

Sistemas de Aprendizagem - Case Based Reasoning, Genetic Algorithms and Artificial Neural Networks

Hugo Carvalho, Marcos Luís, Luís Lima

Universidade do Minho, Departamento de Informática, 4710-057 Braga, Portugal
{a74219, a70676, a74260}@alunos.uminho.pt

Resumo O conceito de aprendizagem caracteriza-se como o processo pelo qual as nossas competências e conhecimentos são adquiridos ou modificados como resultado da experiência, raciocínio ou observação. Deste modo, quando pretendemos elaborar um sistema inteligente, torna-se fundamental uma correta escolha de um sistema ou modelo de aprendizagem que se enquadre nos objetivos pretendidos.

Keywords: Sistemas de Aprendizagem, Raciocínio Baseado em Casos, Algoritmos Genéticos, Redes Neurais Artificiais

1 Introdução

Este trabalho foi desenvolvido no âmbito da unidade curricular de Aprendizagem e Extração de Conhecimento do perfil de especialização de Sistemas Inteligentes tendo como objetivo abordar e compreender o funcionamento dos sistemas de aprendizagem de raciocínio baseado em casos, algoritmos genéticos e redes neuronais artificiais.

Para cada um dos sistemas apresentados, será feita uma descrição característica do seu funcionamento e da sua capacidade de aprendizagem. Para além disso, são indicadas algumas ferramentas de desenvolvimento de cada um dos modelos, bem como apresentados alguns casos reais da sua utilização.

2 Case Based Reasoning

Nos últimos anos, o Raciocínio Baseado em Casos surgiu como uma técnica poderosa para solução automática de problemas, sendo aplicável de forma simples e direta a um amplo espectro de tarefas, todas tipicamente relacionadas à Inteligência Artificial. A ideia básica do Raciocínio Baseado em Casos é resolver um novo problema lembrando uma situação anterior similar, e dessa forma reutilizar informação e conhecimento dessa situação.

Este sistema de aprendizagem foi inspirado a partir da cognição e comportamento dos seres humanos iniciando-se os seus estudos na década de 80 na Universidade de Yale. Em 1983 foi desenvolvido o primeiro sistema CBR, de nome *Cyrus*, pelas mãos de Janet Kolodner.

2.1 Capacidade de Aprendizagem

No Raciocínio Baseado em Casos a capacidade de aprendizagem é feita através de um sistema de bibliotecas onde são guardados casos passados, tal como acontece com uma criança na sua fase de crescimento.

Quando um CBR se depara com um novo caso, pesquisa inicialmente na sua biblioteca por casos similares. Após esta primeira fase do processo, caso se encontre um caso semelhante, a abordagem feita a este mesmo é posta em prática havendo assim uma reutilização da informação e do conhecimento. De seguida a abordagem ao novo caso é verificada por um profissional ou por outro software e conclui-se o processo com o armazenamento na biblioteca do novo caso, agora resolvido.

Este processo pode então ser resumido em quatro etapas:

- *Retrieve* - Analisar a informação de casos anteriores
- *Reuse* - Utilizar caso seja igual ou semelhante
- *Revise* - Analisar a solução do novo caso
- *Retain* - Guardar a informação do novo caso

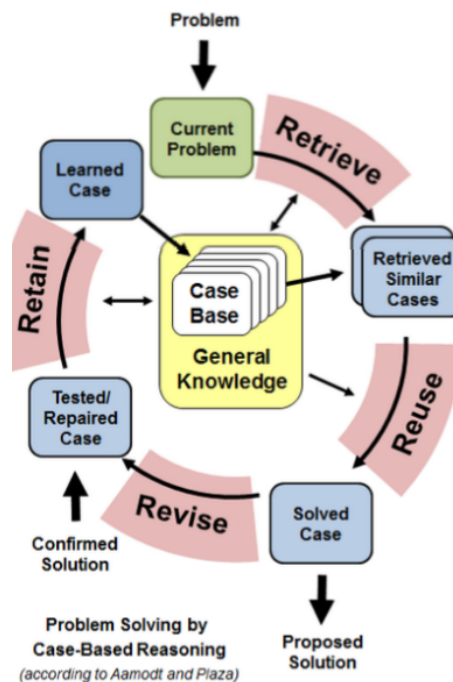


Figura 1. Ciclo de Aprendizagem

Para além de ser intuitivo, este método de aprendizagem não necessita de conhecimento de criação de regras e métodos isto tornando, assim, o seu desenvolvimento simples. O sistema "aprende", guardando novos casos, precisando assim de pouca manutenção e justificando com precedentes. Contudo, este método de aprendizagem tem as suas limitações, como por exemplo, o número de casos já analisados e que se encontram armazenados nas suas bibliotecas ser limitado. Outra desvantagem é por vezes generalizar um caso em concreto.

2.2 Ferramentas de Desenvolvimento

Em termos de ferramentas que nos permitem implementar um sistema de CBR existe uma variedade alargada, das principais destacamos algumas como : ReCall, CBR-Works, ReMind, myCBR, jCOLIBRI, CBR*Tools, CAT-CBR e CASPIAN. Vamos analisar o myCBR e o JCOLIBRI devido à sua maior popularidade de uso e distribuição.

- **myCBR** - Uma das ferramentas mais populares tem as suas capacidades e limitações. Desenvolvido pelo Centro Alemão de Inteligência Artificial, é uma plataforma *open-source* e pode ser modificada de acordo com o propósito do utilizador. O objetivo do myCBR é minimizar o esforço necessário para criar uma aplicação baseada em CBR. Uma das características que justifica a sua popularidade é o facto de não ser necessário saber programar, sendo portanto *friendly-user*.

- **jCOLIBRI** - Desenvolvido em Java e que utiliza tecnologia JavaBeans para representação de casos e geração automática de interfaces para o utilizador. É desenvolvido pelo GAIA, grupo de Inteligência Artificial, sediado na Universidade Complutense em Madrid sendo também uma das ferramentas mais utilizadas.

Comparando as duas ferramentas focadas anteriormente, podemos concluir que o myCBR é direcionado para utilizadores que não estejam familiarizados com a programação, por sua vez, é limitado em termos de dimensão de projeto. Por outro lado, o jColibri é uma ferramenta que pode ser utilizada para projetos com um nível mais complexo e por conseguinte, requer profissionais com conhecimento no campo da programação.

2.3 Soluções no Mercado

A partir dos anos 90 as empresas começaram a implementar CBR e a partir daí houve uma expansão no número de empresas que utilizavam este modelo pois notaram que trazia benefícios e cortes nos seus gastos. De momento o Raciocínio Baseado em Casos é utilizado em áreas como:

- **Diagnóstico** - Sistemas que selecionam casos passados cuja lista de sintomas seja similar ao novo caso e que sugerem um diagnóstico baseado nos casos mais similares.
- **Previsão** - Ajuda a prever acontecimentos futuros analisando acontecimentos passados como por exemplo, na meteorologia.
- **Planeamento** - Ajuda no planeamento inicial de negócios ou empresas com base em historiais de situações semelhantes.
- **Classificação** - Ajuda na classificação tendo em conta notas atribuídas anteriormente.
- **Design** - Estes sistemas assistem o cliente apenas numa parte do processo de design e precisam de ser combinados com outras formas de *reasoning* para suportar um processo de design completo.
- **Lei** - Na lei é utilizado na decisão de sentenças por exemplo, tendo como base casos passados.
- **Centrais de Ajuda** - Estes sistemas são usados na área de serviço ao cliente, com o objetivo de lidar com falhas nos produtos ou nos serviços.

Como se pode verificar, este sistema de aprendizagem incide em vários campos diferentes como exemplos reais no mercado de trabalho. Em 1993 um dos primeiros casos do uso deste método surgiu pelas mãos de McIntyre, Achabal e Miller no qual desenvolveram um sistema que usava raciocínio baseado em casos para prever as vendas num ambiente de descontos no âmbito de vendas em retalho. Este é um dos campos em que a previsão se torna crucial devido a influenciar e a prever a organização de stock, custos associados e decisões de percentagens de descontos. Este exemplo, como um dos primeiros a utilizar este método de aprendizagem, serviu primordialmente para demonstrar a sua utilidade e pertinência do seu uso na área de marketing.

3 Genetic Algorithms

A natureza sempre foi uma ótima fonte de inspiração para toda a humanidade. Algoritmos Genéticos (GAs) são um subconjunto de um ramo muito maior de computação conhecido como Computação Evolutiva, frequentemente utilizados para encontrar soluções aproximadas em problemas de otimização e pesquisa. Estes algoritmos iniciam com um conjunto de soluções possíveis e, através de operações especiais (avaliação, seleção, crossover e mutação), evoluem progressivamente em direção a soluções mais promissoras. Assim como as redes neuronais surgiram inspiradas no funcionamento do cérebro, os algoritmos genéticos foram inspirados na evolução, seleção natural e genética.

GAs têm a capacidade de oferecer uma solução "suficientemente boa" com "o suficiente". Isso torna os algoritmos genéticos atraentes para serem usados na resolução de problemas. Os principais motivos pelos quais os algoritmos genéticos são necessários e importantes são os seguintes:

- **Solução de problemas difíceis** - Na informática, há um grande conjunto de problemas, que são NP-Hard. O que isso significa essencialmente é que, mesmo os sistemas de computação mais poderosos, levam bastante tempo a resolver esse problema. Em tal cenário, os GAs revelam-se uma ferramenta eficiente para fornecer soluções quase ótimas em um curto período de tempo.
- **Obter uma boa solução rapidamente** - Imagine que estamos a usar o sistema de navegação GPS, e leva alguns minutos (ou mesmo algumas horas) para calcular o caminho "ótimo" da origem ao destino. O atraso em tais aplicações do mundo real não é aceitável. Assim, este tipo de aprendizagem é muito útil para combater situações deste género.

Estes algoritmos dependem basicamente de uma função que avalie a qualidade de uma determinada solução para o problema - e esta função pode ser obtida mesmo para problemas difíceis de serem resolvidos através de técnicas convencionais.

3.1 Capacidade de Aprendizagem

O algoritmo é inicializado gerando aleatoriamente um grupo de soluções possíveis, em que cada um destes é chamado de indivíduo (ou fenótipo), enquanto que o grupo é referido como uma população. Cada indivíduo é caracterizado por um conjunto de propriedades, o seu genótipo, cada um destes é representado por vetores binários onde cada elemento de um vetor denota a presença (1) ou ausência (0) de uma determinada característica. Isto é conhecido como a primeira geração e é gerado aleatoriamente. O próximo passo consiste numa avaliação de cada indivíduo como uma solução possível, que é atribuída como um valor numérico conhecido como *fitness*. Poderá haver múltiplas maneiras de avaliar o indivíduo, mas isso dependerá sempre do problema a resolver e dos seus objetivos.

A avaliação de cada solução proposta é feita através da função de *fitness*. Após a avaliação, a população é submetida a um processo de seleção onde um grupo de indivíduos será escolhido para criar a próxima geração. Isso pode ser conseguido através de uma variedade de métodos como crossing over e mutação.

O resultado final deste passo é uma nova geração de novos indivíduos que resultou principalmente dos melhores indivíduos das gerações anteriores. As etapas anteriores serão repetidas em ciclo, em que cada geração seguinte será submetida aos vários métodos já mencionados, a fim de criar indivíduos melhores até encontrar uma solução que seja "suficientemente boa" ou até que um determinado número de gerações criadas. Esta solução geralmente é o indivíduo mais apto da geração final, e se o algoritmo foi bem sucedido, então essa qualidade deve ser a resposta ao problema que o utilizador está a tentar resolver.

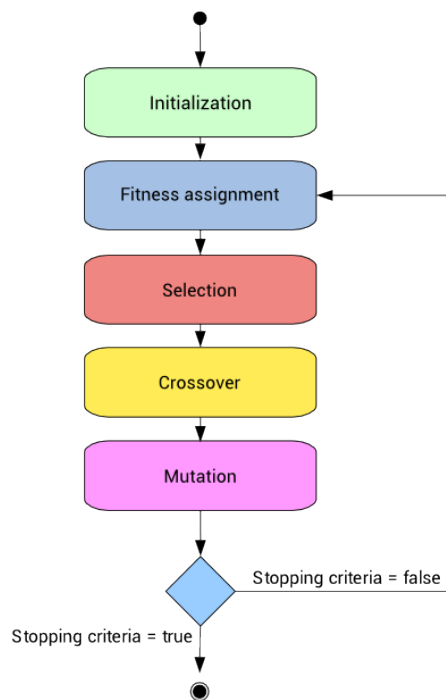


Figura 2. Diagrama de Estados de um Algoritmo Genético

3.2 Ferramentas de Desenvolvimento

Uma das ferramentas de desenvolvimento das GAs é o **JCLEC** (*Java Class Library for Evolutionary Computation*), um sistema de software para pesquisa de Computação Evolutiva, desenvolvido na linguagem de programação Java. Este software fornece uma estrutura de software de alto nível para fazer qualquer tipo de Algoritmo Evolutivo, oferecendo suporte para algoritmos genéticos, programação genética e programação evolutiva.

Outra das ferramentas é um componente de Algoritmos Genéticos e Programação Genética fornecido como uma estrutura Java, o **JGAP**. Ele fornece mecanismos genéticos básicos que podem ser facilmente usados para aplicar princípios evolutivos às soluções problemáticas. O JGAP foi projetado para ser de fácil utilização e intuitivo para os seus utilizadores.

O Genetic Algorithm ToolBox (GAT) usa funções da matriz **MATLAB** para construir um conjunto de ferramentas versáteis para implementar uma ampla gama de métodos de algoritmos genéticos. O GAT é uma coleção de rotinas, escrita principalmente em *m-files*, que implementam as funções mais importantes em algoritmos genéticos.

3.3 Soluções no Mercado

Algoritmos genéticos são usados principalmente em problemas de otimização de diferentes tipos, mas também são frequentemente usados noutras áreas de aplicação. Nesta seção, listamos algumas das áreas nas quais Algoritmos Genéticos são frequentemente usados. Estes são:

- **Otimização** - Algoritmos genéticos são comumente usados em problemas de otimização em que temos que maximizar ou minimizar um determinado valor de função objetivo sob um determinado conjunto de restrições.
- **Aplicações de negocio** – Utilizado em negociações financeiras, avaliações de credito, distribuição de orçamentos, detecção de fraudes.

- **Redes Neurais** - GAs também são usadas para treinar redes neurais, particularmente redes neurais recorrentes.
- **Processamento de imagem** - as GAs são usadas para várias tarefas de processamento de imagem digital (DIP), bem como a correspondência densa de pixels
- **Robot Trajectory Generation** - GAs foram usados para planejar o caminho que um braço do robô leva, movendo-se de um ponto para outro. Projeto paramétrico de aeronaves - GAs foram usados para projetar aeronaves, alterando os parâmetros e evoluindo para melhores soluções.
- **Análise de DNA** - GAs foram utilizados para determinar a estrutura do DNA usando dados espectrométricos (técnica de levantamento de dados físico-químicos através da transmissão, absorção ou reflexão da energia radiante incidente em uma amostra) sobre a amostra.
- **Problema de vendedor ambulante e suas aplicações** - GAs foram usados para resolver o TSP, que é um problema combinatório bem conhecido usando novas estratégias de cruzamento e embalagem.
- **Design** – design de redes neuronais, design e controlo de robôs.

As utilizações dos AG são muitas, todas relacionadas de uma forma ou de outra a uma análise multidimensional, onde se procura conseguir uma solução global. Estes algoritmos, vêm sendo utilizados em várias áreas de pesquisa e em situações do mundo real com bons resultados.

4 Artificial Neural Networks

Artificial Neural Networks (ANN) ou Redes Neuronais Artificiais (RNA), também conhecidas como sistemas computacionais representativos de conhecimento não simbólico (ou conexionista), apresentam um modelo matemático inspirado no sistema nervoso central do ser humano.

A influência do cérebro humano na elaboração deste tipo de sistema de aprendizagem é facilmente justificada se considerarmos que o cérebro é um sistema de processamento de informação altamente complexo, não-linear e paralelo. Através da aptidão em organizar os seus componentes estruturais, este revela a capacidade de realizar operações como reconhecimento de padrões ou controlo motor, muito mais rapidamente que qualquer computador digital existente. Por outro lado, o cérebro tem a capacidade de desenvolver as suas próprias regras através da sua experiência que vai adquirindo ao longo do tempo. Todas estas características tornam-se essenciais quando existe o objetivo de criar um sistema inteligente com capacidade de generalização, eficaz e com capacidade de adaptabilidade.

Posto isto, podemos definir uma RNA como uma estrutura interligada de unidades computacionais, designadas neurónios, com capacidade de aprendizagem. Estas unidades são constituídas por axónios, que correspondem à via de comunicação entre os neurónios, e por sinapses que representam o ponto de ligação entre axónio e neurónios. O valor da sinapse determina o peso (importância) do sinal a entrar no neurónio, e é a sua variação no tempo que determina a aprendizagem da RNA.

Tendo como base esta relação entre uma RNA e o cérebro humano, é fundamental identificar dois aspetos idênticos entre ambos e essências para compreender o funcionamento e processo de aprendizagem deste tipo de modelo: [5]

- O conhecimento é adquirido a partir de um ambiente, através de um processo de aprendizagem.
- O conhecimento é armazenado nas conexões, também designadas por ligações ou sinapse, entre nodos.

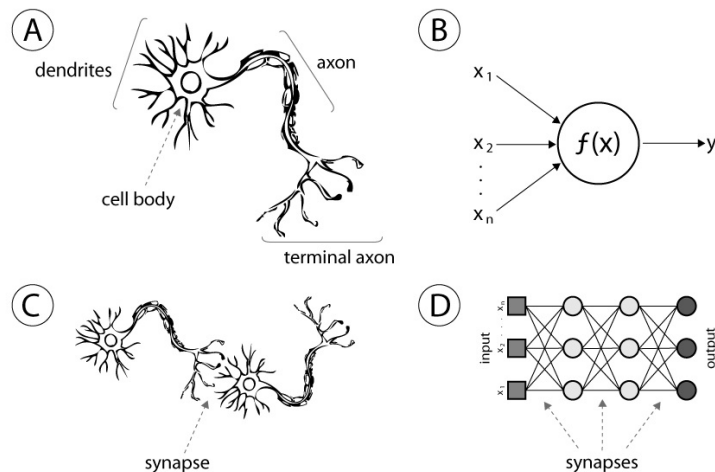


Figura 3. (A) Neurónio Humano; (B) Neurónio Artificial; (C) Sinapse Biológica; (D) Sinapses das RNA

4.1 Capacidade de Aprendizagem

Um dos objetivos deste tipo de sistema centra-se em criar um modelo com capacidade de resolver problemas com base em conhecimento passado ou dados sobre a resolução de outros problemas. Deste modo, a sua capacidade de aprendizagem torna-se fundamental para um melhor desempenho da rede.

Segundo Mendel e McClaren o conceito de aprendizagem é definido por: [5]

Aprendizagem é um processo pelo qual os parâmetros livres de uma rede neuronal são adaptados através de um processo de estimulação pelo ambiente na qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre.

Todo este processo de aprendizagem é executado a partir de um conjunto de regras bem definidas, que chamamos de algoritmo de aprendizagem ou treino. Uma vez que as RNAs são utilizadas em vários contextos, existe uma variedade de algoritmos de aprendizagem, cada um com as suas próprias vantagens, que diferem entre si principalmente na forma como os pesos são modificados e que são adaptados com base no objetivo final do problema em questão. Para além disso, os algoritmos são distinguidos também na forma com operam com o ambiente, tornando-se assim fundamental associar também o conceito de paradigma de aprendizagem.

Paradigmas de Aprendizagem

- Com supervisão: este paradigma consiste na presença de um “professor” que indica à rede a resposta desejada para um determinado padrão de entrada.
- Sem supervisão: está associado ao conceito de auto-organização uma vez que não é fornecida qualquer indicação sobre a resposta correta para o problema.
- De Reforço: neste tipo de aprendizagem também se assume a presença de um “professor”, no entanto este apenas indica se uma determinada resposta está correta ou errada.

Principais Regras de Aprendizagem

- *Hebbian*: esta regra propõe que os pesos das sinapses sejam ajustados se houver sincronismo entre os níveis de atividade das entradas e saídas. Se dois nodos de uma conexão são ativados simultaneamente, a força desta conexão é aumentada. Se estes são ativados de forma assíncrona, então a conexão é enfraquecida ou eliminada. Esta regra de aprendizagem é muito utilizada em aprendizagem sem supervisão.
- Competitiva: as saídas dos nodos da mesma camada competem entre si para se tornarem ativas, com apenas um nodo a ser ativado num dado instante.
- Estocástica: neste tipo de aprendizagem, os pesos são ajustados de um modo probabilístico.
- Baseada na memória: esta regra tem em consideração as experiências passadas, que são guardadas em memória e posteriormente aplicada em casos semelhantes.
- Gradiente decrescente: sempre que um determinado padrão é apresentado à rede, ela produz uma saída. Após calcular o erro entre o valor pretendido e o valor que sai da rede, as conexões são ajustadas de modo a reduzir esta distância. Esta técnica está diretamente ligada à aprendizagem com supervisão, e é muito utilizada.

4.2 Ferramentas de Desenvolvimento

Em seguida apresentamos um conjunto de algumas ferramentas que permitem o desenvolvimento de RNAs.

- **JustNN**: Uma ferramenta simples e intuitiva, que permite construir e testar redes neurais artificiais a partir de um conjunto de dados.
- **MatLAB**: Com o pacote *Neural Network Toolbox* fornece algoritmos e modelos capazes de criar, treinar, visualizar e simular redes neuronais. É possível realizar a classificação, regressão, agrupamento, redução de dimensão e modelagem do sistema.
- **R-Studio**: Através do pacote *neuralnet* permite construir e gerir redes neuronais, utilizando variações da técnica de backpropagation que consiste em utilizar a descida do gradiente para ajustar os parâmetros das redes e assim melhorar o desempenhos nos pares de entrada-saída.

4.3 Soluções no Mercado

No mercado atual é possível entrar um conjunto alargado de aplicações reais onde as redes neuronais artificiais são utilizadas. De seguida apresentamos alguns exemplos que utilizam este tipo de sistema de aprendizagem:

- Reconhecimento Facial – o facto de as redes neuronais estarem organizadas em camadas e possuírem uma grande capacidade de aprendizagem, tornam-se essenciais quando o objetivo é a obtenção de padrões. São também utilizadas no reconhecimento de voz e tratamento de imagens.
- Previsão no mercado financeiro – uma das áreas em que as RNAs estão associadas é nas variáveis, na arquitetura e no algoritmo de aprendizagem, as RNAs podem servir de apoio nas decisões de compra e venda de ações.
- Identificação de fraudes com cartão de crédito – um banco americano chamado Mellon Bank instalou um sistema de deteção de fraudes com cartão bancário baseado nas técnicas das redes neuronais e os prejuízos evitados foram bastante elevados. Desde aí outros bancos começaram a estes sistemas de modo controlar este tipo de fraudes. [7]

Para além destes exemplos, as redes neuronais artificiais estão ligadas a aplicações no âmbito da robótica, análise e processamento de sinais, aplicações meteorológicas, entre outras.

5 Conclusão

Os tópicos abordados ao longo de todo este documento permitem-nos compreender a importância dos sistemas de aprendizagem na elaboração de um sistema inteligente. Como referido anteriormente, cada um dos modelos descritos tem as suas vantagens e desvantagens e devem ser utilizados consoante o objetivo final que utilizador se propõe.

O primeiro método de aprendizagem aqui apresentado, Raciocínio Baseado em Casos, é dos mais intuitivos e fáceis de implementar. Este modelo deve ser adotado para sistemas de previsões ou justificações, no entanto, para garantir a sua coerência e robustez é necessário que exista um conjunto alargado de dados prévios já analisados.

Os sistemas de algoritmos genéticos são caracterizados pela sua eficiência e rapidez quando pretendemos obter uma lista de “boas” soluções e não apenas uma solução única. São úteis quando o espaço de pesquisa é alargado e existe uma grande quantidade de variáveis associadas. Por outro lado, é importante referir que os GAs não são adequados a todos os problemas uma vez que como é necessário fazer muitos cálculos, tornam-se computacionalmente dispendiosos. Como é um processo estocástico, não há garantias totais sobre a otimização e qualidade da solução.

Por fim, em relação às Redes Neurais Artificiais, tal como o modelo de Raciocínio Baseado em Casos, estas devem ser adotadas quando existe uma grande quantidade de dados para treinar a rede e assim poder obter um melhor desempenho. As principais vantagens identificadas são a sua capacidade de generalização, rapidez e a autoaprendizagem. Contudo, existem algumas limitações como treino demorado, ou seja, a solução pode demorar muito a ser apresentada. Por outro lado, torna-se impossível perceber como a rede chega a uma determinada conclusão uma vez que os critérios de decisão e a função objetivo do modelo são encriptados.

6 Bibliografia

1. Aamodt A., Plaza E., *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches*, in *AI Communications*, Vol. 7, Nº 1, pages 39-59, 1994
2. *Leadership and Personnel Management: Concepts, Methodologies, Tools, and Applications*, Usa, 4th Edition, February, 2016
3. David Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, 1989.
4. Tutorials Point - https://www.tutorialspoint.com/genetic_algorithms/index.htm
5. S. Haykin, *Neural Networks – A Comprehensive Foundation*. Prentice-Hall, New Jersey, 2nd Edition, 1999.
6. <http://bit.csc.lsu.edu/jianhua/quang.pdf>
7. <https://www.intechopen.com/books/artificial-neural-networks-architectures-and-applications/applications-of-artificial-neural-networks-in-chemical-problems>