Sobre a Real Contribuição de Modelos Cooperativos Simbólico-Conexionistas

Maria do Carmo Nicoletti[1]
UFSCar-DC
C.P. 676 - 13565-905 São Carlos – SP
Fone fax: (016) 261-8233
e-mail: marian@cse.unsw.edu.au

Júnia Coutinho Anacleto Silva[2]
UFSCar-DC
C.P. 676 - 13565-905 São Carlos SP
Fone fax: (016) 261-8233
e-mail: junia@dc.ufscar.br

Resumo: Este artigo investiga a real contribuição de modelos cooperativos simbólico-conexionistas de aprendizado supervisionado de máquina. Inicialmente são apresentadas as principais características de dois sistemas representativos que implementam esse tipo de colaboração, com o objetivo de subsidiar tal investigação. A seguir, a plausibilidade da colaboração simbólico-conexionista viabilizada em cada um dos sistemas e os eventuais benefícios que tal colaboração pode trazer são analisados, discutidos e questionados.

Palavras-chaves: inteligência artificial, aprendizado de máquina, redes neurais, aprendizado simbólico, modelos cooperativos de aprendizado de máquina.

1. Introdução

As limitações de cada um dos vários modelos de Aprendizado de Máquina — AM — e das técnicas disponibilizadas em cada um deles estão cada vez mais evidentes. Pode se verificar na literatura que a pesquisa em AM se direciona na busca de refinamentos de técnicas já consagradas e estabilizadas, com vistas a contornar e/ou eliminar algumas de suas fragilidades. É fato que por mais refinada e ajustada que uma determinada técnica de AM seja, ela é, entre outros, fortemente determinada pelo domínio de conhecimento em que atua, pelos diferentes tipos de informação de que faz uso e pelas linguagens de representação de conhecimento que emprega. Sob certos aspectos, tais fatores controlam e confinam a atuação da técnica não apenas a determinados domínios mas também ao aprendizado de apenas determinadas classes de conceitos.

Com o objetivo de ampliar o escopo de atuação de sistemas de aprendizado automático, é intuitivo e natural cogitar sistemas que buscam a promoção da diversidade de técnicas, ou disponibilizando maneiras que permitam uma colaboração entre elas ou então viabilizando a integração de algumas delas. Esta é uma das principais justificativas utilizadas para a proposta de sistemas cooperativos de aprendizado de máquina. Este artigo investiga algumas das possíveis colaborações/integrações entre o modelo conexionista e o simbólico, examinando dois sistemas representativos que implementam tais idéias, buscando delinear, para esses sistemas, os seus limites bem como evidenciar se efetivamente eles oferecem uma cooperação significativa, plausível e vantajosa.

O artigo está organizado da seguinte maneira: na Seção 2 são apresentados conceitos relativos a aprendizado de máquina, aprendizado neural e aprendizado simbólico, para contextualizar a área de pesquisa e subsidiar a discussão que se segue. As Seções 3 e 4 apresentam respectivamente as principais características do sistema KBANN [Towell (1991)] e do sistema TREPAN [Craven (1996)]. A Seção 5 discute as características operacionais de ambos sistemas, formas de representação de

[1] Financiado pela FAPESP proc. Nro. 98/05547-6
[2] Financiado pela FAPESP proc. Nro. 97/10072-4

conhecimento utilizadas e as possíveis perdas resultantes das transformações necessárias que o conhecimento representado deve sofrer, com o objetivo de viabilizar a cooperação. Além disso, busca identificar se efetivamente tais tipos de sistemas trazem algum benefício que justifique o investimento no seu desenvolvimento. A Seção 6 apresenta conclusões e linhas para a continuidade da investigação.

2. Aprendizado Empírico: Aprendizado Neural e Aprendizado Simbólico

Aprendizado empírico pode ser caracterizado como o aprendizado realizado por meio de generalizações a partir de um conjunto de exemplos (ou instâncias) do conceito a ser aprendido, conhecido como *conjunto de treinamento*. Cada exemplo do conjunto de treinamento é geralmente descrito como um vetor de pares *atributo-valor de atributo* e uma classe associada, classe esta que identifica o conceito ao qual o exemplo pertence. O fato da descrição do exemplo de treinamento incluir também o conceito ao qual o exemplo pertence caracteriza o chamado *aprendizado supervisionado*. Aprendizado simbólico e aprendizado neural são dois modelos de aprendizado supervisionado. O processo de indução realizado tanto por métodos conexionistas quanto por simbólicos é função do conjunto de treinamento.

O aprendizado usando métodos simbólicos pode ser abordado como uma busca que deve ser realizada no espaço das possíveis expressões do conceito a ser aprendido, de maneira a encontrar aquela(s) expressão(ões) simbólica(s) que "melhor" generaliza(m) o conjunto de treinamento fornecido. Neste contexto, dois dos principais critérios utilizados, que evidenciam a "melhor" hipótese são a precisão na classificação de novos exemplos e a compreensibilidade (por humanos) da expressão do conceito. Em métodos simbólicos a expressão do conceito aprendido geralmente é descrita como regras de produção e/ou árvores de decisão. O sucesso de algumas das implementações do modelo simbólico, tal como o sistema CN2 [Clark (1989)] comprova a efetividade de tal modelo.

Por outro lado, o aprendizado conexionista pode ser caracterizado como o processo de reajuste de pesos de conexões e de valores das funções de ativação nas unidades que definem a rede. O conceito aprendido por uma rede neural a partir de um conjunto de treinamento é a própria rede obtida após a fase de treinamento, representada por sua topologia, pelas funções de ativação e pelos pesos das conexões rede. Redes neurais são inerentemente paralelas, são tolerantes à falhas e geralmente têm um processamento eficiente.

Os primeiros trabalhos que simultaneamente abordam esses dois modelos de aprendizado supervisionado se limitam a comparar o desempenho de ambos (representados pelo ID3 [Quinlan-(1986)] e Backpropagation [Rumelhart et al.(1986)]) em vários domínios de conhecimento. Os resultados obtidos não foram conclusivos (ver, por exemplo, [Fisher (1989)] e [Mooney et al. (1989)]). De qualquer maneira, como comentado em [Shavlik (1994)], "... esses estudos não produziram resultados consistentes e a conclusão final deles é que redes neurais treinadas têm precisão pelo menos comparável à de árvores de decisão, em tarefas que podem ser consideradas simbolicamente orientadas."

O fato de redes neurais serem de difícil compreensibilidade é a principal justificativa usada por todos os métodos e algoritmos que se propõem a investir na combinação e/ou integração dos modelos simbólico e conexionista. Redes neurais são difíceis de serem interpretadas, após o seu treinamento. Em contrapartida, métodos simbólicos oferecem, como resultado, a expressão simbólica do conceito, que pode ser facilmente lida por um ser humano. O principal objetivo desta linha de pesquisa é fazer com que redes neurais ganhem um maior grau de aceitação por parte do usuário. A possibilidade de "visualizar" uma rede neural como um conjunto de regras teoricamente promove o entendimento da rede por parte do usuário, com possíveis ganhos na confiabilidade que tal usuário deposita na rede. Isso contribui para o fortalecimento de redes neurais como uma ferramenta de aprendizado e generalização.

Em [Craven (1993)] é comentado que " para muitos problemas para os quais compreensibilidade é importante, redes neurais fornecem uma generalização melhor do que os algoritmos simbólicos usuais. nesses domínios, é importante existir a possibilidade de extração de descrições compreensíveis de conceito, a partir de redes neurais." Acreditamos que tal comentário é, no mínimo, superficial e tendencioso. É superficial porque faz uma afirmação sem muita fundamentação e é tendencioso porque é utilizado como uma justificativa para o sistema que aqueles autores implementaram.

Não acreditamos que redes neurais precisam necessariamente ser explicadas, no sentido simbólico da palavra "explicação", para serem confiáveis. Mesmo porque o simples fato de uma rede ser "representada" simbolicamente não necessariamente implica o aumento de sua confiabilidade. Não acreditamos que uma regra para ser confiável, necessita ser completamente "entendida" por um especialista humano. Desconhecemos estudos que "quantificam" o quanto um ser humano compreende uma regra que envolve, por exemplo, 15 ou 20 condições (seja ela obtida por um sistema simbólico de aprendizado ou então, obtida como "tradução" de uma rede neural). Acreditamos que seres humanos não têm condições de avaliar completamente a compreensibilidade de tais regras. E mesmo que tivessem, nos perguntamos porque eles próprios não escreveram a regra, ao invés de usar um sistema de aprendizado de máquina para inferi-las.

As duas próximas seções apresentam os sistemas KBANN e TREPAN respectivamente. Devido à limitação de espaço, nos limitamos apenas a fornecer uma visão geral de cada um desses sistemas, buscando mostrar como o modelo cooperativo simbólico-conexionista foi implementado em cada um deles. Esses sistemas têm maneiras diferenciadas de abordar tal colaboração.

3. O Modelo KBANN

O modelo KBANN (Knowledge Based Neural Network) pode ser categorizado como um modelo tipicamente cooperativo, onde o aprendizado neural é usado para o refinamento do conhecimento simbólico preexistente. Neste modelo, o único tipo de aprendizado que efetivamente acontece, é o neural. O sistema KBANN, implementação do modelo, é constituído por três módulos, Inserção, Refinamento e Extração, mostrados na Figura 1 e descritos a seguir. O sistema KBANN recebe como entrada um conjunto de regras simbólicas e gera, como saída, um outro conjunto de regras simbólicas, refinamento do primeiro. A cooperação acontece quando o *backpropagation* é utilizado para refinar o conjunto inicial de regras simbólicas.

3.1 O Módulo de Inserção

Implementa um algoritmo que "traduz" o conjunto de regras simbólicas que é entrada para o sistema, em uma rede neural; cria uma rede neural que se comporta da mesma maneira (com relação à classificação), que o conjunto de regras simbólicas de entrada. Para fazer a tradução o módulo usa a correspondência mostrada na Tabela 1, o que permite definir a topologia da rede bem como o peso inicial das conexões.

A informação simbólica inicial é fornecida como um conjunto de cláusulas de Horn que obedecem uma determinada sintaxe, devendo ser acíclicas e proposicionais (no módulo seguinte a rede gerada é usada com o backpropagation, que não manipula variáveis).

O módulo traduz o conjunto de regras em uma rede neural fazendo inicialmente a tradução individual de cada regra em uma pequena subrede; cada uma destas subredes deve reproduzir, com precisão, o comportamento da regra da qual foi derivada. Feito isso as subredes são organizadas em uma rede única que reflete o comportamento de todo o conjunto de regras. O algoritmo inicialmente traduz um conjunto de regras em uma rede e, então, expande a rede obtida, de maneira que ela possa ser capaz de aprender conceitos não cobertos pelo conjunto inicial de regras. A Figura 2 mostra um

conjunto inicial de regras que representa um disjunto, sua reescrita e a transformação do novo conjunto de regras em uma rede, onde ω indica o peso da conexão. Todos esses passos são realizados pelo módulo de Inserção.

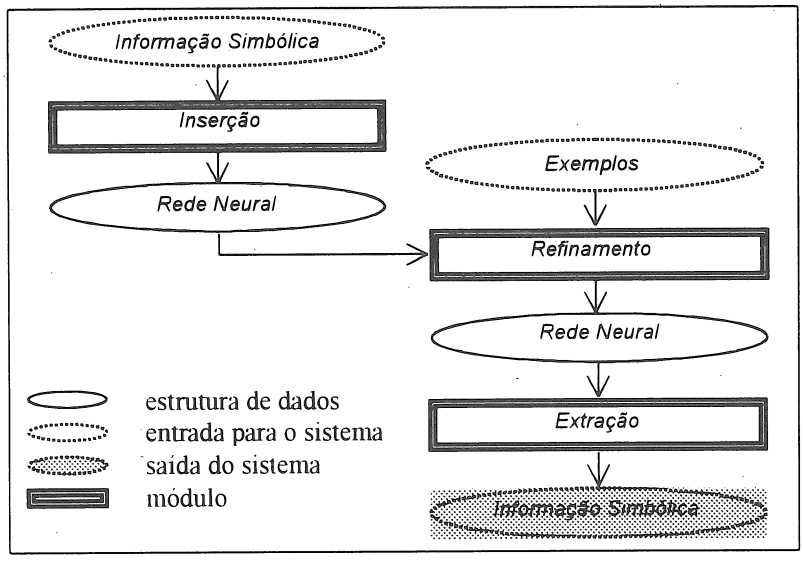

Figura 1. O sistema KBANN

Tabela 1. Correspondência entre uma base de conhecimento (expressa como conjunto de regras) e redes neurais

base de conhecimento	*rede neural*
conclusões finais	unidades de saída
fatos	unidades de entrada
conclusões intermediárias	unidades intermediárias
dependências	conexões ponderadas

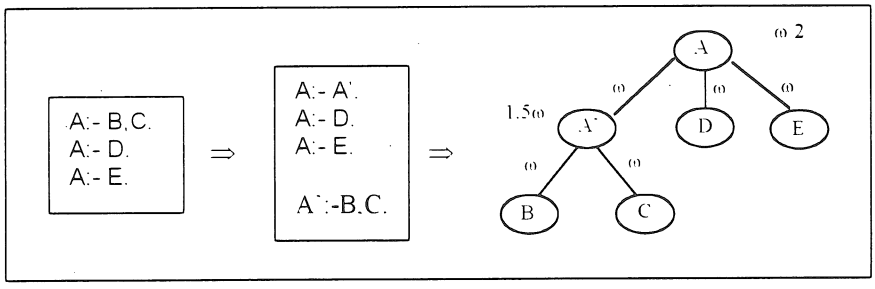

Figura 2 Exemplo de tradução de um conjunto de regras em uma rede neural

3.2 O Módulo de Refinamento

Faz um refinamento da rede criada pelo Módulo de Inserção, usando o algoritmo *backpropagation* e o conjunto de exemplos de treinamento fornecido ao sistema. Enquanto que o mecanismo de aprendizado é essencialmente o *backpropagation* padrão, a rede na qual ele opera não é padrão. É a rede criada pelo módulo Inserção e que deve, teoricamente, refletir o comportamento do conjunto de regras simbólicas. O fato da rede não ser padrão traz uma série de inconveniências quando do uso do *backpropagation*. Devido à maneira como ela foi criada, tal rede apresenta uma "resistência ao desaprendizado" (e

portanto, resistência ao desaprendizado de erros), problema conhecido como *flat spot* do *backpropagation*. A solução adotada pelo KBANN é aquela proposta em [Hinton (1989)]; essa solução é conveniente porque reduz os problemas que existem no início do treinamento, quando a rede gera respostas que estão possivelmente erradas, com alto grau de confiança.

3.3 O Módulo de Extração

Após o seu refinamento uma rede KBANN pode ser usada como um classificador bastante preciso; entretanto, continua apresentando o mesmo problema típico de redes neurais, comentado anteriormente na Seção 2, que é aquele de não fornecer explicação de como derivou a resposta. Este fato naturalmente impede que o resultado de seu aprendizado possa ser compartilhado ou entendido por seres humanos.

O Módulo de Extração tem por objetivo a tradução do conhecimento representado por uma rede KBANN, em regras simbólicas. Para que essa tradução possa ser realizada, algumas pré-condições necessitam ser válidas. A primeira pré-condição é a de que, após o seu refinamento, redes KBANN não mudam o significado de suas unidades. Essa suposição é importante para que métodos de extração de regras possam atribuir rótulos à unidades associadas a conseqüentes e antecedentes da regra, promovendo assim a compreensibilidade das regras a serem extraídas. A utilidade de rótulos é comprometida se o significado das unidades mudam após o refinamento da rede. A segunda pré-condição é a de que quase todas as unidades em uma rede KBANN treinada ou estão completamente ativadas (com ativação quase 1), ou então, estão completamente desativadas (com ativação quase 0). Essa suposição permite tratar cada uma das unidades que não for uma unidade de entrada, como uma função *step* ou uma regra booleana. Portanto, o problema de extração de regras se resume em identificar as situações nas quais a regra é verdade.

O KBANN disponibiliza dois métodos para a extração de regras a partir de redes neurais. O SUBSET, apresentado como uma variante, mas que é praticamente idêntico ao algoritmo KT [Fu (1991)] e o MofN, bem mais elaborado, que pode ser considerado uma evolução do SUBSET. O MofN tira vantagem do fato das regras geradas pelo SUBSET freqüentemente envolverem expressões m-de-n, i.e., expressões da forma *if (M dos seguintes N antecedentes são verdade) then*... Ambos os métodos buscam identificar quais são as combinações de valores de entradas, em uma unidade, que provocam sua ativação próxima de 1. A suposição de que todas as unidades na rede neural têm ativação perto de zero ou perto de um, simplifica essa tarefa, forçando conexões a terem sinal igual ao seu peso, ou então a não terem sinal. Isso faz com que o processo de extração de regras considere apenas os pesos das conexões e ignore a ativação da unidade a partir de onde se origina a conexão. O processo de extração de regras é ainda simplificado pelo fato que unidades sempre têm ativações não negativas. Isto permite que métodos de extração de regras assumam o peso da conexão como um indicador da maneira como o antecedente será usado. Conexões com pesos negativos podem apenas dar origem a antecedentes negados enquanto que conexões com pesos positivos podem apenas dar origem a antecedentes não negados. Isso reduz consideravelmente o tamanho do espaço de busca [Fu (1991)].

4. O Modelo TREPAN

O algoritmo TREPAN (TREes PArroting Networks) e o sistema que o implementa estão propostos em [Craven (1996)]. A justificativa para a proposta do TREPAN é a mesma utilizada pelos outros métodos que propõem uma colaboração simbólica-conexionista, i.e., a de desenvolver classificadores que, além de precisos, possam também produzir resultados facilmente inteligíveis por seres humanos. Entretanto, implementa tal colaboração de maneira totalmente diferenciada daquela do KBANN.

O TREPAN, particularmente, não "cuida" da precisão do conceito induzido, que fica a cargo do próprio treinamento da rede neural. O TREPAN busca, isso sim, tornar uma rede compreensível a seres humanos; sob esse enfoque pode ser tratado mais como um algoritmo de aprendizado, que aprende a *tradução* de uma rede neural para uma árvore de decisão, num processo que se assemelha bastante àquele usado por vários algoritmos simbólicos de aprendizado indutivo de máquina (ver por exemplo o ID3 [Quinlan (1986)]).

A tarefa de aprendizado à qual o sistema TREPAN se propõe está representada na Figura 3 concentrando a descrição nas entradas necessárias e saída esperada. O processo de extração de regras a partir de uma rede neural, realizado pelo TREPAN, é diferenciado dos demais métodos existentes, inclusive daqueles propostos pelo SUBSET e MofN, uma vez que é equacionado e abordado como um processo de aprendizado. O TREPAN pode ser considerado um sistema de aprendizado cujo objetivo é aprender a função que está representada por uma rede neural. A linguagem de representação usada para expressar as possíveis hipóteses aprendidas, que expressam aquela função é a linguagem de árvores de decisão.

Note na Figura 3 que o treinamento da rede (*backpropagation*) é um processo à parte, que não faz parte do TREPAN. Entretanto, o sistema espera como entradas tanto a rede treinada quanto o conjunto de instâncias usado no treinamento. O sistema TREPAN é composto pelos módulos TREPAN, que cuida do aprendizado da árvore de decisão a partir da rede neural, e do ORÁCULO, que colabora com a tarefa de aprendizado. Uma das principais funções do ORÁCULO é a de providenciar mais exemplos de treinamento, de maneira a viabilizar uma melhor "tradução" da rede na árvore de decisão m-de-n.

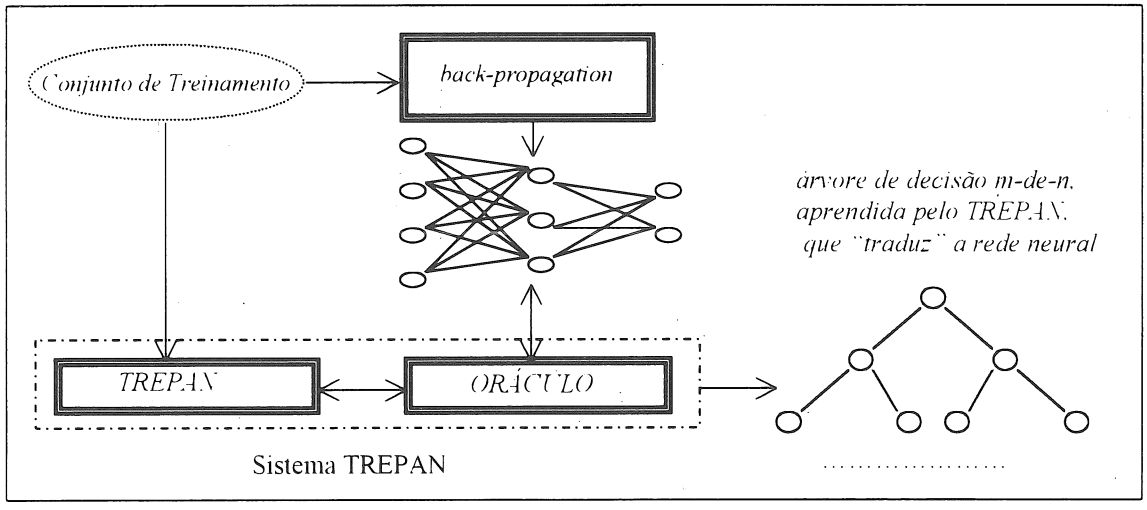

Figura 3. Esquema de funcionamento do TREPAN

O módulo TREPAN constrói a árvore de decisão recursivamente particionando o espaço de instâncias; implementa um processo de expansão de um nó que em muito se assemelha àquele usado por algoritmos convencionais de indução de árvores de decisão: um teste para a ramificação é selecionado para o nó e, para cada resultado do teste, um nó-filho é criado. Cada um desses nós-filhos ou se torna uma folha na árvore ou então é colocado numa fila para expansão futura. Embora o TREPAN tenha muitas similaridades com algoritmos convencionais de indução de árvores de decisão, ele é diferente em alguns aspectos, principalmente naquele relacionado ao uso de um oráculo que colabora com a tarefa de aprendizado.

A interação do sistema TREPAN com a rede se dá através do modulo ORÁCULO, que pode ser invocado para a) reescrever o conjunto de treinamento, substituindo a classe de cada instância pela

classe que a rede atribui à instância; b) criar mais instâncias de treinamento, com o objetivo de obter uma árvore de decisão mais precisa. O ORÁCULO é versátil o suficiente para criar instâncias que satisfazem a determinadas restrições em valores de atributo. Ele gera uma instância completa, selecionando randomicamente valores para cada um daqueles atributos, ao mesmo tempo em que garante a observância das restrições.

Segundo [Craven (1996)], a grande vantagem do aprendizado usando o ORÁCULO é a de que a informação pode ser coletada no momento em que for necessária, durante o aprendizado.

5. Discutindo a Cooperação Simbólico-Conexionista do KBANN e TREPAN

A discussão que será conduzida nesta seção aborda tanto aspectos gerais de ambos os sistemas, focalizando a cooperação simbólico-conexionista, quanto algumas peculiaridades de algoritmos que tais sistemas implementam. Devido à limitação de espaço, os vários algoritmos implementados por ambos os sistemas não foram apresentados (podem ser encontrados em várias das referências listadas no final deste artigo).

O primeiro ponto a considerar nesta discussão foi já mencionado na Seção 1 e diz respeito à principal justificativa que motivou o desenvolvimento dos sistemas KBANN e TREPAN, i.e., a de promover a compreensibilidade do aprendizado conexionista, via abordagem simbólica de uma rede neural. Ambos os sistemas cuidam de traduzir uma rede neural em uma representação simbólica. O quanto de informação pode ser perdida na mudança de representação, as restrições que uma rede deve obedecer para viabilizar tal tradução (caso do sistema KBANN) e o quão "compreensível" as correspondentes regras são, são fatores que devem ser seriamente considerados, antes da utilização de tais sistemas.

O modelo KBANN é bem mais restrito no sentido que usa a colaboração simbólico-conexionista apenas com o objetivo de refinamento da informação simbólica, inicialmente disponível ao sistema. A pré-condição para o uso de tal sistema é ter disponível um conhecimento expresso em regras, sobre um determinado(s) conceito(s), bem como um conjunto de treinamento, com instâncias que representam o(s) conceito(s) em questão. O sistema parte desse conhecimento simbólico, o traduz para uma rede neural, treina a rede com o conjunto de treinamento fornecido e traduz a rede treinada, de maneira a voltar a ter o conhecimento expresso simbolicamente. A criação de uma rede neural a partir de regras já restringe a utilização do sistema, bem como torna os algoritmos que implementam a tradução rede-regra, convenientes apenas para uso em redes específicas, i.e., redes criadas a partir de regras simbólicas.

O modelo KBANN vale mais pelo pioneirismo, do que efetivamente pela sua contribuição como um método efetivo e plausível de aprendizado de máquina. Ao tentar promover a compreensibilidade de redes neurais, a proposta KBANN restringe de maneira dramática o tipo da rede, isto é, promove a compreensibilidade de redes que, a princípio, já eram compreensíveis, dado que foram geradas a partir de um conjunto de regras. E mais, as próprias regras iniciais devem já ser restritas, dado que devem ser proposicionais e aciclicas. E, ainda assim, para viabilizar a cooperação, o sistema precisa fazer concessões, usar uma série de restrições, bem como deixar a cargo do usuário a decisão de alguns aspectos cruciais, aparentemente, sem qualquer orientação de como ele deve proceder.

Na descrição do principal algoritmo implementado pelo módulo de Inserção, dois dos passos estão descritos muito superficialmente em qualquer das referências que apresenta tal algoritmo. A descrição que trata da adição de unidades intermediárias, apenas informa qual é a função desse passo e comenta que a sua execução provê uma rede KBANN com a habilidade de aprender características que não comparecem no conjunto inicial de regras. Como comentado em [Towell (1991)], pg. 20, "Devido ao

fato que unidades intermediárias nem sempre são necessárias, elas são apenas adicionadas mediante solicitação do usuário. Esta solicitação deve especificar o número e a distribuição entre os níveis estabelecidos no passo anterior à adição de unidades." É importante observar aqui que o usuário de tal sistema deve ter uma compreensão bastante profunda do que está acontecendo, para saber se deve (ou não) informar o sistema que um determinado número de unidades intermediárias serão necessárias e, mais ainda, deve saber como o passo anterior do algoritmo, que cuida de rotular as unidades, funciona. Uma vez que existem quatro estratégias diferentes para a atribuição de rótulos às unidades, o usuário precisa também estar informado qual a adotada pela implementação que está usando.

A descrição do passo que trata da adição à rede, de unidades de entrada, sofre dos mesmos inconvenientes já mencionados. A maneira como ele é descrito deixa dúvidas de como efetivamente esse passo é implementado pelo sistema. Não é mencionado como o sistema identifica a necessidade de novas unidades de entrada e de como essas unidades são criadas. Além disso, fica a dúvida de qual símbolo tais unidades de entrada representam, se é que representam algum símbolo. Como deve, então, ser interpretado o conjunto de regras que será gerado no final, que eventualmente faz uso das unidades (intermediárias e/ou de entrada) introduzidas pelo usuário?

O exemplo fornecido pelo autor para motivar a descrição de ambos os passos não esclarece muito pois assume, para facilidade de entendimento, que o usuário do sistema solicitou a introdução de uma unidade intermediária e duas de entrada. Como o usuário detectou tal necessidade e como descobriu o nível com que tal nó intermediário deveria ser introduzido, não é mencionado.

Apesar do módulo de Refinamento ser uma implementação direta do algoritmo *backpropagation*, foram necessárias alterações no cálculo da função do erro, de maneira a evitar o problema do *flat spot* causado pela abordagem discretizada (0 ou 1) de construção da rede KBANN pelo módulo Inserção. Estamos chamando a atenção para esse fato justamente para evidenciar o quanto o autor investiu em pesquisa para tentar encontrar as soluções que contornassem os problemas que o seu modelo de colaboração simbólica-conexionista apresentou.

Para viabilizar a tradução da rede em um conjunto de regras, o modelo KBANN exige que o treinamento realizado pelo módulo de Refinamento não tenha mudado significativamente o significado das unidades (não especificando exatamente o que é uma mudança significativa e/ou fornecendo critérios para avaliar se isso aconteceu), bem como assume que todas as unidades da rede refinada estão ativadas ou desativadas, suposição altamente conveniente (e altamente limitante também) quando se tem em mente a criação de regras. Esta última suposição praticamente exige que a rede seja um conjunto de regras, só que organizado de maneira um pouco diferenciada.

Todas as restrições, indefinições e convenientes suposições nos levam a questionar a efetiva colaboração que o uso de um tal sistema pode trazer, quando aparentemente, a vantagem principal que tem é a de refinar um conjunto inicial de regras. Até que ponto um usuário potencial estaria em condições de fornecer as informações exigidas pelo sistema, saberia conduzir a sua execução, sabendo ao certo se deve (ou não) fornecer novas unidades intermediárias (e seu correspondente nível) ou então unidades de entrada, ao sistema. Enfim, o que nos perguntamos é se um usuário se arriscaria a refinar um conjunto de regras usando um procedimento que, eventualmente, poderia comprometer a qualidade das regras. E nos perguntamos, também, o quanto de refinamento tal modelo pode introduzir nas regras originais. O refinamento, além da rede, é função também do conjunto de treinamento, i.e., dependendo das instâncias desse conjunto, os resultados podem não trazer qualquer refinamento.

Apesar de se propor apenas à extração de regras a partir de redes neurais, o modelo de cooperação proposto pelo TREPAN é mais geral que o KBANN, no sentido que não impõe restrições quanto ao tipo de rede sobre a qual opera. O TREPAN pode ser abordado sob duas perspectivas: como um

sistema simbólico de aprendizado de máquina, que aprende árvores de decisão m-de-n, contando com a cooperação de uma rede neural para realizar esse aprendizado ou então como um sistema que "traduz" uma rede em uma árvore. Isso se deve ao fato do Módulo TREPAN usar, para o aprendizado da árvore, as instâncias de treinamento utilizadas no treinamento da rede. Tais instâncias, entretanto, não são exatamente as mesmas. Para elas serem utilizadas pelo Módulo TREPAN, elas são "corrigidas" (pelo ORÁCULO), i.e., a classe original é substituída pela classe que a rede atribui à instância. O sistema TREPAN nada mais é do que um sistema convencional de aprendizado de árvores de decisão m-de-n. Como as instâncias utilizadas para o aprendizado da árvore são instâncias que tiveram a classe atualizada pela classe que a rede lhes atribuiu, a árvore construída tende a refletir o comportamento da rede. O TREPAN oferece um pouco mais ainda pois permite a geração de mais exemplos de treinamento, quando necessário. Apesar da proposta ter uma certa originalidade, deixa dúvidas com relação à sua efetiva contribuição. O fato de um conceito poder ser expresso como uma árvore (facilmente traduzida como um conjunto de regras de produção) não necessariamente implica que a compreensibilidade do conceito aumenta. Em domínios cujas instâncias de treinamento são descritas por atributos sub-simbólicos, por exemplo, a compreensibilidade da árvore estará comprometida.

6. Conclusões

Apesar de acharmos que ambos os sistemas discutidos não apresentam contribuições significativas a área de aprendizado de máquina, uma vez que estão sujeitos a uma série de limitações e restrições com resultados nem sempre satisfatórios, acreditamos ser ainda prematuro abandonar definitivamente a colaboração simbólico-conexionista proposta por esses dois modelos. Uma avaliação empírica do uso de tais sistemas em um número bem maior de domínios (do que a realizada pelos autores) é ainda necessária para subsidiar melhor qualquer decisão. Apesar disso, é patente o domínio restrito de aplicabilidade do modelo KBANN e é questionável que a cooperação proposta por ele seja realmente efetiva. Além disso, embora os dois sistemas discutidos sejam representativos de dois modelos de colaboração simbólica-conexionista, eles não são os únicos. O estudo e pesquisa descritos neste artigo terão continuidade no exame e avaliação de outras propostas. Entretanto, não pode ser esquecido que modelos cooperativos simbólico-conexionistas precisam ser mais gerais para efetivamente serem de maior utulidade.

7. Referências

[Clark (1989)] Clark, P.; Niblett, T. The CN2 Induction Algorithm. Machine Learning 3, 1989, pp. 261-283.

[Craven (1993)] Craven, M.W.; Shavlik, J. W. Learning Symbolic Rules Using Artificial Neural Networks. In: Proceedings of the 10th ICML, MA, Morgan Kaufmann, 1993, pp 73-80.

[Craven (1996)] Craven, M. W. Extracting Comprehensible Models from Trained Neural Networks. Ph.D. thesis, University of Wisconsin , 1996, 199 pg.

[Fisher (1989)] Fisher, D. H.; McKusick, K. B. An Empirical Comparison of ID3 and Backpropagation. In Proceedings of the 11th IJCAI, Detroit: Morgan Kaufmann, 1989, pp 788-793.

[Fu (1991)] Fu, M.L. Rule Learning by Searching on Adapted Nets. Proc. of the 9th NCAI, CA, pp. 590-595.

[Hinton (1989)] Hinton, G. Connectionist Learning Procedures. Artificial Intelligence, 1986, 40, pp. 185-234.

[Mooney et al.(1989)] Mooney, R.; Shavlik, J.; Towell, G.; Grove, A. An Experimental Comparison of Symbolic and Connectionist Learning Algorithms. In: Proceedings of the 11th IJCAI, Detroit: Morgan Kaufmann, 1989, pp 775-780.

[Quinlan (1986)] Quinlan, J.Q. Induction of Decision Trees. Machine Learning, 1, 1986, pp. 81-106.

[Rumelhart et al. (1986)] D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning Internal Representations by Error Propagation. In: D. E. Rumelhart and J. L. (eds.) Parallel Distributed Processing: Explorations in the Microstructure of Cognition., MIT Press, 1986, pp. 318-363.

[Shavlik (1994)] Shavlik, J.W. Combining Symbolic and Neural Learning. Machine Learning, 14, 1994, pp. 321-331.

[Towell (1991)] Towell, G.G. Symbolic Knowledge and Neural Networks: Insertion, Refinement and Extraction. Doctoral dissertation, University of Wisconsin, 1991.

[Towell (1993)] Towell, G.G.; Shavlik, J. W. Extracting Refined Rules from Knowledge-based Neural Networks. Machine Learning, 13, 1993, pp. 71-101.