Reactive D31:
The Rise of the Ballz

Trabalho Prático 3

Ana Carolina Alves | 2017210460
Leandro País | 2017251509
M. Gabriela Valente | 2017265565

Keywords

Rede neuronal, algoritmo genérico, torneio, evolução, crossover, mutação.

Abstract

No âmbito da cadeira de Introdução à Inteligência Artificial foi implementado utilizando o ambiente Unity um agente cuja função, quando colocado num dado ambiente, é captar todas as informações que são importantes para o aprendizado do nosso agente.
Para captar estas informações é modelada uma rede neuronal, com camadas de entrada, escondidas e de saída. A rede neuronal é responsável por aproximar os inputs do output.
Para melhorar a performance do nosso agente, aplicamos o algoritmo genético. Algoritmo baseado na evolução natural e que passa por várias gerações. Buscamos é a geração com o melhor resultado possível.

Assim, serve o presente relatório para descrever todo o processo desde a implementação, escolhas que foram tomadas, alguns problemas que foram encontrados e como é que foram resolvidos, a experimentação e respetivos métodos e por fim, e provavelmente mais importante, o desempenho verificado e possíveis ilações a retirar dos dados obtidos.

Índice

Introdução

Com o intuito de ajudar nosso agente D31 a praticar bom futebol, pretendemos desenvolver um controlador para o mesmo para realizar tarefas de controlo, defesa e ataque, implementando uma rede neuronal, que nos ajuda a aproximar os inputs dos outputs, nos permitindo captar todas as informações do ambiente que são importantes para o aprendizado do nosso agente e o algoritmo genético, que irá aperfeiçoar a rede, ajudando no aprendizado.

Com este trabalho pretende-se adquirir conhecimentos práticos de análise e desenvolvimento de agentes aprendizes e adaptativos. Perceber como é feita a modelação de uma rede neuronal e o desenvolvimento do algoritmo genético.

Implementação

Neste módulo foi implementado uma rede neural e o algoritmo genético.
Na primeira fase do trabalho nós fixamos a topologia da rede neural. Escolhemos permanecer com a arquitectura que já estava incluída no package com 12 nós na Camada de Entrada, 3 nós da Camada Escondida e 2 nós da Camada de Saída.
Para a mutação e selecção de torneio a implementação foi feita seguindo o pseudocódigo fornecido através do enunciado.

No que toca o Crossover, primeiro verificamos se haveria o mesmo, usando a probabilidade. Caso existisse o Crossover, nós sorteamos aleatoriamente um ponto de corte e trocamos os genes a partir desse ponto. Fazendo com que o filho 1 obtenha metade da mãe e o filho 2 obtenha metade do pai.
Implementamos 3 fitness functions, sendo uma para defender e outras duas para chutar e controlar a bola, sendo uma delas melhorada para ser utilizada no cenário “1x1”.

Attack fitness

fitness *= (distanceToClosestWall.Average()*1)+(distanceToBall.Average()*-2)+ distanceToMyGoal.Average()*1+ (distanceToAdversaryGoal.Average() * -1) + (distancefromBallToAdversaryGoal.Average()*-2)+(GoalsOnAdversaryGoal*5000);*

A fitness de ataque tem o objetivo de fazer o agente vermelho controlar a bola e colocá-la dentro da baliza adversária. O agente foi treinado através de um conjunto de recompensas e punições. A fitness tem impactos positivos quando o agente está afastado das paredes (exemplo: no (distanceToClosestWall.Average()*1) se o agente está a uma distância média de 2 blocos da parede, ele é recompensado apenas com 2 pontos(2*1), mas se ele estiver com uma distância média da parede de 10 blocos é recompensado com 10 pontos (10*1), sendo então mais favorável para o agente se estiver afastado da parede); a fitness também tem impactos positivos se o agente estiver mais próximo da bola e mais próximo da baliza adversária do que da sua baliza, e por fim se ele conseguir marcar golos tem uma maior recompensa para ele "aprender" que é esse o seu objetivo.

Contrariamente a fitness pune negativamente o agente se o inverso acontecer, se ele estiver próximo das paredes, se estiver muito afastado da bola e se a distância da bola à baliza contrária for muito elevada também.

Defense fitness

fitness *= (distanceToClosestWall.Average() * 5) + (distanceToBall.Average() * -2) + (distanceToMyGoal.Average() * -1) + (distancefromBallToMyGoal.Average() * 2) + distancefromBallToAdversaryGoal.Average() + GoalsOnMyGoal * -5000;*

A fitness de defesa tem como objetivo impedir a bola de entrar na baliza do agente vermelho, para esse efeito ser concretizado o agente foi ensinado a manter-se próximo da sua baliza e afastar a bola da mesma. Para isso a fitness pune o agente todas as vezes que a distância da bola à sua baliza é reduzida e tem um impacto ainda mais negativo se a bola entra na mesma. Contrariamente é recompensado se a bola se encontra distante da sua baliza. Depois de várias gerações o agente aprende a posicionar-se na direção da bola para impedir que ela entre na baliza efetuando assim uma "defesa".

1x1 fitness

fitness = *(1/distanceToBall.Average() + 1) + GoalsOnAdversaryGoal +1/(distanceFromBallToAdversaryGoal.Average() + 1)) - (GoalsOnMyGoal + 1/(distanceToClosestWall.Average() +1) + 1/(distancefromBallToMyGoal.Average()+1);*

Com a função de fitness em que o agente foi treinado no mapa de 1 contra 1, procuramos recompensar comportamentos que levam à marcação de um golo na baliza adversária. Para alcançar tal objetivo, a fitness é impactada positivamente quanto a distância do agente à bola. O que na prática significa que o agente irá tentar controlar a bola perto de si, que em conjunto com a distância da bola à baliza adversária irá fazer com que o agente, ao longo das gerações, privilegie o controlo da bola até à baliza adversária.Em contrapartida, o agente é castigado quando o inverso acontece, ou seja, quando a bola está perto da sua baliza a fitness é impactada negativamente fazendo com que, ao longo das várias gerações, o agente procure afastar a bola da sua baliza realizando assim defesa.

Configuração Experimental

No que toca a configuração experimental tendo por base as indicações fornecidas pelos professores foi escolhido um modelo em que primeiro verificamos com um número de gerações menor se a função fitness funcionava de acordo. Depois,fizemos vários testes com 200 gerações e 20 de população para que tivéssemos melhores resultados.

Na maioria utilizamos a random seed 50, mas tiveram alguns testes em que alteramos para 100 e 500, porém não tivemos grandes mudanças.

De modo a tentar manter o relatório sucinto foi decidido incluir apenas os resultados das últimas 200 gerações evoluídas de cada fitness.

Mantivemos o valor da população igual a 20. Já em relação à mutação e ao crossover, tentamos fazer pequenas alterações ao longo do treinamento. Porém os valores mais utilizados foram os valores padrão.

Assim, resta apenas deixar uma breve explicação no que tocas aos seguintes gráficos.

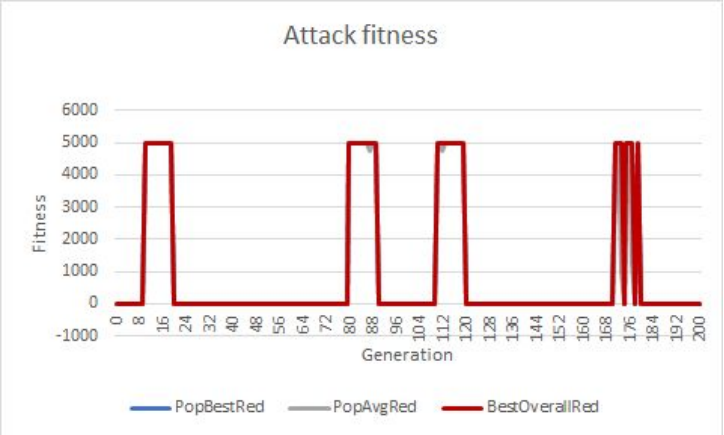
Como se pode ver pelos gráficos os valores de fitness variam bastante, isto pode ser explicado pela mutação e crossover que essencialmente pega numa parte random do genótipo da geração anterior e como é óbvio isto não garante que apenas se resgatem os que têm o melhor fitness.

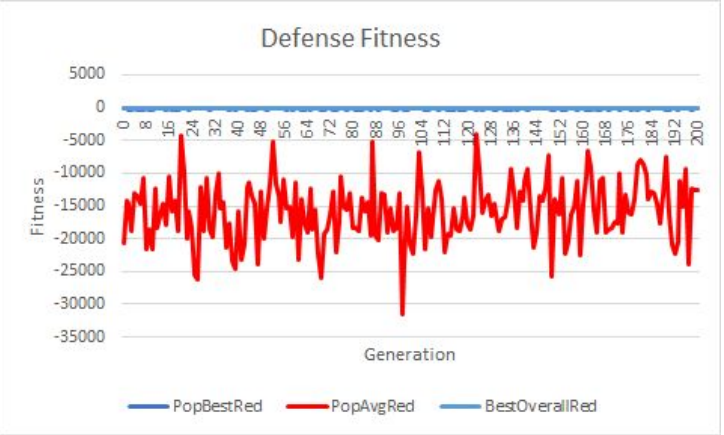
Observando o gráfico do **Attack Fitness** conseguimos concluir que o agente só conseguiu concretizar golos em algumas gerações, uma vez que os pontos elevados do gráfico demonstra que no máximo ele fez apenas 1 golo, uma vez que pudéssemos a recompensa de marcar golo muito elevada (5000), não conseguimos observar as recompensas menores. Ele não teve um desempenho melhor porque se focava mais em se aproximar da baliza adversária do que da bola.

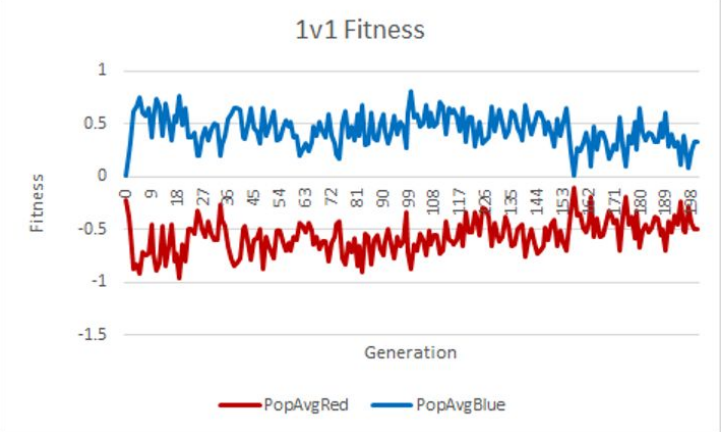
A fitness no cenário de defesa teve resultados tão variados, pois conseguiu realizar algumas defesas, mas, em determinados momentos, também levou golos.

A fitness no cenário 1 contra 1 depende não só do desempenho do agente, mas também do seu adversário. Por fim, é ainda de notar que os valores estão normalizados e como tal variam de -1 a 1.

Resultados Experimentais







Discussão

Em geral, com todos os testes e gerações que obtivemos, tivemos respostas medianas quanto ao nosso agente. Tendo ele realizado todas as funções necessárias, mas de uma forma não tão vantajosa.

Percebemos que o agente durante o ataque ao início realizou o comportamento esperado, mas mais à frente vimos que estava aproximando-se mais da baliza adversária do que da própria bola, nos fazendo acreditar que a compensação quanto à baliza adversária talvez não seja a mais apropriada no contexto.

Quanto à defesa, durante a evolução no mapa “Evolving-defense” conseguimos presenciar algumas.

No que toca o cenário 1 contra 1, ambos os jogadores conseguiram alcançar o objetivo de marcar golos em relativamente poucas gerações. No entanto, em relação à defesa foram precisas bastantes gerações para se observar um comportamento que se assemelhasse a uma defesa. Em retrospectiva, concluímos que de facto nunca chegámos a uma verdadeira defesa, sendo que o comportamento apresentado se tratava de uma tentativa de marcar golos de ambos os jogadores que os levava a interceptar a bola quando esta se deslocava em direção à baliza.

Conclusão

Do ponto de vista técnico, com os resultados obtidos conseguimos tirar algumas conclusões principais.

Concluímos que talvez não tenhamos treinado o agente o bastante para que ele jogasse da melhor forma possível. Como foi dito antes, apesar de termos tido ações esperadas e alguns bons resultados, talvez não tenha sido o melhor que conseguiríamos.

Apesar de tais resultados, é necessário afirmar que saímos deste trabalho com mais e melhores conhecimentos sobre rede neural e algoritmo genético. Tendo sido necessária bastante pesquisa sobre e uma compreensão maior do assunto. E por isso, consideramos que este tipo de trabalho é um ótimo complemento aos conhecimentos teóricos abordados nas aulas. Nos incentivou à procura e ao estudo mais aprofundados do assunto e, acima de tudo, ao contacto prático que de outra maneira não teríamos.

Referências

Abaixo fica uma lista dos links que foram utilizados para melhor perceber melhor como implementar o algoritmo genérico:

Step-by-step Guide to Building Your Own Neural Network From Scratch

Introdução ao Algoritmo Genético - Parte 1t