

Utilização de rede neural artificial otimizada no aprendizado de máquina em jogos digitais

Renan Cunha¹ Luan C. Casagrande¹ Yuri Crotti² Roderval Marcelino² Vilson Gruber¹

¹Universidade Federal de Santa Catarina, Laboratório de Telecomunicações, Brasil ²Universidade Federal de Santa Catarina, Laboratório de Pesquisa Aplicada, Brasil

Abstract

In recent years, a lot of research has been done about the use of artificial intelligence in games. It has been proven that neural network, for example, can be used in a huge number of possibilities in games. Considering this fact, the authors proposed a novel methodology for training neural networks as airplane controller. Our effort is focused to optimize the result using genetic algorithm to find the best topology and parameters for our problem. The proposed system was modeled, tested and validated under a simulation platform using flight simulator FlightGear.

Palavras-chave: Rede Neural Artificial, Algoritmo Genético, FlightGear.

Informações para Contato: {renan1bara,luancasagrande,yuricrotti }@gmail.com

1. Introdução

Nos últimos 20 anos houveram muitas pesquisas e aplicações de soluções baseadas em redes neurais artificiais (RNA) nas áreas de engenharias, finanças, controle e simulação entre outras. Em contrapartida, nesse mesmo período, dentro do mercado de jogos digitais o uso de RNAs não teve toda a sua potencialidade explorada [Charles e McGlinchey 2004].

Porém, de acordo com Bourg e Seemann [2004], esta realidade está mudando. A evolução na capacidade de processamento dos computadores e o surgimento de diversos trabalhos e pesquisas científicas relacionadas ao aprendizado de máquina voltado para jogos acabaram impulsionando o interesse pela área de inteligência artificial (IA) em games, mais especificamente o uso de redes neurais artificiais.

Segundo Macri [2011], tradicionalmente utiliza-se máquinas de estados para se implementar jogadores autônomos (bots). Porém, esta metodologia se torna, de certa maneira, previsível do ponto de vista do jogador real. Há evidências de que é possível complementar este método tradicional baseado em estados, ou até mesmo desenvolver um mecanismo de IA para games, fazendo-se o uso de redes neurais artificiais.

Na maioria das vezes as RNAs são utilizadas em jogos a fim de se criar personagens que agem de maneira autônoma e que consigam apresentar um comportamento similar ao de um jogador real. Em Bourg e Seemann [2004] os autores classificam quatro grandes áreas de aplicação desta técnica de IA dentro de jogos: □ Controle: nesta situação os dados sensoriais de um tanque de guerra, por exemplo, servem como entrada para uma RNA, e a mesma fornece como saída o controle dos motores do mesmo. □ Avaliação de ameaça: utilizado geralmente em jogos de estratégia, neste modelo a RNA, através de poucas informações, consegue inferir e detectar um possível ataque de um oponente. □ Atacar ou fugir: similarmente ao que acontece na situação de controle, a RNA é treinada com experiências prévias de jogadores reais e, a partir deste aprendizado, consegue decidir se em uma dada situação a melhor opção seja atacar ou fugir. □ Antecipação: este é um dos usos mais complexos de redes neurais artificiais em jogos. O mesmo pode ser considerado como uma continuação da avaliação de ameaça. A partir da detecção de um ataque, a RNA consegue montar uma estratégia, em tempo real, de contra-ataque, visando neutralizar o movimento do oponente de maneira eficaz, inteligente e realística.

Portanto, fica evidente que redes neurais artificiais podem ser empregadas de diversas maneiras, para resolver uma vasta gama de problemas de inteligência artificial em jogos digitais. Tendo isto em vista, os autores do presente

trabalho procuraram explorar um mecanismo de otimização topológica de RNAs baseado em computação evolucionária, visando aplicar esta técnica de IA em um game simulador de vôo chamado FlightGear.

Na seção 2 será feito uma revisão de trabalhos relacionados, já na seção 3 os autores demonstram o funcionamento do sistema de otimização. A seção 4 apresenta os resultados obtidos através de testes no simulador. A seção 5 elucida alguns possíveis trabalhos futuros.

2. Trabalhos Relacionados

Existem diversas partes de jogos onde uma RNA pode decidir como um agente vai agir. Na teoria, cada decisão que um jogador possa fazer dentro do jogo, pode ser gerenciada por uma RNA (PETRAKIS; TEFAS, 2010). Embasado neste fato, é possível entender a importância do desenvolvimento de trabalhos baseados em RNAs para o mundo dos games.

Em (VAN HOORN; TOGELIUS; SCHMIDHUBER, 2009), uma RNA foi usada para treinar controle hierárquico para first person shooter (FPS) jogadores autônomos. Três controles para atirar, explorar e seguir caminho foram desenvolvidos individualmente, e então um algoritmo evolucionário foi treinado para decidir quando usar cada um dos controladores. Já em (PETRAKIS; TEFAS, 2010), uma RNA foi usado para treinar o módulo de seleção de armas de bots Unreal Tournament. Os resultados mostraram que usando apenas uma camada oculta na RNA, o desempenho dos jogadores autônomos na escolha das armas aumentou consideravelmente.

Em (VLIEG, 2012), o autor propôs RNA como principal mecanismo para um jogador autônomo de poker. Neste trabalho, o autor concluiu que para alcançar um nível considerável de agente, deve-se levar em conta também a modelagem dos adversários.

Com outra visão de aplicação, e mais próximo do trabalho proposto por estes autores, (TOGELIUS; LUCAS, 2006) propuseram uma RNA com uma camada oculta otimizada para o desenvolvimento de controladores de carros R/C para corrida. Já (ATHANASIADIS; GALANOPOULOS; TEFAS, 2012), também desenvolveu controladores de carros R/C para corrida utilizando RNA, porem evoluiu o treinamento utilizando dados coletados de diferentes bots do jogo The Open Car Simulator.

Com outra visão de aplicação, os autores Furqan et al. (2010) utilizaram o simulador FlightGear para desenvolver controladores de aviões baseados em RNA. Já os autores Esper e Rosa (2015) utilizaram o simulador X-Plane para o desenvolvimento de controladores baseados em RNAs. Cada autor utilizou uma metodologia de modelagem e modelo de avião diferente, atingindo então diferentes resultados. Fundamentado nos trabalhos apresentados, concluiu-se então que RNA e métodos de otimização vem sendo amplamente utilizada em diversas aplicações na área de Games. Percebe-se também que diferentes autores propuseram a utilização de RNA para jogos de simulador de voo, como o FlightGear, porem na maioria dos casos a estrutura da RNA não foi otimizada.

3. Funcionamento Sistema de Otimização

De modo geral, o funcionamento do sistema pode ser representado de acordo com a Figura 1, sendo então dividido entre geração dos indivíduos, análise e ordenação do fitness de cada indivíduo, corte, crossover, mutação e análise do melhor sistema através do cálculo do erro médio quadrático.

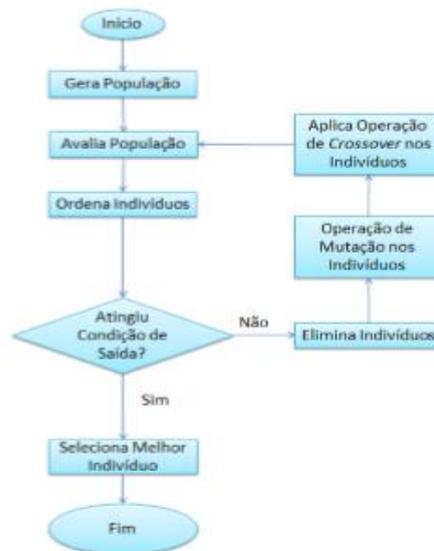


Figura 1: Fluxo do Sistema de Otimização

3.1 Geração dos Indivíduos

Definiu-se para este sistema de otimização que além de randomizar a estrutura em si, ir-se-ia randomizar também as funções de transferência, a função de backpropagation e a taxa de aprendizagem, visando garantir então um sistema ótimo no espaço de busca.

Objetivando a redução do custo computacional, a variação do número de camadas internas da estrutura ficou entre 1 e 10 camadas. Já cada camada pode variar entre 1 e 10 neurônios. Já a função de transferência alternou-se entre função linear, log-sigmoid e hyperbolic tangent sigmoid.

De outro modo, a função de backpropagation foi escolhida aleatoriamente entre uma lista de 10 possibilidades, sendo estas: Levenberg-Marquardt optimization, Bayesian Regularization Backpropagation, BFGS quase-Newton method, Resilient Backpropagation, Scaled conjugate gradient backpropagation, Conjugate gradient backpropagation with Powell-Beale restarts, Conjugate gradient backpropagation with Fletcher-Reeves updates, Conjugate gradient backpropagation with PolakRibière updates, One-step secant backpropagation, Gradient descent backpropagation.. Já a taxa de aprendizagem foi definida entre o valor mínimo e Máximo (0 e 1).

Após a geração da estrutura e das características necessárias para a RNA, organizam-se todos os dados em uma estrutura dinâmica de cromossomos. Nesta estrutura, cada cromossomo representa uma camada de RNA. O primeiro cromossomo representa a camada de entrada e armazena todos os pesos entre os neurônios de entrada e os neurônios da primeira camada interna. Em seguida, vêm os cromossomos que definem as estruturas das camadas internas, sendo que estes além de armazenarem todos os pesos entre os neurônios da camada em questão para a próxima camada armazenam também os valores de bias da camada anterior e a função de transferência. O último cromossomo representa a saída e armazena os valores de bias da última camada interna, a função de transferência, a taxa de aprendizado da estrutura e a função de backpropagation utilizada.

3.2 Avaliação e Ordenação dos Indivíduos

Após a geração dos Indivíduos, faz-se necessário avaliar a população. Com os dados ordenados através da estrutura já discutida, criam-se as RNAs para avaliar a modelagem realizada. Após avaliar tal modelagem, gera-se um erro médio quadrático que é então utilizado como valor referencia de avaliação para a rede.

Este erro médio quadrático gerado por cada RNA é então utilizado para ordenar as RNAs do melhor caso para o pior caso. Tal ordenação é essencial para os próximos passos de eliminação, mutação e crossover. Além disso, é esta ordenação que define o melhor sistema possível na geração de indivíduos em análise.

3.3 Operações de Eliminação, Mutação e Crossover nos Indivíduos

Primeiro, elimina-se 50% dos indivíduos utilizando o erro médio quadrático da rede para os dados da modelagem. Este processo é essencial para eliminar os indivíduos considerados ruins.

Após, utiliza-se os indivíduos considerados bons para a geração de novos indivíduos evoluídos. Para este processo, começando do primeiro, de dois em dois indivíduos eram selecionados para a geração de um novo evoluído. O corte dos indivíduos selecionados foi realizado exatamente no meio.

Após o corte ser realizado, a primeira metade de um dos indivíduos e a segunda metade do segundo indivíduo são repassados ao novo indivíduo. Novos valores randômicos para os pesos e as bias que ficam no limite de corte são então gerados para o indivíduo em análise.

Seguindo o processo, acontece então a operação de mutação. Neste problema, definiu-se uma taxa de mutação de 40%. Sendo selecionado, o indivíduo a ser modificado tem chances iguais de sofrer modificação na taxa de aprendizagem, função de backpropagation e funções de transferência.

4. Simulação e resultados

Conforme foi discutido em seções anteriores, a rede neural otimizada através de algoritmos genéticos foi utilizada para fazer o controle de altitude de uma aeronave do simulador FlightGear. Para realizar as simulações utilizou-se o software MatLab, sendo que este foi executado em paralelo com o FlightGear. Foi estabelecido uma conexão UDP entre ambos de tal maneira que, através do MatLab, tinha-se um domínio total sobre o jogo, ou seja, era possível obter dados de sensores do avião ao mesmo tempo que também se conseguia enviar comandos e alterar parâmetros do simulador via software.

Visando praticidade e eficiência, foi utilizado à implementação de uma rede neural artificial perceptron multilayer presente na Neural Network Toolbox. Esta escolha deu-se levando em consideração que tal implementação já se encontra em estado muito maduro e consolidado na área de estudos de inteligência artificial.

A RNA responsável por fazer o controle de altitude tinha apenas uma entrada, que era a diferença entre a altitude desejada e altitude atual do avião. Como saída, este sistema de IA fornecia a deflexão do profundor (elevator) e a o percentual de ativação do motor da aeronave. O sistema de otimização descrito na Seção 3 tinha como objetivo encontrar a melhor topologia dentro de um espaço de busca limitado pelos autores.

Para se avaliar o funcionamento deste sistema de controle baseado em aprendizado prévio, foi utilizado um modelo de avião bastante conhecido no FlightGear que é o Rascall 110. Colocou-se tal aeronave a uma altitude de 5000 pés e então ativou-se o controlador configurando o mesmo para atingir uma altitude de 6000 pés. A Figura 2 demonstra o gráfico de tempo versus altitude.

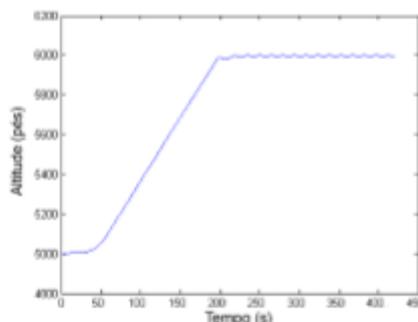


Figura 2: Resultado da simulação

A análise do gráfico disposto na Figura 2 deixa evidente que o resultado obtido foi satisfatório e correspondeu com o esperado. A aeronave subiu de 5000 pés para 6000 pés obedecendo ao comportamento encontrado nos dados de treinamento.

5. Conclusão

A partir do levantamento bibliográfico feito pelos autores deste trabalho ficou evidente que o setor de jogos digitais possui uma infinidade de áreas onde se pode fazer a aplicação de sistemas de inteligência artificial. As RNAs, em particular, são capazes de agregar a capacidade de aprendizagem a um determinado agente autônomo dentro de um jogo. A partir de um processo de aprendizagem utilizando dados prévios, a rede neural consegue fazer futuras inferências, agindo de maneira “racional”.

Os jogos estão em constante evolução, principalmente no quesito gráfico. Porém, visando atender a expectativa dos jogadores, a tendência é que os desenvolvedores venham investir também na questão lógica. Espera-se que os mesmos venham a agregar cada vez mais sistemas de IA a fim de tornar a experiência do usuário mais rica e interativa.

Referências

- CHARLES, D., MCGLINCHEY, S., 2004. “The Past Present and Future of Artificial Neural Networks in Digital Games. In: International Conference on Computer Games: Artificial Intelligence, Design and Education, Microsoft Campus, Reading, 77.
- BOURG, DM., SEEMANN, G., 2004. AI for Game Developers. O'Reilly Media, Inc.
- MACRI, D., 2011. An Introduction to Neural Networks with an Application to Games [online] Bournemouth University. Disponível em: < <https://software.intel.com/en-us/articles/an-introductionto-neural-networks-with-an-application-to-games>> [Acesso em: 25 julho 2016].
- PETRAKIS, Stelios; TEFAS, Anastasios. Neural Networks Training for Weapon Selection in First-Person Shooter Games. Greece, p. 1-10. 2010. Disponível em: <<http://poseidon.csd.auth.gr/papers/PUBLISHED/CONFERENCE/pdf/Petrakis10a.pdf>>. Acesso em: 28 jul. 2016.
- VAN HOORN, N.; TOGELIUS, J.; SCHMIDHUBER, J.; "Hierarchical controller learning in a First-Person Shooter," 2009 IEEE Symposium on Computational Intelligence and Games, Milano, 2009, pp. 294-301. doi: 10.1109/CIG.2009.5286463
- TOGELIUS, J.; LUCAS, S.m.. Evolving robust and specialized car racing skills. 2006 Ieee International Conference On Evolutionary Computation, [s.l.], p.1187-1194, jan. 2006. Institute of Electrical & Electronics