

Marcos Vinícius da Silva

# **Aplicação de Redes Neurais e Aprendizado Por Reforço Profundo em Agentes Inteligentes em Jogos Eletrônicos**

Divinópolis - Brasil

31 de agosto de 2024

Marcos Vinícius da Silva

# **Aplicação de Redes Neurais e Aprendizado Por Reforço Profundo em Agentes Inteligentes em Jogos Eletrônicos**

Monografia apresentada ao Curso de Engenharia de Computação da UEMG Unidade Divinópolis, como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação, sob a orientação do Prof. Arismar

Universidade Estadual de Minas Gerais – UEMG

Unidade Divinópolis

Curso de Engenharia da Computação

Orientador: Arismar Moraes Gonçalves Junior

Divinópolis - Brasil

31 de agosto de 2024

Marcos Vinícius da Silva

# **Aplicação de Redes Neurais e Aprendizado Por Reforço Profundo em Agentes Inteligentes em Jogos Eletrônicos**

Monografia apresentada ao Curso de Engenharia de Computação da UEMG Unidade Divinópolis, como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação, sob a orientação do Prof. Arismar

Trabalho aprovado. Divinópolis - Brasil, 31 de agosto de 2024:

---

**Arismar Moraes Gonçalves Junior**  
Orientador

---

**Professor**  
Convidado 1

---

**Professor**  
Convidado 2

Divinópolis - Brasil  
31 de agosto de 2024

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
	Introdução	1
<b>1.1</b>	<b>Objetivos</b>	<b>1</b>
1.1.1	Objetivo Geral	1
1.1.2	Objetivos Específicos	1
<b>2</b>	<b>REVISÃO DE LITERATURA</b>	<b>3</b>
	Revisão de Literatura	3
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>5</b>
	Fundamentação Teórica	5
<b>3.1</b>	<b>Inteligência Artificial e Sua Aplicação em Jogos</b>	<b>5</b>
<b>3.2</b>	<b>Redes Neurais Artificiais</b>	<b>6</b>
<b>3.3</b>	<b>Aprendizado Por Reforço Profundo</b>	<b>10</b>
3.3.1	Apredizado Por Reforço	10
3.3.2	Aprendizado Profundo	11
3.3.3	Integração do Aprendizado por Reforço com Aprendizado Profundo	11
<b>4</b>	<b>METODOLOGIA</b>	<b>13</b>
	Metodologia	13
<b>5</b>	<b>RESULTADOS ESPERADOS</b>	<b>15</b>
	Resultados Esperados	15
<b>6</b>	<b>CRONOGRAMA</b>	<b>16</b>
	Cronograma	16
	<b>REFERÊNCIAS</b>	<b>17</b>

# Lista de ilustrações

Figura 1 – Representação de uma rede neural biológica. Fonte: (CUTTER, 2000).	6
Figura 2 – Representação de um neurônio artificial. Fonte: (AGGARWAL, 2018).	7

# Lista de tabelas

Tabela 1 – Cronograma . . . . .	16
---------------------------------	----

# Lista de abreviaturas e siglas

IA	Inteligência artificial
DRL	<i>Deep reinforcement learning</i> , aprendizado por reforço profundo
GOAP	<i>Goal-oriented action planners</i> , planejadores de ações orientados a metas
NPC	<i>Non-playable character</i> , personagem não-jogável
ReLU	<i>Rectified linear unit</i> , unidade linear retificada

# Lista de símbolos

$z$	Letra latina $z$
$\Sigma$	Operador de somatório
$i$	Índice do somatório
$n$	Limite superior do somatório
$w_i$	Peso com índice $i$
$\cdot$	Símbolo de multiplicação
$x_i$	Entrada com índice $i$
$b$	Termo de polarização
$f(z)$	Função que aplica a transformação sobre $z$
$\max$	Função máxima
$a$	Saída do neurônio
$f$	Função de ativação
$\eta$	Taxa de aprendizado
$\frac{\partial J}{\partial w_i}$	Gradiente da função de custo em relação ao peso $w_i$
$\frac{\partial E}{\partial w_i}$	Gradiente do erro em relação ao peso $w_i$
$\frac{\partial E}{\partial a}$	Gradiente do erro em relação à saída $a$
$\frac{\partial a}{\partial z}$	Gradiente da saída $a$ em relação à entrada $z$
$\frac{\partial z}{\partial w_i}$	Gradiente da entrada $z$ em relação ao peso $w_i$



# Resumo

Este trabalho tem como objetivo explorar a aplicação de redes neurais artificiais (RNAs) e aprendizado por reforço profundo (DRL) na criação de agentes inteligentes em jogos eletrônicos, utilizando técnicas de aprendizado de máquina para desenvolver agentes que interajam de maneira dinâmica com o ambiente, jogadores e outros agentes, adaptando seus comportamentos com base nas experiências adquiridas e acumuladas. A metodologia envolve a implementação de uma rede neural artificial que processa entradas complexas do ambiente e ajusta o comportamento dos agentes através do uso de aprendizado por reforço profundo. A função de ativação ReLU foi escolhida para garantir eficiência no treinamento das redes neurais. O desempenho dos agentes será avaliado através de sua adaptabilidade, a singularidade de seus comportamentos e a coerência dos mesmos para com o ambiente virtual em que se encontram, com ajustes contínuos para melhorar esses parâmetros.

**Palavras-chaves:** redes neurais artificiais, aprendizado por reforço profundo, agentes inteligentes, desenvolvimento de jogos.

# Abstract

This work aims to explore the application of artificial neural networks (ANNs) and deep reinforcement learning (DRL) in the creation of intelligent agents in electronic games, using machine learning techniques to develop agents that interact dynamically with the environment, players and other agents, adapting their behaviors based on acquired and accumulated experiences. The methodology involves implementing an artificial neural network that processes complex inputs from the environment and adjusts the behavior of agents through the use of deep reinforcement learning. The ReLU activation function was chosen to ensure efficiency in training the neural networks. The agents' performance will be evaluated through their adaptability, the uniqueness of their behaviors and their coherence with the virtual environment in which they find themselves, with continuous adjustments to improve these parameters.

**Key-words:** artificial neural networks, deep reinforcement learning, intelligent agents, game development.

# 1 Introdução

A inteligência artificial (IA) é parte integral do desenvolvimento de jogos, onde sua aplicação impacta diretamente a experiência do jogador. Esse campo de estudo está em constante evolução, e suas décadas de conhecimento acumulado são usadas das mais diversas maneiras para a criação de variados agentes inteligentes em jogos eletrônicos (DILL, 2015).

Uma das formas mais sofisticadas de inteligência artificial existentes atualmente são as redes neurais artificiais, que podem corroborar para a criação de IAs mais inteligentes e com a capacidade de tomar decisões mais complexas, através da captação de dados do ambiente virtual em que se encontra, o processamento desses dados pelo modelo de rede aplicado e a determinação de ações a serem tomadas em sua saída (NIELSEN, 2015).

O modelo de aprendizado por reforço profundo é especialmente interessante como estrutura de tomada de decisões para agentes inteligentes em jogos eletrônicos por sua capacidade de aprendizado e adaptação dinâmicas, sua consideração sequencial de tomadas de decisão ao longo do tempo, considerando as consequências a longo prazo, e sua boa performance em ambientes altamente mutáveis e complexos (MNIH et al., 2015).

## 1.1 Objetivos

Este trabalho tem como objetivo o estudo e aplicação de redes neurais e aprendizado por reforço profundo na criação de agentes inteligentes em um jogo eletrônico que, ao decorrer de diferentes interações com o “mundo” em que se encontram, seja essa interação com o jogador, elementos estáticos do “mundo”, ou outros agentes inteligentes, modifique seu comportamento de forma a se adaptar às novas condições.

### 1.1.1 Objetivo Geral

Aprofundamento no estudo de redes neurais artificiais (RNAs) e aprendizado por reforço profundo (*deep reinforcement learning*, ou DRL) com foco em sua aplicação na criação de inteligência artificial para agentes inteligentes em jogos eletrônicos, criando agentes mais adaptáveis ao ambiente atual e à interferência de terceiros.

### 1.1.2 Objetivos Específicos

- Implementar um modelo de inteligência artificial que gerencie o comportamento de agentes inteligentes em um jogo eletrônico.

- Aplicar técnicas de aprendizado por reforço profundo para dar a capacidade aos agentes de aprenderem e modificarem seus comportamentos em tempo real através da interação com o jogador, outros agentes, ou o ambiente.
- Implementar um modelo de inteligência artificial que dê individualidade aos agentes no jogo através de suas diferentes experiências “vividas”.
- Comparar a aplicação da abordagem desenvolvida com abordagens já presentes no mercado e sua eficácia em criar agentes inteligentes e altamente adaptáveis.
- Gerar uma maior imersão do jogador no ambiente virtual através da realização desses agentes mais inteligentes e adaptáveis.

## 2 Revisão de Literatura

A inteligência artificial é uma parte inseparável da maioria dos jogos eletrônicos existentes, pois muitos deles possuem adversários ou auxiliares que definem parte integral da experiência do jogo, e é amplamente utilizada para criar comportamentos interessantes para os agentes inteligentes, contribuindo para a imersão e complexidade dos desafios oferecidos aos jogadores (DILL, 2015).

Diversas aplicações de IA, incluindo redes neurais artificiais, já são usadas em jogos de diferentes tipos. No entanto, muitos jogos não disponibilizam seu código-fonte publicamente, nem os métodos utilizados para alcançar os resultados vistos. Essa falta de transparência limita a análise e compreensão dos métodos de IA aplicados em diferentes contextos de jogos eletrônicos.

O estudo "*Player-IA Interaction: What Neural Network Games Reveal About AI as Play*", de Zhu et al. (2021), apresenta uma coletânea de jogos onde redes neurais são utilizadas de diferentes formas e discute a interação humano-máquina. O estudo classifica as redes neurais em quatro tipos principais: aprendizes (a rede aprende com o jogador a executar uma tarefa), competidores (a rede aprende como o jogador joga para gerar desafios), projetistas (a rede neural cria elementos para interação do jogador) e parceiros (a rede e o jogador colaboram em direção a um objetivo comum). Esse estudo é particularmente relevante para dar a capacidade aos agentes de se encaixarem dinamicamente nas funções descritas em tempo real através da interação com o jogador, outros agentes, ou o ambiente se ajustando a diferentes papéis conforme o contexto.

O livro "*Game AI Pro*", editado por Rabin et al. (2015), aborda uma vasta gama de tópicos relacionados à IA em jogos, desde a base teórica do funcionamento dos neurônios biológicos até aplicações práticas e complexas, como o uso de redes neurais para criar experiências imersivas. O livro é dividido em capítulos, cada um escrito por diferentes profissionais da indústria de jogos e pesquisadores da área de IA, oferecendo uma visão diversificada e abrangente sobre o impacto da IA em jogos. Tanto os tópicos teóricos quanto os tópicos mais abstratos são de extrema relevância para o projeto atual, já que as aplicações dessas técnicas de IA devem oferecer ao jogador uma experiência singular, mas ainda assim agradável e imersiva.

O livro "*Artificial Intelligence: A Modern Approach*", de Russell e Norvig (2020), é uma das obras mais abrangentes sobre IA. Foca nos fundamentos da IA, agentes inteligentes e sua interação com o ambiente, resolução de problemas e lógica. Além disso, o livro aborda a aplicação de IA em jogos estratégicos, como o xadrez, no contexto de algoritmos de busca e raciocínio estratégico, fornecendo uma base sólida para a compreensão dos conceitos

fundamentais da IA aplicados a jogos. Esses conceitos fornecem a base teórica necessária para desenvolver algoritmos que permitam aos agentes aprender e se adaptar a partir de suas interações.

O livro "*Neural Networks and Deep Learning: A Textbook*", de Aggarwal (2018), explora o uso de redes neurais profundas para aproximar funções de valor e funções de ação-valor em aprendizado por reforço profundo. O livro oferece uma análise detalhada sobre como as redes neurais podem ser empregadas para aprender e otimizar políticas de decisão em ambientes complexos. Isso é crucial para o objetivo de implementar um modelo de inteligência artificial que dê individualidade aos agentes no jogo através de suas diferentes experiências "vividas", ao permitir que cada agente aprenda e se adapte de forma única, criando comportamentos emergentes que refletem suas interações e experiências passadas.

O artigo "*Human-level Control Through Deep Reinforcement Learning*", de Mnih et al. (2015), demonstra a aplicação de redes neurais profundas para alcançar controle em nível humano em jogos, como Atari. O estudo é fundamental para compreender a eficácia das técnicas de aprendizado por reforço profundo em jogos complexos, permitindo que agentes aprendam e otimizem estratégias avançadas, se correlacionando com o objetivo de dar a capacidade aos agentes de aprenderem e modificarem seus comportamentos em tempo real.

O artigo "*Continuous Control With Deep Reinforcement Learning*", de Lillicrap et al. (2016), explora o uso de DRL para controlar sistemas contínuos, como robôs. O estudo detalha como o DRL pode ser aplicado para realizar tarefas complexas em ambientes variados, incluindo manipulação de objetos e navegação, ajudando com o fornecimento de bases técnicas para o desenvolvimento de agentes que gerenciem o próprio comportamento.

O livro "*Neural Networks and Learning Machines*", de Haykin (2008), fornece uma visão abrangente sobre redes neurais e suas aplicações, discutindo funções de ativação e métodos de treinamento. O artigo "*Learning Representations by Back-Propagating Errors*", de Rumelhart, Hinton e Williams (1986), descreve o algoritmo de retropropagação, uma técnica fundamental para o treinamento de redes neurais. Ambos os estudos são cruciais para formar uma base técnica para o desenvolvimento das redes neurais artificiais.

O livro "*Neural Networks and Deep Learning: A Textbook*", de Nielsen (2015), discute o treinamento de redes neurais e a detecção de overfitting, oferecendo detalhes sobre como garantir que a rede neural generalize bem e não apenas memorize os dados de treinamento. Isso é essencial para a implementação de um modelo de IA que dê individualidade aos agentes no jogo, assegurando que o comportamento dos agentes seja único e dinâmico.

## 3 Fundamentação Teórica

### 3.1 Inteligência Artificial e Sua Aplicação em Jogos

Inteligência artificial (IA) é um campo de estudo que tem como objetivo o desenvolvimento de sistemas que apresentem a capacidade de realizarem tarefas complexas realizadas por seres humanos e que requerem certo nível de inteligência. Dentre essas tarefas podem ser citadas a tomada de decisões a partir da análise de dados, aprendizagem contínua, reconhecimento de padrões, adaptação a situações inusitadas, dentre outras. IA, como conceito, diz respeito tanto ao desenvolvimento de sistemas que superam a capacidade humana em suas atividades, quanto à simulação em máquinas de processos cognitivos humanos (RUSSEL; NORVIG, 2020).

Na década de 1950, pesquisadores como Alan Turing apresentaram a teoria de que as máquinas poderiam replicar o funcionamento da mente humana (TURING, 1950). A evolução tecnológica das últimas décadas, pareada com o constante aprofundamento dos estudos na área, levaram a uma grande evolução da IA, estando hoje presente em diversas áreas, como a medicina, automação industrial, sistemas autônomos, e vários outros setores e aplicações (GOODFELLOW; BENGIO; COURVILLE, 2016), como o mercado de jogos eletrônicos, que é o foco deste trabalho.

Jogos eletrônicos têm como um de seus principais objetivos criar uma experiência para o jogador, seja através de sua história, personagens, ou qualquer outro elemento. A IA desempenha um importante papel na criação dessa imersão. Uma boa IA se torna indispensável para que não haja a quebra da suspensão de descrença, já que a partir do momento em que o jogador começa a tratar as interações do jogo apenas como saídas de um programa em uma máquina e não como interações orgânicas, o objetivo da IA no jogo falhou (DILL, 2015).

Um dos usos mais comuns de inteligência artificial em jogos é para a definição de ações de personagens não-jogáveis (*non-playable characters*, ou NPCs). Máquinas de estado finito (*finite-state machines*, ou FSMs) são comumente usadas para a definição de comportamentos e tomadas de decisões simples através da avaliação do estado atual do “mundo”. Árvores de comportamento (*behavior trees*) podem ser usadas para modelar uma seleção de decisões mais complexa através de sua estrutura de dados de nós e filhos, permitindo a filtragem das ações corretas a serem tomadas. Planejadores de ações orientados a metas (*goal-oriented action planners*, ou GOAPs) podem ser usados para determinar uma sequência de ações a partir do estado atual do “mundo” e as ações disponíveis. Há uma vasta gama de opções de algoritmos de seleção de comportamento,

e sua escolha depende dos objetivos específicos que se querem alcançar com os NPCs designados (DAWE et al., 2015).

## 3.2 Redes Neurais Artificiais

Redes neurais artificiais são ferramentas modeladas para simular a maneira como o cérebro humano processa informações, o que permite que sistemas de IA aprendam com dados, identifiquem padrões e tomem decisões de forma autônoma e eficiente. As redes neurais são a base de muitos avanços recentes em IA, especialmente no campo do aprendizado profundo (*deep learning*), que impulsionou inovações em reconhecimento de fala, visão computacional, tradução automática e até mesmo em jogos, tema foco deste trabalho.

Os neurônios biológicos são as células do sistema nervoso responsáveis pela transmissão de informações. Um neurônio é constituído por três partes principais: o soma (corpo celular), os dendritos e o axônio. O corpo celular contém o núcleo e é essencial para a sobrevivência da célula. Os dendritos são prolongamentos que recebem sinais de outros neurônios e os transmitem para o soma. O axônio, por sua vez, é uma extensão longa que conduz impulsos elétricos do corpo celular para outros neurônios, músculos ou glândulas (AGGARWAL, 2018).

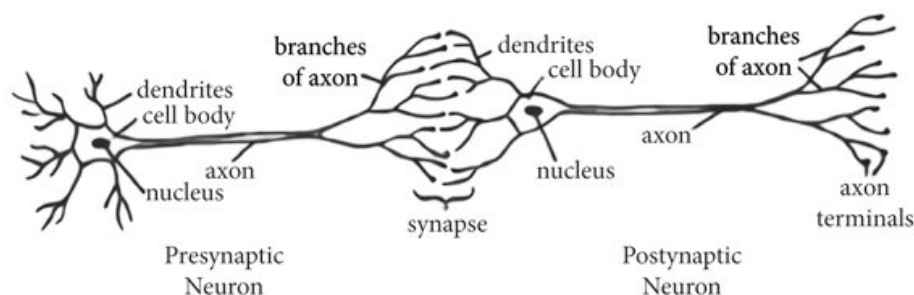


Figura 1 – Representação de uma rede neural biológica. Fonte: (CUTTER, 2000).

A comunicação entre os neurônios se dá por meio de sinais elétricos e químicos. Quando um neurônio recebe um estímulo suficiente, ele gera um impulso elétrico chamado potencial de ação, que se propaga ao longo do axônio até as terminações axonais. Nessas terminações, o potencial de ação provoca a liberação de neurotransmissores, substâncias químicas que atravessam a sinapse, a junção entre dois neurônios, e se ligam a receptores nos dendritos do próximo neurônio, continuando o processo de transmissão do sinal (AGGARWAL, 2018).

As redes neurais artificiais são formadas por neurônios artificiais inspirados nos neurônios biológicos. Esses neurônios artificiais recebem entradas (que podem ser dados ou



sinais de outros neurônios), processam essas entradas através de uma função de ativação e geram uma saída. Cada entrada é multiplicada por um peso, que determina sua importância. As entradas ponderadas são somadas, resultando em uma soma ponderada. Esta soma é então passada por uma função de ativação que decide se o neurônio deve "disparar" e passar sua saída adiante na rede (GOODFELLOW; BENGIO; COURVILLE, 2016).

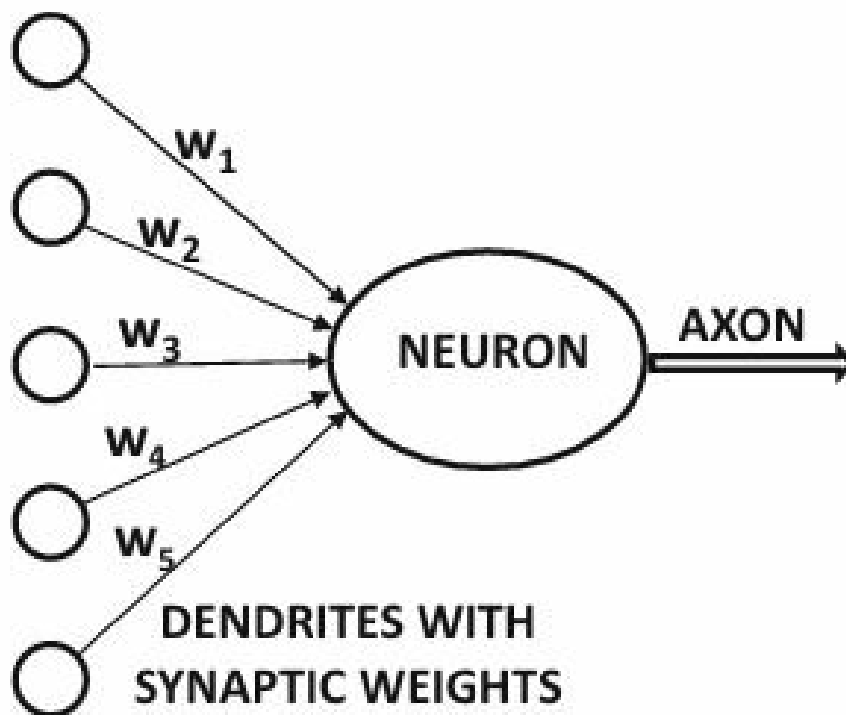


Figura 2 – Representação de um neurônio artificial. Fonte: (AGGARWAL, 2018).

Embora os neurônios artificiais sejam inspirados nos biológicos, eles são simplificações extremas. Os neurônios biológicos operam através de interações eletroquímicas complexas e possuem milhares de conexões com outros neurônios, enquanto os neurônios artificiais são modelados usando operações matemáticas básicas e têm um número limitado de conexões, dependendo da arquitetura da rede neural (AGGARWAL, 2018).

O funcionamento dos neurônios biológicos é contínuo e não linear, enquanto os neurônios artificiais geralmente funcionam de maneira discreta com uma função de ativação definida, que é muito mais simples que o comportamento real de um neurônio biológico. Essa simplificação permite que os neurônios artificiais sejam computacionalmente eficientes, embora não capturem toda a complexidade dos neurônios biológicos (GOODFELLOW; BENGIO; COURVILLE, 2016).

As redes neurais artificiais (RNAs) são sistemas computacionais inspirados na estrutura e no funcionamento do cérebro humano, projetadas para reconhecer padrões, aprender com dados e tomar decisões. Essas redes são compostas por unidades chamadas

"neurônios artificiais", organizadas em camadas. As principais camadas são a camada de entrada, a camada oculta (ou camadas ocultas) e a camada de saída.

Cada neurônio recebe um ou mais sinais de entrada, que são multiplicados por pesos sinápticos. Esses sinais ponderados são somados junto a um termo de polarização, formando o valor de entrada total do neurônio, que pode ser representado pela equação (3.1):

$$z = \sum_{i=1}^n w_i \cdot x_i + b \quad (3.1)$$

Onde  $z$  é a soma ponderada,  $w_i$  são os pesos,  $x_i$  são as entradas, e  $b$  é o viés (ou *bias*), que é um valor constante que é adicionado à combinação das entradas ponderadas para permitir o ajuste da saída do neurônio de forma mais flexível, deslocando a função de ativação (GOODFELLOW; BENGIO; COURVILLE, 2016).

O resultado  $z$  é então passado por uma função de ativação  $f(z)$ , que determina a saída do neurônio. A função de ativação pode ser linear ou não linear, sendo as funções mais comuns a função sigmoide, ReLU (Rectified Linear Unit) e tangente hiperbólica (HAYKIN, 2008).

A função de ativação ReLU, por exemplo, pode ser expressa de acordo com a equação (3.2):

$$f(z) = \max(0, z) \quad (3.2)$$

O processo de aprendizado das RNAs envolve a minimização de uma função de custo, que mede o quão longe as previsões da rede estão dos valores reais. Um método amplamente utilizado para esse ajuste é o algoritmo de retropropagação, que ajusta os pesos de cada neurônio com base no erro da saída (RUMELHART; HINTON; WILLIAMS, 1986). Esse ajuste é feito utilizando o gradiente descendente, onde os pesos são atualizados de acordo com a equação (3.3):

$$w_{i+1} = w_i - \eta \cdot \frac{\partial J}{\partial w_i} \quad (3.3)$$

Onde  $\eta$  é a taxa de aprendizado e  $\frac{\partial J}{\partial w_i}$  é o gradiente da função de custo em relação ao peso  $w_i$  (GOODFELLOW; BENGIO; COURVILLE, 2016).

Assim, o treinamento de uma rede neural é um processo iterativo, onde os pesos são continuamente ajustados até que a função de custo atinja um mínimo, indicando que a rede foi treinada com sucesso (LECUN; BENGIO; HINTON, 2015).

Após entender a estrutura básica de uma rede neural artificial (RNA) e as funções de ativação, é importante explorar o processo de propagação e aprendizado, que inclui o *forward pass*, o *backpropagation*, o treinamento e a validação.

O *forward pass* é a etapa em que os dados de entrada são passados pela rede, camada por camada, até que se obtenha uma previsão na camada de saída. Nessa fase, as entradas são multiplicadas pelos pesos sinápticos, somadas ao termo de polarização, e então transformadas por uma função de ativação em cada neurônio, resultando na previsão da rede. A fórmula geral para a saída de um neurônio é dada de acordo com a equação (3.4):

$$a = f \left( \sum_{i=1}^n w_i \cdot x_i + b \right) \quad (3.4)$$

Onde  $a$  representa a saída do neurônio após a função de ativação, e  $f$  é a função de ativação.

O processo de *backpropagation* é responsável por ajustar os pesos da rede com base no erro entre a previsão da rede e o valor real. O erro é calculado usando uma função de custo, e o gradiente do erro em relação a cada peso é calculado usando a regra da cadeia. Os pesos são então ajustados na direção que minimiza o erro. O erro  $E$  pode ser expresso pela função de custo, e o gradiente do erro em relação ao peso  $w_i$  é dado de acordo com a equação (3.5):

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_i} \quad (3.5)$$

Onde  $a$  é a saída da rede e  $z$  é a soma ponderada das entradas. A atualização do peso é feita de acordo com a equação (3.3), onde  $\eta$  é a taxa de aprendizado (GOODFELLOW; BENGIO; COURVILLE, 2016).

O treinamento de uma RNA envolve aplicar repetidamente o *forward pass* e o *backpropagation* em um conjunto de dados de treinamento, ajustando os pesos com base nos erros calculados. Este processo ocorre ao longo de várias iterações chamadas de épocas. Durante cada época, a rede processa todo o conjunto de dados de treinamento, ajustando os pesos a cada iteração. O objetivo do treinamento é encontrar o conjunto de pesos que minimize a função de custo, permitindo que a rede generalize bem para dados não vistos anteriormente (LECUN; BENGIO; HINTON, 2015).

A taxa de aprendizado  $\eta$  é um parâmetro importante durante o treinamento, pois determina o tamanho dos ajustes nos pesos em cada iteração. Uma taxa de aprendizado muito alta pode fazer com que a rede oscile em torno do mínimo da função de custo, enquanto uma taxa muito baixa pode resultar em um treinamento muito lento (HAYKIN, 2008).

A validação é um passo essencial no treinamento de redes neurais para garantir que o modelo treinado generalize bem para novos dados. Para isso, o conjunto de dados é geralmente dividido em três subconjuntos: treinamento, validação e teste. O conjunto de validação é usado para avaliar o desempenho da rede após cada época de treinamento e ajustar hiperparâmetros, como a taxa de aprendizado e a arquitetura da rede (GOODFELLOW; BENGIO; COURVILLE, 2016).

Durante a validação, a função de custo é calculada no conjunto de validação, mas os pesos não são ajustados. Isso permite verificar se a rede está começando a superajustar (*overfitting*) os dados de treinamento, ou seja, aprender os ruídos ou detalhes específicos desse conjunto, em vez de captar padrões generalizáveis. Um sinal de *overfitting* é quando o erro no conjunto de validação começa a aumentar enquanto o erro no conjunto de treinamento continua a diminuir (NIELSEN, 2015).

Ao final do treinamento, o desempenho do modelo é finalmente avaliado utilizando o conjunto de teste, que contém dados nunca vistos pela rede, fornecendo uma estimativa da capacidade de generalização do modelo (GOODFELLOW; BENGIO; COURVILLE, 2016).

### 3.3 Aprendizado Por Reforço Profundo

O aprendizado por reforço profundo (*Deep Reinforcement Learning*, ou DRL) é uma abordagem que combina o aprendizado por reforço (*Reinforcement Learning*, ou RL) com aprendizado profundo (*Deep Learning*). O DRL é eficaz para treinar agentes a tomar decisões complexas e otimizar comportamentos em ambientes dinâmicos, sem a necessidade de supervisão explícita.

#### 3.3.1 Aprendizado Por Reforço

##### 3.3.1.1. Aprendizado por Reforço

O aprendizado por reforço é uma abordagem de aprendizado de máquina em que um agente aprende a tomar decisões otimizadas através de interações com um ambiente. O objetivo é maximizar uma função de recompensa ao longo do tempo, ajustando suas ações com base nos feedbacks recebidos do ambiente. O processo é definido por quatro componentes principais:

- i) **Agente:** O sistema que toma decisões.
- ii) **Ambiente:** O contexto com o qual o agente interage.
- iii) **Ação:** As escolhas que o agente pode fazer.

- iv) **Recompensa:** O feedback do ambiente que indica a qualidade das ações do agente.

O agente segue uma política, que é uma estratégia para escolher ações com base no estado atual do ambiente. A política pode ser determinística ou estocástica. A função de valor mede a qualidade de um estado ou de uma ação, e o objetivo do aprendizado é descobrir uma política que maximize a recompensa acumulada ao longo do tempo (SUTTON; BARTO, 2018).

### 3.3.2 Aprendizado Profundo

O aprendizado profundo envolve o uso de redes neurais profundas para extrair características complexas e hierárquicas dos dados. As redes neurais profundas, compostas por múltiplas camadas de neurônios, são capazes de capturar representações sofisticadas dos dados e realizar tarefas como reconhecimento de imagem, processamento de linguagem natural e, mais recentemente, aprendizado por reforço.

No contexto do DRL, redes neurais profundas são usadas para aproximar funções de valor e políticas. Isso é especialmente útil em ambientes com grandes espaços de estado e ação, onde métodos tradicionais de aprendizado por reforço se tornam impraticáveis (GOODFELLOW; BENGIO; COURVILLE, 2016).

### 3.3.3 Integração do Aprendizado por Reforço com Aprendizado Profundo

No DRL, redes neurais profundas são empregadas para aproximar funções de valor ( $V(s)$ ) e funções de ação-valor ( $Q(s, a)$ ). A função de valor  $V(s)$  estima a recompensa esperada a partir de um estado  $s$ , enquanto a função de ação-valor  $Q(s, a)$  estima a recompensa esperada para uma ação  $a$  em um estado  $s$ . As redes neurais profundas permitem que essas funções sejam aproximadas de forma eficaz, mesmo em espaços de estado e ação muito grandes (Mnih et al., 2015).

Alguns dos algoritmos mais notáveis de DRL incluem:

- **Deep Q-Networks (DQN):** Introduzido por Mnih et al. (2015), o DQN usa uma rede neural profunda para aproximar a função  $Q$  e é capaz de aprender políticas eficientes para uma variedade de jogos e tarefas. O algoritmo incorpora técnicas como a experiência de replay e a atualização de alvo fixo para melhorar a estabilidade do treinamento.
- **Proximal Policy Optimization (PPO):** Proposto por Schulman et al. (2017), o PPO é um algoritmo de política que busca otimizar a política de maneira estável e eficiente. O PPO é conhecido por sua simplicidade e desempenho robusto em ambientes contínuos e discretos.

- **Trust Region Policy Optimization (TRPO):** Desenvolvido por Schulman et al. (2015), o TRPO melhora a estabilidade do treinamento de políticas ao garantir que as atualizações da política não desviem muito da política anterior.

O DRL tem sido aplicado em uma ampla gama de áreas, incluindo:

- **Jogos:** O DRL tem alcançado sucesso em jogos complexos como o Atari, Go e StarCraft II, demonstrando a capacidade dos agentes para aprender e otimizar estratégias avançadas (SILVER et al., 2016).
- **Robótica:** Em robótica, o DRL é utilizado para ensinar robôs a realizar tarefas complexas, como manipulação de objetos e navegação em ambientes desconhecidos (LILLICRAP et al., 2016).
- **Controle de Sistemas:** O DRL também é utilizado para otimizar o controle de sistemas em tempo real, como a gestão de redes elétricas e sistemas de tráfego (MNIH et al., 2015).

## 4 Metodologia

A abordagem metodológica para a criação de agentes inteligentes em um jogo eletrônico, utilizando redes neurais e aprendizado por reforço profundo, com o objetivo de desenvolver agentes que possam interagir de forma dinâmica com o cenário, jogadores e outros agentes, adaptando seu comportamento com base nas experiências acumuladas, passou primeiro por um extenso período de pesquisa sobre os tópicos envolvidos, com foco em redes neurais artificiais (RNAs) e aprendizado por reforço profundo (DRL).

Como é possível perceber pelo referencial teórico apresentado anteriormente, RNAs se encaixam muito bem para o objetivo deste trabalho. A rede neural irá receber entradas, processar esses dados e então alterar o comportamento do agente inteligente de acordo. Essas entradas serão observações do estado atual do ambiente, ações realizadas pelo jogador ou outros agentes, ou consequências de ações realizadas em momentos passados.

Jogos eletrônicos frequentemente apresentam ambientes complexos, o que resulta em entradas complexas para a rede neural. O aprendizado por reforço profundo é eficaz para lidar com essas complexidades porque combina o aprendizado por reforço (*reinforcement learning*, ou RL) com redes neurais profundas (*deep neural networks*, ou DNN), permitindo que os agentes aprendam e tomem decisões a partir de grandes quantidades de dados e identifiquem padrões complexos que seriam difíceis de modelar com técnicas tradicionais de RL (MNIH et al., 2015).

Inicialmente, será definida uma política de comportamento padrão para cada tipo de agente, a fim de dar a eles um comportamento inicial esperado. Definir uma política inicial permitirá um certo nível de controle sobre o design dos agentes, e também garantirá uma variedade de comportamentos iniciais. Esse comportamento será então alterado dinamicamente a partir de suas interações e as saídas da rede neural.

O DRL é capaz de melhorar continuamente essa política através de feedback contínuo, otimizando as ações para maximizar a recompensa acumulada ao longo do tempo. Essa capacidade de adaptação é crucial para criar agentes que se comportem de maneira adequada ao meio em que se encontram (SUTTON; BARTO, 2018).

Para as funções de ativação da rede neural a ser implementada, será utilizada a função ReLU (rectified linear unit), definida pela equação (3.2), onde  $x$  é a entrada para o neurônio. Essa simplicidade permite cálculos rápidos e eficientes, o que é crucial para o treinamento de redes neurais profundas, onde a velocidade de processamento pode ser um fator limitante. Se a entrada  $x$  for maior ou igual a zero, a saída da ReLU é igual à entrada. Se a entrada  $x$  for menor que zero, a saída da ReLU é zero, inibindo a propagação de valores negativos (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Em redes neurais profundas, funções de ativação como a sigmoide e a tanh podem sofrer com o problema de gradiente desvanecido, onde os gradientes se tornam muito pequenos, dificultando o aprendizado. A ReLU, por outro lado, não sofre desse problema da mesma maneira porque seu gradiente é constante (1) para valores positivos. Isso ajuda a manter gradientes significativos durante o treinamento e permite um fluxo mais eficiente de informações através das camadas da rede neural (GLOROT; BENGIO; ALLAUZEN, 2011). Essa função também promove um treinamento mais eficiente e uma melhor capacidade de generalização, pois ativa somente uma fração dos neurônios em qualquer dado ponto, pois todos os valores negativos são transformados em zero, reduzindo a redundância (HINTON; NAIR; RUSSELL, 2012).

Após o processamento da rede neural, o comportamento do agente será modificado de acordo com as entradas recebidas. Como exemplo, se o agente continuamente se beneficia por ficar junto de outro agente, por exemplo, com acesso mais fácil a comida, esse agente aprenderá que sua relação com o outro é benéfica para si, e tenderá a ficar próximo desse agente. Da mesma forma, um agente carnívoro, que caça como fonte principal de comida, poderá passar a evitar certos locais onde ele sofreu muito dano do ambiente, pois sua recompensa por estar naquele local era muito baixa.

Para evitar que todos os agentes convergirem nos mesmos comportamentos, será introduzida na política padrão de comportamento de cada um um nível de tolerância a mudanças, de forma randômica, para garantir uma vasta gama de diferentes comportamentos.

O ambiente e seus agentes, assim como o jogador, serão componentes 3D, desenvolvidos com o aplicativo de modelagem Blender, uma ferramenta de código aberto com ampla quantidade de materiais educativos disponíveis online, e com uma grande quantidade de ferramentas disponíveis para utilização.

O jogo em geral, sua programação, design, mecânicas e execução da rede neural farão uso do motor gráfico Godot, também de código aberto, e que oferece ferramentas para facilitar o desenvolvimento, como renderizadores de partículas e simuladores de física, dentre outras ferramentas.

A eficácia dos agentes será avaliada com base em seu desempenho em diferentes cenários e sua capacidade de adaptação às mudanças. Métricas de desempenho incluirão a capacidade de maximizar recompensas, a adaptabilidade a diferentes estratégias do jogador, a interação com outros agentes e a variedade de comportamentos gerados. Ajustes serão feitos nas redes neurais e nos algoritmos de aprendizado por reforço para melhorar o desempenho dos agentes conforme necessário. Passados os testes iniciais, serão realizados também testes com outras pessoas, a fim de coletar feedback sobre a implementação.



## 5 Resultados Esperados

Ao fim do projeto, espera-se que a rede neural artificial esteja funcionando devidamente, e o aprendizado por reforço profundo esteja produzindo cenários interessantes criados através da experiência de cada agente.

Os agentes devem demonstrar inteligência superior se comparados a agentes que utilizam técnicas mais simples, como máquinas de estado finito. Também devem ser altamente adaptáveis, mesmo que ocorram mudanças bruscas no ambiente e em suas alterações.

Mesmo que os agentes tenham completa autonomia para modificar suas políticas de comportamento, ainda assim deve haver espaço para o design intencional de comportamento durante o desenvolvimento, a fim de criar cenários e interações mais interessantes.

Também é imprescindível que haja uma evolução do conhecimento sobre o desenvolvimento de jogos, redes neurais e aprendizado de máquina ao fim do projeto.

Por fim, o resultado mais importante é o de que o jogo esteja em um estado jogável, e que apresente engajamento e imersão suficientemente grandes, e que os comportamentos não se afunilem em certos padrões depois de certo tempo.

[illegible]

# Referências

- AGGARWAL, C. C. *Neural Networks and Deep Learning: A Textbook*. 1st. ed. Cham: Springer, 2018. Citado 3 vezes nas páginas ii, 6 e 7.
- CUTTER, M. A. G. *Brain: Understanding Neurobiology through the Study of Addiction*. Colorado Springs: BSCS, 2000. Citado 2 vezes nas páginas ii e 6.
- DAWE, M. et al. Behavior selection algorithms: An overview. In: RABIN, S. (Ed.). *Game AI Pro: Collected Wisdom of Game AI Professionals*. 1st. ed. [S.l.]: CRC Press, 2015. p. 47–60. Citado na página 6.
- DILL, K. What is game ai. In: RABIN, S. (Ed.). *Game AI Pro: Collected Wisdom of Game AI Professionals*. 1st. ed. [S.l.]: CRC Press, 2015. p. 3–9. Citado 3 vezes nas páginas 1, 3 e 5.
- GLOROT, X.; BENGIO, Y.; ALLAUZEN, A. Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*. [S.l.: s.n.], 2011. Citado na página 14.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge: MIT Press, 2016. Citado 6 vezes nas páginas 5, 7, 8, 9, 10 e 11.
- HAYKIN, S. *Neural Networks and Learning Machines*. 3rd. ed. Upper Saddle River: Pearson Education, 2008. Citado 2 vezes nas páginas 8 e 9.
- HINTON, G. E.; NAIR, V.; RUSSELL, B. Layer normalization. 2012, 2012. Citado na página 14.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS)*. [S.l.: s.n.], 2012. Citado na página 13.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, 2015. Citado 2 vezes nas páginas 8 e 9.
- LILLICRAP, T. et al. Continuous control with deep reinforcement learning. In: *Proceedings of the 4th International Conference on Learning Representations (ICLR)*. San Juan: [s.n.], 2016. Citado na página 12.
- MNIH, V. et al. Human-level control through deep reinforcement learning. *Nature*, v. 518, n. 7540, p. 529–533, 2015. Citado 3 vezes nas páginas 1, 12 e 13.
- NIELSEN, M. A. *Neural Networks and Deep Learning: A Textbook*. [s.n.], 2015. Determination of whether a neural network is overfitting or not, based on validation and training errors. Disponível em: <<http://neuralnetworksanddeeplearning.com/>>. Citado 2 vezes nas páginas 1 e 10.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, n. 6088, p. 533–536, 1986. Citado na página 8.

RUSSEL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 4th. ed. [S.l.]: Pearson, 2020. Citado na página 5.

SILVER, D. et al. Mastering the game of go with deep neural networks and tree search. *Nature*, v. 529, n. 7587, p. 484–489, 2016. Citado na página 12.

SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. 2nd. ed. Cambridge: MIT Press, 2018. Citado 2 vezes nas páginas 11 e 13.

TURING, A. M. Computing machinery and intelligence. *Mind*, v. 59, n. 236, p. 433–460, 1950. Citado na página 5.