

APRENDIZADO DE MÁQUINA APLICADO A JOGOS DIGITAIS: CRIAÇÃO DE UM AGENTE CAPAZ DE APRENDER COM BASE NA OBSERVAÇÃO.

Daniel Carlos SOUZA de Oliveira

Instituto UFC Virtual - Universidade Federal do Ceará, Fortaleza, CE

daniel.hammerdev@gmail.com

1 INTRODUÇÃO

No desenvolvimento de jogos digitais, tanto em relação a visual quanto em mecânicas de jogo, é nítido nível de realismo e imersão aplicado em boa parte desses jogos. Como explica (MURRAY, 2003), quando estamos imersos em um universo ficcional usamos nossa inteligência mais para reforçar do que para questionar a veracidade da experiência. Para tanto, utilizam-se diversos elementos do jogo para alcançar a suspensão da descrença e manter o jogador focado na tarefa que o jogo exige. Um elemento que não pode ser negligenciado é o comportamento dos agentes do jogo, ou seja, entidades do jogo com capacidade de receber estímulos e agir no ambiente (SILVA, 2005). Por mais que o visual, as animações, a física e os efeitos especiais de um jogo busquem um alto nível de realismo, de nada valem os esforços em realismo se o comportamento dos agentes não for coerente e suas decisões não forem adequadas a situação em que os personagens do jogo estão inseridos.

O comportamento de um agente é determinado por um processo dividido em três etapas: perceber, pensar e agir (RABIN, 2012). No primeiro, o agente coleta informações a respeito do ambiente e da situação a qual esse agente está inserido (entrada). Assim como utilizamos nossos sentidos para perceber o ambiente ao nosso redor, um agente do jogo utiliza sensores com finalidade de simular os sentidos humanos. Uma vez que toda informação necessária para o agente já está disponível na memória do computador, esses sensores servem para filtrar as informações coletadas, permitindo ao agente utilizar apenas aquelas informações que forem lícitas e coerentes; na segunda etapa, o agente processa as informações obtidas e as converte em uma decisão (saída). Esse processamento é feito por uma função $f(X)=Y$, onde X é o conjunto contendo as informações coletadas na etapa anterior e Y é o objeto de saída, a decisão (podendo

ser tanto um único valor numérico ou nominal, quanto um conjunto de valores); agir é a etapa em que o agente põe em prática o que foi decidido na etapa de pensamento (RABIN, 2012).

Para obterem resultados satisfatórios no que diz respeito a comportamento de agentes de jogo, utilizam-se diversas técnicas diferentes de inteligência artificial (IA). Dentre essas técnicas, as Redes Neurais Artificiais (RNAs) sobressaem-se por fornecer a um agente a capacidade de aprender a jogar, adaptar-se a uma situação completamente nova dentro do jogo e imitar características cognitivas que apenas um cérebro humano seria capaz (SATHYANARAYANA, 2014). Uma RNA é uma estrutura montada a partir de nós conectados entre si (neurônios) e organizada em camadas (*layers*). A primeira camada de uma RNA é chamada de Camada de Entrada (*input layer*) e tem por finalidade receber os valores de entrada (X) onde cada neurônio dessa camada é responsável por um único tipo de informação. A última camada, representa as saídas da rede e, conseqüentemente, a saída Y da função. Cada neurônio da camada de saída é responsável por apresentar um tipo de informação diferente que estará associada a uma possível decisão; as demais camadas são chamadas de camadas escondidas (*hidden layers*), ficam entre a *input layer* e a *output layer* e têm por finalidade modificar os valores de entrada convertendo-os em valores de saída ($f(X) = Y$) (SATHYANARAYANA, 2014). As camadas de entrada, escondidas e de saída representam, respectivamente, as etapas da tomada de decisão de um agente do jogo.

Além de ser composta por neurônios, uma RNA também possui conexões direcionadas que ligam dois neurônios, o neurônio de entrada e o neurônio de saída. Cada conexão possui um valor numérico denominado de peso sináptico. O conjunto com os pesos de todas as conexões de uma RNA formam os parâmetros dessa rede e influenciam diretamente sua tomada de decisão. O processo de modificar os parâmetros (pesos) de uma RNA com finalidade de melhorar sua acurácia na tomada de decisão é chamado de ajuste dos pesos. Uma das técnicas utilizada para o ajuste de pesos de uma RNA é a *Backpropagation* (ou retro propagação do erro) oferece meios para ajustar os pesos da rede com base na comparação dos resultados obtidos pela rede com os resultados desejados

(SATHYANARAYANA, 2014). Em outras palavras, se esperávamos que um determinado agente decidisse atacar e a decisão desse agente diverge do esperado, conclui-se que a rede neural não está tomando boas decisões e aplica-se um ajuste nos seus pesos. Esse

processo é repetido exaustivas vezes com diferentes situações até que a rede consiga um desempenho desejável. O ato de testar uma rede e efetuar ajustes repetidamente é chamado de treinamento da rede.

Para uma maior compreensão dos conceitos e aplicação de redes neurais, foi criado o game chamado ANNs Volley, um jogo onde dois adversários se enfrentam numa quadra de vôleibol com regras parecidas com as do esporte original.

Embora o game possua regras simples, uma mecânica simples e uma gama limitada de movimentos, criar um agente capaz de jogá-lo de maneira semelhante a um humano torna-se difícil devido à dinâmica do jogo e da quantidade de possibilidades de interação e combinações de movimentos. O objetivo deste estudo é criar uma IA capaz de aprender a jogar esse jogo com base na observação de um jogador humano e de reproduzir seu estilo de jogo, analisar os resultados obtidos e melhorar a compreensão acerca do aprendizado de máquina e suas possibilidades.

2 MÉTODO

Tendo em vista a complexidade de ações em um jogo competitivo, seria difícil construir uma IA baseada em regras que o pudesse jogar de maneira eficiente. Há muitas variáveis envolvidas e muitas possibilidades de jogadas e ações que um jogador pode realizar que não poderiam ser previstas durante a criação de uma IA.

Nesse caso, a utilização de aprendizado máquinas poderia ser a solução, pois os desenvolvedores não precisariam pensar na estratégia ou nas regras da IA. O próprio agente aprenderia a associar cada situação a uma ação adequada.

Para o treinamento, foi construída uma rede neural do tipo Multilayer Perceptron (MLP) com duas camadas escondidas com oito neurônios em cada, uma camada de entrada com sete neurônios e uma camada de saída com quatro neurônios.

2.1 Camada de Entrada

Para cada neurônio da camada de entrada foi designado um tipo de característica do ambiente.

Tabela 1. Neurônios de entrada e suas funções

Neurônio de Entrada	Característica observada
Py	Componente vertical da posição do agente.*
Bx	Componente horizontal da posição da bola.*
By	Componente vertical da posição da bola.*
Vx	Componente horizontal da velocidade da bola.*
Vy	Componente vertical da velocidade da bola.*
Ex	Posição do adversário.*
Fx	Posição estimada de queda da bola.**

* Valores normalizados com base nas dimensões da quadra onde 0.0 é a posição da rede, 1.0 é o limite da quadra do agente e -1.0 é o limite da quadra do adversário.

** Valor calculado com base na extrapolação da posição da bola levando em consideração sua velocidade e a aceleração da gravidade.

Cada uma das características do ambiente da Tabela 1 representa uma característica que os jogadores humanos observam na hora de tomar uma decisão.

2.2 Camada de Saída

Para cada neurônio da camada de saída, uma ação dentro do jogo.

Tabela 1. Neurônios de saída e suas funções

Neurônio de Saída	Característica observada
PosX	Posição onde o agente se encontra (ou deveria se encontrar)*
Jump	Ação de saltar. Varia de 0.0 a 1.0
Strike	Ação de saque. Varia de 0.0 a 1.0
Upper	Ação de levantar a bola. Varia de 0.0 a 1.0

* Valor normalizado com base nas dimensões da quadra onde 0.0 é a posição da rede, 1.0 é o limite da quadra do agente e -1.0 é o limite da quadra do adversário.

2.3 Coleta de Dados

Para a criação do conjunto de treinamento, foi criada uma cena separada onde um único jogador humano é exposto a uma série de situações e suas reações são então registradas.

O registro é feito através da coleta de amostras contendo informações sobre o ambiente (que serão usadas para preencher a camada de entrada da rede neural) e as ações do jogador (que preencherão a camada de saída da rede neural).

No lugar do adversário, foi posto uma zona vermelha na quadra que seria desse adversário. A bola é posicionada aleatoriamente e sua velocidade é indicada por uma seta. A bola permanecerá imóvel até que o jogador se posicione e dê o comando para que a bola seja lançada e a coleta comece.

Será coletada uma amostra sempre que a bola for lançada, sempre que o jogador executar uma ação de saque, levantamento ou pulo e em intervalos regulares de tempo.

A coleta se encerra quando o jogador marca ponto, o jogador sofre um ponto ou quando a bola atinge a zona vermelha.

Após a coleta dos dados, será feita uma avaliação do desempenho do jogador e decidir quais amostras utilizar e quais descartar.

Se o jogador errou, marcou ponto contra ou se a bola atingiu a zona vermelha, todas as amostras coletadas serão descartadas, pois considera-se que seu desempenho não foi satisfatório a ponto de ser aprendido pela IA.

Se o jogador marcar ponto, apenas as amostras anteriores à última vez em que ele tocou a bola serão consideradas. As restantes serão descartadas, pois considera-se que depois de dar seu último toque, o jogador não terá motivação para executar boas jogadas podendo realizar ações aleatórias que gerariam ruídos nos dados coletados dificultando, assim, o processo de treinamento da IA.

Também serão descartadas amostras muito antigas, pois considera-se que elas não fazem parte da jogada que levou ao êxito do jogador.

Por fim, as amostras que sobrarem, serão armazenadas e comporão o conjunto de treinamento.

2.4 Treinamento

O treinamento da rede foi feito utilizando *Backpropagation* repetidas vezes até que a rede neural se ajuste e seu erro fique abaixo do valor desejado

2.5 Aplicação

Para utilizarmos a rede neural treinada, foi construído um agente que possui uma réplica dessa rede neural.

A todo instante, o agente irá então coletar informações sobre o ambiente e preencher a camada de entrada de sua rede neural como descrito na etapa de percepção do agente. Em seguida, a rede será estimulada a produzir uma saída para os valores que ela recebeu como descrito na etapa de pensamento do agente. Essa saída, é levada em consideração e converter-se-á em uma ação dentro do jogo.

O primeiro valor de saída da rede (PosX) refere-se à posição onde o agente deve estar. Caso o agente não esteja nessa posição, ele deve mover-se para um dos lados a fim de chegar a posição desejada.

Os três valores de saída restantes serão avaliados e o que tiver o maior valor, terá sua ação realizada, desde que tenha um valor maior que 0.2.

Se nenhuma das saídas referentes as ações obtiver valor maior que 0.2, nenhuma ação será realizada pelo agente.

3 RESULTADOS E DISCUSSÃO

Durante os testes, o agente conseguiu reproduzir alguns dos movimentos básicos dos jogadores humanos, porém, com algumas falhas. Algumas de suas jogadas pareciam adequadas para a situação, mas fora de sincronia. Ora realizava uma jogada antes da bola se aproximar, ora realizava a jogada com atraso, quando a bola já não estava mais ao alcance.

O desempenho do agente foi considerado insatisfatório em termos competitividade. As falhas do agente podem ser facilmente exploradas por um jogador humano.

A característica que mais chamou atenção foi a capacidade de o agente reproduzir sequências de movimentos como um pulo para frente seguido de um saque como os jogadores de vôlei fazem.

4 CONCLUSÃO

Com esse experimento, foi possível constatar que é razoável a utilização de aprendizado de máquina para jogos competitivos que envolvem uma complexidade maior de ações possíveis ao jogador.

Embora o agente ainda não represente um desafio aos adversários humanos, é notável o potencial da tecnologia utilizada. Por tratar-se de uma técnica que depende diretamente de uma base de dados, as RNAs podem evoluir progressivamente a medida em que for utilizada, portanto, é natural que as falhas do agente sejam gradativamente corrigidas ou atenuadas ao longo do tempo.

REFERÊNCIAS

MURRAY, Janet H. **Hamlet no Holodeck: o futuro da narrativa no ciberespaço**. São Paulo, Itaú Cultural, UNESP, 2003.

SILVA, Felipe Correa Lima da. MAC5701 - Tópicos em Ciência da Computação: Agentes Inteligentes em Jogos de Computador, 2005.

RABIN, Steve. Introdução ao Desenvolvimento de Games vol. 2 Programação: técnica, linguagem e arquitetura. CENGAGE Learning.

SATHYANARAYANA, Shash. **A Gentle Introduction to Backpropagation**, 2014.

Disponível em:

<http://numericinsight.com/uploads/A_Gentle_Introduction_to_Backpropagation.pdf>

Acesso em: 13 jun. 2018, 09:41:30.