ações podem afetar não apenas a recompensa imediata, mas também a próxima situação e, com isso, todas as recompensas subsequentes. Essas duas características - pesquisa por tentativa e erro e recompensa atrasada - são as duas características distintivas mais importantes do aprendizado por

reforço.

O aprendizado por reforço é diferente do aprendizado supervisionado, o tipo de aprendizado estudado na maioria das pesquisas atuais no campo do aprendizado de máquina. Aprendizado supervisionado é aprender com um conjunto de treinamento de exemplos rotulados fornecidos por um supervisor externo qualificado. O objetivo desse tipo de aprendizado é o sistema extrapolar ou generalizar suas respostas para que ele atue corretamente em situações não presentes no conjunto de treinamento. Este é um tipo importante de aprendizado, mas por si só não é adequado para aprender com a interação. Em problemas interativos, muitas vezes é impraticável obter exemplos do comportamento desejado que sejam corretos e representativos de todas as situações nas quais o agente precisa agir. Em um território desconhecido - onde se espera que a aprendizagem seja mais benéfica - um agente deve ser capaz de aprender com sua própria experiência.

O aprendizado por reforço também é diferente do que os pesquisadores de aprendizado de máquina chamam de aprendizado não supervisionado, que geralmente consiste em encontrar estruturas ocultas em coleções de dados não rotulados. Os termos aprendizado supervisionado e aprendizado não supervisionado parecem classificar exaustivamente os paradigmas de aprendizado de máquina, mas não o fazem. Embora se possa ficar tentado a pensar no aprendizado por reforço como um tipo de aprendizado não supervisionado, porque não se baseia em exemplos de comportamento correto, o aprendizado por reforço está tentando maximizar um sinal de recompensa em vez de tentar encontrar uma estrutura oculta. Descobrir a estrutura na experiência de um agente certamente pode ser útil no aprendizado por reforço, mas por si só não aborda o problema do aprendizado por reforço de maximizar um sinal de recompensa. Portanto, o aprendizado por reforço é considerado como um terceiro paradigma de aprendizado de máquina, ao lado de aprendizado supervisionado, aprendizado não supervisionado e talvez outros paradigmas (SUTTON; BARTO, 2018).

A **Figura 3** mostra um diagrama de aprendizagem por reforço relacionando o agente de aprendizado com o ambiente no qual ele é inserido. O ambiente representa o mundo pelo qual o agente se move. O ambiente nada mais é do que um sistema que toma o estado atual e a ação do agente como entrada e retorna como saída a recompensa do agente e seu próximo estado.

Ambientes podem ser modelados como funções que transformam uma ação executada no estado atual, no próximo estado e uma recompensa. Já os agentes podem ser modelados como funções que transformam o novo estado e recompensam na próxima ação. Podemos conhecer a função do agente, mas não podemos conhecer a função do ambiente. É uma caixa preta onde só vemos as entradas e saídas. O aprendizado por reforço representa a tentativa de um agente de aproximar a função do ambiente, para que possamos enviar ações para o ambiente de caixa preta que maximize as recompensas que ele distribui

(NICHOLSON, 2016).

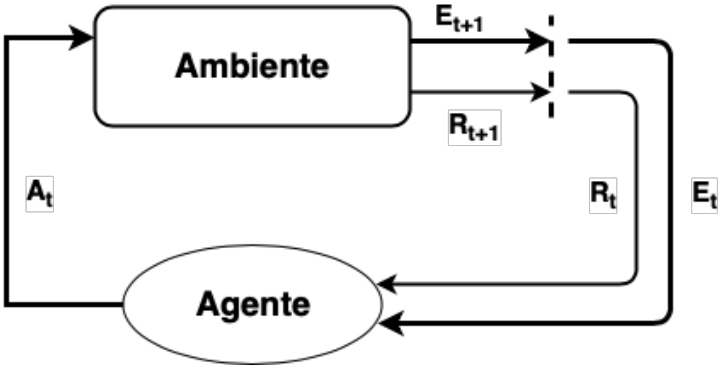

Figura 3: Diagrama de aprendizagem por reforço. No *loop* de *feedback* acima, os subscritos indicam as etapas de tempo t e $t+1$, cada uma das quais se refere a estados diferentes: o estado no momento t e o estado no momento $t+1$. A ação A_t de um agente é determinada por sua **política**, que por sua vez é uma função que depende do estado atual do sistema E_t. A política de um agente tem como objetivo maximizar a **função de valor** que é calculada utilizando o **sinal de recompensa** R_t. O ambiente se comporta como um sistema caixa preta que transforma uma ação executada no estado atual A_t, no próximo estado E_{t+1} e uma recompensa R_{t+1}

As escolhas de ação são feitas com base em julgamentos de valor. Buscamos ações que gerem estados de maior valor, e não de maior recompensa, porque essas ações obtêm a maior quantidade de recompensa a longo prazo. As recompensas são basicamente dadas diretamente pelo ambiente, mas os valores devem ser estimados e re-estimados a partir das sequências de observações que um agente faz ao longo de toda a sua vida útil.

Além do agente e do ambiente, é interessante ressaltar outros elementos importantes de um sistema de aprendizado por reforço: a **política**, o **sinal de recompensa** e a **função de valor**.

A **política** define a maneira que o agente deve se comportar em um determinado momento. Uma política é basicamente um mapeamento dos estados do ambiente para as ações a serem tomadas quando nesses estados. A política em casos mais simples ter a forma de uma função simples ou uma tabela de pesquisa, enquanto em casos mais complexos pode envolver cálculos mais extensivos. Em geral, as políticas podem ser estocásticas, especificando probabilidades para cada ação.

Um **sinal de recompensa** define o objetivo de um problema de aprendizado por reforço. Em cada etapa, o ambiente envia ao agente de aprendizado por reforço um único número que funciona como uma recompensa para o agente. O único objetivo do agente é maximizar a recompensa total que recebe a longo prazo. O sinal de recompensa define, portanto, quais são os eventos bons e ruins para o agente. O sinal de recompensa é a base principal para alterar a política - se uma ação selecionada pela política for seguida por

uma baixa recompensa, a política poderá ser alterada para selecionar outra ação nessa situação no futuro. Em geral, os sinais de recompensa podem ser funções estocásticas do estado do ambiente e das ações tomadas.

Enquanto o sinal de recompensa indica o que é bom em um sentido imediato, uma **função de valor** especifica o que é bom a longo prazo. O valor de um estado representa a quantidade total de recompensa que um agente pode esperar acumular no futuro, a partir desse estado. Enquanto as recompensas determinam a conveniência imediata e intrínseca dos estados ambientais, os valores indicam a conveniência a longo prazo dos estados após levar em conta os estados que provavelmente seguirão e as recompensas disponíveis nesses estados. Por exemplo, um estado sempre pode gerar uma recompensa imediata baixa, mas ainda tem um valor alto porque é seguido regularmente por outros estados que produzem recompensas altas. Ou o contrário poderia ser verdade.

O *deep reinforcement learning* (DRL) é uma abordagem do *deep learning* que, em contraste a abordagens mais tradicionais como o aprendizado supervisionado e não supervisionado, utiliza as técnicas de aprendizagem por reforço para treinar o agente. Essa abordagem consiste em fornecer ao sistema parâmetros relacionados ao seu estado e uma recompensa positiva ou negativa com base em suas ações. Nenhuma regra sobre o jogo é dada e, inicialmente, o agente não tem nenhuma informação sobre o que precisa fazer. O objetivo do sistema é descobrir e elaborar uma estratégia para maximizar sua pontuação - ou recompensa.

3.3 *Allegro*

O *Allegro* é uma biblioteca multiplataforma destinada principalmente a jogos de vídeo e programação multimídia. A biblioteca fornece rotinas de baixo nível comumente necessárias na programação de jogos, como a criação de janelas, aceitação de entrada do usuário, carregamento de dados, desenho de imagens, reprodução de sons etc (HARGREAVES, 1990). Algumas outras características da biblioteca que facilitam a implementação de jogos são:

- Suportada em Windows, Linux, Mac OS, iPhone e Android;
- API intuitiva e amigável, utilizável em C, C ++ e em muitas outras linguagens;
- Bitmap acelerado por hardware e suporte a desenho gráfico primitivo (via OpenGL ou Direct3D);
- Suporte de gravação de áudio;
- Carregamento e desenho de fontes;

- Reprodução de vídeo.

A implementação de uma ferramenta de aprendizado voltada para jogos em *Allegro*, é facilitada pelo fato de o *Allegro* ser simples e amigável, o que permite a extração dos comandos básicos do jogo a partir do seu código fonte, funcionalidade essencial para a implementação de uma ferramenta genérica.

Outro motivo que levou à escolha da biblioteca como requisito importante do projeto, é a funcionalidade de controle de *frame rates*. Um jogo que possua uma alta taxa de quadros, ou *frames per second* (FPS), apresenta uma saída com um grande número de informação em um intervalo curto de tempo. Esse alto fluxo de informações pode sobrecarregar a IA, ou até mesmo apresentar perda de informação (*frame drops*), o que pode resultar em um treinamento ineficiente em uma arquitetura com baixo poder computacional. Ao reduzir o FPS do jogo, podemos diminuir a velocidade com que os estados do jogo são atualizados, diminuindo a quantidade de informação que deve ser tratada e permitindo o desenvolvimento de uma IA em arquiteturas com menor poder computacional.

3.4 Aplicação de DRL em um *Allegro Learning Enviroment*

O *Arcade Learning Enviroment* é uma ferramenta de software que oferece uma interface para interagir com ambientes de jogos Atari 2600 emulados. Seu objetivo é oferecer uma plataforma que facilite o desenvolvimento de agentes de aprendizado para aprender a jogar jogos Atari. Essa ferramenta também fornece uma camada de manipulação de jogos que transforma cada jogo em um problema padrão de aprendizado por reforço, identificando a pontuação acumulada e se o jogo terminou. (BELLEMARE et al., 2012)

Inspirado na plataforma descrita acima, este trabalho visa a utilização de um *Allegro Learning Enviroment*, que funcionaria de forma semelhante ao *Arcade Learning Enviroment*, com a distinção de que o primeiro seria uma plataforma voltada para jogos implementados exclusivamente em *Allegro*.

O *Allegro Learning Enviroment* (ALE) terá como base a ferramenta implementada por (SILVA, 2019), que oferece um ambiente facilitador ao estudo de soluções de IA aplicada em jogos. Essa ferramenta fornece funcionalidades como a exportação dos comandos básicos de um jogo, que precisam ser passados para o agente para que o mesmo tenha conhecimento dos limites físicos do ambiente no qual está inserido. Isso permite que o pesquisador não fique limitado a um jogo existente, mas possa usar qualquer jogo que ele tenha acesso ao código fonte e feito em *Allegro*. Outra funcionalidade fornecida pelo ALE, é a possibilidade de se extrair a pontuação e obter capturas de tela em cada estado do jogo, informações que devem ser fornecidas para o treinamento do agente.

Para o treinamento do agente, serão utilizados capturas da tela em cada estado do jogo, obtidas pelo ALE. A partir dessas imagens serão extraídas as informações do estado atual do jogo (posição do jogador, obstáculos, etc), de forma a determinar qual a melhor ação do agente para a situação na qual ele se encontra. A utilização de capturas de tela como entradas para o agente permite que a IA seja treinada para situações em que hajam obstáculos gerados de forma aleatória. A partir dessas imagens, o agente deverá ser capaz de identificar tais obstáculos, sua localização e a melhor maneira de lidar com os mesmos.

Em relação aos jogos nos quais o agente será treinado, a proposta é de inicializar o treinamento com jogos mais simples (e.g. *Snake, Frogger, Agar.io*). Se possível, neste trabalho ou em trabalhos posteriores, a ideia é de se utilizar diferentes jogos de diferentes complexidades para avaliar o potencial do sistema. Os jogos serão obtidos de fontes de código aberto disponíveis *online* ou, caso seja necessário, serão implementados com os requisitos necessário para o projeto.

4 Conclusões

A inteligência artificial e o aprendizado de máquina são ferramentas muito poderosas e com inúmeras aplicações práticas. A IA busca fornecer softwares que sejam capazes de realizar atividades como seres humanos para automatizar e otimizar o trabalho de rotina. Diante do grande potencial da IA, além do crescimento exponencial de pesquisa que a área vêm sofrendo, grandes empresas no mercado estão investindo na tecnologia, seja para propor novos serviços ou aprimorar produtos existentes e garantir uma vantagem competitiva no mercado.

Situações do mundo real são muitas vezes complexas e apresentam problemas com um número muito grande de variáveis, o que dificulta a solução utilizando algoritmos de otimização tradicionais. Treinar um agente em jogos digitais para superar os jogadores humanos e otimizar sua pontuação pode nos ensinar como otimizar processos variados com múltiplas aplicações. Uma vez que se tenha uma IA que possa aprender a jogar e a otimizar estratégias para maximizar a pontuação de um jogo, pode-se facilmente implementar um jogo que simule uma situação real e aplicar o sistema para que este encontre a melhor resposta ou solução para um dado problema. Com isso em mente, o problema proposto nesse trabalho é o de implementar uma IA que, utilizando algoritmos de *deep reinforcement learning*, seja capaz de aprender e desenvolver estratégias para jogar diferentes jogos digitais.

A IA proposta deverá ser genérica, ou seja, capaz de aprender a jogar diferentes jogos, desde que se tenha acesso ao codigo fonte e que sejam implementados em *Allegro*. Para auxiliar na implementação do sistema será utilizado um *Allegro Learning Enviroment*, plataforma que irá facilitar a implementação da ferramenta para o treinamento do agente.

Diante disso, utilizando técnicas de DRL existentes, espera-se produzir uma IA que seja flexível e que possa ser adaptada para diferentes cenários. Neste sentido, espera-se uma IA que seja genérica e capaz de ser treinada para diversos jogos. Por fim, será feita uma análise crítica dos resultados e uma comparação dos mesmos com trabalhos semelhantes realizados por outras entidades.

4.1 Proposta de Continuidade

Este trabalho tem como continuidade o desenvolvimento do Trabalho de Conclusão de Curso II, onde haverá um maior detalhamento sobre a modelagem matemática do problema, além de especificadas as decisões de implementação da ferramenta elaborada, bem como uma análise dos resultados.

A abordagem, além do que já foi exposto, consistirá na implementação da rede neural proposta, utilizando os algoritmos DRL mencionados para o treinamento do agente. Também serão implementados (se necessário), diferentes jogos em *Allegro* para a validação do sistema. Por fim, o agente será treinado em jogos simples e de baixa complexidade e serão apresentados os resultados obtidos para avaliar o potencial do sistema. Em trabalhos futuros, a ferramenta poderá também ser utilizada para o treinamento em jogos de diferentes complexidades, ampliando ainda mais o alcance do sistema.

4.2 Cronograma TCC 2

Atividades	Meses			
	Agosto	Setembro	Outubro	Novembro
Levantamento bibliográfico	X			
Pesquisa e implementação da rede neural proposta do trabalho	X			
Aplicação do estudo realizado no assunto do TCC a ser desenvolvido	X			
Entrega da Visão Geral do Trabalho	21/08/2020			
Desenvolvimento do trabalho e implementação da rede neural proposta do trabalho	X	X		
Elaboração do corpo principal do TCC		X	X	
Entrega do FomulárioPonto de Controle		11/09/2020		
Marcação da defesa		25/09/2020		
Ajustes finais e conclusão to trabalho implementado			X	
Emissão da versão inicial do TCC			17/10/2020	
Preparação do material referente à apresentação do TCC			X	
Apresentação oral para banca examinadora			Semana de 26 a 30/10	
Ajuste no material relativo ao trabalho escrito				06/11/2020

5 Modelagem

Como mencionado no **Capítulo 3**, a proposta deste trabalho consiste na implementação de um algoritmo de *deep reinforcement learning* para treinar um agente que seja capaz de aprender a jogar um jogo em *Allegro*. A inspiração para o presente projeto vem do trabalho realizado pelo *Deep Mind* e publicado no artigo (MNIH et al., 2013), onde foi implementado uma IA capaz de jogar diferentes jogos Atari 2600. Assim, será implementado um sistema semelhante voltado para jogos em *Allegro*.

O atual capítulo consiste na modelagem matemática da abordagem proposta no **Capítulo 3**, além de um maior aprofundamento de alguns conceitos de DRL e um breve detalhamento sobre a implementação.

5.1 *Contextualização*

5.1.1 Aprendizagem por Reforço

No aprendizado por reforço, é criado um agente que executa ações dentro de um ambiente. O ambiente, por sua vez, é alterado de acordo com a ação realizada, e o agente recebe várias recompensas de acordo com o estado em que se encontra dentro do ambiente. Em outras palavras, um agente explora um jogo, e é treinado tentando maximizar as recompensas nesse jogo. Este ciclo é ilustrado na **Figura 4**.

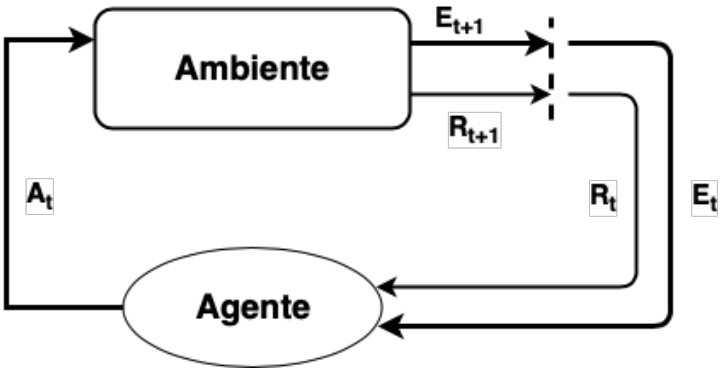

Figura 4: Diagrama de aprendizagem por reforço. No *loop* de *feedback*, os subscritos indicam as etapas de tempo t e $t+1$, cada uma das quais se refere a estados diferentes: o estado no momento t e o estado no momento $t+1$. A ação A_t de um agente é determinada por sua **política**, que por sua vez é uma função que depende do estado atual do sistema E_t. A política de um agente tem como objetivo maximizar a **função de valor** que é calculada utilizando o **sinal de recompensa** R_t. O ambiente se comporta como um sistema caixa preta que transforma uma ação executada no estado atual A_t, no próximo estado E_{t+1} e uma recompensa R_{t+1}

5.1.2 Processos de Decisão de Markov e a Equação de Bellman

5.1.3 O Jogo

O jogo utilizado para teste do modelo foi o *Frogger*. A escolha do mesmo foi feita tendo em vista sua simplicidade, tendo em vista as limitações de implementação (**Seção 5.3.2**). A **Figura 5** mostra o jogo utilizado. O objetivo do agente é partir do estado inicial mostrado na figura, e alcançar o topo da tela sem colidir com nenhum obstáculo.

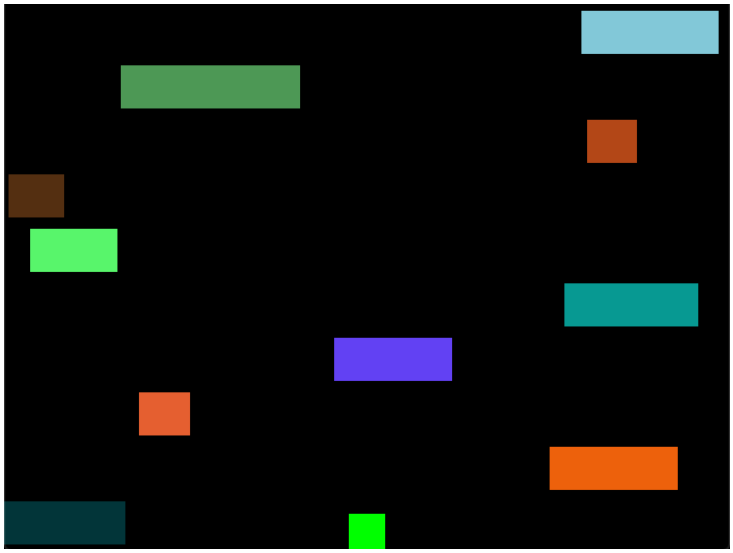

Figura 5: Exemplo do jogo *Frogger* utilizado. O jogador controla o quadrado verde no centro inferior da tela, enquanto os outros retângulos coloridos são os obstáculos

As recompensas para ações durante o jogo foram definidas inicialmente da seguinte forma:

- $r = 0$ caso o agente faça uma ação que o mantenha na mesma linha que se encontrava previamente;
- $r = 1$ caso o agente faça uma ação que o aproxime verticalmente de seu objetivo;
- $r = -1$ caso o agente faça uma ação que o distancie verticalmente de seu objetivo;
- $r = 10$ caso o agente alcance seu objetivo.

5.2 Modelagem Matemática

Conforme descrito em (MNIH et al., 2013), são consideradas tarefas em que um agente interage com um ambiente ε, nesse caso o jogo em *Allegro*, como uma sequência de ações, observações e recompensas. Em cada etapa de tempo, o agente seleciona uma ação em do conjunto de ações legais do jogo, $\mathcal{A} = \{1, \cdots, K\}$. A ação é executada, modificando

o estado e pontuação do jogo. O estado interno do jogo não é observado pelo agente, este observa apenas uma imagem $x_t \in \mathbb{R}^d$, que é um vetor de valores de pixel brutos que representam a tela do estado atual do jogo. Além disso, o agente recebe uma recompensa r que representa a alteração na pontuação do jogo.

É importante ressaltar que a pontuação do jogo pode depender de toda a sequência anterior de ações e observações. O feedback sobre uma ação só pode ser recebido depois de decorridos múltiplos de intervalos de tempo. Uma vez que o agente apenas observa as imagens da tela atual, a análise do atual estado do jogo pode ser mal-representada, ou seja, é difícil para o agente compreender totalmente a situação atual apenas da tela atual x_t. Para solucionar esse problema, considera-se como um estado s_t do jogo, uma sequência de ações e observações $s_t = (x_{t-n}, a_{t-n}, \cdots, a_{t-1}, x_t)$, as quais serão utilizadas para treinar o agente, fornecendo-o um melhor contexto do estado em que se encontra. Esse formalismo dá origem a um processo de decisão de Markov (MDP), no qual cada sequência é um estado distinto. Como resultado, podemos aplicar métodos de aprendizado por reforço padrão para MDPs, simplesmente usando a sequência completa s_t como a representação do estado no tempo t.

O objetivo do agente é interagir com o jogo, selecionando ações de uma forma que maximize recompensas futuras. É feita a suposição padrão de que as recompensas futuras são descontadas por um fator de γ por intervalo de tempo, e que o retorno descontado futuro é definido por:

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} \cdot r_{t'} \tag{5.1}$$

onde T é o intervalo de tempo em que o jogo termina.

A função de valor de ação ótima $Q^*(s, a)$ pode ser definida como o máximo retorno esperado alcançável de uma estratégia, depois de ver a sequência s e se tomar alguma ação a:

$$Q^*(s, a) = max_\pi(\mathbb{E}[R_t | s_t = s, a_t = a, \pi]) \tag{5.2}$$

onde π é uma política que mapeia sequências para ações e $\mathbb{E}$ é a função de retorno esperado para um estado s dado uma ação a.

A função de valor de ação ótima obedece a identidade da equação de Bellman. Essa se baseia na seguinte intuição: se o valor ótimo $Q^*(s_{t+1}, a_{t+1})$ da sequência s_{t+1} na próxima etapa de tempo for conhecido para todas as ações possíveis ações a_{t+1}, então a estratégia ótima para o estado s_t consiste em selecionar a ação a_t que maximize o valor esperado futuro:

$$Q^*(s_t, a_t) = r + \gamma \cdot max(Q^*(s_{t+1}, a_{t+1})|\forall a_{t+1}) \tag{5.3}$$

A ideia básica por trás de muitos algoritmos de aprendizagem por reforço é estimar a função de valor de ação, usando a equação de Bellman como uma atualização iterativa. Assim, dado um fator de aprendizagem α, o valor de $Q(s, a)$ é atualizado durante o treinamento da seguinte forma:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r + \gamma \cdot max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{5.4}$$

sendo que a subtração de $\gamma \cdot max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$ por $Q(s_t, a_t)$ é realizada para normalizar a atualização.

5.3 Implementação

5.3.1 Pré-processamento de Dados

5.3.2 Limitações

Referências

AFRAZ DANIEL L.K. YAMINS, J. J. D. A. Neural mechanisms underlying visual object recognition. *Cold Spring Harb Symp Quant*, Cold Spring Harbor Laboratory Press; all rights reserved, 2014. Disponível em: <https://doi.org/10.1101/sqb.2014.79.024729>. Acesso em: 2 ago 2019. Citado na página 18.

BALDI, P.; SADOWSKI, P.; WHITESON, D. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, v. 5, n. 1, p. 4308, 2014. Disponível em: <https://doi.org/10.1038/ncomms5308>. Citado 2 vezes nas páginas 18 e 21.

BELLEMARE, M. G. et al. The arcade learning environment: An evaluation platform for general agents. *CoRR*, abs/1207.4708, 2012. Disponível em: <http://arxiv.org/abs/1207.4708>. Citado na página 29.

BOTVINICK, M. et al. Reinforcement learning, fast and slow. *Trends in Cognitive Sciences*, Elsevier, v. 23, n. 5, p. 408–422, 2019/10/17 2019. Disponível em: <https://doi.org/10.1016/j.tics.2019.02.006>. Citado na página 18.

COMI, M. *How to teach AI to play Games: Deep Reinforcement Learning*. 2018. Disponível em: <https://towardsdatascience.com/how-to-teach-an-ai-to-play-games-deep-reinforcement-learning-28f9b920440a>. Acesso em: 8 out 2019. Citado 2 vezes nas páginas 22 e 23.

DAHL, G. E.; JAITLY, N.; SALAKHUTDINOV, R. *Multi-task Neural Networks for QSAR Predictions*. 2014. Citado na página 18.

DWIBEDI, D.; VEMULA, A. 2016. Disponível em: <https://pdfs.semanticscholar.org/179d/04d9da112c16b6fa5310c273d66de65e5768.pdf>. Acesso em: 18 ago 2019. Citado na página 19.

GEBRU, T. et al. Using deep learning and google street view to estimate the demographic makeup of neighborhoods across the united states. *Proceedings of the National Academy of Sciences*, National Academy of Sciences, v. 114, n. 50, p. 13108–13113, 2017. ISSN 0027-8424. Disponível em: <https://www.pnas.org/content/114/50/13108>. Citado na página 21.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado 3 vezes nas páginas 13, 17 e 25.

HARGREAVES, S. *Allegro*. 1990. Disponível em: <https://liballeg.org/index.html>. Acesso em: 8 out 2019. Citado 3 vezes nas páginas 17, 23 e 28.

HSU, F.-H. *Behind Deep Blue: Building the Computer That Defeated the World Chess Champion*. Princeton, NJ, USA: Princeton University Press, 2002. ISBN 0691090653. Citado na página 15.

Karavolos, D.; Liapis, A.; Yannakakis, G. N. Using a surrogate model of gameplay for automated level design. In: *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. [S.l.: s.n.], 2018. p. 1–8. Citado na página 18.

MARR, B. *Is Artificial Intelligence Dangerous? 6 AI Risks Everyone Should Know About*. 2018. Disponível em: <https://www.forbes.com/sites/bernardmarr/2018/11/19/is-artificial-intelligence-dangerous-6-ai-risks-everyone-should-know-about/#256480952404>. Acesso em: 11 nov 2019. Citado na página 21.

MENABREA, L. et al. *Sketch of the Analytical Engine invented by Charles Babbage ... with notes by the translator. Extracted from the 'Scientific Memoirs,' etc. [The translator's notes signed: A.L.L. ie. Augusta Ada King, Countess Lovelace.]*. R. & J. E. Taylor, 1843. Disponível em: <https://books.google.com.br/books?id=hPRmnQEACAAJ>. Citado na página 13.

MILLINGTON, I.; FUNGE, J. *Artificial Intelligence for Games, Second Edition*. 2nd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009. ISBN 0123747317, 9780123747310. Citado na página 18.

MNIH, V. et al. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. Disponível em: <http://arxiv.org/abs/1312.5602>. Citado 2 vezes nas páginas 33 e 34.

MNIH, V. et al. Human-level control through deep reinforcement learning. *Nature*, Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved. SN -, v. 518, p. 529 EP –, 02 2015. Disponível em: <https://doi.org/10.1038/nature14236>. Acesso em: 2 ago 2019. Citado na página 18.

Nassif, A. B. et al. Speech recognition using deep neural networks: A systematic review. *IEEE Access*, v. 7, p. 19143–19165, 2019. Citado 2 vezes nas páginas 17 e 21.

NICHOLSON, C. *A Beginner's Guide to Deep Reinforcement Learning*. 2016. Disponível em: <https://skymind.ai/wiki/deep-reinforcement-learning>. Acesso em: 8 out 2019. Citado na página 27.

Piergigli, D. et al. Deep reinforcement learning to train agents in a multiplayer first person shooter: some preliminary results. In: *2019 IEEE Conference on Games (CoG)*. [S.l.: s.n.], 2019. p. 1–8. Citado na página 18.

Ripamonti, L. A. et al. Believable group behaviours for npcs in fps games. In: *2017 IEEE Symposium on Computers and Communications (ISCC)*. [S.l.: s.n.], 2017. p. 12–17. Citado na página 18.

RIPAMONTI, L. A. et al. Procedural content generation for platformers: designing and testing fun pledge. *Multimedia Tools and Applications*, v. 76, n. 4, p. 5001–5050, Feb 2017. ISSN 1573-7721. Disponível em: <https://doi.org/10.1007/s11042-016-3636-3>. Citado na página 18.

ROBU, V. et al. Consider ethical and social challenges in smart grid research. *Nature Machine Intelligence*, 2019. Disponível em: <https://doi.org/10.1038/s42256-019-0120-6>. Citado na página 13.

RODRIGUES, J. *O que é o Processamento de Linguagem Natural?* 2017. Disponível em: <https://medium.com/botsbrasil/o-que-é-o-processamento-de-linguagem-natural-49ece9371cff>. Acesso em: 2 set 2019. Citado na página 13.

SILVA, A. P. Ambiente para desenvolvimento de inteligência artificial em jogos allegro. Departmento de Ciência da Computação (DCC) da Universidade Federal de Minas Gerais (UFMG), Belo horizonte, Brasil, 2019. Disponível em: <https://github.com/artphil/allegro_game_ai>. Acesso em: 8 out 2019. Citado na página 29.

SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. 2. ed. [S.l.]: MIT Press, 2018. <http://incompleteideas.net/book/the-book.html>. Citado 3 vezes nas páginas 13, 25 e 26.

Tang, J. et al. Enabling deep learning on iot devices. *Computer*, v. 50, n. 10, p. 92–96, 2017. Citado na página 21.

TAO, Y. et al. A deep neural network modeling framework to reduce bias in satellite precipitation products. *Journal of Hydrometeorology*, v. 17, n. 3, p. 931–945, 2016. Disponível em: <https://doi.org/10.1175/JHM-D-15-0075.1>. Citado na página 21.

VINYALS, O. et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, v. 575, n. 7782, p. 350–354, 2019. Disponível em: <https://doi.org/10.1038/s41586-019-1724-z>. Citado na página 19.

YANNAKAKIS, G. N. Game ai revisited. In: *Proceedings of the 9th Conference on Computing Frontiers*. New York, NY, USA: ACM, 2012. (CF '12), p. 285–292. ISBN 978-1-4503-1215-8. Disponível em: <http://doi.acm.org/10.1145/2212908.2212954>. Citado na página 18.

YEUNG, S. et al. A computer vision system for deep learning-based detection of patient mobilization activities in the icu. *npj Digital Medicine*, v. 2, n. 1, p. 11, 2019. Disponível em: <https://doi.org/10.1038/s41746-019-0087-z>. Citado na página 18.

YOUNG, T. et al. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, Institute of Electrical and Electronics Engineers (IEEE), v. 13, n. 3, p. 55–75, Aug 2018. ISSN 1556-6048. Disponível em: <http://dx.doi.org/10.1109/mci.2018.2840738>. Citado na página 18.