

Sistemas de Aprendizagem

Aprendizagem e Extração de Conhecimento

Universidade do Minho, Portugal

“How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?” [1]

Resumo A área de Sistemas de Aprendizagem ou *Machine Learning*, pode ser vista como um paradigma de computação dentro da campo da inteligência artificial, onde o principal objetivo é a conceção e o desenvolvimento de algoritmos e técnicas que permitam que os computadores adquiram conhecimento de forma automática. Estes sistemas têm a função de analisar informações e generalizá-las para extrair novas aprendizagens, representando uma promessa no que diz respeito à sua aplicação na sociedade, conduzindo a uma mudança importante nos ‘ecossistemas digitais’. Das várias abordagens inseridas neste método autónomo, neste documento, iremos tratar a Aprendizagem por Reforço, as Redes Neurais Artificiais e as Máquinas de Vetores de Suporte.

Keywords: Sistemas de Aprendizagem · Machine Learning · Reinforcement Learning · Aprendizagem por Reforço · Artificial Neural Networks · Redes Neurais Artificiais · Support Vector Machines · Máquina de Vetores de Suporte

1 Aprendizagem por Reforço

1.1 Introdução

Desde de que nascemos, aprendemos através do ambiente onde nos encontramos e das interações que temos com este ambiente. Com isto em conta trazemos o tópico de Aprendizagem por Reforço, onde se aplica esta mesma ideia de interação com o ambiente e análise das respostas que as ações neste ambiente têm; tudo isto tendo como ideal o atingir um objetivo final, uma recompensa. [2]

1.2 Descrição característica

” We have no idea how to do something, but we can say whether it has been done right or wrong.” [3]

A aprendizagem por reforço caracteriza-se pela aprendizagem de como responder/agir a determinadas situações de forma a maximizar a recompensa.

Sendo que, não sabe quais são as melhores ou piores ações a tomar e só descobre experimentando essas opções. Em situações mais complexas esta recompensa pode só se demonstrar no futuro impossibilitando assim uma inferência direta da ação a tomar. Assim, este tipo de aprendizagem é caracterizado por 'trial-and-error' e por 'delayed reward'. [4]

Assim, dado o 'delayed reward' desta forma de aprendizagem, uma recompensa não pode ser diretamente relacionada com determinada ação, pode ser o resultado de outra ação passada, a única forma que o agente tem é o 'trial-and-error' com o intuito de melhorar a recompensa final. [5]

Esta ideia de recompensa pode ser vista também em nós; de forma análoga ao agente, a nossa recompensa pode ser vista como a sensação de prazer, para uma recompensa elevada, ou dor para uma recompensa baixa. [2]

Um dos problemas é o equilíbrio entre a 'exploration' e 'exploitation', onde o agente tende a procurar mais informações e onde o agente tende a maximizar a recompensa, respetivamente. No entanto, se o agente optar pela 'exploitation' para maximizar a recompensa ele tem que experimentar ações que não tomou no passado de formar a ser eficaz nesta tarefa. Se optar pela 'exploration' nunca irá tirar o proveito do que já descobriu. Logo, seguir exclusivamente por um caminho não é viável, o ideal é um equilíbrio entre ambos: um sistema onde o agente experimenta várias ações e vai progressivamente favorecendo as que lhe são mais vantajosas. [2]

Em suma, a aprendizagem por reforço considera o problema como um todo e está orientada a um objetivo, através da maximização da recompensa.

1.3 Capacidade de Aprendizagem

O problema que é tratado por este tipo de aprendizagem pode ser formalizado através dos modelos markovianos de decisão com informação parcial. Com base nesta formalização, o agente de acordo com o estado do ambiente que o rodeia é capaz de tomar decisão que afetam o estado em que se encontra. Tudo isto acontece tendo em conta que o agente tem um ou mais objetivos em relação ao estado do meio.

De forma muito simplificada, os modelos markovianos de decisão apenas incluem três aspetos: a sensação, a ação e o objetivo (ver Fig. 1). De acordo com a sensação, o agente toma a ação mais adequada de forma a receber a maior recompensa, de acordo com o objetivo pretendido. [2,6]

Desta forma, todos os problemas que se enquadrem nesta forma de resolução podem-se considerar problemas de aprendizagem por reforço. Assim, a aprendizagem por reforço pode ser como caracterizada como outro paradigma de aprendizagem de 'machine learning', para além da aprendizagem com supervisão e da aprendizagem sem supervisão, diferenciando-se de ambos pelo objetivo final que consiste na maximização da recompensa e do facto de não se ter dados iniciais. [7]

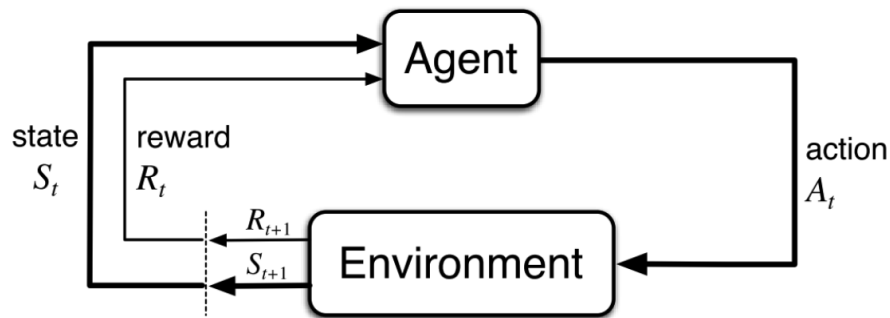


Figura 1. Relação entre o agente e meio num modelo markoviano de decisão.

1.4 Ferramentas de Desenvolvimento

As ferramentas que se encontram no mercado para a aprendizagem por reforço são bastante vastas. Bibliotecas, como o OpenAI Gym, TensorFlow, DeepMind Lab e Pytorch, que são o resultado do trabalho da comunidade *open-source* em conjunto com as grandes empresas tecnológicas como a *Google* e *Facebook*.

1.5 Soluções no Mercado

Este tipo de problemas existe quando não é possível obter um conjunto de dados representativo (ideal para a aprendizagem com supervisão, que se baseia num conjunto representativo de dados de teste e generaliza a solução para a situação geral); ou quando a interação com o meio é indispensável para a aprendizagem ou porque é um campo novo ou porque é uma solução mais adequada como por exemplo nos bots de jogos com inteligência artificial, na navegação de robôs, entre outros.

Um desses casos esteve recentemente em destaque na comunicação social quando um *bot* venceu o campeão mundial do jogo de tabuleiro *Go*, jogo esse bastante conhecido dada a sua dificuldade devida a existência de inúmeras possibilidades de jogo e os respetivos jogos que daí advêm. Sendo que recentemente já esse *bot* foi derrotado por outros.



Figura 2. Logo AlphaGo.

2 Redes Neuronais Artificiais

2.1 Descrição Característica

As Redes Neuronais Artificiais (RNA) são modelos computacionais baseados na biologia, mais propriamente nos neurónios humanos, com o intuito de replicá-los e simulá-los. [8]

As RNA são, essencialmente, constituídas por duas camadas principais, sendo elas as camadas de entrada e de saída (ver Fig. 3). No entanto, na maioria dos casos, possuem uma ou mais camadas ocultas que transformam os dados da entrada em dados úteis para as saídas. [9]

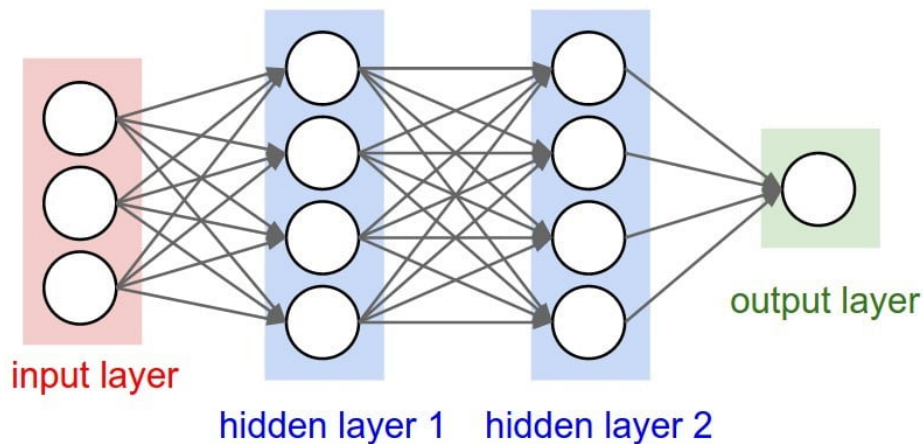


Figura 3. Representação das camadas de uma RNA

Estas camadas são formadas por vários neurónios artificiais, os quais se encontram conectados por coeficientes, constituindo, assim, uma estrutura neuronal. [8]

Cada neurónio apresenta entradas ponderadas, uma função de transferência e uma saída. As entradas têm pesos associados, os quais são parâmetros ajustáveis, tornando as redes neuronais num sistema parametrizado (ver Fig. 4). [8]

A soma destas entradas cria um sinal de ativação do neurónio, sendo ele propagado pela função de transferência com o intuito de produzir uma única saída do neurónio. [8]

As RNA são muito usadas em *Machine Learning*, uma vez que apresentam uma capacidade de aprendizagem baseada em experiências, e não da programação, detetando padrões e relações nos dados. [8]

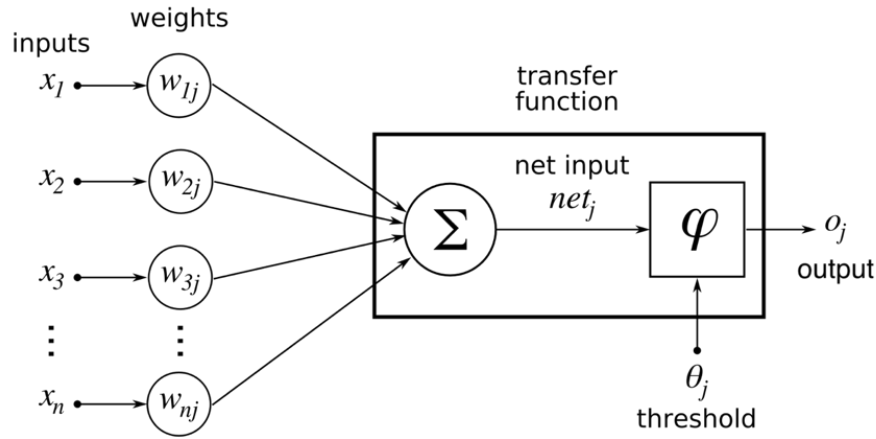


Figura 4. Representação de um neurónio artificial

2.2 Capacidade de Aprendizagem

A rede neuronal artificial exhibe os seus comportamentos com base nas funções de transferência dos neurónios, pelas regras de aprendizagem e pela própria arquitetura. [8]

Durante a aprendizagem ou treino, as conexões entre os neurónios são otimizadas, de forma a reduzir o erro nas previsões, aumentando, assim, a precisão. [8]

As RNA podem ser ensinadas de várias formas, tais como aprendizagem supervisionada, aprendizagem não supervisionada ou aprendizagem por reforço. Desta forma, as redes neuronais artificiais requerem dados para aprender e, na maioria dos casos, quanto mais dados forem introduzidos numa rede, mais precisos serão os resultados. [9]

A tarefa de ensinar as RNA está dividida em 3 etapas. A primeira diz respeito à realização de um conjunto de treinos, os quais vão ajudar a rede a estabelecer os vários pesos entre os nós. A segunda consiste em ajustar a rede, usando dados válidos. Por fim, na terceira e última etapa, realiza-se um conjunto de testes com o intuito de averiguar se a rede consegue transformar a(s) entrada(s) numa saída desejada. [9]

Um dos maiores desafios das RNA é o tempo excessivo que elas demoram a aprender. No entanto, o maior problema é o facto de não ser possível aceder ao processo de tomada de decisão, tendo apenas acesso aos dados fornecidos e às saídas obtidas. [8]

2.3 Ferramentas de Desenvolvimento

Existem várias ferramentas de desenvolvimento de RNA, sendo que a sua maioria são Bibliotecas *open-source* para diversas linguagens de programação, sendo que algumas delas são:

- Theano
- Torch
- Caffe
- TensorFlow
- MXNet
- FANN
- Neuroph
- OpenNN

2.4 Soluções no Mercado

Existem várias soluções implementadas no mercado, sendo que as mais relevantes são:

- Medicina
 - Diagnóstico de doenças;
 - Previsão do tempo de recuperação de pacientes;
- Biologia
 - Interpretação de sequências do genoma humano;
- Engenharia
 - Detecção de falhas em estruturas de pontes;
- Economia
 - Previsão do valor de ações na bolsa;
- Robótica
 - Visão por computador;
 - Condução autónoma.

3 Support Vector Machines

3.1 Descrição característica

Support Vector Machines, também denominadas como *Support Vector Networks*, são modelos de aprendizagem supervisionados associados a algoritmos de aprendizagem com o propósito de analisar dados para classificação e análise de regressão. Existindo um *training set* de exemplos, cada um marcado como uma de duas categorias, um algoritmo de treino *SVM* escolhido, constrói um modelo que classifica novos exemplos com uma categoria ou com outra. Este tipo de atribuição, torna-o um classificador linear binário não probabilístico, no entanto, existem métodos como o escalonamento de *Platt* para usar *SVM* com uma configuração de classificação probabilística.

Um modelo *SVM* é uma representação dos exemplos como pontos no espaço, sendo eles mapeados com o intuito de que os exemplos de cada categoria sejam separados por uma divisão clara o mais ampla possível. Aquando da aparição de novos exemplos, os mesmos são mapeados no mesmo espaço e previstos numa determinada categoria com base no lado da divisão que se inserem.

3.2 De que modo exhibe a capacidade de aprendizagem

Formalmente, um *SVM*, constrói um hiperplano ou um conjunto de hiperplanos em um espaço de alta ou infinita dimensão, que pode, posteriormente, ser usado para classificação, regressão ou outras funções como detecção de *outliers*. Por intuição, uma boa separação é alcançada pelo hiperplano que possuir a maior distância até o ponto do *training set* mais próximo de qualquer classe, denominada margem funcional. Na generalidade, quanto maior a margem menor o erro de generalização do classificador.

Partindo do pressuposto que o problema original podia ser declarado num espaço dimensional finito, frequentemente acontece que os conjuntos para discriminar não são linearmente separáveis nesse espaço. De modo a manter a carga computacional razoável, os mapeamentos usados pelos esquemas *SVM* foram desenhados para garantir que produtos escalares de pares de dados de entrada possam ser computados facilmente em termos das variáveis no espaço original, definindo-os em termos de uma *kernel function* $k(x, y)$ selecionados de acordo com o problema. Os hiperplanos de dimensão superior no espaço são definidos como o conjunto de pontos cujo produto de um ponto com um vetor nesse espaço seja constante, sendo o tal conjunto de vetores um conjunto de vetores ortogonais que definem o hiperplano.

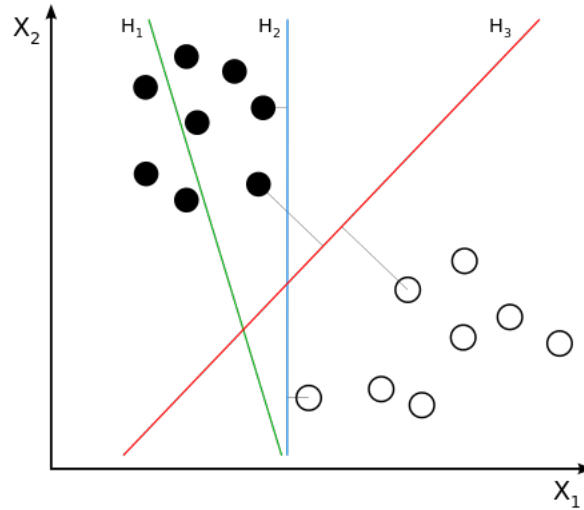


Figura 5. SVM separação hiperplano

Os vetores que definem os hiperplanos podem ser escolhidos de forma a serem combinações lineares com os parâmetros α_i de vetores de imagem x_i que ocorram na base de dados. Com a escolha do hiperplano efetuada, os pontos x no *feature space* que são mapeados no hiperplano são definidos pela relação $\sum_i \alpha_i k(x_i, x) = constant$.

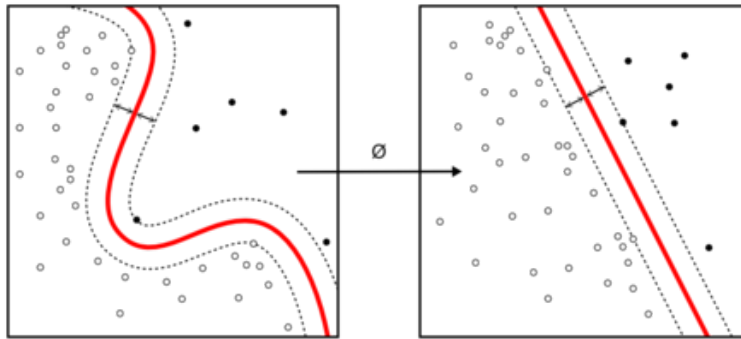


Figura 6. Kernel Machine

3.3 Ferramentas de Desenvolvimento

Tratando-se de uma técnica de aprendizagem com casos de uso bastante abrangentes, é possível encontrar uma implementação da mesma em várias bibliotecas de matemática ou *machine learning*, nomeadamente:

- LIBSVM
- MATLAB
- TensorFlow
- Shogun
- Weka
- Shark
- OpenCV

3.4 Soluções existentes no mercado

As *SVMs* podem ser usadas em várias aplicações no mundo real como por exemplo:

- **Deteção de faces** - SVM classifica partes da imagem como um face e uma não face e criam um quadrado em volta da face.
- **Categorização de texto** - SVM permitem que o texto seja categorizado para ambos modelos indutivos e transdutivos.
- **Classificação de imagens** - Permitem uma melhor precisão de procura quando comparado com métodos tradicionais de procura por interrogação.
- **Bio-informática** - Inclui classificação de proteínas e cancro. É usada para identificar a classificação de genes, pacientes e outros problemas biológicos.
- **Reconhecimento de escrita** - Usados para reconhecer caracteres manuscritos.

Referências

1. Tom M. Mitchell (2006) The Discipline of Machine Learning, Carnegie Mellon University. <http://www.cs.cmu.edu/~tom/pubs/MachineLearning.pdf>
2. Richard S. Sutton and Andrew G. Barto (2017) Reinforcement Learning: An Introduction, The MIT Press, Second edition, in progress. <http://incompleteideas.net/book/bookdraft2017nov5.pdf>.
3. Supervised learning, unsupervised learning and reinforcement learning (2016) <https://stats.stackexchange.com/q/144154>. Último acesso: 12/10/2018.
4. César Analide, Filipe Gonçalves (2018) Aprendizagem por Reforço, Aprendizagem e Extração de Conhecimento, ISLAB, Universidade do Minho.
5. D. J. Finton and Y. H. Hu (1995) Importance-Based Feature Extraction for Reinforcement Learning. <http://www.cs.wisc.edu/~finton/clnl.ps>.
6. Modelos Markovianos de Decisão, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (2009). <https://moodle.fct.unl.pt/mod/folder/view.php?id=101140>.
7. Andrew G. Barto and Thomas G. Dietterich (2004) Reinforcement Learning and its Relationship to Supervised, University of Massachusetts. http://www-anw.cs.umass.edu/pubs/2004/barto_d_04.pdf.
8. S. Agatonovic-Kustrin, R. Beresford (1999) "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research". <https://www.sciencedirect.com/science/article/abs/pii/S0731708599002721> Último acesso: 11/10/2018.
9. Luke Dormehl (2018) "What is an artificial neural network? Here's everything you need to know". <https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/> Último acesso: 12/10/2018.
10. DF Team (2017) Real-Life Applications of SVM (Support Vector Machines) . <https://data-flair.training/blogs/applications-of-svm/>. Último acesso: 11/10/2018.
11. Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks". Machine Learning. 20 (3): 273–297. <https://doi.org/10.1007/BF00994018>