

CENTRO UNIVERSITÁRIO FEI
MURILO MENDEL COSTA

**REDE NEURAL ARTIFICIAL UTILIZADA COMO ESTRUTURA DE SELEÇÃO DE
ALVO NO DOMÍNIO DO JOGO *ANGRY BIRDS***

São Bernardo do Campo

2019

MURILO MENDEL COSTA

**REDE NEURAL ARTIFICIAL UTILIZADA COMO ESTRUTURA DE SELEÇÃO DE
ALVO NO DOMÍNIO DO JOGO *ANGRY BIRDS***

Dissertação de Mestrado apresentada ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Mestre em Engenharia Elétrica. Orientado pelo Prof. Dr. Paulo Eduardo Santos.

São Bernardo do Campo

2019

Costa, Murilo Mendel.

Rede neural artificial utilizada como estrutura de seleção de alvo no domínio do jogo angry birds / Murilo Mendel Costa. São Bernardo do Campo, 2019.

74 f. : il.

Dissertação - Centro Universitário FEI.

Orientador: Prof. Dr. Paulo Eduardo Santos.

1. Rede Neural Artificial. 2. Agente Inteligente. 3. Angry Birds. I. Santos, Paulo Eduardo, orient. II. Título.

Aluno: Murilo Mendel Costa

Matrícula: 117117-2

Título do Trabalho: Rede neural artificial utilizada como estrutura de seleção de alvo no domínio do jogo angry birds.

Área de Concentração: Inteligência Artificial Aplicada à Automação e Robótica

Orientador: Prof. Dr. Paulo Eduardo Santos

Data da realização da defesa: 08/08/2019

ORIGINAL ASSINADA

Avaliação da Banca Examinadora:

São Bernardo do Campo, / / .

MEMBROS DA BANCA EXAMINADORA

Prof. Dr. Paulo Eduardo Santos

Ass.: _____

Prof. Dr. Danilo Hernani Perico

Ass.: _____

Profª. Drª. Anna Helena Reali Costa

Ass.: _____

A Banca Julgadora acima-assinada atribuiu ao aluno o seguinte resultado:

APROVADO

REPROVADO

VERSÃO FINAL DA DISSERTAÇÃO

**APROVO A VERSÃO FINAL DA DISSERTAÇÃO EM QUE
FORAM INCLUÍDAS AS RECOMENDAÇÕES DA BANCA
EXAMINADORA**

Aprovação do Coordenador do Programa de Pós-graduação

Prof. Dr. Carlos Eduardo Thomaz

A Deus pela graça da vida, aos meus pais pela educação, e à minha grande amiga e companheira Eloisa France Santos, pela paciência e incentivo imensurável até a conquista deste objetivo.

AGRADECIMENTOS

Aos professores do Centro Universitário FEI, que desde o início contribuíram para a construção de conceitos teóricos essenciais para desenvolvimento deste trabalho: Carlos Eduardo Thomaz, Flavio Tonidandel e Reinaldo da Costa Bianchi.

Aos colegas do curso de mestrado. Aos meus familiares e amigos pelo apoio.

Ao meu orientador, Paulo Eduardo Santos, pelo conhecimento compartilhado, paciência, incentivo e orientação.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

“Por vezes sentimos que aquilo que fazemos
não é senão uma gota de água no mar. Mas o
mar seria menor se lhe faltasse uma gota”
Madre Teresa de Calcutá

RESUMO

A inteligência artificial é um campo da ciência da computação dedicado ao desenvolvimento de sistemas capazes de reproduzir ou até mesmo superar as capacidades de raciocínio e solução de problemas dos seres humanos. Uma abordagem bastante explorada por essa comunidade científica é o uso de Redes Neurais Artificiais como instrumento de estudo de sistemas complexos, difíceis de serem modelados matematicamente, na tentativa de aprender a representação do conteúdo de um conjunto de dados de treinamento, utilizados para: classificação, clusterização e tomada de decisões. Este trabalho pretende desenvolver um agente inteligente capaz de operar de forma autônoma no domínio do jogo digital *Angry Birds*, através do desenvolvimento de uma Rede Neural Artificial utilizada como ferramenta de tomada de decisões no processo de seleção do alvo de um lançamento. Foi proposta a delimitação da região onde estão localizados os alvos de um lançamento em 98 zonas. O sistema treinado é capaz de identificar, a partir de uma cena do jogo, um valor numérico de pertinência para cada uma das zonas criadas se tornar o alvo de um lançamento, selecionando, por fim, a zona mais promissora em busca da maior pontuação possível.

Palavras-chave: Rede Neural Artificial. Agente Inteligente. Jogo *Angry Birds*.

ABSTRACT

Artificial intelligence is a computer science field dedicated to the development of systems capable to reproduce or even overcome human beings capabilities of thinking and problem-solving. A widely explored approach by this scientific community is the use of Artificial Neural Networks as a tool to study complex systems, difficult to be mathematically modeled, attempting to learn the feature of a set of training data, generally used for: classification, clustering and decision making. This work intends to develop an autonomously intelligent agent capable to operate in the Angry Birds digital game domain, through the development of an Artificial Neural Network used as a tool for decision making in the target selection process. It was proposed the target region delimitation in 98 launch zones. The trained system is able to identify, from a game scene, a numerical value of pertinence for each created zone to become the target of a launch, and finally select the most promising zone for the highest possible score.

Keywords: Artificial Neural Network. Intelligent Agent. Angry Birds Game.

LISTA DE ILUSTRAÇÕES

Figura 1 – Cenário do Jogo <i>Angry Birds</i>	14
Figura 2 – Estrutura do Ambiente de Desenvolvimento	17
Figura 3 – Sistema de coordenadas cartesianas e polares para lançamento dos pássaros	18
Figura 4 – Imagem segmentada pelo método MBR	20
Figura 5 – Imagem segmentada pelo método de contorno real	20
Figura 6 – Previsão de trajetória de lançamento em função da posição do pássaro no estilingue	23
Figura 7 – Modelo matemático de neurônio artificial	27
Figura 8 – Funções de ativação degrau e sigmoide	28
Figura 9 – Rede <i>Perceptron</i> Multicamadas	29
Figura 10 – Imagem original capturada pelo agente	38
Figura 11 – Imagem recortada, segmentada e redimensionada	38
Figura 12 – Zona ativa de presença de alvos	39
Figura 13 – Divisão da zona de lançamento em sub-regiões	40
Figura 14 – Estrutura da base de dados	41
Figura 15 – Inicialização do servidor	43
Figura 16 – Inicialização do agente	44
Figura 17 – Recorte da imagem original	45
Figura 18 – Segmentação da imagem recortada	46
Figura 19 – Redimensionamento da imagem segmentada	46
Figura 20 – Zonas inalcançáveis por lançamentos baixos	47
Figura 21 – Arquitetura do nível 17	50
Figura 22 – Inferência 1 no nível 17	52
Figura 23 – Inferência 2 no nível 17	53
Figura 24 – Desempenho dos três agentes avaliados no nível 17	55
Figura 25 – Arquitetura do nível 10	56
Figura 26 – Inferência 1 no nível 10	57
Figura 27 – Inferência 2 no nível 10	58
Figura 28 – Desempenho dos três agentes avaliados no nível 10	60
Figura 29 – Configuração do nível 13	63
Figura 30 – Configuração do nível 20	64

LISTA DE TABELAS

Tabela 1 – Resistência dos blocos ao impacto com pássaros	15
Tabela 2 – Resposta da rede à inferência apresentada na Figura 22	51
Tabela 3 – Comparação de desempenho dos três agentes no nível 17	54
Tabela 4 – Comparação de desempenho dos três agentes no nível 10	59
Tabela 5 – Desempenho dos agentes na conclusão na primeira parte do Episódio <i>Poached Eggs</i>	62

SUMÁRIO

1	INTRODUÇÃO	11
2	REVISÃO TEÓRICA	14
2.1	ANGRY BIRDS	14
2.2	AGENTE INTELIGENTE	16
2.2.1	Protocolo de comunicação Servidor/Cliente	17
2.2.2	Sistema de Visão	19
2.2.3	Planejamento de Trajetória	21
2.2.3	Agente Ingênuo	24
2.2.4	Agente Campeão da Competição de Angry Birds (2017/2018): <i>Eagle Wing</i>	24
2.3	REDES NEURAIS ARTIFICIAIS	27
2.3.1	Rede Perceptron Multicamadas	28
2.3.2	Algoritmo de aprendizado <i>Back Propagation</i>	30
2.3.3	Algoritmo ADADELTA	31
2.4	TRABALHOS CORRELATOS	34
3	DESENVOLVIMENTO	37
3.1	MODELAGEM DO PROBLEMA	37
3.1.1	Entrada	37
3.1.2	Topologia da Rede Neural Artificial (RNA)	39
3.1.3	Saída	39
3.1.4	Treinamento	40
3.2	FUNCIONAMENTO DO AGENTE	43
3.2.1	Captura da imagem de representação do cenário atual	44
3.2.2	Extração de informação e segmentação	45
3.2.3	Seleção do alvo	46
3.2.4	Conduta de lançamento	47
3.2.5	Ativação das habilidades específicas	48

3.2.6 Registro de lançamento	48
4 RESULTADOS	49
4.1 TREINAMENTO PRELIMINAR	50
4.1.1 Nível 17	50
4.1.2 Nível 10	55
4.2 TREINAMENTO FINAL	60
5 CONCLUSÃO	65
5.1 TRABALHOS FUTUROS	66
REFERÊNCIAS	68

1 INTRODUÇÃO

Jogos eletrônicos são considerados um ótimo ambiente de testes para muitos problemas de estudo nas áreas de Engenharia, Ciência da Computação e Inteligência Artificial, principalmente quando o desafio é a criação de agentes inteligentes autônomos em domínios específicos (LAIRD; LENT, 1999; LAIRD, 2002; KAMINKA et al., 2002).

O desenvolvimento de soluções no ambiente virtual possui vantagens em relação às realizadas no mundo real quando consideramos os custos de implementação, facilidade e segurança nos testes e validação dos trabalhos propostos. Além disso, apresentam complexidade de modelagem equivalente aos problemas de aplicação no mundo real, permitindo explorar conceitos correntes com o estado da arte das diversas áreas do conhecimento científico.

A maior motivação para desenvolvimento de agentes inteligentes autônomos é a tentativa de superar o desempenho dos seres humanos, desde tarefas simples e rotineiras às extremamente complexas e específicas, com o principal objetivo de entender, modelar e até mesmo exceder as capacidades de raciocínio, tomada de decisões, compreensão e previsão.

O *Angry Birds* é um jogo popular desenvolvido para dispositivos móveis em 2009 pela Rovio, cujo principal objetivo é lançar objetos com um estilingue em alvos específicos. O cenário possui estruturas de diferentes tipos de materiais que são organizadas para proteger esses alvos. Este cenário simula as leis mecânicas da física para o lançamento de projéteis sob ação da gravidade e transferência de energia proveniente da colisão entre dois corpos, (AIBIRDS, 2019a).

O domínio em questão é considerado complexo quando tenta-se submeter um agente inteligente para jogá-lo de forma autônoma, considerando a existência de diferentes tipos de projéteis, inúmeros alvos potenciais e consequências incertas para as relações entre eles, caracterizando um conjunto de ações extremamente grande aplicadas em um ambiente físico cujo conhecimento do domínio é incompleto.

As dificuldades desse problema envolvem: extração das características do cenário por um sistema de visão, representação do conhecimento extraído, tomada de decisões, estratégias para escolha dos alvos, técnicas de aprendizado de máquina, planejamento, buscas heurísticas e previsão de trajetória em função das grandezas físicas atuantes e intrínsecas ao cenário (YOON; KIM, 2015).

As características desse domínio despertaram interesse na comunidade científica no que diz respeito ao desenvolvimento de agentes autônomos capazes de jogar *Angry Birds*, pois

permite promover pesquisas nas áreas de inteligência artificial, ciência da computação e engenharias com baixo custo de implementação, além de ser considerado um jogo fácil de ser aprendido porém difícil de ser completamente dominado em virtude combinação de um sistema de comandos simples e níveis complexos (YOON; KIM, 2015).

Para medir o desempenho desses agentes, anualmente é realizada a competição de Inteligência Artificial de *Angry Birds* (*Angry Birds AI Competition*, 2019), organizada pela Universidade Nacional da Austrália desde o ano de 2012 e realizada durante a Conferência Internacional de Inteligência Artificial (IJCAI), (AIBIRDS, 2019a).

A competição submete os agentes a solucionar uma sequência de níveis inéditos do jogo, e os classificam em função da pontuação total obtida. Os quatro agentes mais promissores são colocados em um desafio com os melhores jogadores humanos com o propósito de avaliar se o objetivo da competição de superar o desempenho humano foi atingido, (AIBIRDS, 2019a).

Neste trabalho foi desenvolvido uma Rede Neural Artificial utilizada como estrutura de tomada de decisões de um agente no processo de seleção do alvo de um lançamento a partir de uma imagem de representação do cenário do jogo que antecede esta ação.

O agente foi desenvolvido a partir da estrutura do Agente Ingênuo disponibilizada pelos organizadores da competição, (GE et al., 2014).

O Agente Ingênuo (seção 2.2) é um agente padrão desenvolvido e disponibilizado pelos organizadores da competição internacional, utilizado como ponto de partida pelos interessados em desenvolver soluções no cenário do *Angry Birds* (seção 2.1), executado para este fim em uma versão *offline* do jogo rodado no navegador do *Google Chrome*.

O agente é dotado de um sistema de visão capaz de extrair o tipo e posicionamento de todos os objetos que compõem o cenário do jogo. A partir das informações extraídas, o agente padrão escolhe um alvo e calcula os parâmetros de lançamento para o atingir. Todos os comandos do agente no jogo são codificados em mensagens, enviadas para um servidor *proxy* que as executa através de comandos em *javascript* no navegador.

As Redes Neurais Artificiais (seção 2.3) são geralmente utilizadas em problemas cuja modelagem matemática para o reconhecimento de padrões é de alta complexidade, geralmente não lineares, ou que apresentem dados ruidosos ou incompletos para a representação do problema, (BEZDEK, 1992)

A construção desse agente envolveu:

- Definição das características dos dados de entrada/saída e topologia da rede;
- Coleta de dados experimentais suficientes para compor uma base de dados de treinamento;

- Desenvolvimento de interface de inferência do agente na rede (*Java/Python*);
- Implementação, treinamento e teste de funcionamento da rede (*Python*);

Além de avaliar o desempenho do agente desenvolvido nos moldes da competição em comparação aos agentes existentes, o trabalho também teve como objetivo secundário avaliar a capacidade de representação da Rede Neural desenvolvida no domínio estudado, por se tratar de uma nova abordagem para a literatura.

Os resultados mostram que a estrutura de aprendizado desenvolvida foi capaz de aprender a representação entre o conteúdo do cenário do jogo e as ações mais promissoras para cada configuração possível desse cenário. A estrutura de aprendizado foi treinada a partir de 36 mil dados de treinamento coletados em 9 níveis do jogo, e testada nos 21 primeiros níveis do episódio *Poached Eggs*, conferindo desempenho equivalente ao agente padrão disponibilizado e ao atual campeão da competição.

A equivalência quanto ao desempenho dos agentes conferindo a eficácia do sistema desenvolvido está relacionada à capacidade de solução de 90,5% dos níveis em que foi submetido, à pontuação geral obtida pelo agente proposto nos testes realizados ser muito próxima dos agentes comparados e às condutas apresentadas pelo processo de inferência da rede treinada diante das inúmeras situações apresentadas.

As seções a seguir são organizadas da seguinte maneira: o capítulo 2 desenvolve os conceitos teóricos pertinentes ao trabalho e contextualiza o estado da arte para a utilização de jogos no cenário científico e desenvolvimento de agentes atuantes no domínio do jogo *Angry Birds*.

No capítulo 3 são descritos a arquitetura da Rede Neural Artificial proposta, assim como a forma com que a mesma foi incorporada ao agente inteligente.

Os resultados são discutidos e analisados no capítulo 4, e finalmente, no capítulo 5 são apresentadas as conclusões do trabalho e potenciais trabalhos futuros.

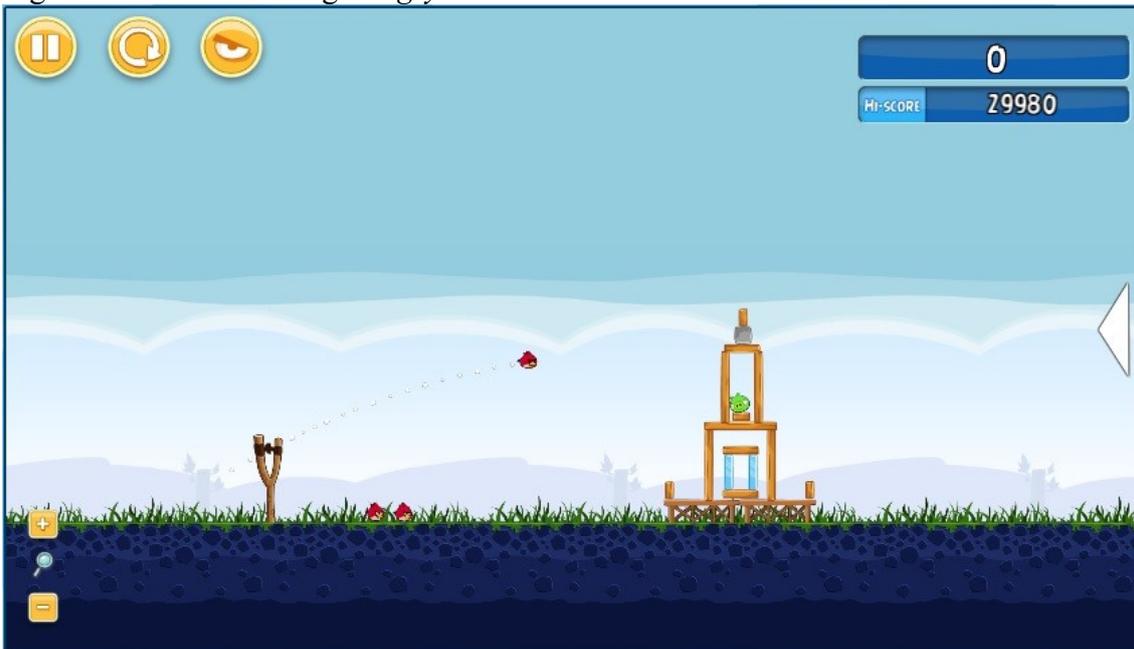
2 REVISÃO TEÓRICA

Este capítulo apresenta o domínio do jogo *Angry Birds* e a arquitetura do agente inteligente onde será aplicado o método de aprendizado supervisionado desenvolvido neste trabalho. Além disso, serão apresentados os conceitos essenciais utilizados na construção da Rede Neural Artificial que compõe a estrutura de tomada de decisões proposta.

2.1 ANGRY BIRDS

O objetivo do jogo *Angry Birds* é a destruição alvos, protegidos e rodeados por edificações, através do lançamento de projéteis. No domínio do jogo, os projéteis são pássaros, lançados por um estilingue a fim de atingir porcos como alvo, (AIBIRDS, 2019a). Os edifícios são formados por pequenos blocos geométricos apoiados verticalmente, conforme ilustrado na Figura 1.

Figura 1 – Cenário do Jogo *Angry Birds*.



Fonte: Autor.

Os blocos que compõem o cenário podem assumir quatro formas geométricas: retângulos, quadrados, círculos e triângulos e possuir três tipos de composição material: madeira, gelo e concreto. A composição material dos blocos indica sua resistência aos impactos das colisões com o solo e com os pássaros.

Existem cinco tipos de pássaros disponíveis na versão clássica do jogo para lançamento, classificados nas cores: vermelho, amarelo, azul, branco e preto. Além da distinção em cores, cada um desses cinco pássaros possui vantagens ou desvantagens para destruir determinado

tipo de material e uma habilidade especial que poderá ser ativada durante sua trajetória de lançamento.

- a) Pássaro Vermelho: O pássaro vermelho é o único que não possui habilidade de ativação secundária;
- b) Pássaro Amarelo: A habilidade secundária desse pássaro diminui o efeito da aceleração da gravidade sobre ele e aumenta de sua velocidade. Essa característica facilita a destruição de blocos de madeira.
- c) Pássaro Azul: Quando ativado a habilidade secundária, o pássaro azul se divide em outro três pássaros menores, facilitando a destruição de blocos de gelo.
- d) Pássaro Branco: O pássaro branco solta um ovo em queda livre vertical que explode ao atingir qualquer alvo ou o solo. Ao soltar o ovo, o pássaro muda de trajetória sendo lançado para cima em alta velocidade.
- e) Pássaro Preto: O pássaro preto explode quando ativada a habilidade secundária ou quando atingir alguma estrutura, utilizado geralmente para destruir as estruturas de concreto, por serem mais resistentes.

A resistência dos blocos ao impacto de pássaros é um fator bastante relevante quando deseja-se selecionar um, visto que alguns pássaros destroem com mais facilidade certos tipos de estruturas. A Tabela 1 classifica a resistência de cada tipo de bloco quando atingido por um pássaro específico.

Tabela 1 – Resistência dos blocos ao impacto com pássaros.

Tipo de Pássaro	Estrutura de Madeira	Estrutura de Gelo	Estrutura de Concreto
Vermelho	Baixa	Média	Alta
Amarelo	Baixa	Baixa	Alta
Azul	Média	Baixa	Alta
Branco	Média	Baixa	Alta
Preto	Média	Baixa	Média

Fonte: Autor “adaptado de” Dasgupta et al., 2016.

O jogo é composto por inúmeros níveis, agrupados atualmente em 12 episódios (2018), porém, o cenário de desenvolvimento deste trabalho será na primeira parte do Episódio denominado *Poached Eggs*, formado por 21 níveis de complexidade variada.

A conclusão de um nível é atingida ao destruir todos os porcos do cenário, liberando assim o nível seguinte para ser jogado.

A competitividade do jogo, apesar do objetivo simples de eliminação de porcos, é garantida graças ao sistema de pontuação para avaliação do desempenho do jogador. A ideia do sistema de pontuação proposto pelo jogo é estimular o jogador a concluir o nível, utilizando o menor número de pássaros possível, causando a maior destruição possível no cenário além do objetivo principal de eliminação de todos os porcos. Sendo assim, será atribuído ao jogador 5 mil pontos para cada porco eliminado, 10 mil pontos para cada pássaro remanescente após a conclusão do nível e uma pontuação proporcional à destruição das estruturas que compõem o cenário.

Alguns níveis possuem elementos explosivos distribuídos pelo cenário, denominados TNTs. Ao serem atingidos, uma explosão é gerada causando grande destruição aos objetos na proximidade o tornando uma boa escolha de alvo para um lançamento como contribuição para conclusão do objetivo e maximização da pontuação.

A partir da definição do domínio do jogo apresentada, a seção 2.2 irá apresentar a estrutura utilizada para o desenvolvimento de agentes inteligentes capazes de jogar o *Angry Birds* de forma autônoma, especificando a arquitetura de comunicação entre o agente e o jogo, processo aquisição de informações visuais, tomada de decisões e previsão e planejamento de trajetórias.

2.2 AGENTE INTELIGENTE

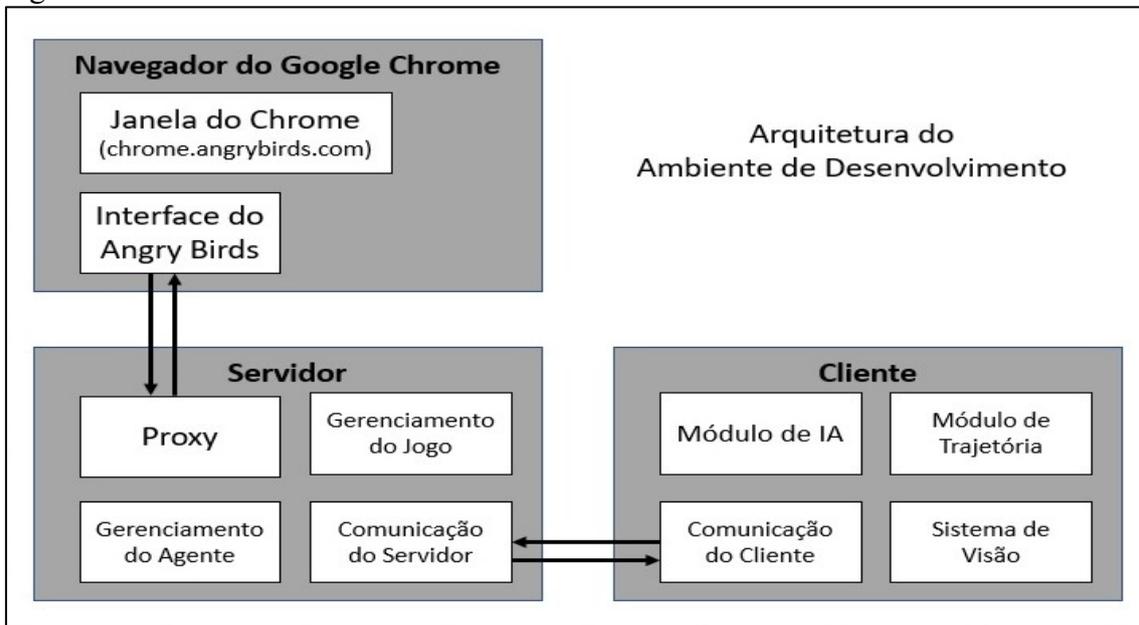
O objeto de estudo do presente trabalho é o desenvolvimento de um agente inteligente capaz de jogar de forma autônoma o jogo *Angry Birds*. Ao longo do tempo, inúmeras competições foram idealizadas com o objetivo de explorar o domínio de desenvolvimento de agentes autônomos tanto em aplicações reais como a *RoboCup*, (MACKWORTH, 1993), como no domínio dos jogos eletrônicos de tiro em primeira pessoa (HINGSTON, 2009), simulador 2D de corrida de carros (LOIACONO et al., 2009), Super Mario (TOGELIUS et al., 2013), e jogos de estratégia em tempo real (ONTANON; SYNNAEVE, 2013).

Os agentes inteligentes no domínio do jogo *Angry Birds* protagonizam anualmente a competição internacional realizada durante o IJCAI. A competição foi idealizada em 2012 pela Universidade Nacional da Austrália e possui edições anuais para avaliar os avanços científicos inerentes ao desenvolvimento de agente inteligentes em vista de superar o desempenho humano no jogo, (AIBIRDS, 2019a).

Os organizadores da competição desenvolveram uma estrutura base para um agente inteligente, (GE et al., 2014)., disponibilizada aos interessados no assunto como ponto de partida. A estrutura desenvolvida será descrita nesta seção.

O agente é dividido em dois módulos, um servidor e um cliente, e foi construído para jogar na versão do *Angry Birds* rodada no navegador do *Google Chrome*, conforme representado pela Figura 2.

Figura 2 – Estrutura do Ambiente de Desenvolvimento



Fonte: Autor adaptado de “*Angry Birds Basic Playing Software v1.32*”, (GE et al., 2014).

A estrutura desenvolvida permite ao agente realizar ações no navegador através de mensagens transmitidas pelo servidor para um *plugin* em javascript. As ações executadas são essencialmente para suprir a impossibilidade de utilização do mouse e teclado pelo agente, ou seja, servem para navegar na tela do jogo, clicar, utilizar a barra de rolagem e realizar captura de imagem.

Os comandos executados pelo servidor através do módulo *proxy* são realizados através das mensagens:

- CLICK: Clicar com o botão esquerdo do mouse;
- DRAG: Arrastar o cursor do mouse de um lugar para outro;
- MOUSEWHELL: Girar a barra de rolagem do mouse;
- SCREENSHOT: Capturar da tela atual do jogo.

2.2.1 Protocolo de comunicação Servidor/Cliente

O servidor interage com o jogo através de um plugin específico instalado no Google Chrome, dessa forma, o agente é capaz de enviar mensagens de três tipos diferentes para o servidor: ações a serem realizadas, consultas de estado do jogo e mensagens de configuração.

As mensagens recebidas do cliente deverão sempre possuir um prefixo e um corpo, que equivalem respectivamente ao ID da mensagem (MID) e os parâmetros que compõem a função solicitada.

A mensagem de configuração é utilizada durante a realização da competição logo que o agente é conectado ao servidor, contendo o respectivo ID do agente para verificação de sua validade e status na competição.

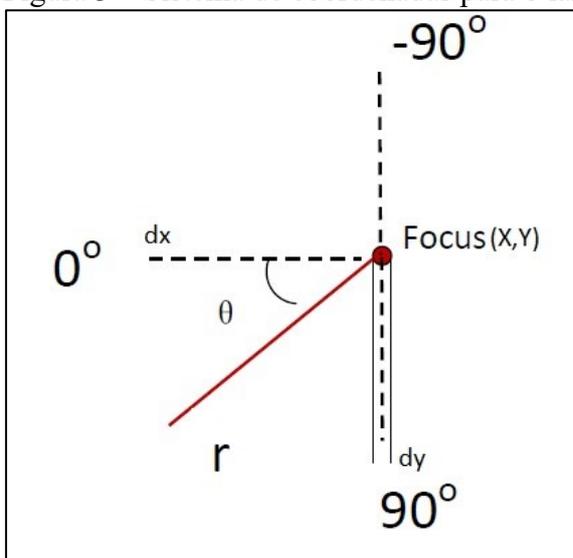
As mensagens de ações do agente para o servidor são utilizadas para realização de lançamentos e ajuste da distância da câmera do cenário.

O agente pode realizar os lançamentos de pássaros de duas formas diferentes: modo seguro e modo rápido. No modo seguro, o agente irá aguardar 15 segundos após o lançamento do pássaro aguardando a tela de nível concluído, e em seguida irá armazenar a pontuação. Já no modo rápido, a mensagem de resposta de confirmação de lançamento é o suficiente para permitir que o próximo lançamento seja realizado.

Para realizar um lançamento, o agente deve formular uma mensagem ao servidor contendo as coordenadas espaciais do estilingue na tela, as coordenadas de liberação do pássaro, um tempo de liberação do pássaro que será lançado e o tempo entre o lançamento e ativação da habilidade secundária do pássaro.

As coordenadas para o lançamento do pássaro são determinadas por um sistema de previsão de trajetória em função do alvo selecionado (seção 2.2.3) e podem ser definidas em coordenadas polares ou retangulares referenciadas pelo posicionamento do estilingue, conforme ilustrado pela Figura 3.

Figura 3 – Sistema de coordenadas para o lançamento dos pássaros.



Fonte: Autor adaptado “Server/Client Protocol”, Ge, X. et al., 2016.

A Figura 3 apresenta um plano cartesiano para a representação do mecanismo de lançamento dos pássaros através do estilingue, sendo $Focus(x, y)$ representa a origem do sistema de coordenadas de lançamento e diz respeito a posição inicial do pássaro no estilingue, dx e dy são as coordenadas retangulares para um potencial lançamento em relação ao eixo das abcissas e ordenadas respectivamente e os termos r e θ são as coordenadas polares de um lançamento representando o módulo do tensionamento do estilingue e o ângulo de lançamento respectivamente.

Além das mensagens de lançamento de pássaros, o agente pode solicitar ao servidor ações de ajuste focal do cenário, através de comandos de aproximação e distanciamento da perspectiva de visualização do cenário.

A função de distanciamento “*ZoomOut*” é executada sempre que um novo nível é carregado, solicitando ao servidor o distanciamento da perspectiva de visualização da tela. Os comandos de aproximação da perspectiva do jogador e de navegação lateral pelo cenário também são possíveis, apesar de serem utilizados com menor frequência.

O último tipo de mensagem disponível em um agente é a realização de consultas, informando sobre: o estado atual do jogo, melhores pontuações, as pontuações do agente em questão em todos os níveis, o nível atual, e a representação atual da tela.

2.2.2 Sistema de Visão

O sistema de visão é composto por dois componentes de segmentação de imagens distintos. Um desses componentes processa e segmenta a imagem identificando os principais objetos do cenário pelos menores retângulos que os contém, conhecido como *Minimum Bounding Rectangles* (MBR). Os objetos identificados por esse método incluem o estilingue, pássaros vermelhos, amarelos, azuis, pretos e brancos, porcos, blocos de gelo, madeira e pedra, TNTs e espectro de trajetórias anteriores.

A Figura 4 ilustra o método identificação de objetos no cenário pelo método do MBR.

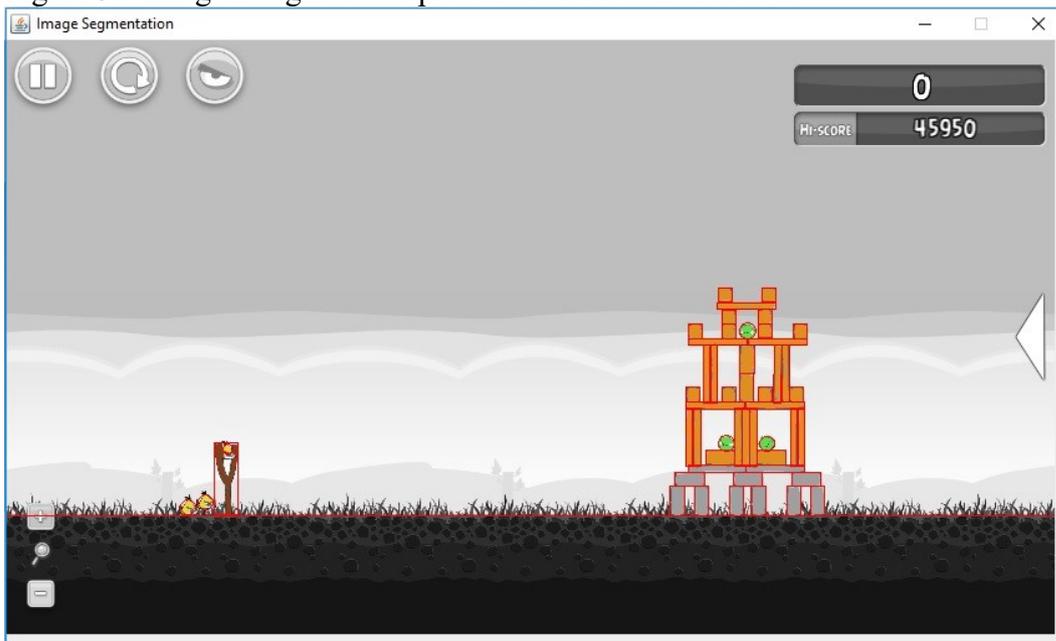
Figura 4 – Imagem segmentada pelo método MBR.



Fonte: Autor.

O outro componente de segmentação de imagens identifica a posição exata dos objetos descritos acima, permitindo a distinção e identificação da superfície das estruturas fixas do cenário, conforme representado na Figura 5. A segmentação real dos objetos demanda de 300ms à 500ms enquanto o método do MBR leva apenas 100ms para concluir o processamento.

Figura 5 – Imagem segmentada pelo método de contorno real.



Fonte: Autor.

Para cada elemento identificado através da análise da imagem capturada, o módulo de visão irá criar uma instância da classe objeto denominada `ABObject.java`, que possui quatro atributos principais: coordenada x e y da diagonal superior esquerda, comprimento e largura.

Cada objeto é identificado também por um tipo específico: morro, estilingue, pássaros vermelho, amarelo, azul, preto, branco, porco, gelo, madeira, pedra e TNT. Os objetos que são vazados no meio possuem identificação específica quando instanciados. Esses atributos são inerentes à superclasse de `ABObject.java` denominada `java.awt.Rectangle`.

A classe `ABObject.java` possui três subclasses: `Circ`, `Rect` e `Poly`.

- a) `Circ`: Representa uma estrutura de formato circular com atributos definindo as coordenadas do centro e raio.
- b) `Rect`: Essa subclasse introduz três novos parâmetros para definir um retângulo, com três atributos específicos para identificar o comprimento, a largura e o ângulo do retângulo em relação ao solo, com variação entre 0° e 180° .
- c) `Poly`: Utilizada para representar um polígono, por exemplo, um triângulo.

Sabendo a posição espacial de todos os elementos do cenário, o agente é capaz de escolher o alvo que deseja atingir. O grande desafio nesse momento para o jogador é estimar as coordenadas de lançamento do pássaro para que o alvo escolhido seja atingido. A seção a seguir irá tratar dos conceitos relevantes para a definição da abordagem de previsão de trajetória de lançamento para o jogo.

2.2.3 Planejamento de Trajetória

O estudo da trajetória descrita pelos pássaros ao serem lançados depende basicamente de sua velocidade inicial em função da tensão no elástico do estilingue e de seu ângulo de lançamento, ou seja, o módulo de planejamento de trajetória estima a trajetória realizada por um pássaro em função das coordenadas cartesianas que o pássaro é liberado em relação à posição do estilingue.

A velocidade inicial do lançamento pode ser decomposta em duas dimensões e é proporcional ao tensionamento do elástico do estilingue.

$$\vec{v}_0 = (v_{0x}, v_{0y}) \quad (1)$$

Sendo:

- a) v_{0x} : a componente horizontal da velocidade inicial;
- b) v_{0y} : a componente horizontal da velocidade inicial;

A componente horizontal da velocidade é descrita por um movimento retilíneo uniforme visto a ausência da ação gravitacional e resistência do ar, logo, pode-se definir o movimento do pássaro em função do tempo no eixo das abcissas por:

$$x_{final} = (v_{0x} \cos \theta)t + x_{inicial} \quad (2)$$

Sendo:

- a) x_{final} : a posição final do pássaro no eixo x no instante de tempo t ;
- b) $x_{inicial}$: a posição inicial do movimento do pássaro no eixo x ;
- c) θ : o ângulo de lançamento do pássaro no estilingue.

A componente vertical da velocidade do pássaro é descrita por um movimento uniformemente variado devido à ação da aceleração da gravidade sobre o pássaro lançado, permitindo descrever a posição vertical do pássaro em função do tempo da seguinte forma:

$$y_{final} = (v_{0y} \sin \theta)t - \frac{1}{2}gt^2 + y_{inicial} \quad (3)$$

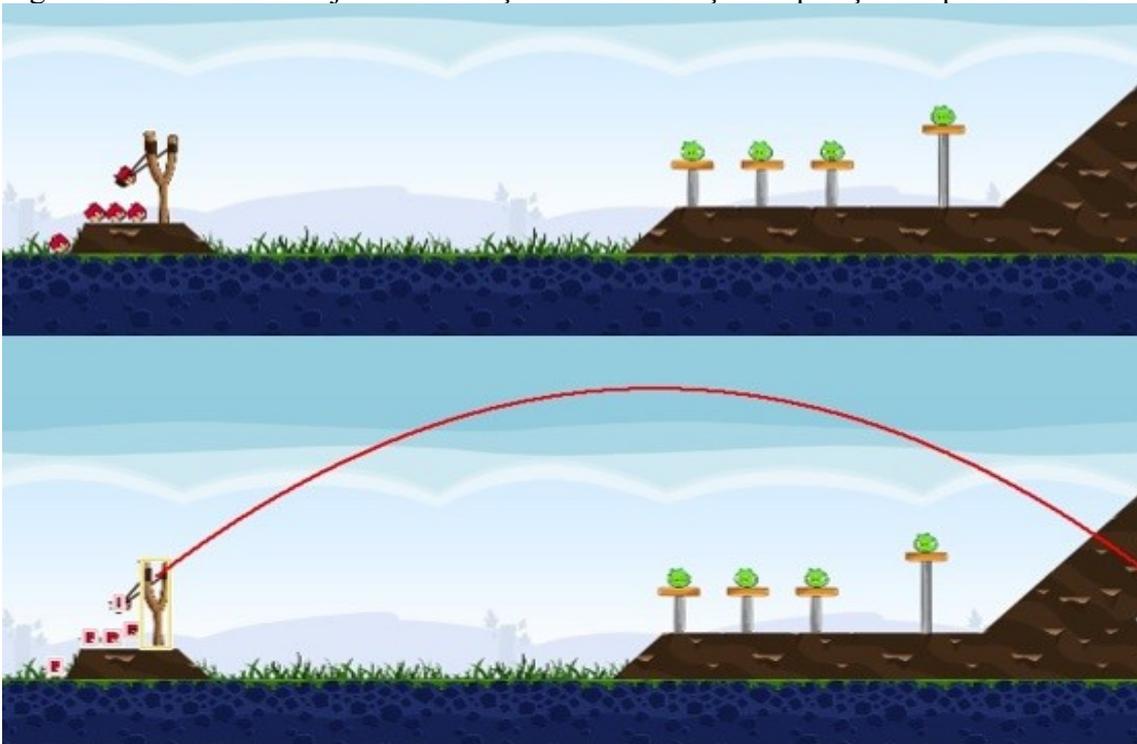
Sendo:

- a) y_{final} : a altura atingida pelo pássaro lançado no instante de tempo t ;
- b) g : a constante de aceleração da gravidade;
- c) $y_{inicial}$: a posição inicial do movimento do pássaro no eixo y .

O agente é dotado de uma função capaz de determinar uma lista de pares de coordenadas (x, y) que representam as posições percorridas pelo pássaro em cada instante de sua trajetória parabólica a partir das coordenadas do ponto de liberação do pássaro no estilingue.

É possível visualizar essa previsão de trajetória pelo módulo de visão quando solicitado pelo comando de inicialização do agente conforme representado na Figura 6.

Figura 6 – Previsão de trajetória de lançamento em função da posição do pássaro no estilingue.



Fonte: Autor.

Combinando as equações (2) e (3), podemos descobrir ângulo de lançamento apropriado para atingir um alvo do cenário a partir das coordenadas de posicionamento de um alvo e do módulo da velocidade do pássaro que será lançado.

$$\theta = \arctan\left(\frac{v^2 \pm \sqrt{v^4 - g(gx^2 + 2yv^2)}}{gx}\right) \quad (4)$$

Sendo:

- a) v : a velocidade inicial do pássaro, representada pela sua posição relativa no estilingue antes do lançamento;
- b) x : coordenada normalizada do alvo no eixo x em relação ao estilingue;
- c) y : coordenada normalizada do alvo no eixo y em relação ao estilingue;

O módulo de planejamento de trajetória é uma ferramenta aproximada para definir a configuração do lançamento de um pássaro dado o alvo que se deseja atingir. Alguns parâmetros como ruídos na extração do posicionamento dos alvos, diferença entre o ângulo desejado e o ângulo real do disparo e as variações do valor da velocidade inicial do pássaro para diferentes níveis do jogo justificam a imprecisão do método descrito.

2.2.3 Agente Ingênuo

O Agente Ingênuo é uma criação dos idealizadores da competição como ponto de partida para os interessados no desenvolvimento de trabalhos neste domínio. A construção deste agente inclui os módulos essenciais de visão, planejamento de trajetória e interface com o Chrome conforme descritos ao longo deste capítulo.

O agente é denominado ingênuo pelo fato de utilizar como estratégia de seleção de alvo a escolha aleatória de um dos porcos vivos independentemente dos obstáculos que o cercam.

Assim que o alvo do lançamento é selecionado, o agente realiza o cálculo de planejamento de trajetória para o lançamento. O agente sempre utilizará um lançamento alto para na primeira jogada, e nas jogadas seguintes seleciona com probabilidade 0,167 lançamentos altos e 0,833 lançamentos baixos.

As configurações para seleção das habilidades específicas de cada pássaro estão configuradas da seguinte forma:

- Pássaros Amarelos: Escolha aleatória entre 65% e 90% da trajetória;
- Pássaros Azuis: Escolha aleatória entre 50% e 70% da trajetória;
- Pássaros Pretos: 100% da trajetória;
- Pássaros Brancos: Escolha aleatória entre 65% e 85% da trajetória.

A estratégia de seleção de níveis do Agente Ingênuo consiste em jogar um nível por 3 vezes na tentativa de concluí-lo, prosseguindo para o próximo nível caso a conclusão não seja alcançada.

2.2.4 Agente Campeão da Competição de Angry Birds (2017/2018): *Eagle Wing*

O agente *Eagle Wing* foi o vencedor da competição nas edições de 2017 e 2018, desenvolvido por Wang (2017), construído a partir do código disponibilizado pelo Time *DataLab* do agente campeão da competição em 2014, *DataLab Birds*.

A proposta do atual campeão foi manter a ideia de análise estrutural e decisões de múltiplas estratégias. O agente possui um sistema de avaliação de utilidade para cada estratégia em função da posição espacial dos alvos potenciais do lançamento e os objetos em sua redondeza, *e.g.* a estratégia de atingir os TNTs levará em consideração a análise da quantidade de objetos, estruturas e porcos por perto.

O agente desenvolvido possui 5 abordagens estratégicas diferentes para tentar concluir um nível:

- a) Mirar diretamente nos porcos: Baseado na estratégia padrão do Agente Ingênuo, essa estratégia tenta encontrar uma trajetória para atingir um porco desprotegido, ou um aglomerado de porcos no cenário. A abordagem de mirar diretamente nos porcos pode ser uma boa estratégia inicial para cenários mais simples com poucas estruturas obstruentes ou até como último tiro para eliminar porcos isolados e remanescentes.
- b) Destruir os TNTs: Quando há estruturas explosivas no cenário, atingi-las é geralmente uma boa estratégia visto o potencial destrutivo para o cenário e consequentemente eliminação de porcos
- c) Destruir o maior número de blocos possíveis: Em alguns níveis, geralmente compostos por muitos blocos empilhados e poucos pássaros para serem lançados, pontuar destruindo as estruturas pode ser considerada a melhor estratégia, visto que concluir o nível com pássaros remanescentes é improvável.
- d) Destruir objetos próximos de estruturas circulares posicionadas no topo do cenário, na tentativa de destruir o cenário e eliminar porcos a partir de sua queda.
- e) Estratégia estrutural: Essa estratégia avalia todos os blocos do cenário identificados pelo sistema de visão e escolhe o melhor bloco possível de ser acertado com maior potencial de destruição, ou seja, blocos essenciais para a estabilidade da estrutura que apresentem alto potencial de destruição pelo pássaro que será lançado.

Muitos lançamentos de teste foram realizados para definir as cinco estratégias descritas acima e para otimizar o tempo de ativação da habilidade secundária dos pássaros. Através do método de aprendizado supervisionado de aumento de gradiente, e outras milhares de jogadas de treinamento, foi definida uma árvore de decisão para cada estratégia. Em análise às árvores de decisão, dado uma situação específica, cada estratégia pode ser classificada como:

- a) Ser escolhida com certeza, atribuindo utilidade máxima para a estratégia;
- b) Não ser escolhida com certeza, atribuindo utilidade mínima para a estratégia;
- c) Calcular um valor utilidade em função de características específicas do cenário.

Caso algum objeto obstrua a trajetória do alvo selecionado, o agente seleciona um conjunto de blocos para se tornar alvo do lançamento. Além disso, caso o agente consiga selecionar a melhor estratégia, um lançamento é realizado selecionando um bloco aleatório identificado no cenário.

Os assuntos tratados nesta seção descreveram os módulos utilizados pelos agentes existentes para interagir com o domínio do jogo, assim como as estratégias específicas do Agente Ingênuo e do atual campeão da competição. Esses dois agentes serão utilizados como referência para avaliação de desempenho da solução proposta neste trabalho.

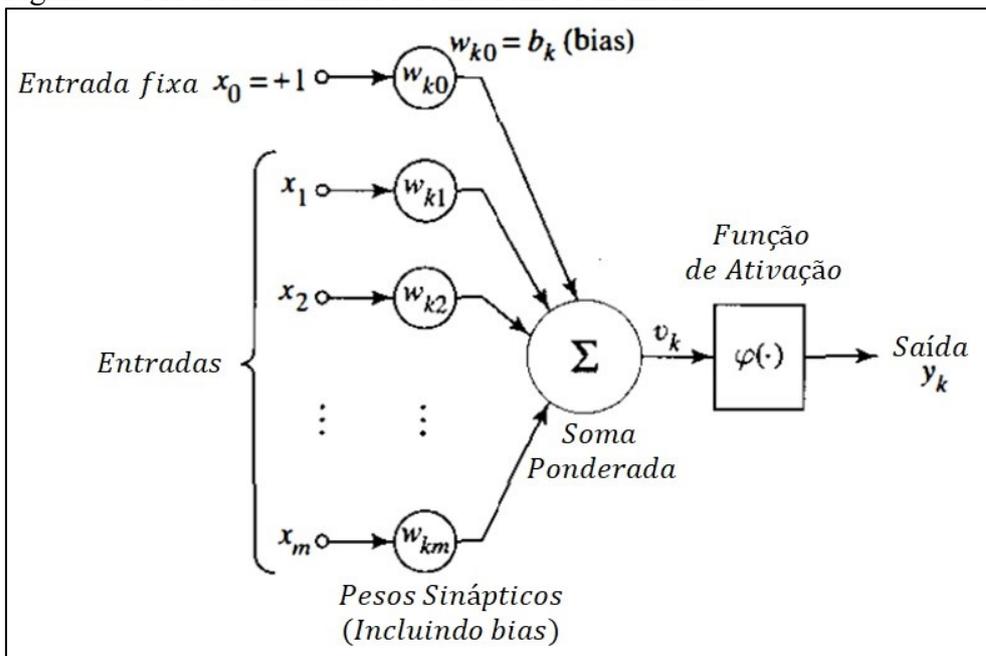
A seção seguinte irá desenvolver os conceitos teóricos sobre Redes Neurais Artificiais, utilizada neste trabalho como estrutura de tomada de decisões para a seleção do alvo de lançamento.

2.3 REDES NEURAIS ARTIFICIAIS

Redes Neurais Artificiais (RNA) são modelos computacionais inspirados na estrutura neurológica do cérebro humano com as capacidades de adquirir, armazenar e utilizar conhecimento experimental. São empregadas geralmente em problemas de reconhecimento de padrões quando os dados são ruidosos ou em sistemas complexos de difícil modelagem das regras gerais de funcionamento, (HAYKIN, 1998).

Na tentativa de entender a forma com que o cérebro humano armazena e utiliza informações, e como essas influenciam nossa percepção e comportamento, Rosenblatt (1957) propõe um modelo probabilístico para armazenamento de informações denominado *perceptron*, composto por uma estrutura de processamento capaz de apresentar respostas à um conjunto de estímulos a ela apresentados. Esse modelo está representado na Figura 7.

Figura 7 – Modelo matemático de neurônio o artificial.



Fonte: Adaptado de S. Haykin. Neural Networks: A Comprehensive Foundation, Prentice Hall, second edition, 1999.

Sendo:

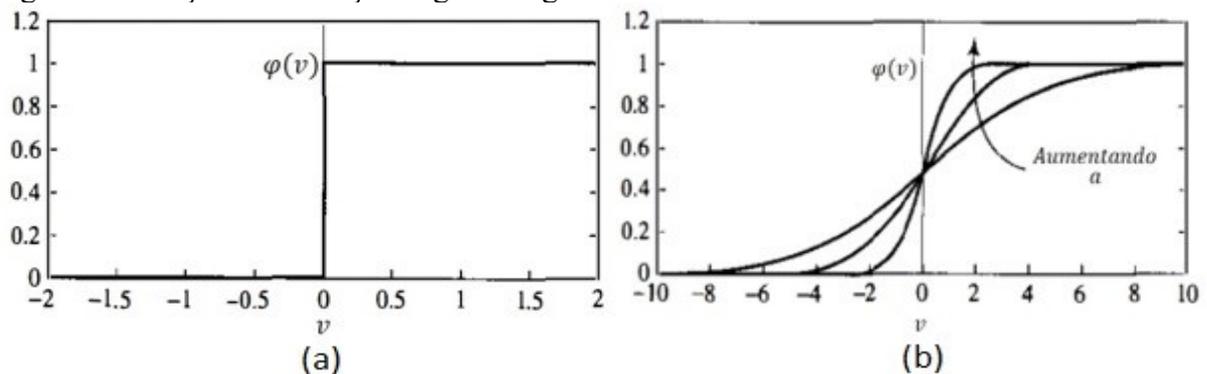
- a) X_m : o vetor de entradas;
- b) W_{km} : o vetor de pesos sinápticos;
- c) v_k : Soma ponderada das entradas;
- d) $\varphi(\cdot)$: Função de ativação;
- e) $y_k = Saída = \varphi(v_k) = f(W_{km}^t \cdot X_m)$

As entradas são submetidas à influência de pesos sinápticos antes de serem processadas. Essas conexões sinápticas são fundamentais no processo de mapeamento de entradas em suas respectivas saídas, garantindo às redes neurais a capacidade de representar modelos complexos de difíceis de serem representados matematicamente.

As entradas ponderadas são somadas, e aplicadas à uma função de ativação com o objetivo normalizar o dado de saída que será propagado. As funções degrau e sigmoide são duas das mais utilizadas em aplicações envolvendo redes neurais (Figura 8).

A função degrau, representada pela Figura 8a possui característica classificatória, utilizada inicialmente por McCulloch e Pitts (1943) quando se acreditava que o sistema nervoso biológico possuía características de decisão binária em função de um limiar v , enquanto a função de ativação sigmoide, representada pela Figura 8b permite utilizar redes neurais para modelos não-lineares.

Figura 8 – Funções de ativação degrau e sigmoide.



Fonte: Adaptado de S. Haykin. Neural Networks: A Comprehensive Foundation, Prentice Hall, second edition, 1999.

Os neurônios artificiais definidos anteriormente podem ser organizados em redes multicamadas com fluxo de sinais unidirecional. Essa topologia é utilizada no processo de cálculo da saída da rede para determinado padrão de entrada, denominado *recall* e será apresentado na próxima seção.

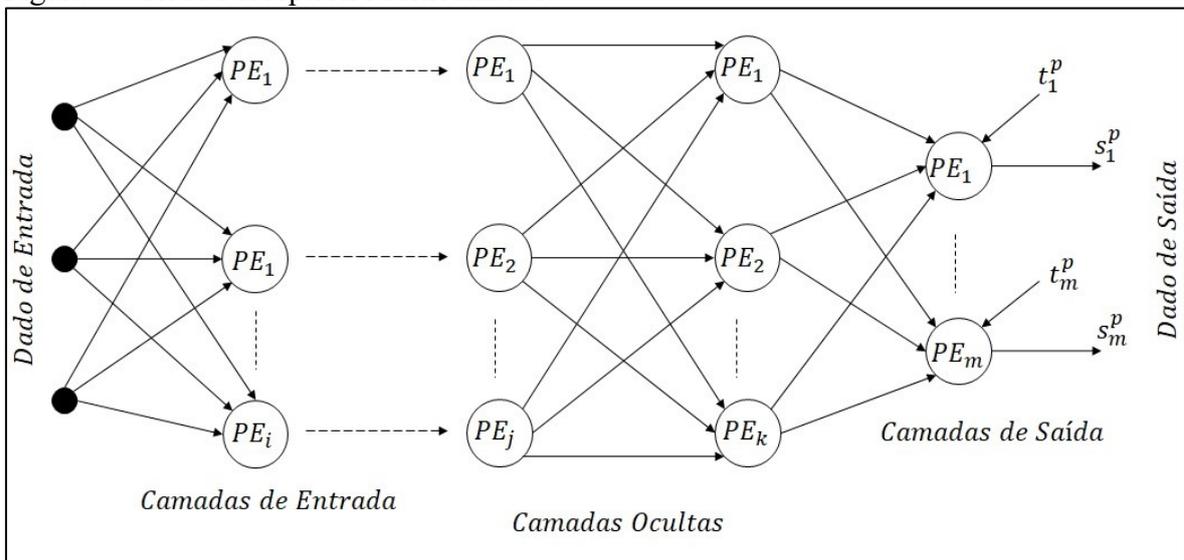
2.3.1 Rede Perceptron Multicamadas

A ideia de modelar um sistema computacional equivalente ao sistema nervoso biológico teve como ponto de partida a concepção do neurônio artificial, denominado Perceptron, representado pela Figura 7.

A Rede Perceptron de Multicamadas, do inglês: *Multilayer Perceptron* (MLP) é uma associação de neurônios artificiais, dispostos em camadas e interligados, com fluxo de propagação de sinal unidirecional utilizada geralmente em problemas cuja classificação não é

linearmente separável (HOPFIELD, 1982). Um padrão é aplicado nos neurônios de entrada da rede, processado e propagado por todas as camadas ocultas até que seja encontrado uma resposta equivalente nos neurônios da camada de saída. A topologia de uma MLP está ilustrada na Figura 9.

Figura 9 – Rede Perceptron Multicamadas.



Fonte: Autor.

Essa topologia de rede surgiu após a conclusão de Minsk e Papert (1969) de que as redes *Perceptron* de uma única camada possuem restrição de classificação exclusiva a problemas linearmente separáveis, demonstrado pela tentativa de modelagem da função XOR.

O processamento e propagação das informações de uma rede neural é baseada na regra de aprendizado de Hebb (1949), se dois neurônios entre si são ativados simultaneamente de forma síncrona então a força da sinapse entre eles deve ser seletivamente aumentada, e por outro lado, se dois neurônios conectados entre si são ativados assincronamente então aquela sinapse deve ser seletivamente enfraquecida ou cancelada.

O processo seletivo de mudança no peso das sinapses nos remete ao processo de aprendizado das redes neurais, que consiste na atualização de seus pesos sinápticos a partir da aplicação de dados de treinamento em uma MLP, análise do erro da saída encontrada em relação à saída esperada. Esse processo de aprendizado é denominado *back propagation*, e será apresentado a seguir.

2.3.2 Algoritmo de aprendizado *Back Propagation*

O algoritmo *back propagation* é um caso especial do método de diferenciação automática que consiste em determinar o gradiente da função erro para atualização e ajuste dos pesos da rede, ou seja, em um problema de aprendizado supervisionado, o erro na saída da rede é retro propagado para todas as camadas anteriores, corrigindo proporcionalmente todas as sinapses neurais.

O processo de aprendizado é feito em duas etapas. Em um primeiro momento, um conjunto de dados p são aplicados à entrada da rede, propagada até a obtenção de uma resposta nos m processadores da camada de saída. A saída obtida é comparada com a saída esperada caracterizando uma variável de erro, que será utilizada para atualização dos pesos sinápticos da rede através do método do gradiente descendente. A Equação 5 apresenta o cálculo do erro na camada de saída da rede a partir da comparação entre o padrão obtido e o padrão esperado.

$$E = \frac{1}{2} \sum_p \sum_m (t_m^p - s_m^p)^2 \quad (5)$$

sendo:

- a) t_m^p o valor de saída esperado da aplicação do padrão p para os m processador da camada de saída;
- b) s_m^p o valor de saída obtido da aplicação do padrão p para os m processadores da camada de saída.

O ajuste dos pesos é realizado na direção contrária do gradiente. Cada peso sináptico i de cada processador da camada de saída j é atualizado em função da derivada parcial do erro encontrado E_p em relação à cada peso w_{ji} .

$$\Delta w_{ji} = -\eta \frac{\partial E_p}{\partial w_{ji}} \quad (6)$$

sendo η a taxa de aprendizado da rede.

O desenvolvimento da equação de variação do erro nos permite formular uma equação generalizada para cálculo do erro de cada neurônio da rede. Para os neurônios da camada de saída, pode-se representar o erro de saída do processador propagado como:

$$e_m = (t_m^p - s_m^p) \cdot \varphi'(v_m) \quad (7)$$

sendo e_m o erro da camada de saída do processador m propagado para as entradas e $\varphi'(v_m)$ a derivada da função de ativação do neurônio.

Para os neurônios das camadas ocultas da rede, o erro propagado da saída do processador é representado por:

$$e_k = \left(\sum_k e_m \cdot w_{mk} \right) \cdot \varphi'(v_k) \quad (8)$$

sendo e_k o erro da saída dos processadores das camadas intermediárias propagados para suas entradas e w_{mk} os pesos sinápticos das conexões entre as saídas intermediárias de k e as entradas da camada seguinte m .

Através do processo de propagação de erros descrito podemos estipular um método de aprendizado para uma rede neural. Esse processo envolve aplicar um padrão de entrada que será propagado por todas as camadas da rede neural até obter de uma resposta equivalente na saída. O erro da aproximação da saída encontrada em relação à saída desejada é utilizado para avaliar a capacidade de representação de conhecimento da rede. O processo de aprendizado pode ser expressado pelo algoritmo 1 apresentado a seguir.

Algoritmo 1 – Algoritmo *Back Propagation*.

```

1 Inicializar pesos com valores aleatórios e pequenos  $|w| \leq 0,1$ 
2 Repetir (enquanto erro de ajuste  $\leq$  limite determinado)
3   Aplicar um padrão de entrada à rede neural
4   Calcular as saídas dos processadores até a camada de saída
5   Calcular o erro dos processadores das camadas ocultas e de saída
6   Se (erro  $\geq$  tolerância) para qualquer um dos processadores
7     Atualizar os erros a partir da camada de saída até a primeira camada oculta
8 fim

```

Fonte: Adaptado de Rumelhart et al. (1986).

2.3.3 Algoritmo ADADELTA

O aprendizado pelo método do gradiente descendente apresenta notória sensibilidade à escolha da taxa de aprendizado η da rede, sujeito à paralisia da rede ou aprendizado muito lento quando a taxa de aprendizado escolhida é muito pequena, e grandes oscilações ou não convergência quando a taxa de aprendizado escolhida é muito grande.

Uma forma de lidar com esse problema considera iniciar o treinamento com uma taxa de aprendizado η elevada, e ao longo do processo de aprendizado reduzir gradativamente através de um termo de momento α . A Equação 9 apresenta o processo de atualização de pesos em função da taxa de aprendizado sobre o erro obtido nos processadores da camada de saída da rede e o termo de momento sobre a atualização realizada no instante anterior.

$$\Delta w_{ji}(t+1) = \eta e_j s_i + \alpha \Delta w_{ji}(t) \quad (9)$$

Duchi et al. (2011) propôs uma forma de controle dinâmico no processo de atualização de pesos, proporcional ao gradiente calculado em cada etapa do processo de aprendizado, logo, a taxa de aprendizado evoluirá sempre ao inverso do gradiente calculado.

Derivado da ideia descrita acima, Zeiler (2012) propõe um algoritmo capaz de suprir as ineficiências no uso de uma taxa de aprendizado global e da utilização do decaimento contínuo da taxa de aprendizado durante o treinamento.

A proposta do método de taxa de aprendizagem adaptativa, denominado algoritmo ADADELTA, é utilizar uma quantidade limitada de gradientes obtidos durante o processo de treinamento para ajuste da taxa de aprendizado. A janela de gradientes utilizada é implementada por um decaimento exponencial médio em função do tempo de treinamento. Assumindo no instante t , o gradiente médio quadrático $E[g^2]_t$ é calculado por:

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1 - \rho)g_t^2 \quad (10)$$

sendo ρ uma constante de decaimento similar ao termo de momento em (9).

Para atualização paramétrica de pesos, é utilizada a raiz quadrada do termo em (11):

$$RMS[g]_t = \sqrt{E[g^2]_t + \epsilon} \quad (11)$$

sendo a constante ϵ utilizada para melhor condicionamento do denominador na atualização de pesos:

$$\Delta w_{ji}(t) = -\frac{\eta}{RMS[g]_t} g_t \quad (12)$$

A seguir está representado a laço de repetição que descreve o funcionamento algoritmo de taxa de aprendizado adaptativa, ADADELTA.

Algoritmo 2 – Algoritmo de aprendizado ADADELTA.

- 1 *Inicializar taxa de decaimento ρ e constante ϵ*
- 2 *Inicializar parâmetro de entrada x_1*
- 3 *Inicializar variáveis acumuláveis: $E[g^2]_0 = 0, E[\Delta x^2]_0 = 0$*
- 4 ***Para $t = 1:T$ fazer***
- 5 *Calcular o gradiente g_t*
- 6 *Acumular gradiente: $E[g^2]_t = \rho E[g^2]_{t-1} + (1 - \rho)g_t^2$*
- 7 *Calcular atualização: $\Delta x_t = -\frac{RMS[\Delta x]_{t-1}}{RMS[g]_t} g_t$*
- 8 *Acumular atualização: $E[\Delta x^2]_t = \rho E[\Delta x^2]_{t-1} + (1 - \rho)\Delta x_t^2$*
- 9 *Aplicar atualização: $x_{t+1} = x_t + \Delta x_t$*
- 10 ***Fim***

Fonte: Adaptado de Zeiler (2012).

Este capítulo introduziu as características do jogo *Angry Birds* utilizado como ambiente de desenvolvimento deste trabalho, a estrutura do agente inteligente capaz de atuar de forma autônoma nesse cenário e suas particularidades funcionais.

Também foi introduzida a teoria envolvendo o funcionamento das Redes Neurais Artificiais, utilizada neste trabalho como estrutura de tomadas de decisões pelo agente desenvolvido para atuação no domínio especificado.

No próximo capítulo será apresentado uma breve revisão bibliográfica sobre as motivações para utilização dos jogos como ambiente de desenvolvimento científico, as diferentes abordagens dos principais agentes participantes ao longo da história da competição até os assuntos correntes que caracterizam o estado da arte e sustentam a relevância do trabalho apresentado.

2.4 TRABALHOS CORRELATOS

O *Angry Birds* é um dos inúmeros jogos utilizados como ambiente de desenvolvimento de temas correntes na comunidade científica, motivado pela competição anual realizada durante a Conferência Internacional de Inteligência Artificial (IJCAI). O foco dos trabalhos que envolvem esse ambiente está relacionado principalmente ao desenvolvimento de agentes inteligentes capazes de jogar de forma autônoma e sem que haja qualquer intervenção humana.

Segundo Yoon et al. (2015), pesquisas científicas em ambientes de jogos são ferramentas motivacionais para se despertar interesse de alunos em nível de graduação do curso de Ciência da Computação, pois permite aplicar competências teóricas essenciais para a formação de um profissional da área. Dentre essas competências estão: a introdução do cenário complexo de criação de um jogo e oportunidade de trabalhar conceitos avançados utilizados no âmbito científico, com a garantia de proporcionar estímulos positivos ao aprendizado através do entretenimento oferecido pelos jogos.

Sob uma perspectiva menos focada no processo de concepção de um jogo, Robič e Repnik (2015) exploraram a temática da utilização do conteúdo digital de jogos como ferramenta de aprendizado em salas de aula de ensino médio. O alvo do trabalho são jogos que não foram desenvolvidos com propósito educativo, e.g. o jogo *Angry Birds*, que pode ser utilizado para explicar conceitos de cinemática e dinâmica no estudo de física.

Outra abordagem possível no cenário dos jogos digitais envolve a problemática relacionada à geração de níveis para jogos, inserida no contexto de Geração Processual de Conteúdo (PCG), Shaker et al. (2013), que compartilha o interesse de desenvolvedores frente à geração de conteúdo para os jogos, e pesquisadores no desenvolvimento científico de alternativas para o desenvolvimento automático e inteligente desse conteúdo.

Ferreira e Toledo (2016) propuseram o desenvolvimento de um algoritmo de geração de níveis para o jogo *Angry Birds*, fundamentado em algoritmos genéticos focado principalmente na geração de níveis estáveis, solucionáveis e compatíveis com a habilidade do jogador, enquanto Jiang et al. (2017) propuseram uma solução focada exclusivamente na satisfação do jogador com a criação de níveis exclusivamente divertidos. A complexidade da tarefa proposta pela competição oficial de geração automática de níveis para o jogo *Angry Birds* é combinar a criatividade, diversão e dificuldade dos níveis gerados em uma única solução.

Quando o objeto é o desenvolvimento de agentes inteligentes, percebemos abordagens distintas aplicadas ao cenário do *Angry Birds* com o propósito comum de desenvolver agentes suficientemente inteligentes, autônomos e competitivos através do aperfeiçoamento da

funcionalidade dos módulos básicos que compõem um agente: sistema de visão para aquisição de imagens, tomada de decisões, estratégias de lançamento e previsões de trajetória.

O agente padrão desenvolvido e disponibilizado aos interessados em trabalhar no domínio do jogo *Angry Birds* é denominado Agente Ingênuo, dotado da estratégia elementar de mirar diretamente nos porcos independentemente da configuração estrutural que o cercam (GE et al., 2014). O Agente Ingênuo foi o vencedor do primeiro desafio realizado em 2012, composto de 9 participantes (AIBIRDS, 2019a).

O agente IHSEV, vice-campeão da competição anual de *Angry Birds* em 2015 e 2016, desenvolvido por Polceanu et al. (2014) é modelado a partir de uma formulação e representação do domínio do jogo em um contexto computacional que tenta de reproduzir o processo de raciocínio e tomada de decisões humana. O agente teve seu sistema de visão implementado aprimorando a qualidade dos dados extraídos pelo agente, utilizados em um processo de simulações em busca da melhor descrição de estratégia de lançamento e tomada de decisões nesse “mundo imaginário” em busca do objetivo do jogo.

Borovička et al. (2014) desenvolveram um agente denominado DataLab Birds dotado de múltiplas estratégias dada a configuração do cenário, assim como avaliação heurística da utilidade das trajetórias de lançamento possíveis em função do número de blocos, distância entre o lançamento e o alvo e o número de blocos de interferência na trajetória avaliada. A abordagem em questão garantiu o título das competições de 2014 e 2015, e serviu de base para o atual campeão da competição.

O agente Eagle Wing, vencedor da competição em 2017 e 2018, proposto por Wang (2017) foi construído a partir da abordagem do campeão da competição em 2014 (DataLab Birds), conforme descrito na seção 2.2.4. O atual campeão da competição modificou a abordagem de múltiplas estratégias com a formulação de nova avaliação de utilidade e predição de lançamento. O aprendizado das melhores condutas foi realizado através de milhares de exemplos de treinamento jogados nos 42 níveis do episódio *Poached Eggs* utilizando método de aprendizado supervisionado *xgboost*.

Além das abordagens utilizadas pelos agentes finalistas e vencedores descritos, diversas abordagens foram exploradas por outros agentes em busca de melhores desempenhos como: técnicas de representação de conhecimento e programação lógica em Tziortziotis et al. (2016), raciocínio espacial qualitativo em Ferreira et al. (2013), Walega et al. (2014) e Zhang e Renz (2014), métodos avançados de simulação em Schiffer et al. (2016), análises estruturais e predição de dano em Dasgupta et al. (2016).

Recentemente, alguns trabalhos surgiram com a proposta de aplicação do método de aprendizado por reforço no cenário do jogo *Angry Birds*. Yuan et al. (2017) desenvolveram um algoritmo de aprendizado por reforço para um agente com o objetivo de aprender as melhores configurações de tipo de pássaro para lançamento, distância do alvo e ponto de ativação da habilidade especial, utilizando o algoritmo *Deep Deterministic Policy Gradient* (DDPG). Os resultados apresentados foram satisfatórios apesar de não superar o desempenho dos agentes campeões.

As redes neurais foram muito pouco exploradas em agentes no ambiente de jogo *Angry Birds*, e podem ser um caminho para o desenvolvimento de agentes com a capacidade de aprendizado melhoradas. Segundo Bezdek (1992), a literatura indica a prevalência do uso de Redes Neurais nas tarefas de reconhecimento de padrões, e é nesse contexto que será explorada neste trabalho com o intuito aproximar a capacidade de atuação de um agente inteligente à dos seres humanos.

3 DESENVOLVIMENTO

Neste capítulo, será apresentada a implementação do agente jogador de *Angry Birds* através da modelagem de uma Rede Neural Artificial para a tomada de decisões de seleção de alvo. O ambiente de testes e validação da proposta utilizado é o mesmo utilizado pela competição oficial, rodado no navegador *Google Chrome* em sua versão off-line, e a solução implementada a partir do Agente Ingênuo disponibilizado, *v1.32* (2018). A avaliação de seu desempenho será feita através da comparação entre agentes existentes e da submissão do agente desenvolvido para participação da competição anual de *Angry Birds* de 2019.

3.1 MODELAGEM DO PROBLEMA

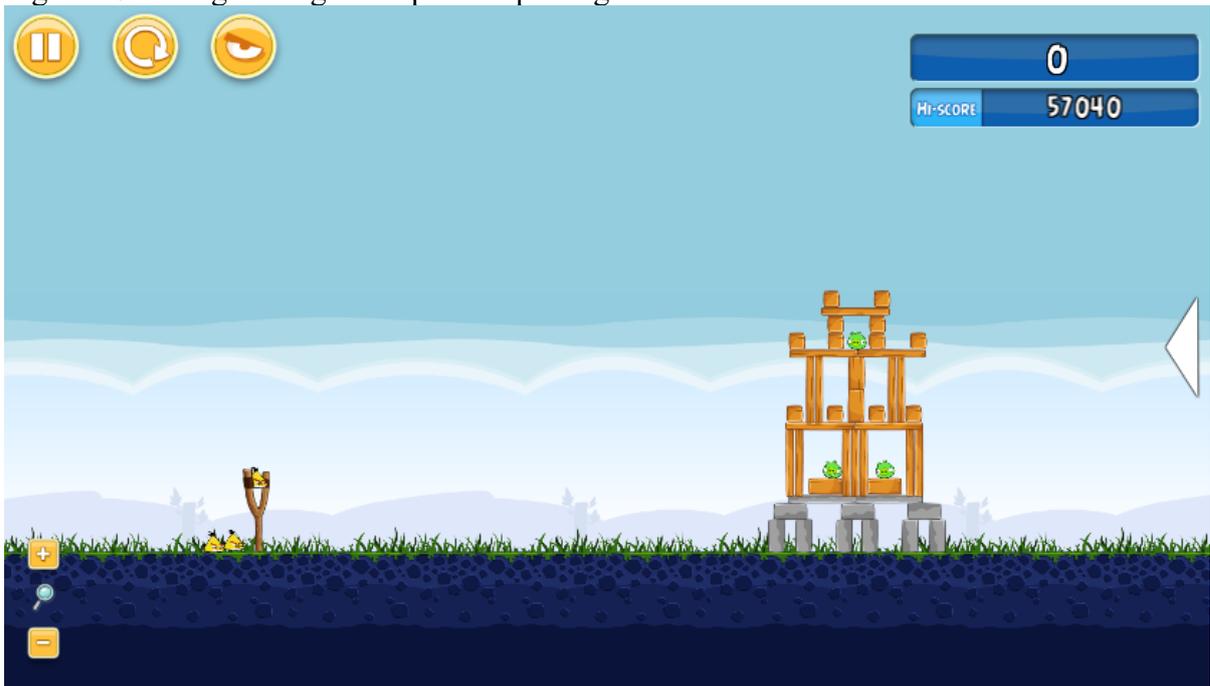
Este trabalho propõe o desenvolvimento de uma Rede Neural Artificial como estrutura de tomada de decisões para a seleção de alvo a partir da representação do cenário do jogo por uma imagem capturada, segmentada e redimensionada. A seguir, serão apresentados os detalhes de construção da rede desenvolvida e utilizada pelo agente proposto.

3.1.1 Entrada

Os dados de entrada da RNA para o problema proposto são uma representação do estado atual do jogo em que a ação será tomada, ou seja, uma imagem processada, segmentada e redimensionada capaz de representar o estado atual do jogo.

A imagem extraída originalmente pelo agente está no formato PNG, com sistema RGB de representação de cores e dimensões 840x480 *pixels*. A figura 10 representa a imagem original extraída pelo agente.

Figura 10 – Imagem original capturada pelo agente.



Fonte: Autor.

Essa imagem é processada, recortando regiões irrelevantes para o treinamento, segmentada por cores, simplificando a representação dos objetos que a compõem, e redimensionada em 90×23 pixels para serem utilizadas como dados de entrada de uma Rede Neural Artificial. O resultado do processo de tratamento da imagem está representado pela Figura 11.

Figura 11 – Imagem recortada, segmentada e redimensionada.



Fonte: Autor.

O método MBR de identificação de objetos foi utilizado para identificar o tipo de cada objeto do cenário e sua posição espacial para a composição da nova imagem segmentada.

Finalmente, durante o processo de treinamento as imagens são convertidas para o sistema de representação em tons de cinza e serializadas, tornando cada um dos 2070 pixels de composição da imagem uma entrada da rede, assumindo valores numéricos inteiros entre 0 e 255.

$$\text{Escala de Cinza} = \frac{(R + G + B)}{3} \quad (14)$$

A equação 14 apresenta a função de transição do sistema de representação RGB para a escala em tons de cinza através do cálculo da média aritmética entre as três componentes originais da imagem colorida: vermelho, verde e azul.

3.1.2 Topologia da Rede Neural Artificial (RNA)

A Rede Neural foi implementada na linguagem *python v3.6.2*, através da biblioteca de código aberto para redes neurais *Keras v2.2.2*, rodada pelo *Tensorflow v1.10.0*.

A RNA possui duas camadas ocultas completamente conectadas, com 800 neurônios na primeira e 500 neurônios na segunda, garantindo à rede quantidade de neurônios suficiente para o aprendizado da função de representação dos dados e dentro da capacidade de processamento do sistema computacional utilizado. O algoritmo de aprendizado utilizado é o ADADELTA (ZEILER, 2012) e utiliza função de ativação linear e correção dos pesos sinápticos através do método dos mínimos quadrados para avaliação do erro nas camadas de saída.

3.1.3 Saída

A partir de observações sobre a distribuição dos objetos nos níveis do jogo, constatou-se que existe uma região específica capaz de delimitar as posições padrões dos alvos dos lançamentos.

Portanto, utilizando os *pixels* da imagem como grandeza de medida, foi possível demarcar essa região a partir do *pixel* 450 e antes do *pixel* 787 na horizontal, e abaixo do *pixel* 200 na vertical, conforme representado na Figura 12.

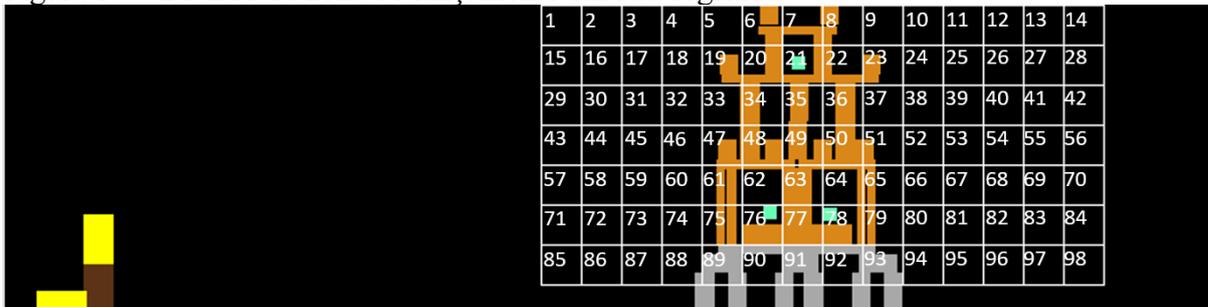
Figura 12 – Zona ativa de presença de alvos.



Fonte: Autor.

A partir da demarcação apresentada, a região de lançamento foi subdividida em 98 regiões idênticas, de 4x4 *pixels* na escala da imagem de dimensionalidade reduzida, equivalente a regiões de 32x32 *pixels* na escala da imagem original. Essa subdivisão pode ser observada na Figura 13.

Figura 13 – Divisão da zona de lançamento em sub-regiões.



Fonte: Autor.

Sendo assim, para cada estado estacionário do jogo o agente possui 98 ações, equivalentes aos 98 alvos de lançamento. Alguns testes durante a implementação das zonas de lançamento revelaram que: utilizar menos zonas de maior área tornam a seleção de alvos imprecisa, e mais zonas de menor área demandam alto poder computacional durante o processo de treinamento da rede.

3.1.4 Treinamento

O treinamento supervisionado encontra uma função de mapeamento entre dados de entrada e suas respectivas saídas desejadas. O processo de treinamento da rede proposta exigiu a modelagem de uma base de dados, construída através da exploração experimental do domínio do jogo.

Um agente foi adaptado para jogar ininterruptamente diversos níveis da versão clássica do jogo e realizar lançamentos aleatórios em todas as 98 zonas pré-definidas. A cada lançamento realizado, o agente armazena a imagem de representação do estado atual do jogo antes do lançamento e registra em um arquivo texto o nome dessa imagem, as coordenadas das abscissas e ordenadas do alvo desse lançamento, e a pontuação obtida desta ação.

A figura 14 apresenta estrutura dos registros que compõem a base de dados.

Figura 14 – Estrutura da base de dados.



Arquivo	Editar	Formatar	Exibir	Ajuda
img000000.png	659	333	15130	
img000001.png	452	309	29640	
img000002.png	561	203	2200	
img000003.png	655	243	9790	
img000004.png	691	301	3850	
img000005.png	454	248	12450	
img000006.png	486	218	0	
img000007.png	600	322	22360	
img000008.png	452	230	1100	
img000009.png	468	322	9750	
img000010.png	674	366	25080	

Fonte: Autor.

Para o treinamento da RNA, as informações das coordenadas do lançamento contidas na base de dados devem ser transformadas em um valor numérico que represente a zona que essa coordenada está contida. Observando a Figura 13, é possível comparar a estrutura de representação das zonas de lançamento com uma matriz $A_{7 \times 14}$ e definir a relação entre a posição espacial na imagem com a posição do termo equivalente na matriz.

$$A_{m \times n} \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{pmatrix} \quad (15)$$

sendo n as colunas da matriz, que mapeiam as coordenadas no eixo horizontal das zonas de lançamento, com $\{n \in \mathbb{Z} \mid 0 < n < 14\}$, e m as linhas da matriz, que mapeiam as coordenadas no eixo vertical das zonas de lançamentos, com $\{m \in \mathbb{Z} \mid 0 < m < 7\}$.

$$n = \frac{\frac{(alvo_x - recorte_x)}{fr} - d_{e \rightarrow z}}{l_z - 1} \quad m = \frac{\frac{alvo_y - recorte_y}{fr}}{h_z - 1} \quad (16)$$

Sendo:

- $alvo_x$: o pixel de coordenada do centro do alvo na horizontal;
- $recorte_x$: a zona desconsiderada da imagem original na extremidade horizontal esquerda, de 119 *pixels* de comprimento;
- $recorte_y$: a zona desconsiderada da imagem original na extremidade vertical superior, de 199 *pixels* de comprimento;
- $d_{e \rightarrow z}$: a distância do estilingue ao início da demarcação das zonas de lançamento na imagem redimensionada, equivalente à 40 *pixels* de comprimento;

- fr : o fator de redução da imagem original para a imagem redimensionada, reduzida em 8 vezes;
- l_z : a largura de cada zona na imagem redimensionada, de 4 *pixels* de comprimento;
- h_z : a altura de cada zona na imagem redimensionada, de 4 *pixels* de comprimento.

Através desse mapeamento apresentado na equação 16, é possível definir a zona que cada alvo ocupa a partir de suas coordenadas cartesianas. As operações aritméticas de divisão apresentadas na equação 16 consideram o maior quociente inteiro possível.

$$ZONA = 14m + n + 1 \quad (17)$$

A base de dados contém o valor da pontuação de cada lançamento realizado, porém, fez-se necessário classificar cada lançamento entre bom, regular e ruim. Após estudo do domínio, verificou-se que as maiores pontuações observadas para um único lançamento estão em torno de 30.000 pontos, portanto, cada lançamento contido na base de dados foi classificado da seguinte forma:

$$R(\text{pontuação}) = \begin{cases} -1 & \text{se } \frac{\text{pontuação}}{30.000} \leq 0,17 \\ -0,5 & \text{se } 0,17 < \frac{\text{pontuação}}{30.000} \leq 0,34 \\ 0 & \text{se } 0,34 < \frac{\text{pontuação}}{30.000} \leq 0,5 \\ +0,5 & \text{se } 0,5 < \frac{\text{pontuação}}{30.000} \leq 0,7 \\ +1 & \text{se } \frac{\text{pontuação}}{30.000} > 0,7 \end{cases}$$

As informações da base de dados são lidas e organizadas em três conjuntos: os vetores de representação da imagem, o número inteiro de representação da zona alvo do lançamento, e o valor da classificação desse lançamento. Essas informações compõem os dados utilizados para o treinamento da rede.

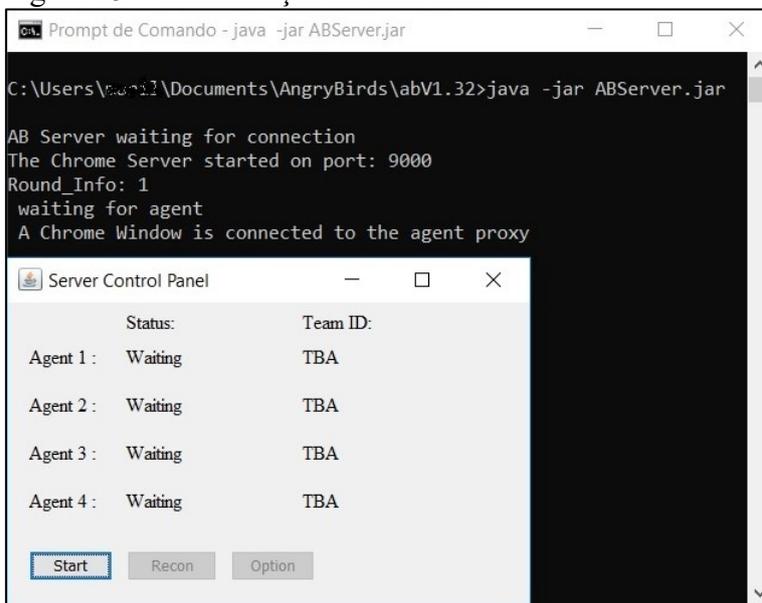
A rede criada utiliza o processo de treinamento incremental, com atualização do gradiente a cada dado aplicado à rede, e que apenas o erro da saída do padrão aplicado seja propagado para atualização dos pesos das camadas ocultas da rede, pois o resultado de um lançamento possui informações de pertinência de apenas uma zona.

3.2 FUNCIONAMENTO DO AGENTE

Para que o agente inteligente possa jogar *Angry Birds* de forma autônoma, o ambiente e a estrutura de atuação devem ser previamente inicializados.

O jogo deve ser inicializado na janela do *Google Chrome*, em sua versão *offline*. Em seguida, uma janela do *prompt* de comando deverá ser aberta no diretório raiz do agente. Neste diretório está presente um arquivo executável denominado *ABServer.jar*, responsável pela inicialização do servidor que será conectado ao jogo já aberto no navegador do *Chrome*, digitando o comando: *java -jar ABServer.jar*.

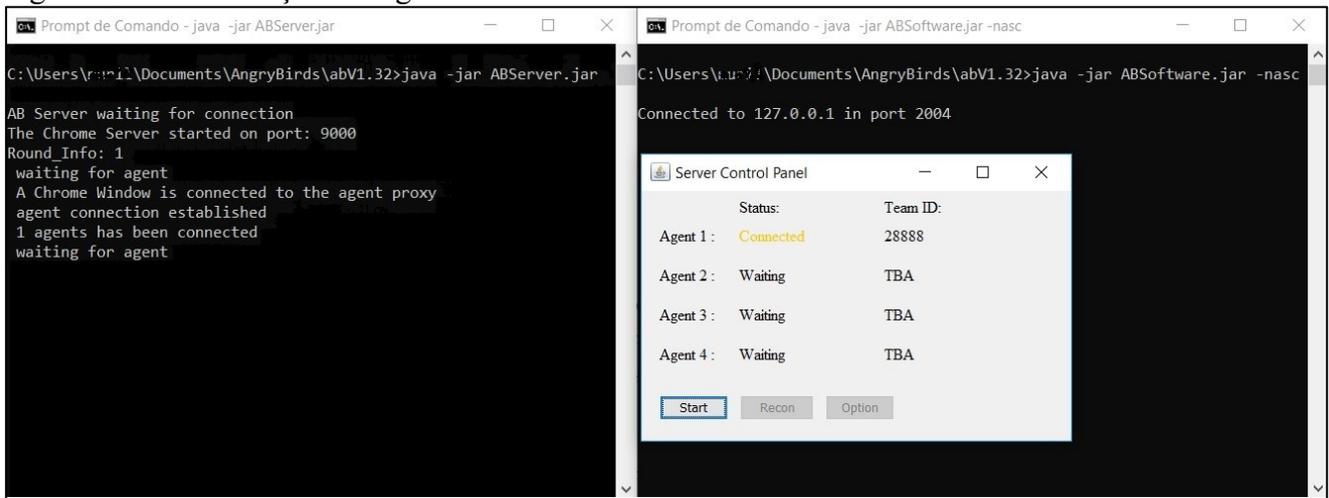
Figura 15 – Inicialização do Servidor.



Fonte: Autor.

Outra janela do Prompt de Comando deve ser aberta para que o agente seja inicializado e conectado ao servidor. Para isso, o arquivo executável *ABSoftware.jar* localizado no diretório raiz do agente deve ser inicializado através do comando: *java -jar ABSoftware.jar -nasc*.

Figura 16 – Inicialização do Agente.



Fonte: Autor.

Assim que o agente é conectado ao servidor, todos os módulos necessários para sua atuação são instanciados, sendo eles: módulo de ações no navegador, módulo de previsão de trajetória e o módulo desenvolvido denominado RNA.

O módulo RNA é uma nova classe incorporada à estrutura do Agente Ingênuo, responsável pela tomada de decisões, composto por métodos responsáveis pelo: recebimento da imagem extraída do jogo, segmentação e redimensionamento dessa imagem, coleta de dados de treinamento baseado em experiências e inferência à Rede Neural desenvolvida.

Pressionando o botão *Start* da janela aberta, representado na Figura 16, um nível é inicializado e o agente começa a jogar, executando uma sequência de operações na tentativa de concluí-lo:

1. Captura da imagem de representação do cenário atual;
2. Extração de informações e segmentação;
3. Seleção do alvo do lançamento;
4. Cálculo das coordenadas de lançamento para atingir o alvo desejado;
5. Seleção do momento de ativação da habilidade secundária do pássaro lançado;
6. Resultado e registro em base de dados do lançamento;

3.2.1 Captura da imagem de representação do cenário atual.

A primeira ação realizada pelo agente é capturar a imagem atual do jogo, instanciá-la em seu módulo de visão. A imagem obtida é utilizada pelo módulo de visão do agente para: identificar a posição espacial dos objetos, distinguir a tipo de objeto identificado, extrair a pontuação do agente e identificar os botões de navegação no ambiente do jogo.

A imagem capturada possui validade apenas se for possível identificar a posição espacial do estilingue, permanecendo em um laço de repetição até que o mesmo seja identificado.

3.2.2 Extração de informação e segmentação.

O módulo RNA recebe a imagem original extraída do jogo, e realiza a segmentação dessa imagem. O processo de segmentação da imagem tem como objetivo suprimir detalhes irrelevantes da imagem e reduzir sua dimensionalidade, oferecendo uma imagem simplificada de representação da cena do jogo que será utilizada como dado de entrada da rede neural proposta.

A imagem original de 840x480 *pixels* é cortada na horizontal e na vertical, resultando em uma nova imagem de 720x184 *pixels*. Esse recorte suprime as áreas desnecessárias para análise de cenário, como o solo na parte inferior, os painéis de pontuação e navegação na parte superior e a região que antecede a posição do estilingue. A Figura 17 apresenta as regiões desconsideradas pelo recorte.

Figura 17 – Recorte da imagem original.

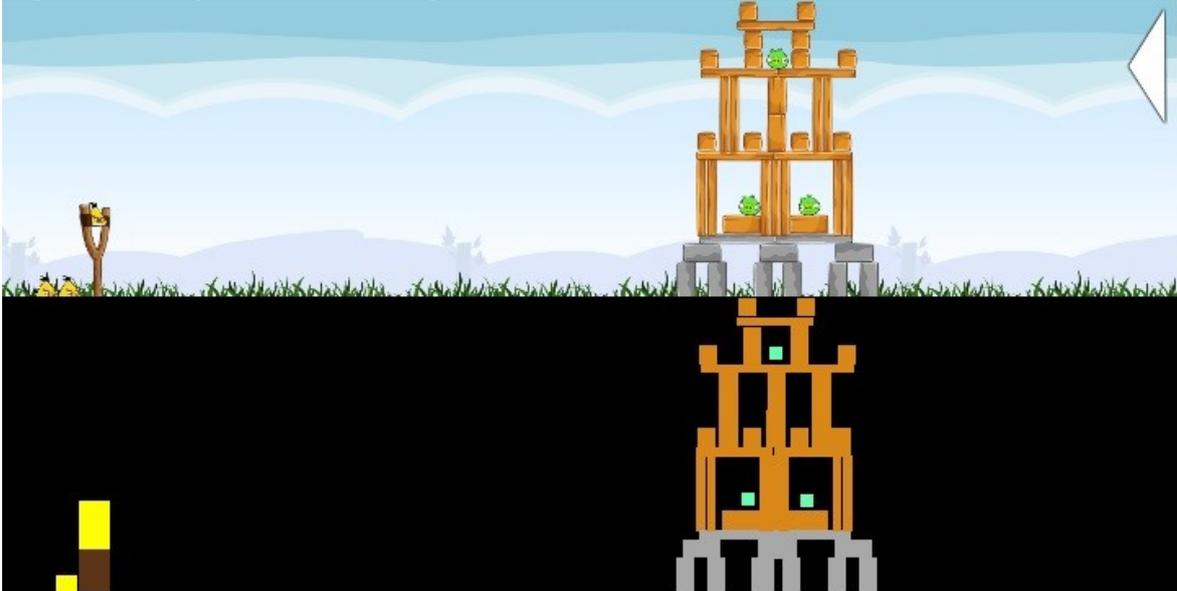


Fonte: Autor.

A imagem segmentada possui fundo preto, e apresenta cada objeto da imagem original representado por retângulos, mantendo a proporcionalidade de largura, altura e coloração dos

objetos originais. A Figura 18 apresenta a transição da imagem original para a imagem segmentada.

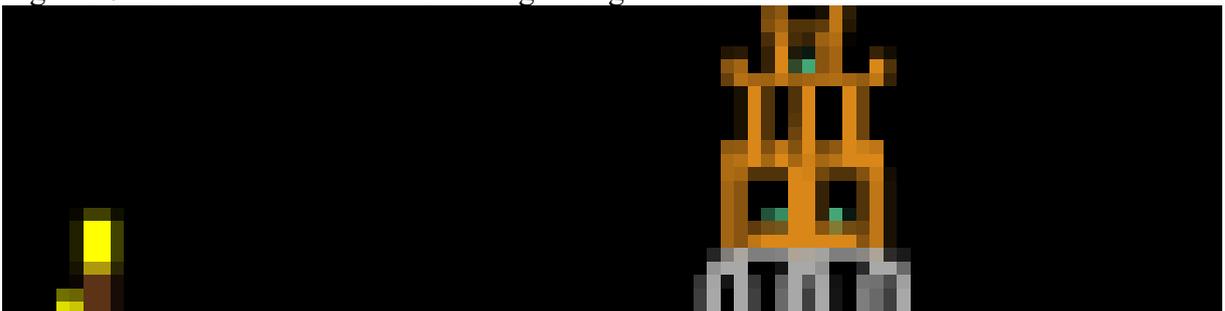
Figura 18 – Segmentação da imagem cortada.



Fonte: Autor.

Por fim, a imagem é redimensionada de 720×184 pixels para 90×23 pixels. A redução da imagem em 8 vezes de seu tamanho original tem como objetivo tornar viável seu processamento computacional como entrada de uma RNA, preservando as informações nela contida. A figura 19 ilustra a imagem redimensionada.

Figura 19 – Redimensionamento da imagem segmentada.



Fonte: Autor.

3.2.3 Seleção do alvo

A estrutura de tomada de decisões para identificação do alvo de cada lançamento é a principal proposta desse trabalho, realizado através da construção de uma Rede Neural Artificial, treinada de forma supervisionada a partir de dados de treinamento extraídos da exploração aleatória e experimental do domínio.

O agente salva a imagem que representa a situação atual do jogo em um diretório específico, e executa um *script* em *python* contendo o nome dessa imagem para que seja utilizada como inferência na rede.

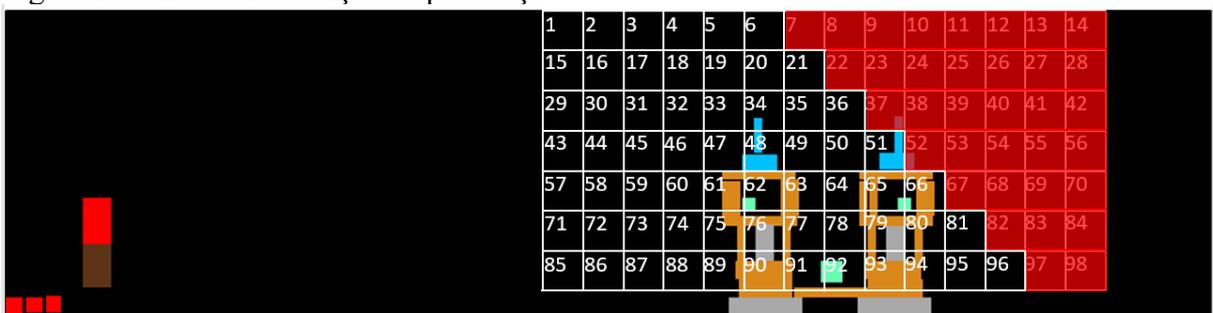
A rede treinada processa essa imagem de entrada, e apresenta um o valor estimado de vantagem de cada uma das 98 zonas saídas ser escolhida, baseado no treinamento previamente realizado. Considerando que para uma condição de lançamento possa existir mais de uma região cujo lançamento seja promissor, o agente seleciona, com probabilidade de 0,75, a zona mais promissora ou, com probabilidade 0,25, a segunda zona mais promissora, e retorna o resultado da inferência para o agente.

3.2.4 Conduta de lançamento

O número da zona de lançamento recebido pela rede de inferências é transformado em coordenadas cartesianas para que a trajetória de lançamento seja calculada. O sistema de cálculo de trajetória apresenta no máximo dois ângulos possíveis para atingir determinado alvo, sendo assim, o agente foi parametrizado para sempre selecionar o menor ângulo de possível calculado, visto que o potencial de destruição para esse tipo de lançamento é geralmente maior, pois atinge um número maior de estruturas de sustentação das edificações do cenário.

Esse critério de seleção do menor ângulo de lançamento torna impossível atingir determinadas zonas segundo o modelo de divisão proposto, reduzindo o número total de ações possíveis da modelagem de 98 para 63, conforme representado na Figura 20.

Figura 20 – Zonas inalcançáveis por lançamentos baixos.



Fonte: Autor.

Essas zonas inalcançáveis foram mantidas na modelagem da solução do problema visto a possibilidade de melhoria do agente em trabalhos futuros, podendo o agente aprender o tipo de lançamento mais adequado para determinada situação.

A região selecionada como alvo do lançamento possui em escala real 24x24 *pixels*, ou seja, existem 576 pontos de lançamento possíveis contidos em uma única zona, concentrados

em uma região quadrada, muito próximos uns dos outros ou pertencentes a trajetórias semelhantes.

A princípio, o agente foi configurado para selecionar aleatoriamente qualquer um dos pontos pertencentes à zona escolhida, porém, foi observado durante seu desenvolvimento que esse tipo de escolha acentuava a imprecisão dos lançamentos, impactando negativamente o processo de treinamento, visto que esse efeito possibilita consequências diferentes para lançamentos realizados dentro da mesma região.

Portanto, o agente foi configurado para utilizar como coordenadas de lançamento as coordenadas do centro de cada zona.

3.2.5 Ativação das habilidades específicas

A escolha de ativação das habilidades secundárias dos pássaros foi fixada para serem ativadas em 85% e 95% da trajetória do lançamento para pássaros amarelos e azuis respectivamente. A finalidade da adoção desses parâmetros é minimizar a imprecisão de lançamentos nas condições de funcionamento do agente proposto e maximizar o poder destrutivo de cada lançamento.

O tipo do pássaro para cada lançamento está implícito na imagem segmentada disponibilizada à rede, portanto, espera-se que a rede treinada seja capaz de considerar essas informações contidas na imagem para seleção dos alvos mais promissores.

3.2.6 Registro de lançamento.

O resultado de todo lançamento é registrado em uma base de dados. A base de dados é composta de um diretório com imagens do cenário do jogo que precedem um lançamento e um arquivo de texto atrelando o nome dessa imagem, as coordenadas do alvo do lançamento realizado e a pontuação obtida.

O módulo de visão do agente possui uma função para extrair a pontuação obtida após um lançamento, portanto, é possível obter e registrar isoladamente a pontuação de cada lançamento. O registro de pontuação inclui a condição de conclusão de um nível, onde são considerados bônus por concluir o nível sem utilizar todos os pássaros.

4 RESULTADOS

Para validar a solução proposta, a rede foi inicialmente treinada e testada separadamente em dois níveis distintos, com o objetivo de avaliar a capacidade de representação dos exemplos de treinamento apresentados à rede. Essa primeira etapa de testes está descrita na seção 4.1, quando o agente foi treinado e submetido a jogar os níveis 10 e 17 do primeiro episódio da versão clássica do jogo.

Os resultados preliminares obtidos durante a validação do processo de treinamento da rede desenvolvida neste trabalho mostraram que o método de taxa de aprendizado adaptativo apresentou melhor desempenho em relação aos demais métodos testados.

O algoritmo ADADELTA facilitou a realização do processo de treinamento de um conjunto de dados de treinamento novo e pouco explorado, além de apresentar melhor eficiência no tempo necessário para o treinamento da rede.

O critério de escolha desses dois níveis para os testes preliminares foi a dificuldade que os agentes Ingênuo e *Eagle Wings* apresentaram para solucioná-los. Dessa forma foi possível avaliar a capacidade e limitações do reconhecimento e representação de padrões da Rede Neural Artificial desenvolvida.

Em seguida, a rede foi treinada com um conjunto de dados de treinamento de 35 mil exemplos, coletados dos níveis ímpares de 3 a 17 e nível 10 do episódio *Poached Eggs*. Nesta segunda etapa de testes o agente treinado foi colocado pra jogar em todos os 21 níveis do episódio, permitindo assim avaliar o desempenho do agente quando submetido em níveis nunca vistos e certificando que a rede efetivamente foi capaz de generalizar o aprendizado. A seção 4.2 contém a metodologia e resultados desta segunda etapa de testes.

Em geral, a acurácia ou a precisão de uma Rede Neural Artificial são medidas estatísticas eficientes para avaliação de seu desempenho, utilizados para avaliar quão bem uma classificação binária identifica ou exclui corretamente determinado padrão.

Entretanto, não é possível avaliar o desempenho da rede proposta nesse trabalho através dessas métricas, pois não se trata essencialmente de um processo de classificação binária, mas da capacidade de identificar as vantagens de cada ação possível dado um padrão apresentado.

Além disso, as ações tomadas pelo agente e suas consequências não se comportam de forma determinística devido à natureza física que modela as relações entre os objetos do jogo.

4.1 TREINAMENTO PRELIMINAR

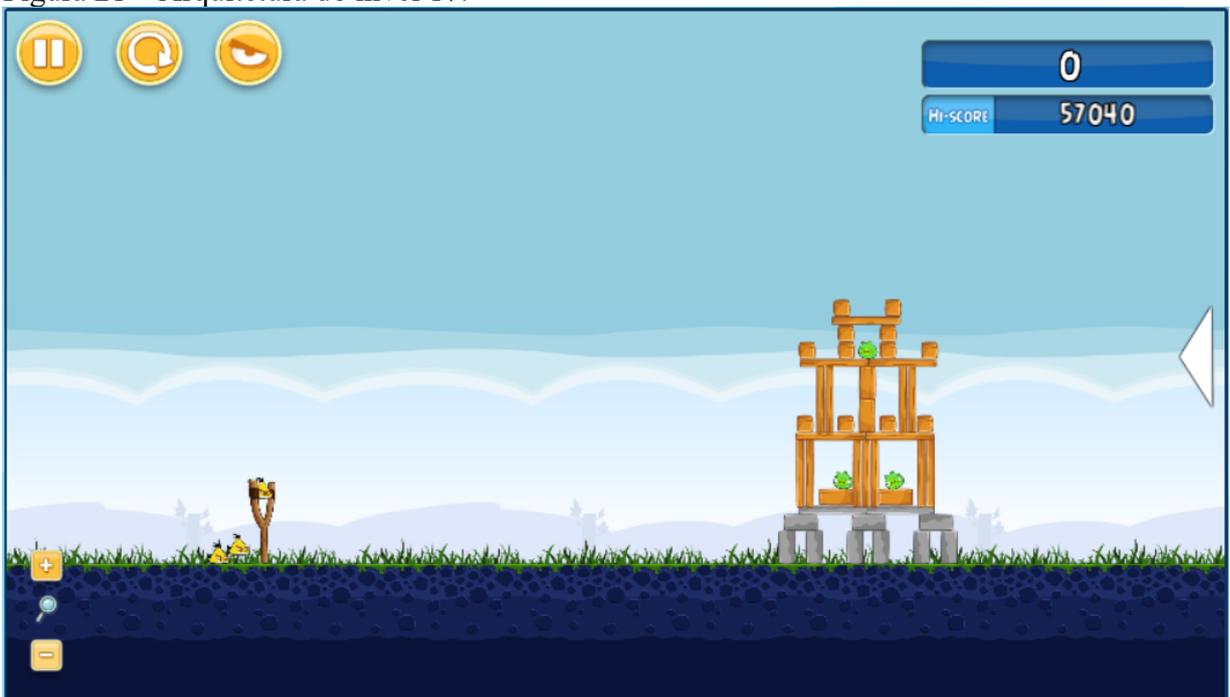
Está descrito nessa seção o processo de treinamento da rede para operar exclusivamente nos níveis 17 e 10 da primeira parte do episódio *Poached Eggs*. Estão apresentadas a configuração física dos objetos que compõem o cenário dos níveis, as decisões tomadas pela rede treinada diante de um padrão apresentado e a análise qualitativa da pertinência das ações obtidas pelo processo de inferência. Por fim, o desempenho dos agentes é comparado a fim de avaliar se a proposta apresentada foi capaz de traduzir as melhores ações possíveis para cada situação.

4.1.1 Nível 17

O nível 17 foi o primeiro cenário em que o agente desenvolvido foi submetido, através da coleta de dados de treinamento, treinamento e inferência na rede de seleção de alvos e avaliação de tendências e resultados. Esse nível possui apenas um tipo de pássaro para lançamento, um tipo de material na composição da estrutura física e não exige estratégias complexas para ser solucionado.

O nível disponibiliza três pássaros amarelos para a eliminação de três porcos cercados por uma estrutura alta de madeira, conforme representação na Figura 21.

Figura 21 – Arquitetura do nível 17.



Fonte: Autor.

Apesar da simplicidade do nível, a estratégia utilizada pelo Agente Ingênuo não apresenta bons resultados, pois a profundidade dos alvos no interior da estrutura impede que sua habilidade secundária seja efetivamente ativada, demandando na maioria das vezes todos os pássaros para a conclusão do nível, minimizando assim a pontuação conquistada.

Já o agente *Eagle Wing* (2017), por vezes apresentou dificuldade para concluir o nível, pois a seleção de uma estratégia ruim pode ocasionar o desmoronamento das estruturas sem que os alvos sejam eliminados, dificultando que sejam atingidos nas duas tentativas seguintes.

Foram utilizados 4.000 dados para o treinamento da rede de seleção de alvos. Esses dados foram coletados a partir de lançamentos aleatórios utilizando todos os pássaros disponíveis, em diferentes condições do cenário e em todas as zonas acessíveis.

Para avaliar a capacidade de representação da rede treinada, foram aplicados diferentes padrões de entrada e avaliado as respectivas respostas obtidas dessa inferência. A avaliação da resposta da rede ao teste foi baseada no conhecimento humano aprendido sobre o nível durante a exploração experimental do domínio.

O primeiro padrão de entrada aplicado à rede foi o vetor de representação da condição inicial do nível. Os valores de cada uma das 98 saídas da rede em resposta ao padrão apresentado estão representados na Tabela 2.

Tabela 2 – Resposta da rede à inferência apresentada na Figura 22.

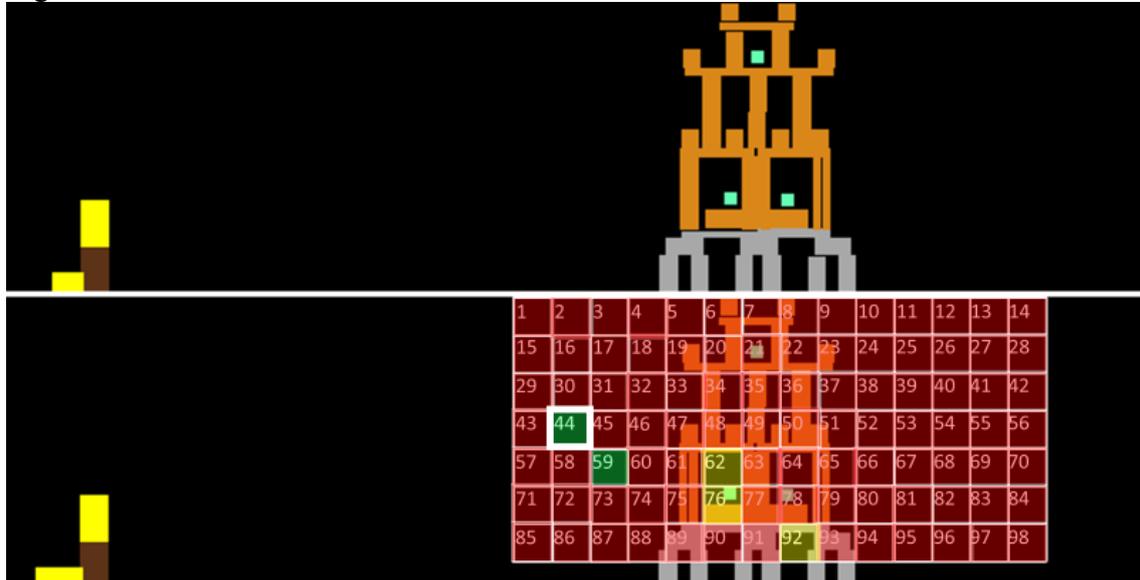
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Inferência	-0,53	-0,80	-0,74	-0,55	-0,73	-0,74	-1,00	-1,00	-1,00	-1,00	-1,00	-1,00	-1,00	-1,00
Zona	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Inferência	-0,75	-0,10	-1,40	-0,83	0,82	-0,50	-0,46	-1,00	-1,00	-1,00	-1,00	-1,00	-1,00	-1,00
Zona	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Inferência	-0,73	-0,35	-2,10	-0,07	-1,36	-0,40	0,00	-0,92	-1,00	-1,00	-1,00	-1,00	-1,00	-1,00
Zona	43	44	45	46	47	48	49	50	51	52	53	54	55	56
Inferência	-0,62	2,04	0,06	-0,73	-0,34	-0,42	-1,68	-0,64	0,00	-1,00	-1,00	-1,00	-1,00	-1,00
Zona	57	58	59	60	61	62	63	64	65	66	67	68	69	70
Inferência	1,11	0,49	1,89	0,10	-0,60	1,67	-0,72	0,82	-1,02	-1,16	-1,00	-1,00	-1,00	-1,00
Zona	71	72	73	74	75	76	77	78	79	80	81	82	83	84
Inferência	-0,12	-1,34	-0,98	0,12	0,16	1,85	1,24	0,64	0,69	0,21	0,73	-1,00	-1,00	-1,00
Zona	85	86	87	88	89	90	91	92	93	94	95	96	97	98
Inferência	-0,88	-1,36	-0,47	-0,97	-1,65	0,70	-0,62	1,37	0,29	-0,10	-0,67	0,07	-1,00	-1,00

Fonte: Autor.

A tabela possui dois valores destacados na cor verde referente as duas saídas mais promissoras para a situação apresentada. Em amarelo a terceira, quarta e quinta saída com maior valor de pertinência para o lançamento. Essas zonas foram destacadas para facilitar a visualização das potenciais ações da rede neural treinada para essa situação.

A Figura 22 apresenta o padrão de entrada aplicada à rede na parte superior e a representação de pertinência de cada uma das 98 saídas para essa entrada na parte inferior.

Figura 22 – Inferência 1 no nível 17.



Fonte: Autor.

A ilustração dos valores obtidos pela inferência realizada revela que a melhor estratégia aprendida para a situação inicial é atingir a parte frontal inferior da estrutura, priorizando alvejar a zona 44, considerada a melhor escolha visto a possibilidade de atingir a aresta superior da estrutura de sustentação inferior. Esta ação possui alto potencial destrutivo e apresenta a possibilidade de conclusão do nível no primeiro lançamento.

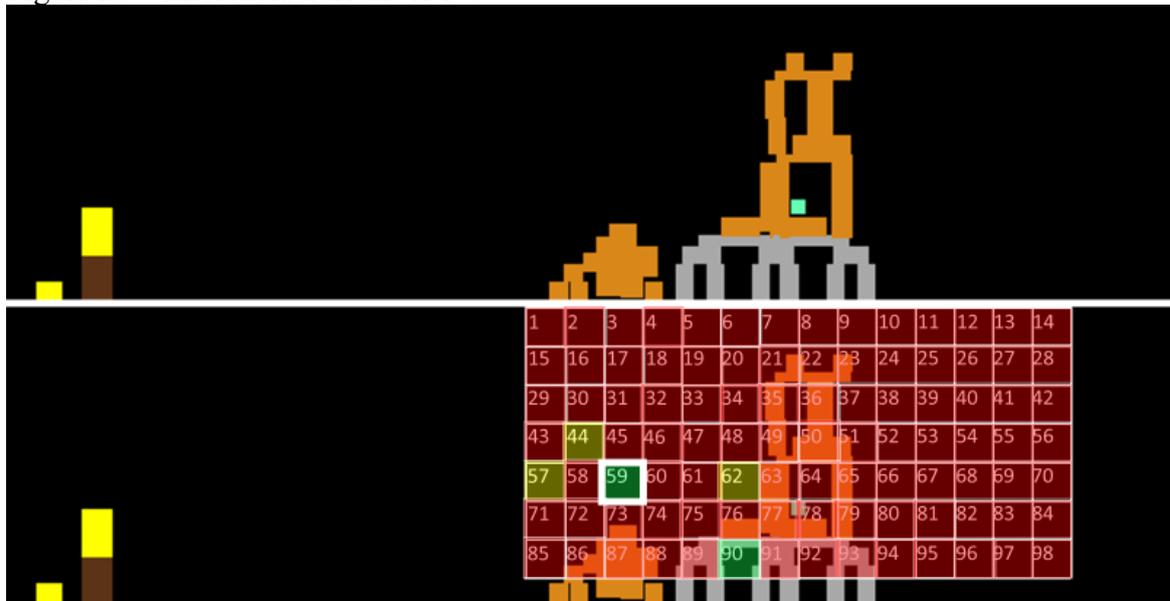
Alguns testes preliminares constataram a presença de ruído na precisão dos lançamentos, pois foi verificada a ocorrência de dois lançamentos no mesmo alvo apresentarem trajetórias distintas.

Além disso, foi verificado que lançamentos realizados na mesma região, na mesma condição de cenário podem apresentar consequências de desmoronamento e destruição diferentes, devido à natureza física do jogo. Esse efeito impossibilita testar a eficiência da rede aplicando um conjunto de dados de testes e avaliando sua acurácia, pois a mesma zona pode apresentar variações consideráveis na pontuação obtida para dois lançamentos similares.

De modo geral, as regiões mais promissoras estão localizadas antes da estrutura em consequência da necessidade de ativação da habilidade secundária do pássaro amarelo para obtenção de melhores resultados. Além disso, todas as regiões de destaque da Figura 22 compartilham a mesma trajetória, garantindo a eficiência da rede em aprender a melhor conduta de lançamento possível.

A figura 23 apresenta na parte superior o segundo padrão de entrada aplicado em caráter de teste da rede treinada e a representação de pertinência de cada uma das 98 saídas para essa entrada na parte inferior. Essa segunda imagem apresenta o cenário parcialmente destruído após a conclusão do primeiro lançamento realizado pelo agente.

Figura 23 – Inferência 2 no nível 17.



Fonte: Autor.

Algumas zonas foram mantidas em posições de destaque, pois diante da destruição apresentada, a mesma trajetória do primeiro lançamento é efetiva para a conclusão do nível. Nesse caso a inferência apresentou a zona 59 como zona de destaque que atingirá a parte inferior da estrutura remanescente.

O aprendizado da rede pelo método proposto revela nas duas configurações de lançamento apresentados na Figura 22 e Figura 23 que a estratégia aprendida pela agente envolve mirar nos blocos estruturais na região inferior, equivalente a uma das estratégias utilizadas pelo atual campeão.

Para qualificar a eficiência da estrutura de tomada de decisões desenvolvida, os três agentes foram colocados para jogar por 30 vezes o nível 17, registrando o número de pássaros utilizados e pontuação final obtida na conclusão de cada tentativa. A Tabela 3 apresenta o resultado obtido por cada um dos agentes para o teste realizado.

Tabela 3 – Comparação de desempenho dos três agentes no nível 17.

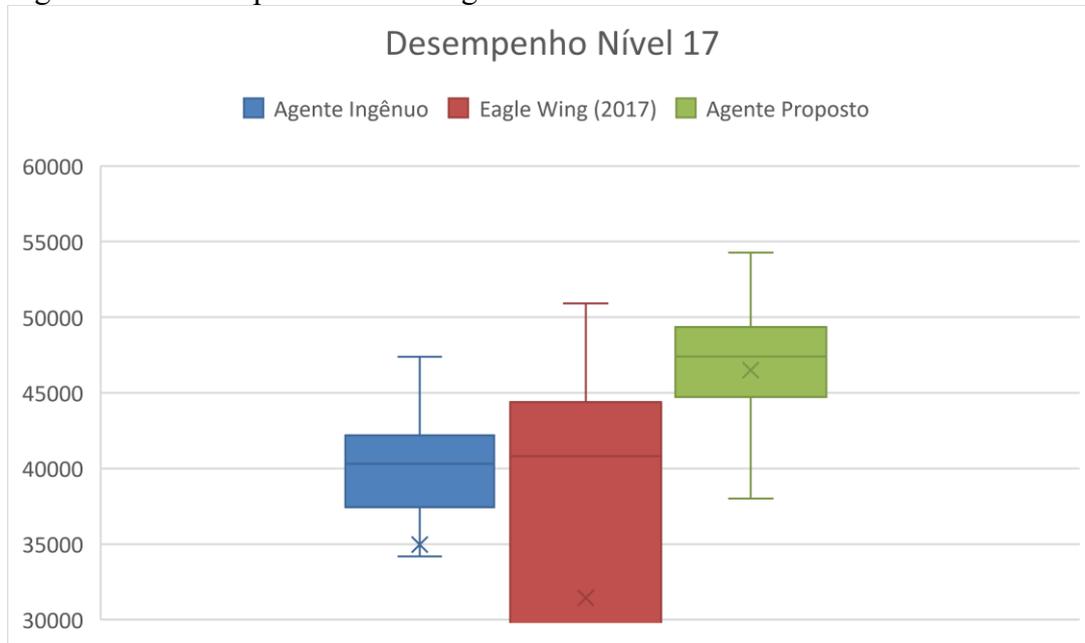
Rodada	Pássaros Utilizados			Pontuação		
	Agente Ingênuo	<i>Eagle Wing</i> (2017)	Agente Proposto	Agente Ingênuo	<i>Eagle Wing</i> (2017)	Agente Proposto
1	3	2	2	38820	46510	49700
2	3	3	3	37140	0	41160
3	3	2	2	40880	50920	51710
4	3	2	3	41920	43290	38000
5	3	3	2	0	38340	48800
6	3	2	3	42240	43700	39660
7	3	2	2	42080	45910	54270
8	3	3	2	37320	39950	46320
9	2	3	3	47030	0	39080
10	3	3	3	38670	0	40020
11	3	3	2	40310	41000	49230
12	3	2	2	40580	45540	51370
13	3	3	2	0	0	48760
14	3	3	2	41690	40620	48860
15	3	3	3	0	39650	37060
16	3	3	2	40320	38500	46690
17	2	3	2	46280	36820	46630
18	2	3	2	52540	43130	46220
19	3	2	2	38500	44460	49570
20	3	3	2	37840	0	49400
21	2	3	2	46000	42070	43540
22	3	2	2	39370	45390	48140
23	3	2	2	34190	47620	49780
24	3	2	2	40160	45240	46870
25	2	3	2	52680	41620	44500
26	3	3	2	41320	0	50860
27	3	3	2	44000	0	46110
28	3	3	2	0	38720	45410
29	3	3	2	0	0	49040
30	2	2	2	47390	44150	47950
Média	2,80	2,63	2,20	34976	31438	46490
			Desvio Padrão	16437	19511	4289

Fonte: Autor.

A tabela apresenta as média e desvio padrão das pontuações obtidas para cada um dos agentes submetidos ao teste. Os valores que apresentam pontuação igual a zero caracterizam a não conclusão do nível, ou seja, os alvos não foram completamente eliminados apesar da utilização de todos os pássaros disponíveis.

A Figura 24 apresenta graficamente o desempenho de cada agente avaliado no Nível 17.

Figura 24 – Desempenho dos três agentes avaliados no Nível 17.



Fonte: Autor.

Os resultados mostram que a média apresentada pelo agente proposto é superior aos dois agentes comparados, permitindo concluir que as estratégias de utilizar a estrutura como alvo dos lançamentos é melhor para este nível.

A análise do desvio padrão indica que mesmo considerando o pior resultado do agente proposto, seu desempenho se mostrará melhor do que os dois agentes comparados.

Outro fator positivo do agente proposto está na eficiência de conclusão do nível treinado, visto que concluiu o nível em 100% das vezes em que foi submetido, enquanto o Agente Ingênuo e *Eagle Wing* (2017) apresentaram percentuais de conclusão de 83,5% e 73,5% respectivamente.

4.1.2 Nível 10

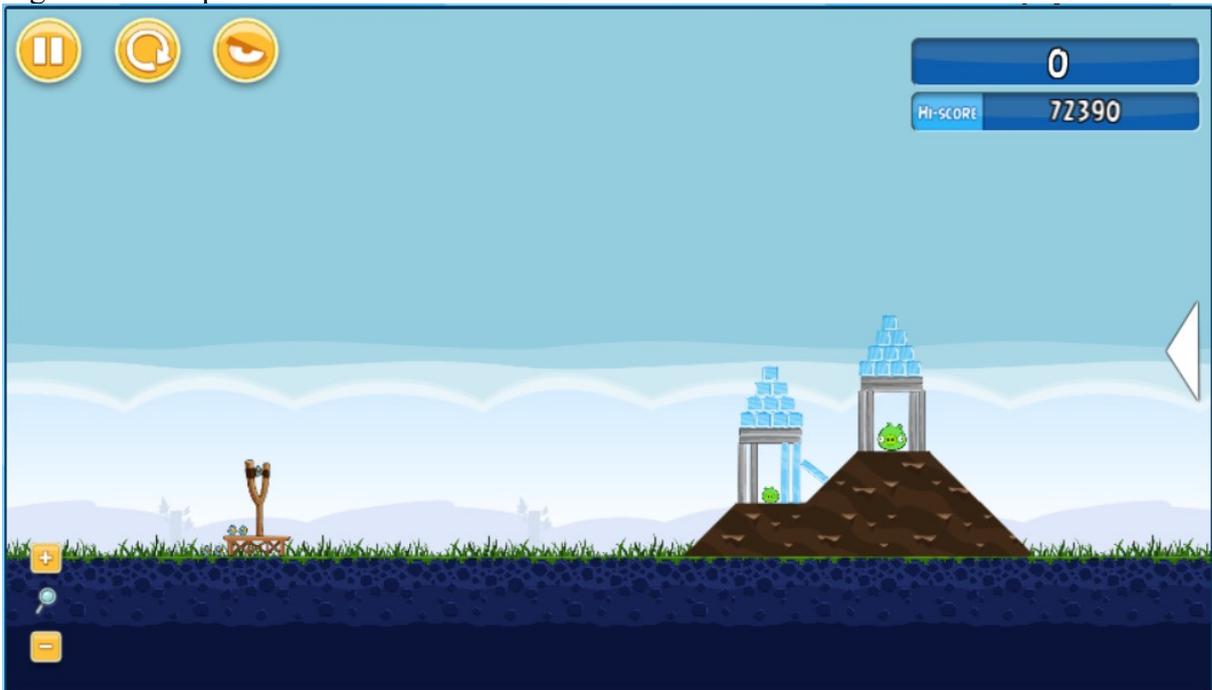
O segundo cenário para avaliar a capacidade de reconhecimento e representação de padrões da rede foi o nível 10. Esse nível possui apenas um tipo de pássaro para lançamento e uma combinação de dois tipos de materiais na composição da estrutura física.

O nível 10 foi selecionado como cenário de teste das capacidades da rede neural por envolver uma estratégia conclusão mais complexa em relação ao nível 17. O Agente Ingênuo

apresentou dificuldade para solucioná-lo, e só foi capaz de concluí-lo utilizando todos os pássaros disponíveis, enquanto o atual campeão sequer foi capaz de concluí-lo.

A configuração inicial do nível 10 está representada na Figura 25, com 5 pássaros azuis disponíveis para eliminação de dois porcos protegidos por duas estruturas compostas por blocos de gelo e concreto.

Figura 25 – Arquitetura do nível 10.



Fonte: Autor.

A particularidade do nível está nos apoios de gelo na parte posterior da primeira estrutura. A melhor estratégia para esse nível é realizar o primeiro lançamento entre as duas estruturas, garantindo que os apoios de gelo mencionados sejam destruídos. Em seguida, atingir frontalmente as estruturas garantindo seu desmoronamento e a eliminação dos alvos.

O Agente Ingênuo é capaz de solucionar esse nível quando o alvo do primeiro lançamento é o porco mais distante do estilingue, garantindo a destruição involuntária dos apoios de gelo da estrutura mais próxima.

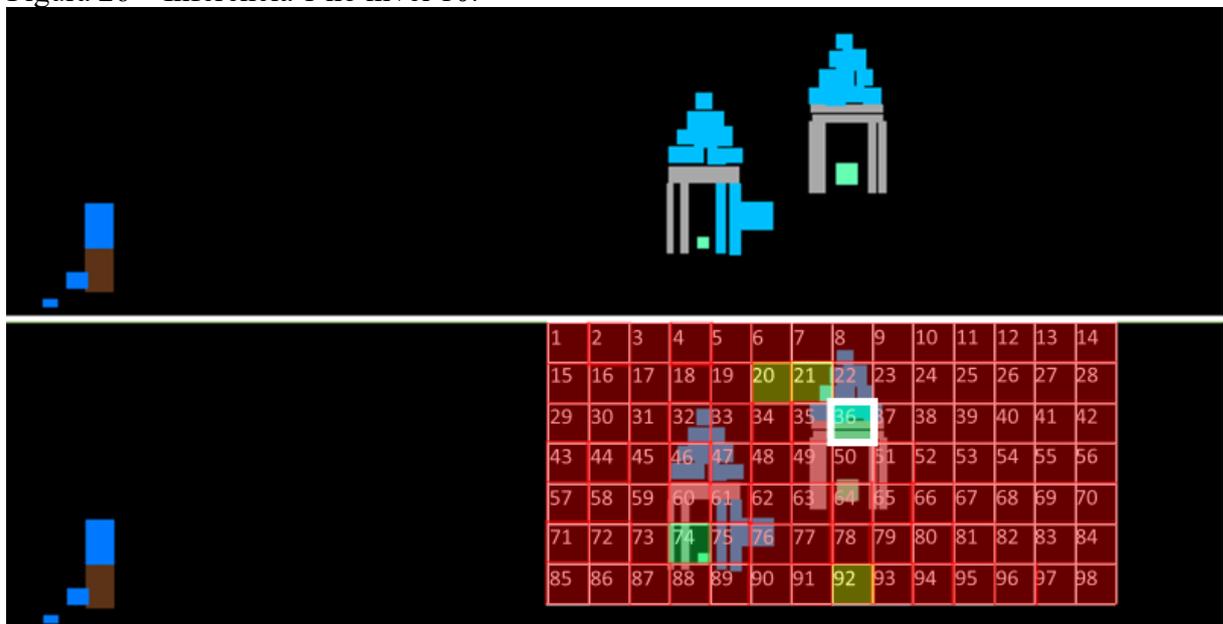
Por outro lado, o agente *Eagle Wing* utiliza a estratégia de destruição de estruturas e acaba utilizando os blocos de concreto na vertical da primeira estrutura como alvo para todos os lançamentos disponíveis, impedindo que o nível seja concluído.

O agente proposto foi treinado nesse nível com 4.000 dados de treinamento contendo as experiências de todas as configurações de cenário possíveis, realizando lançamentos aleatórios nas 98 zonas disponíveis e registrando a pontuação obtida.

Para avaliar a capacidade de representação da rede treinada, foram aplicados diferentes padrões de entrada e avaliadas as respectivas respostas obtidas dessa inferência. A avaliação da resposta da rede ao teste foi baseada no conhecimento humano aprendido sobre o nível durante a exploração experimental do domínio.

O primeiro padrão de entrada aplicado à rede foi o vetor de representação da condição inicial do nível. A figura 26 apresenta na parte superior o padrão de entrada aplicado em caráter de teste da rede treinada e a representação de pertinência de cada uma das 98 saídas para essa entrada na parte inferior.

Figura 26 – Inferência 1 no nível 10.



Fonte: Autor.

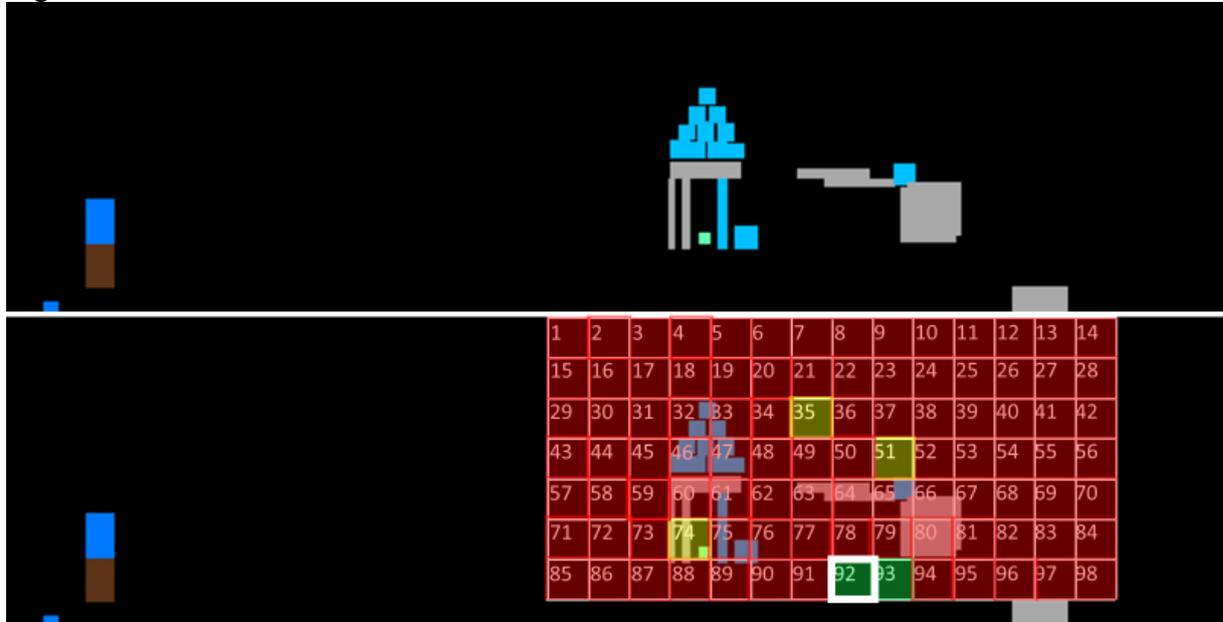
A análise da Figura 26 revela que a melhor estratégia aprendida para a situação inicial é atingir a aresta superior da estrutura mais distante do estilingue, destacado pela zona 36, considerada a melhor escolha visto a possibilidade de demolição dessa estrutura mais distante, eliminando o alvo protegido por ela e destruindo os apoios na parte frontal da estrutura mais próxima. Essa conduta contém a possibilidade de concluir o nível com apenas um lançamento.

Percebeu-se durante esse teste que a zona 74 foi considerada a segunda zona mais pertinente, permitindo concluir que a rede treinada não conseguiu relacionar a vantagem de seleção dessa zona com a ausência do apoio na parte frontal da primeira estrutura, apesar da preferência pela escolha da zona 36 no primeiro lançamento sempre prevalecer.

A segunda condição de teste aplicada à rede está apresentada na Figura 27, aplicado o padrão de entrada representado a parte superior, contendo a estrutura parcialmente

comprometida, com apenas um alvo remanescente, e a representação das saídas apresentadas como resposta pela rede na parte inferior.

Figura 27 – Inferência 2 no nível 17.



Fonte: Autor.

Apesar de algumas zonas serem mantidas em posições de destaque, o comprometimento estrutural e a posição do único alvo remanescente tornam regiões na parte inferior mais vantajosas, visto que sua trajetória resulta na destruição da primeira estrutura e conclusão do nível.

Os três agentes foram colocados para jogar por 10 vezes o nível 10, registrando o número de pássaros utilizados e pontuação final obtida na conclusão de cada tentativa. A Tabela 4 apresenta o resultado obtido por cada um dos agentes para o teste realizado.

Tabela 4 – Comparação de desempenho dos três agentes no nível 10.

Rodada	Pássaros Utilizados			Pontuação		
	Agente Ingênuo	<i>Eagle Wing</i> (2017)	Agente Proposto	Agente Ingênuo	<i>Eagle Wing</i> (2017)	Agente Proposto
1	4	5	2	43980	0	56410
2	4	5	4	46440	0	34580
3	5	5	2	32780	0	59120
4	5	5	2	0	0	54700
5	5	5	2	0	0	58380
6	4	5	2	40550	0	56870
7	5	5	5	0	0	0
8	5	5	1	32880	0	68370
9	5	5	2	0	0	53780
10	5	5	5	0	0	0
11	4	5	2	43700	0	56800
12	5	5	2	0	0	63940
13	5	5	5	0	0	0
14	5	5	3	0	0	59450
15	3	5	2	48930	0	64170
16	4	5	5	43990	0	0
17	5	5	3	0	0	61180
18	5	5	5	0	0	0
19	5	5	5	37120	0	0
20	5	5	2	0	0	60530
21	5	5	5	0	0	0
22	5	5	2	0	0	56620
23	5	5	5	0	0	0
24	5	5	3	0	0	56510
25	5	5	5	0	0	0
26	5	5	5	0	0	0
27	5	5	5	0	0	0
28	4	5	2	43970	0	59170
29	5	5	2	0	0	63370
30	5	5	4	0	0	43390
Média	4,73	5,00	3,30	13811	0	36245
			Desvio Padrão	19764	0	28188

Fonte: Autor.

A tabela apresenta as média e desvio padrão das pontuações obtidas. Os valores que apresentam pontuação igual a zero caracterizam a não conclusão do nível, ou seja, os alvos não foram completamente eliminados apesar da utilização de todos os pássaros disponíveis.

A Figura 28 apresenta graficamente o desempenho de cada agente avaliado para no Nível 10.

Figura 28 – Desempenho dos três agentes avaliados no Nível 10.



Fonte: Autor.

Os resultados mostram que média obtida pelo agente proposto é consideravelmente superior do que a média apresentada pelo Agente Ingênuo, enquanto o código do atual campeão disponibilizado pelos organizadores da competição no site oficial foi incapaz de concluir o nível.

O agente treinado apresentou 63,5% de eficiência de conclusão do nível treinado, enquanto o Agente Ingênuo e *Eagle Wing* (2017) apresentaram percentuais de conclusão de 33,5% e 0% respectivamente.

Na seção 4.2 está descrita a metodologia e resultados para o processo de treinamento final da Rede Neural proposta. A rede foi treinada com um conjunto de dados de treinamento coletados em 9 níveis distintos e o agente foi colocado para jogar tanto nos níveis conhecidos quanto em outros 12 níveis desconhecidos.

4.2 TREINAMENTO FINAL

O treinamento final da rede demandou a coleta de em média 4.000 exemplos de treinamento de cada um dos 9 níveis selecionados, totalizando aproximadamente 36.000 dados exemplos para treinar a rede.

Os dados foram extraídos dos níveis ímpares de 3 a 17 mais o nível 10 da primeira parte do Episódio *Poached Eggs*, garantindo ao conjunto de dados de treinamento a inúmeras combinações distintas da quantidade e natureza dos blocos, número de porcos, organização física dos objetos, quantidade e natureza dos pássaros disponíveis.

O método de avaliação de desempenho do agente proposto foi realizado através da comparação com os agentes Ingênuo e *Eagle Wing* (2017). Os três agentes foram colocados para jogar os 21 níveis disponíveis do episódio *Poached Eggs* e foram avaliados pelo número de tentativas para conclusão cada nível, o número de pássaros utilizados e a pontuação obtida.

O agente jogou cada nível o suficiente para concluí-lo a primeira vez, portanto, o desempenho do agente pode apresentar pequenas variações caso seja submetido para atuar novamente nos níveis testados.

A fim de simular as condições estabelecidas pela competição oficial, cada agente foi submetido a jogar no máximo 5 vezes na tentativa de solucionar cada um dos 21 níveis, logo, caso o agente não conseguisse atingir seu objetivo dentro desse intervalo pré-definido, sua pontuação foi considerada igual a zero.

A Tabela 5 apresenta os resultados obtidos pelo teste descrito, contendo o desempenho dos três agente em questão, nos três itens avaliados para os 21 níveis da primeira parte do episódio *Poached Eggs*.

Tabela 5 – Desempenho dos agentes na conclusão na primeira parte do Episódio *Poached Eggs*.

Nível	Pássaros Disponíveis	Pássaros Utilizados			Número de Tentativas			Pontuação		
		Agente Ingênuo	<i>Eagle Wing</i> (2017)	Agente Proposto	Agente Ingênuo	<i>Eagle Wing</i> (2017)	Agente Proposto	Agente Ingênuo	<i>Eagle Wing</i> (2017)	Agente Proposto
1	3	1	1	1	1	1	1	29380	28620	28860
2	5	4	2	3	1	1	1	35120	52230	43420
3	4	1	2	1	1	1	1	40560	30910	41480
4	4	2	2	3	1	1	1	29160	27930	18790
5	4	1	1	1	1	1	2	66770	62620	67950
6	4	2	2	4	1	1	3	33830	33360	18770
7	4	4	4	1	2	1	1	24890	21600	45620
8	4	2	2	2	1	1	2	47900	45470	48680
9	4	4	2	3	1	1	1	26240	42350	36430
10	5	5	5	3	2	5	3	34420	0	52760
11	4	2	2	1	1	1	1	46020	50390	60210
12	4	1	3	1	1	1	1	56470	45460	63990
13	4	4	2	4	1	1	5	26610	45130	0
14	4	3	3	4	1	1	1	57980	55640	55270
15	4	3	4	3	1	1	1	38230	32520	39200
16	5	2	4	2	1	3	1	61760	56120	63740
17	3	3	2	3	2	1	1	42080	44620	39540
18	5	4	4	4	1	3	2	46940	46210	44570
19	4	2	3	3	1	1	1	37760	31640	27820
20	5	4	5	5	3	5	5	51050	0	0
21	8	8	8	8	4	1	1	55080	57850	57400
Total								888250	810670	854500

Fonte: Autor.

Os valores destacados em verde demonstram os níveis em que o agente proposto obteve a maior pontuação entre os três agentes analisados, e os valores destacados na cor amarela configuram os níveis em que a pontuação obtida superou apenas o desempenho de um dos agentes.

Observando a soma da pontuação obtida de cada agente, percebe-se que o agente proposto apresentou desempenho consideravelmente superior ao atual campeão da competição, e que o Agente Ingênuo apresentou o melhor resultado no teste.

Está disponibilizado no site oficial da competição uma referência dos resultados obtidos pelos 57 agentes que já participaram da competição, quando submetidos aos 21 níveis iniciais do episódio *Poached Eggs*.

A análise comparativa entre o agente proposto e os demais participantes revela que, apesar de se tratar de uma abordagem nova e pouco explorada, o resultado obtido ocupa a 35ª

colocação entre 57 participantes desta lista de referência, apresentando resultado equivalente a agentes que já chegaram ao pódio de competições passadas.

A partir da análise das pontuações obtidas, podemos concluir que de fato, o agente proposto apresentou desempenho compatível com os dois agentes comparados.

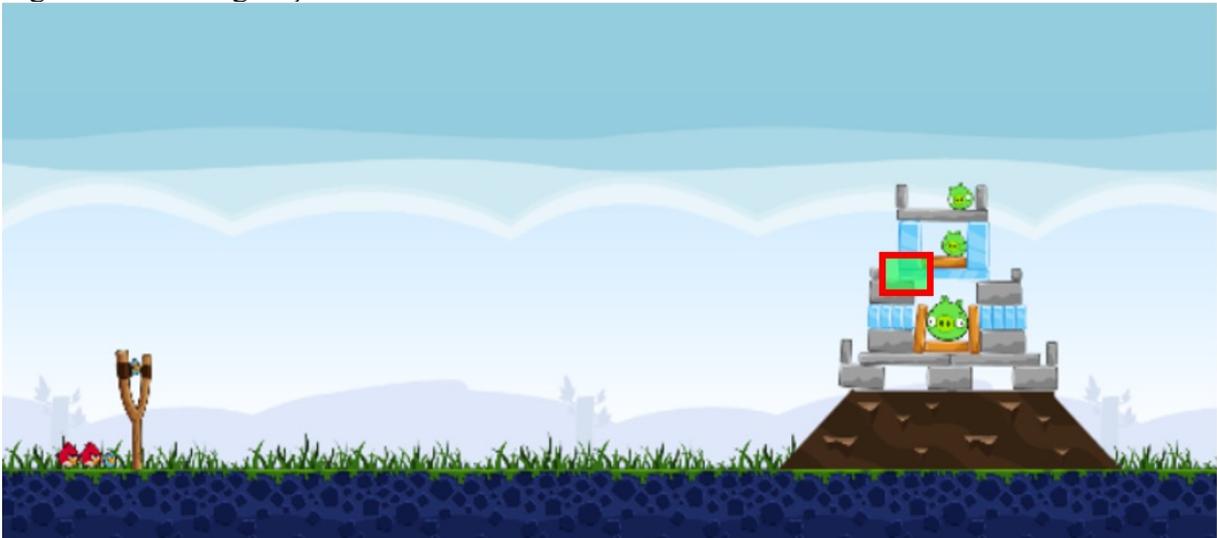
Entretanto, percebeu-se que apesar de obter boas pontuações na conclusão dos níveis em que foi submetido, o agente demandou mais tentativas para concluir diversos deles, configurando uma característica indesejada no cenário da competição, onde cada agente possui intervalo de tempo pré-definido de aproximadamente 3 minutos para cada nível proposto.

O desempenho do agente proposto nos níveis desconhecidos foi extremamente satisfatório, pois apesar de apresentar algumas pontuações baixas nos níveis iniciais, percebeu-se que em geral atingiu ótimas pontuações, e as condutas adotadas em cada situação de lançamento foram fiéis ao esperado para conclusão do nível.

O agente foi incapaz de concluir os níveis 13 e 20, entretanto, foi possível identificar duas limitações da solução proposta nesses dois níveis.

O nível 13 possui uma estrutura formada por blocos de gelo e de concreto protegendo um total de 3 porcos. A melhor conduta neste nível é atingir a base da estrutura de gelo intermediária sob o porco maior, representada pela Figura 29.

Figura 29 – Configuração do Nível 13.



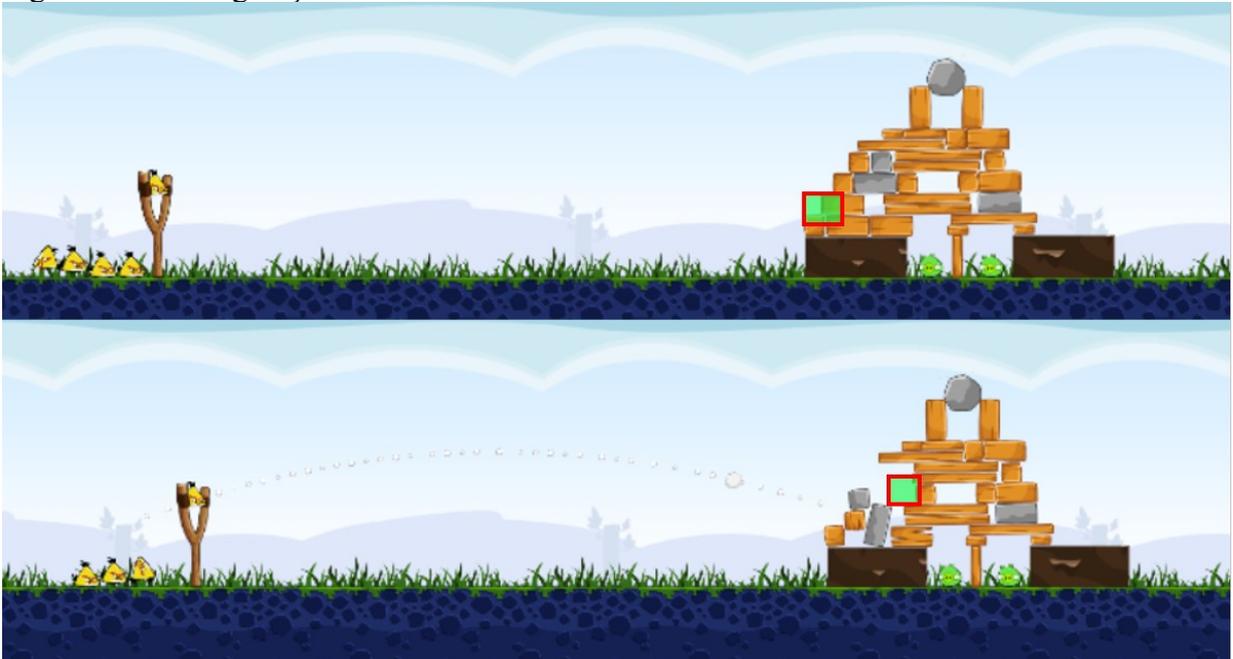
Fonte: Autor.

O agente identificou a região mais promissora de ser alvejada, porém, os lançamentos não foram capazes de atingir a placa de gelo desejada, pois os tiros baixos sempre atingem primeiro os blocos de concreto, mantendo a estrutura intacta.

Neste caso, as dimensões das zonas de lançamento utilizadas impossibilitam que a região desejada seja atingida precisamente por um tiro baixo, portanto, acredita-se que o aumento a resolução da grade de zonas disponíveis ou atribuir ao agente a capacidade de realizar tiros altos podem solucionar a ineficiência de desempenho identificada neste nível.

O nível 20 foi o outro nível não concluído, que possui uma configuração estrutural essencialmente de madeira com alguns blocos de concreto posicionados estrategicamente, fazendo com que as melhores escolhas de lançamento sejam atingir a princípio os blocos de madeira abaixo dos blocos de concreto, e em seguida atingir a base da estrutura de madeira que protege os porcos com um lançamento alto. Essa configuração está representada na Figura 30.

Figura 30 – Configuração do Nível 20.



Fonte: Autor.

Neste caso, o agente foi capaz de identificar as regiões de lançamento mais promissoras, porém a ausência do recurso para realizar lançamentos altos impossibilita que a zona desejada seja atingida de forma que favoreça a conclusão do nível

O capítulo a seguir apresenta as conclusões relacionadas ao desempenho e limitações do trabalho desenvolvido, assim como os potenciais trabalhos futuros pretendidos.

5 CONCLUSÃO

Este trabalho propôs a modelagem de uma Rede Neural Artificial para ser utilizada como ferramenta de tomada de decisões no processo de seleção do alvo de um lançamento no domínio do jogo *Angry Birds*. A estrutura de seleção de alvos recebe como dado de entrada uma imagem de representação atual do cenário e apresenta como saídas a pertinência de seleção de cada uma das 98 zonas delimitadas como alvo desse lançamento.

Dois pontos foram analisados para qualificar o desempenho da solução apresentada: a capacidade de aprendizado e representação de padrões da Rede Neural Artificial e sua contribuição na melhoria de desempenho quando utilizada por um agente autônomo inteligente.

Dois experimentos foram realizados. O primeiro experimento consistiu em identificar dois níveis de elevada complexidade de solução para que fosse possível avaliar a real capacidade de aprendizado da RNA desenvolvida. O segundo experimento nos mostrou a capacidade de generalização do aprendizado da rede, condicionando o agente treinado a jogar níveis jamais vistos nos dados de treinamento.

Os resultados para as duas etapas do primeiro experimento revelaram que a arquitetura da rede é suficientemente capaz de aprender a representação de padrões apresentadas pelos dados de treinamento, além de permitir estimar a quantidade média de 4.000 dados de treinamento para que seja possível ao agente treinado apresentar desempenho superior aos agentes comparados, tanto em relação ao baixo índice de falhas apresentado quanto na pontuação média obtida.

O segundo experimento confirmou a capacidade de generalização de aprendizado da rede proposta, visto que o agente treinado foi capaz de concluir 90,5% dos níveis que jogou, além de apresentar desempenho superior ao atual campeão em alguns níveis em que foi submetido. Esses níveis, em linhas gerais apresentam como melhor estratégia atingir estruturas para a eliminação em massa dos alvos do cenário.

O agente falhou na solução de dois níveis, dentre eles um desconhecido, e outro presente nos dados de treinamento. Nos dois casos, as melhores condutas exigiam do agente a capacidade percepção e realização lançamentos altos para a conclusão dos objetivos propostos pelo nível.

Observou-se a necessidade de jogar diversos níveis por mais de uma vez para que fossem concluídos, o que é indesejável nas configurações da competição oficial. Esses resultados, somados às observações realizadas durante os testes do agente revelaram que a dimensão e delimitações das zonas utilizadas neste trabalho causam imprecisão nos lançamentos realizados. Por outro lado, melhorar a precisão do agente significa delimitar a

região de lançamento em um número maior zonas de menor área, o que conseqüentemente aumentará a demanda computacional para o treinamento e processamento de dados pela rede, inviabilizando sua participação nos moldes da competição oficial.

Finalmente, ao observar as pontuações alcançadas pelos três agentes em comparação, percebe-se que o Agente Ingênuo foi o melhor, o que pode ser justificado pelo fato de os níveis de teste pertencerem ao primeiro episódio do jogo, composto por níveis de baixa complexidade na solução.

O fato de a solução proposta superar em alguns pontos o desempenho do atual campeão da competição oficial não permite considera-lo melhor, visto que o atual campeão foi treinado com dados dos 42 níveis iniciais do episódio *Poached Eggs*, enquanto os agente proposto foi treinado em apenas 9 dos 21 níveis iniciais, ou seja, o desempenho do atual campeão em níveis mais complexos será provavelmente melhor.

O agente desenvolvido foi submetido para participar da competição anual de Angry Birds de 2019, ocorrida durante o IJCAI em Macau, China, para avaliação de seu desempenho nos moldes oficiais da competição.

A competição de 2019 teve 7 times participantes, dentre eles: os agentes *IHSEV* e *AngryHex*, participantes de todas as competições desde o ano de 2013, o campeão da competição em 2016 *BamBirds*, o campeão da competição em 2017 e 2018 *Eagle Wing*, e três agentes novatos: *Shiro*, *SimbaDD* e o agente proposto neste trabalho *MMC*.

Os agentes foram submetidos a solucionar 8 níveis criados pelos organizadores da competição em no máximo 30 minutos, com autonomia de seleção da estratégia para a solução dos níveis apresentados.

O agente proposto terminou a fase qualificatória na sétima colocação, com desempenho pouco promissor. Era esperado que a oitava edição da competição apresentasse níveis mais complexos de serem solucionados, e adversários bastante desenvolvidos no cenário do domínio, o que culminasse em um desempenho pouco satisfatório.

Entretanto, a solução apresentada tem potencial para ser explorada a fim de ter seu desempenho melhorado nas próximas competições, visto a possibilidade de cada agente ser rodado exclusivamente em um *notebook* de alto poder computacional.

5.1 TRABALHOS FUTUROS

O desenvolvimento desse trabalho gerou discussões sobre potenciais trabalhos futuros, a fim de corrigir eventuais ineficiências observadas nos testes realizados.

A partir da coleta de mais dados de treinamento em níveis de maior complexidade, podemos testar a potencial melhoria de desempenho do agente proposto em níveis mais complexos.

Melhorar a resolução dos dados de entrada ou aumentar o número de zonas de lançamento de menor área útil podem gerar um agente com maior precisão na seleção dos alvos do lançamento, apesar de consequentemente aumentar a demanda computacional operacional.

Finalmente, adicionar à estrutura de aprendizado desenvolvida a capacidade de avaliar e escolher realizar lançamentos baixos ou altos para atingir o alvo desejado pode gerar um impacto bastante positivo no desempenho do agente.

REFERÊNCIAS

_____. **Site Oficial da Angry Birds AI Competition**. 2019 [Online]. Disponível em: <<http://aibirds.org>>. Acesso em: 16 jun. 2019.

_____. **Site Oficial da Angry Birds AI Competition**. Referência de Pontuações. 2017 [Online]. Disponível em: <<http://aibirds.org/benchmarks.html>>. Acesso em: 26 jun. 2019.

BEZDEK, J.C. On the relationship between neural networks, pattern recognition and intelligence., v.6, n. 2, p. 85 – 107, fev. 1992.

BOROVÍČKA, T.; ŠPETLÍK, R.; RYMEŠ, K. DataLab Birds Angry Birds AI. 2014. Disponível em: <<https://aibirds.org/2014-papers/datalabbirds.pdf>>. Acesso em: 9 mar. 2018.

CALIMERI, F. *et al.* AngryHEX: An artificial player for Angry Birds based on declarative knowledge bases. **IEEE Transactions on Computational Intelligence and AI in Games**, v. 8, n. 2, p. 128-139, jun. 2016.

DASGUPTA, S.; VAGHELA, S.; MODI, V.; KANAKIA, H. s-Birds avengers: A dynamic heuristic engine-based agent for the Angry Birds problem. **IEEE Transactions on Computational Intelligence and AI in Games**, v. 8, n. 2, p. 140-151, jun. 2016.

DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. **Journal of Machine Learning Research**. P. 2121–2159. jul. 2011.

FERREIRA, L. A.; LOPES, G. A. W.; SANTOS, P. E. Combining qualitative spatial representation utility function and decision making under uncertainty on the Angry Birds domain. 2013. Disponível em: <<https://aibirds.org/2013-Papers/Symposium/fei2.pdf>>. Acesso em: 9 mar. 2018.

FERREIRA, L. N.; TOLEDO, C. Tanager: A generator of feasible and engaging levels for Angry Birds. **IEEE Transactions on Computational Intelligence and AI in Games**. v. 14, n. 8, p. 1-14, out. 2016.

GE, X.; GOULD, S.; RENZ, J.; ABEYASINGHE, S.; KEYS, J.; WANG, A.; ZHANG, P. Angry Birds basic game playing software v1.32. 2014. Disponível em: <<https://aibirds.org/basic-game-playing-software.html>>. Acesso em: 4 jun. 2018.

GE, X.; GOULD, S.; RENZ, J.; ABEYASINGHE, S.; KEYS, J.; WANG, A.; ZHANG, P. Server/Client Protocol. Disponível em: <<https://aibirds.org/basic-game-playing-software.html>>. Acesso em: 4 jun. 2018.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. 2nd. ed. Prentice Hall PTR Upper Saddle River, NJ, USA. 1998. ISBN 0132733501.

HEBB, D. O. The organization of behavior: A neuropsychological theory. New York: John Wiley and Sons Inc. 1949.

HINGSTON, P. A Turing test for computer game bots. **IEEE Transaction on Computational Intelligence and AI Games**. v. 1, n. 3, p. 169-186, set. 2009.

HOPFIELD, J. J. Neural Networks and Physical Systems with Emergence Collective Computational Abilities. **Proceedings of the National Academy of Sciences of the United States of America**. v. 79, p. 2554-2558, abr. 1982.

JIANG, Y.; HARADA, T.; THAWONMAS, R. Procedural generation of Angry Birds fun levels using pattern-struct and preset-model. **IEEE Conference on Computational Intelligence and Games**. p. 154-161, ago. 2017.

KAMINKA, G. A.; SCHAFFER, F.; SOLLITTO, C.; ADOBBATI, R.; MARSHAL, A. N.; SCHOLER, A. S.; TEJADA, S. Gamebots: a flexible testbed for multiagent team research. *Communications of the ACM*, v. 45(1), jan. 2002

LAIRD, J. E. Research in human-level ai using computer games. **Communications of the ACM**, v.45(1), p. 32–35, jan. 2002.

LAIRD, J. E.; LENT, M. Intelligent agents in computer games. **In Proceedings of the National Conference on Artificial Intelligence**, p. 929–930. AAAI Press, jul. 1999.

LOIACONO, D.; LANZI, P. L.; TOGELIUS, J.; ONIEVA, E.; PELTA, D. A.; BUTZ, M. V.; LONNEKER, T. D.; CARDAMONE, L.; PEREZ, D.; SAEZ, Y.; PREUSS, M.; QUADFLIEG, J. The 2009 simulated car racing championship. **IEEE Transaction on Computational Intelligence and AI Games**. v. 2, n. 2, p. 131-147, jun. 2009.

MCCULLOCH, W. S.; PITTS, W.H. A logical calculus of the ideas immanente in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 115-133, 1943.

MINSKY, M.; PAPERT, S. A. Perceptrons: An introduction to Computational Geometry. Cambridge, MA, USA: MIT Press, 1969.

MITCHELL, T. M. **Machine Learning**. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072.

ONTANON, S.; SYNNAEVE, G.; URIARTE, F.; RICHOUX, D.; CHURCHILL, D.; PREUSS, M. A survey of real-time strategy game AI research and competition in StarCraft. **IEEE Transaction on Computational Intelligence and AI Games**. v. 5, n. 4, p. 293-311, dez. 2013.

POLCEANU, M. et al. Towards a Theory-Of-Mind-Inspired Generic Decision-Making Framework. **IJCAI 2013 Symposium on AI in Angry Birds**. 2014

ROBIČ, D.; REPNIK, R. Computer game Angry Birds in teaching physics. **International Convention on Information and Communication Technology, Electronics and Microelectronics**, v. 38, p. 680-684, jul. 2015.

ROSENBLATT, F. The Perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 8, 1958.

RUMELHART, D. E. HINTON, G. E. WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**. v. 323, p. 533-536. out. 1986

SCHIFFER, S.; JOURENKO, M.; LAKEMEYER, G. Akbaba – An agent for the Angry Birds challenge based on search and simulation. **IEEE Transactions on Computational Intelligence and AI in Games**. v. 8, n. 2, p 116-127, jun. 2016

SHAKER, N.; TOGELIUS, J.; NELSON, M. J. **Procedural Content Generation in Games: a Textbook and an Overview of Current Research**. Disponível em: <pcgbook.com>. Acessado em: 16 maio 2018.

TOGELIUS, J.; SHAKER, N.; KARAKOVSKIY, S.; YANNAKAKIS, N. The Mario AI championship 2009-2012. **AI Magazine**. v. 34, n. 3, p. 89-92, 2013.

TSITSIKLIS, J. N.; VAN ROY, B. An analysis of temporal-difference learning with function approximation. **IEEE Transaction on Automatic Control**. v. 42, no. 5, p. 674-690. maio 1997.

TZIORTZIOTIS, K. B. N.; PAPAGIANNIS, G.; BLEKAS, K. A Bayesian ensemble regression framework on the Angry Birds game. **IEEE Transactions on Computational Intelligence and AI in Games**, v. 8, n. 2, p. 104-115, jun. 2016.

WALEGA, P.; LECHOWSKY, T.; ZAWIDZKI, M. Qualitative physics in Angry Birds: first results. 2014. Disponível em: <<https://aibirds.org/2014-papers/ImpactVactor-2014.pdf>>. Acesso em: 9 mar. 2018.

WANG, T. J. Description for Eagle's Wing, an AI Angry Bird Playing Agent. 2017. Disponível em: <<https://aibirds.org/2017/eaglewing2017.pdf>>. Acesso em: 9 mar. 2018.

WATKINS, C. J. C. H.; DAYAN, P. Q-learning, Machine Learning, vol. 8, no. 3–4, p. 279-292, maio 1992.

YOON, D.; KIM, K. Challenges and opportunities in game artificial intelligence education using Angry Birds. **IEEE Access**. v. 3, p. 793-804, jun. 2015.

YUAN, Y. *et al.* Enhancing Deep Reinforcement Learning Agent for Angry Birds. 2017. Disponível em: <https://aibirds.org/2017/aibirds_BNU.pdf>. Acesso em: 9 Mar. 2018.

ZEILEIR, M. D. ADADELTA: An Adaptive Learning Rate Method, CoRR, abs/1212.5701. dez. 2012

ZHANG, P.; RENZ, J. Qualitative spatial representation and reasoning in Angry Birds: The extended rectangle algebra. **Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning**. v. 14, p. 378-387, jul. 2014.