

PUCRS – Inteligência Artificial

T2 – Aprendizagem por Reforço: RN + AG + Minimax

Profa. Silvia Moraes

Objetivo: Construir uma solução de IA usando aprendizagem por reforço que faça uma rede neural aprender a jogar o jogo da Velha.

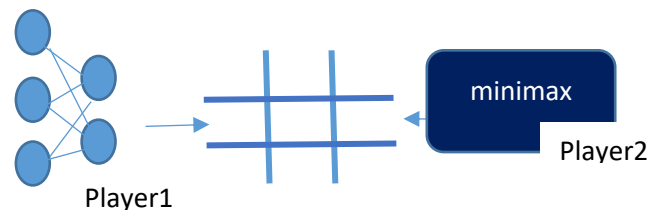
Arquitetura: Na arquitetura da solução, você deve usar uma rede neural e evoluir os seus pesos usando o algoritmo genético (AG). O algoritmo backpropagation não será usado. Caberá ao seu AG encontrar o melhor conjunto de pesos para a rede. Para facilitar a aprendizagem vamos acoplar o algoritmo Minimax (dado em aula) como usuário que treinará a rede. O algoritmo Minimax irá ensinar a rede a jogar. Para isso, o Minimax será adaptado para jogar em 3 modos: Fácil, Médio e Difícil. No modo Fácil, o minimax é usado em apenas 25% das jogadas da máquina, as demais jogadas serão aleatórias. Já no modo Médio, o minimax é executado em apenas 50% das jogadas. E no modo Difícil, ele é sempre usado. Esses modos de jogo do Minimax devem ser aplicados para evoluir a rede neural. Faz parte do trabalho definir a ordem em que esses modos são aplicados durante o aprendizado da rede. Lembrando que a IA durante o treinamento, deve sempre iniciar jogando.

- **Rede Neural:** Construa uma rede MLP de 2 camadas, a topologia inicial será discutida em aula. Implemente apenas a parte de propagação. Você pode se inspirar nos códigos de rede dados nos exercícios. A rede será um dos players.
- **Algoritmo Genético:** Os cromossomos serão os pesos da rede. Você terá que adaptar os exemplos dados em aula. Além da codificação da solução, você deve mudar a operação de cruzamento para uma que trabalhe como dados em double (consulte o moodle). Você precisará também definir uma função de aptidão que indique o quão bem a rede se saiu a cada jogo da velha finalizado. Será necessário ajustar os parâmetros do algoritmo também para este problema.
- **Front End:** Construir um front end mínimo para o jogo da velha. O controle sobre o estado do jogo é feito de forma algorítmica, ou seja, seu programa deve verificar se alguém venceu a cada jogada ou se ainda há jogo. Seu Front End, deve permitir que:
 - Você jogue com o minimax
 - A rede aprenda jogar com o minimax
 - Você jogue com a rede treinada pelo minimax
- **Desenvolvimento:** Mostrar a evolução da aprendizagem da rede, mostrando a evolução da população do Algoritmo Genético (crie uma forma para se consiga acompanhar passo a passo). Ao final da aprendizagem, permite que o usuário jogue contra a rede neural, nesse caso calcule a acurácia da rede.

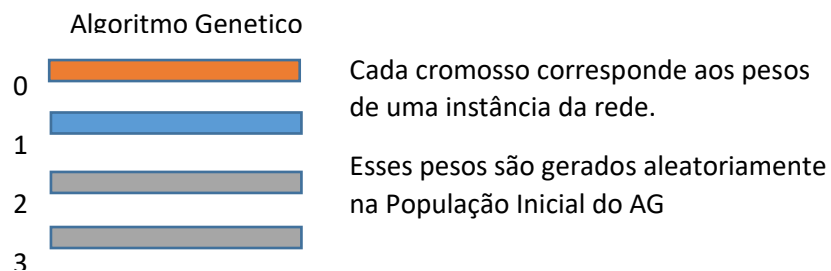
Etapas de Desenvolvimento:

1. Sua solução de IA terá um front end (mínimo) onde uma rede neural aprende a jogar o jogo da velha com o minimax. Portanto, sua solução deve permitir a alternância de turnos entre o a rede neural e o minimax, ou seja, ora a rede joga, ora o minimax joga.

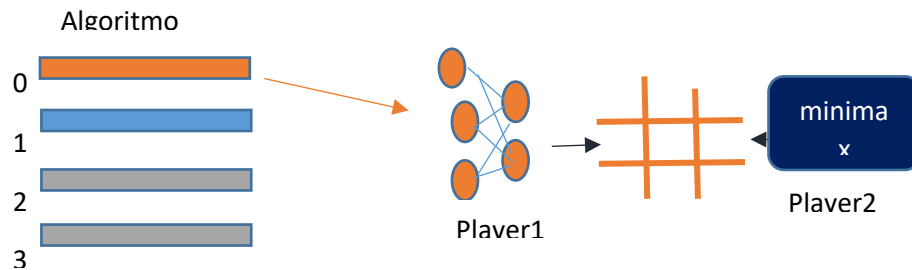
Jogo da Velha: o Minimax vai ensinar a rede a jogar



2. A rede recebe como entrada o tabuleiro atual e indica a jogada (a célula do tabuleiro que deve receber o seu símbolo X ou O).
3. Para iniciar o processo de aprendizagem por reforço, no algoritmo genético, crie uma população de tamanho N, onde cada cromossomo contém os pesos de uma rede neural. Cada cromossomo representa uma instância de rede. Portanto a quantidade de pesos corresponde a quantidade total de pesos da rede independente da camada. O N deve ser configurável.



4. Crie uma função de aptidão para medir o desempenho de cada cromossomo ao jogar o jogo da velha contra o minimax. A função deve contabilizar acertos e erros da rede ao final de um jogo. Por exemplo, se ela escolher célula já ocupada, penalize. Se ela perder, penalize.



Cada cromossomo representa uma instancia da RN

Cada cromossomo(pesos) são carregados na rede.

A rede com os pesos do cromossomo atual jogará com o minimax até atingir o estado fim de jogo: empate, vitória do Player 1 ou vitória do Player2. Você pode encerrar o jogo em caso de jogada inválida da rede também.

A função de aptidão deve conseguir medir o desempenho da rede.

5. Ao final de uma geração, execute o ciclo do AG aplicando os operadores vistos em aula: seleção (elitismo e torneio), cruzamento (para valores reais, ver documentação no moodle) e mutação.
6. Execute o AG por várias gerações e observe a evolução da rede. Seu algoritmo deve parar por número de gerações e convergência.
7. Ao final do processo de treino, ou seja, evolução da rede. Teste a rede. Jogue com ela e meça a sua acurácia.

Pontuação:

- (1,0) Adaptação do Minimax (modos) para jogar com a rede neural no seu front end
- (2,0) Construção da topologia da rede neural e implementação da propagação
- (2,0) Ciclo do AG para evolução da rede (operadores e função de aptidão)
- (1,5) Módulos de teste do Front End: 1,0 pt (permite usuário jogar contra minimax vale 0,3; no processo de aprendizagem rede joga com minimax vale 0,6; e rede joga com usuário, após treino da rede, vale 0,6).
- (2,5) Relatório de desenvolvimento: detalhando e evidenciando escolhas, descreve topologia da rede, estrutura e operadores do AG, etapas de desenvolvimento e análise de resultados. Considerações finais destacando pontos positivos e negativos do trabalho,
- (1,0) Vídeo (máximo de 10min) demonstrando a solução com a evolução da rede pelo AG e etapas de desenvolvimento.
- A falta de domínio na apresentação pode resultar em perda de até 70% da nota.

Orientações:

- Mesmo grupo do T1
- Data de entrega/apresentação no cronograma da disciplina