

Extração e Representação de Conhecimento de Redes Neurais Artificiais Utilizando o Reticulado Conceitual Iceberg e Extração de Regras de Implicação no Método FCANN

Sérgio M. Dias¹, Luis E. Zárate², Bruno M. Nogueira³, Newton J. Vieira¹

¹Universidade Federal de Minas Gerais (UFMG)

Av. Antônio Carlos 6627 - ICEX - 4010 – 31.270-010 – Belo Horizonte – MG – Brasil

²Pontifícia Universidade Católica de Minas Gerais (PUC MG)
Belo Horizonte – MG – Brasil

³Universidade de São Paulo (USP)
São Carlos – SP – Brasil

mariano@dcc.ufmg.br, zárate@pucminas.br, brunomn@icmc.usp.br

Resumo. *Uma vez que o conhecimento adquirido por Redes Neurais Artificiais não é trivialmente comprehensível pelos humanos, diversas iniciativas têm surgido para extrair conhecimento destas redes. Neste trabalho, destaca-se o método FCANN, o qual utiliza Análise Formal de Conceitos para extrair e representar conhecimento interpretável das redes neurais. Entretanto, a extração de regras neste método apresenta um alto custo computacional e, muitas vezes, apresenta regras cuja interpretabilidade seja difícil. Assim sendo, este trabalho propõe a extração de regras de implicação de reticulados conceituais construídos a partir de itemsets frequentes, aumentando a escalabilidade do método e a clareza das regras para o usuário. Essa nova proposta foi aplicada a um sistema de energia solar (termossifão), obtendo bons resultados.*

1. Introdução

Redes Neurais Artificiais (RNAs) têm sido muito utilizadas para a representação de processos reais complexos, especialmente industriais. Estas redes possuem a capacidade de obter relações entre os parâmetros de entrada e saída de um domínio, nas quais parâmetros de entrada são mapeados em parâmetros de saída da rede por uma função implícita. Este comportamento torna as RNAs uma “caixa preta” e nenhuma informação pode ser explicitamente obtida de sua estrutura interna.

Diversas pesquisas são encontradas na busca de extrair conhecimento de RNAs, como, por exemplo, [Tickle et al. 1998],[Benítez et al. 1997] e [Craven 1996]. Recentemente, a Análise Formal de Conceitos (AFC), uma poderosa ferramenta para representação de conhecimento, vem sendo utilizada com sucesso para extrair conhecimento de RNAs previamente treinadas [Vimieiro et al. 2005].

Dentre os métodos que utilizam a AFC para extração de conhecimento de redes neurais, destaca-se, aqui, o método FCANN [Zárate et al. 2008], o qual extrai relações qualitativas aprendidas pela RNA, independente de sua estrutura, obtendo bons resultados em diversos domínios de aplicação [Zárate and Dias 2009]. Este método aplica o algoritmo *Next Closure* [Ganter and Stumme 2003] para a obtenção de regras de implicação

do tipo “SE ... ENTÃO” a partir de uma base canônica não redundante ou mínima, denominada *Stem Base* ou *Duquenne-Guigues Base*. Entretanto, a obtenção dessa base mínima não é trivial, sendo pouco encontrada em casos práticos [Godin et al. 1986] e com custo computacional proibitivo para muitas aplicações [Carpinetto and Romano 2004].

Além da questão computacional outros problemas são observados. Não existe uma relação direta entre o conjunto mínimo de regras (*stem base*) e o reticulado conceitual, o que pode dificultar uma posterior análise dessas regras. Frequentemente, também é observado uma grande quantidade de regras e conceitos formais, o que dificulta sua análise visual. Outro problema é a dificuldade de derivar logicamente o conjunto mínimo de regras utilizando axiomas para inferir novas regras [Carpinetto and Romano 2004].

Assim, esse trabalho aborda os dois principais problemas observados no método FCANN: 1) a dificuldade de extrair um grande conjunto de conceitos formais e regras, e 2) a dificuldade de análise do conhecimento obtido através do conjunto mínimo de regras e do reticulado conceitual. Para o primeiro problema, propõe-se a construção do reticulado conceitual usando *itemsets* frequentes (reticulados *Iceberg* [Stumme et al. 2002]), o qual provê um limiar no número de conceitos formais. Já para o segundo problema, extraem-se regras de implicação do reticulado conceitual.

Este artigo está dividido da seguinte maneira: na Seção 2, são apresentados alguns conceitos primordiais para o entendimento deste trabalho. Na Seção 3, apresenta-se o método FCANN. Na Seção 4, as melhorias do método FCANN aqui propostas são mostradas, seguida por um estudo de caso apresentado na Seção 5. Por fim, algumas conclusões sobre este trabalho são apresentadas na Seção 6.

2. Conceitos Iniciais: Análise Formal de Conceitos e Algoritmo *Next Closure*

Nesta seção, são apresentados dois conceitos essenciais para o total entendimento deste trabalho: a Análise Formal de Conceitos e o Algoritmo *Next Closure*.

2.1. Análise Formal de Conceitos

A Análise Formal de Conceitos (AFC) [Ganter and Wille 1999] é um método matemático voltado para a representação de conhecimento por meio de diagramas de linha, os quais são representações de um reticulado conceitual. O conceito inicial de AFC se refere aos contextos formais. Estes são representados por tabelas cruzadas e possuem a notação (G, M, I) , na qual G é um conjunto de objetos (linhas da tabela), M é um conjunto de atributos (colunas) e I é a relação de incidência ($I \subseteq G \times M$). Se um objeto $g \in G$ possui um atributo $m \in M$, estes estão na relação I , representando-se por gIm ou $(g, m) \in I$.

Dado um conjunto de objetos $A \subseteq G$ de um contexto formal (G, M, I) , pode-se obter os atributos de M que são comum a todos os objetos. Da mesma forma, pode-se obter, para um conjunto $B \subseteq M$, quais objetos possuem os atributos de B . Estas operações são possíveis usando os operadores de derivação, mostrados na Equação 1.

$$A' = \{ m \in M \mid gIm \text{ } g \in A \} \quad B' = \{ g \in G \mid gIm \text{ } m \in B \} \quad (1)$$

Diversos algoritmos podem ser aplicados a contextos formais (tabelas cruzadas) para obter seus conceitos formais e seus diagramas de linha. Conceitos formais são pares (A, B) , nos quais $A \in G$ é denominado extensão e $B \in M$ intenção. Quando o conjunto de todos

os conceitos formais de um contexto formal (G, M, I) é ordenado hierarquicamente, este conjunto é denominado Reticulado Conceitual, com a notação $\underline{\beta}(G, M, I)$. Dois conceitos formais possuem a relação hierárquica $(A_1, B_1) \leq (A_2, B_2)$ quando $A_1 \subseteq A_2$ e $B_2 \subseteq B_1$.

2.2. Algoritmo *Next Closure*

Sistemas de fechamento (*closure systems*) são, de maneira simples, conjuntos de conjuntos. Na Equação 2, apresenta-se um sistema de fechamento C dado um conjunto M :

$$C \subseteq \mathfrak{P}(M), M \in C, D \subseteq C \rightarrow \cap D \in C \quad (2)$$

na qual $\mathfrak{P}(M)$ é o conjunto potência do conjunto M e $\cap D \leftarrow \{x | \forall S \in D, x \in S\}$

Operadores de derivação da AFC podem ser considerados, por suas características, como operadores de fechamento [Vimieiro et al. 2005]. Assim, dois conjuntos são considerados fechados se $B'' = B, B \subseteq M$.

Como um conjunto de subconjuntos de um conjunto tem cardinalidade exponencial (2^n subconjuntos), o desenvolvimento de algoritmos que verificam se um fechamento já foi gerado resulta em um algoritmo de complexidade exponencial. Assim, é necessário gerar fechamentos em uma ordem predefinida para evitar essa verificação, a qual é a ideia principal do algoritmo *Next Closure* [Ganter and Stumme 2003]. Considerando $A \subseteq M$ e $m_i \in M$, os passos deste algoritmo podem ser resumidos da seguinte forma:

1. Remova, de A , todos os elementos maiores que $m_i(A \cap \{m_1, \dots, m_{i-1}\})$;
2. Una o conjunto de resultado do passo 1 ao conjunto $\{m_i\}$;
3. Aplique o operador de fechamento $(.)''$ ao conjunto resultante do passo 2.

3. Método FCANN para Extração de Conhecimento de RNAs

Nesta seção, os passos do método FCANN para extração de conhecimento de RNAs previamente treinadas é discutido brevemente. Assume-se, aqui, uma rede neural *perceptron* em múltiplas camadas, *feedforward*, totalmente conectada, com n entradas e $m = 1$ saída.

1. Selecione um conjunto representativo de dados para treinar a rede (Equação 3):

$$X = [x_{ij}]k \times n \quad (3)$$

no qual x_{ij} para $i = 1, \dots, k$ e $j = 1, \dots, n - 1$ são os parâmetros de entrada e x_{in} para $i = 1, \dots, k$ a saída, a qual deve ter a distribuição de probabilidade conhecida.

2. Defina a estrutura da RNA, com n entradas, h camadas e m saídas, e treine-a.
3. Construa uma base de dados sintética, considerando os limites dos domínios dos parâmetros de entrada. A base de dados sintética é construída para operar a rede neural treinada e obter os valores do parâmetro de saída, tentando revelar o conhecimento adquirido. Esta base pode ser definido conforme a Equação 4:

$$Y = [y_{ij}]p \times n - 1 \quad (4)$$

na qual Y possui elementos gerados e p representa o número de registros (Equação 7). Cada parâmetro de entrada possui valores mínimos e máximos conhecidos. O vetor W (Equação 5) define o número de intervalos que vai ser gerado para cada parâmetro, entre os valores mínimos e máximos:

$$W = [W_j]; j = 1, \dots, n - 1 \quad (5)$$

Assim, o intervalo de variação de cada parâmetro na composição da base sintética pode ser expresso conforme a Equação 6:

$$Int = \{I_1, I_2, \dots, I_{n-1}\} \text{ onde } I_j = \frac{|v_j - u_j|}{w_i} \text{ for } j = 1, \dots, n - 1 \quad (6)$$

Os valores de cada parâmetro usados para gerar a base de dados sintética podem ser representados por:

$$S = \begin{bmatrix} S_{1,1} & S_{1,2} & \cdots & S_{1,n-1} \\ S_{2,1} & S_{2,2} & \cdots & S_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{w_1,1} & S_{w_2,2} & \cdots & S_{w_{n-1},n-1} \end{bmatrix}$$

Pode-se observar que o número de conjuntos p a serem gerados depende do número de dados para cada parâmetro, tal como na Equação 7

$$p = W_1 \times W_2 \times \dots \times W_{n-1} = \prod_{i=1}^{n-1} W_i \quad (7)$$

4. Apresente a base sintética Y à rede para obter o parâmetro de saída $Z = [z_{ij}]_{p \times 1}$, p qual tem de possuir a mesma função de probabilidade já conhecida. Se $e_{x,z} = |\bar{x}_1 - \bar{z}_2|$ e $e_{S(x,z)} = |S(x)_1 - S(z)_2|$ representarem grandes erros, volte ao passo 1.
5. Classifique os parâmetros da matriz $U = [Y, Z]_{p \times n}$ em intervalos discretos.
6. Construa uma tabela cruzada do contexto formal para discretizar as n variáveis, estabelecendo uma relação binária de incidência entre objetos e atributos.
7. Obtenha o conceito formal e construa seu diagrama de linha.
8. Aplique o algoritmo *Next Closure* a fim de obter o conjunto de regras denominado *Duquenne-Guigues Base*, composto por regras do tipo “SE ... ENTÃO”.

4. Melhorias Propostas para o Modelo FCANN

Nesta seção, são apresentadas as melhorias adotadas a fim de suprir os problemas detectados no método FCANN. O primeiro problema detectado remete-se à escalabilidade do método, o qual define um contexto formal (G, M, I) , no qual $|M| = \sum_i^{n+m} D_i$, $|G| = p$, $n+m$ é o número de parâmetros (entrada e saída) da RNA e D_i é o intervalo de discretização utilizado em cada parâmetro. O segundo problema detectado é a falta de correlação direta entre as regras extraídas e o reticulado conceitual, dificultando a análise do conhecimento obtido por meio do conjunto mínimos de regras e do reticulado.

Para resolver o primeiro problema, propõe-se a adoção associações entre *item-sets* frequentes e conceitos formais frequentes [Stumme et al. 2002]. Reticulados conceituais construídos apenas com conceitos formais frequentes são conhecidos na literatura como reticulados conceituais *Iceberg* [Stumme et al. 2002]. Considerando um conceito formal (X, Y) , o conjunto $Y \subseteq M$ (e (X, Y)) será frequente se, e somente se, $|Y| > minSup$ [Carpinetto and Romano 2004], no qual $minSup$ é um limite de suporte mínimo.

Como a frequência de cada atributo $m_i \in Y$ é fixa, adota-se um peso adicional para estes atributos, o qual corresponde à frequência de m_i em um contexto formal

construído com o conjunto de dados original. Para tal, utilizam-se os mesmos intervalos de discretização utilizados no passo 5 do FCANN. Assim, considera-se um conceito formal (X, Y) frequente se, e somente se, os conceitos formais são frequentes e $(|Y| + weight(m_i)) > minSup$. É importante observar que *itemsets* e conceitos frequentes são bastante correlacionados. O suporte de qualquer *itemset* Y é igual ao suporte de seu fechamento Y'' , o que garante que quaisquer *itemsets* frequentes são unicamente determinados por conceitos frequentes [Stumme et al. 2002].

O segundo problema, referente à falta de correlação direta entre as regras extraídas e o reticulado conceitual, é aqui abordado utilizando regras de implicação diretamente do reticulado conceitual. Uma regra de implicação $A \rightarrow B$ pertence a um contexto formal (G, M, I) se, e somente se, $(B', B'') \geq (A', A'')$ [Carpineto and Romano 2004]. Entretanto, o conjunto de regras extraído pode ser redundante [Maier 1983]. Essas redundâncias, entretanto, podem ser removidas em tempo polinomial [Carpineto and Romano 2004].

Para gerar todos os conceitos formais frequentes e extrair regras de implicação, pode-se utilizar a ferramenta Sophiann [Zárate et al. 2006b], bem como os algoritmos *Frequent Next Neighbors* e *Find Implications* [Carpineto and Romano 2004].

5. Estudo de Caso

Nesta seção, é apresentado um estudo de caso de extração de regras de um sistema real utilizando o método FCANN modificado. Para tal, escolheu-se o sistema termosifão [Zárate et al. 2006a], o sistema mais utilizado de aquecimento solar de água. Sua eficiência é influenciada por parâmetros operacionais e de instalação. A extração de conhecimento aqui visa compreender o comportamento desses parâmetros.

5.1. Representação Neural

Temperatura de entrada da água (Tin), irradiação solar (G), temperatura do ambiente ($Tamb$), taxa de fluxo (\bar{m}), inclinação do coletor solar (I) e altura do tanque de armazenamento de água (H) são variáveis utilizadas como entradas da rede neural. Todas essas variáveis são mapeadas na temperatura de saída da água ($Tout$), conforme a Equação 8:

$$f(Tin, G, Tamb, \bar{m}, I, H) \xrightarrow{RNN} (Tout) \quad (8)$$

Utilizou-se uma rede neural do tipo *perceptron* em múltiplas camadas (MLP), *feedforward*, uma camada intermediária e com neurônios de camadas consecutivas totalmente conectados. Para o treinamento, utilizou-se o algoritmo *backpropagation*, com função log-sigmoide de ativação nos neurônios. Como sugerido em [Kovács 1996], utilizou-se $2N + 1$ neurônios na camada intermediária, na qual N é o número de entradas da rede neural. Portanto, a estrutura da rede possui $N = 6$ neurônios na camada de entrada, 13 neurônios na camada intermediária e $M = 1$ neurônio na camada de saída.

Para o processo de treinamento, 117 instâncias foram coletadas do sistema, sendo 90% utilizadas no conjunto de treinamento, que continha as instâncias com os valores máximos e mínimos de cada parâmetro e outras escolhidas aleatoriamente. Os dados foram normalizados no intervalo $[0.2, 0.8]$ para facilitar a convergência da rede neural, como sugerido em [Altincay and Demirekler 2002] e mostrado na Equação 9, na qual L_n é o valor normalizado, L_o o original e L_{min} e L_{max} o mínimo e o máximo do parâmetro.

$$Ln = \frac{(Lo - Lmin)}{(Lmax - Lmin)} \quad e \quad Lo = Ln * Lmax + (1 - Ln) * Lmin \quad (9)$$

Para sistemas de termossifão, o máximo erro médio na saída ($Tout$) recomendado é de 1°C, sendo este o critério adotado para parada do treinamento da RNA. Na Tabela 1, é possível observar os valores de erro obtidos tanto no treinamento quanto no teste.

Tabela 1. Resultados de treinamento e teste da rede neural

Erro (°C)	Treinamento	Teste
Mínimo	0.01	0.16
Máximo	1	3.38
Média	0.32	0.99
Desvio Padrão	0.18	1.15

5.2. Extração de Conhecimento Utilizando o Método FCANN Modificado

Após o treinamento satisfatório da rede neural, passa-se à aplicação do método FCANN modificado. Para fins de comparação da eficiência, a configuração original do método FCANN também foi aplicada à rede neural. Na Tabela 2 é possível observar o número de objetos, atributos, conceitos e regras (*Stem Base*) obtidas em relação ao número de valores por parâmetro e intervalos de discretização utilizando o FCANN original. Pode-se observar que a partir de 6 valores por parâmetros e intervalos de discretização, não foi possível obter regras devido ao custo computacional. Além disso, é possível perceber que o número de conceitos e regras cresce rapidamente, inviabilizando a correta aprendizagem. Com isso, mostra-se uma necessidade de adoção de valores limiares, podando o reticulado conceitual e extraíndo regras apenas de conceitos formais mais relevantes.

Tabela 2. Número de objetos, atributos, conceitos e regras obtidos

	Dados por parâmetro e intervalos de discretização						
	2	3	4	5	6	7	8
Objetos G	64	729	4096	15625	46656	117649	262144
Atributos M	14	21	28	35	42	49	56
Número de conceitos	1084	8174	34396	109205	282829	643944	1303887
Número de regras	80	1039	5667	22266	-	-	-

Adotando as modificações, aplicou-se um limiar de suporte mínimo ao contexto formal. Ao incrementar o suporte mínimo, o número de regras decresce, filtrando apenas os conceitos formais mais relevantes e, consequentemente, as regras de implicação mais relevantes. A Figura 1 mostra, em escala logarítmica, esta relação.

Com isso, consegue-se um compromisso entre o suporte mínimo, a qualidade das regras e, consequentemente, do conhecimento extraído. Além disso, é possível aumentar a escalabilidade do método FCANN. Em relação ao conjunto de regras extraídas, apesar de este não ser um conjunto mínimo, completo e não-redundante, este pode ser facilmente analisado pelo usuário. Além disso, apresenta regras mais representativas, uma vez que foram extraídas apenas dos conceitos formais mais relevantes. Outro ganho remete-se à possibilidade da análise das regras diretamente do reticulado conceitual.

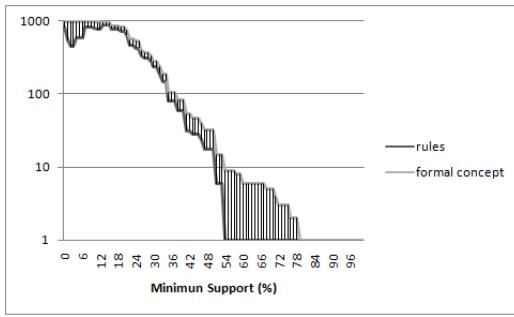


Figura 1. Relação entre números de conceitos formais, regras e suporte mínimo

Para o contexto formal deste exemplo, 18 regras de implicação foram obtidas, adotando um suporte mínimo de 50%. Foram obtidas, por exemplo, as seguintes regras:

If $G=8$ Then $I=3$ **If $M=3$ Then $I=3$** **If $G=8$ Then $H=8$**
If $M=3$ Then $H=8$ **If $G=8$ Then $H=3$** **If $M=3$ Then $H=3$**

No reticulado conceitual *Iceberg* gerado para este exemplo, não mostrado aqui por limitações de espaço, somente os conceitos formais mais representativos estão presentes, existindo uma relação direta entre as regras e o reticulado. Além das 18 regras obtidas, outras podem ser inferidas por implicações lógicas ou por análise visual do reticulado.

6. Conclusões

Neste trabalho, adaptou-se o método FCANN para dar-lhe maior escalabilidade e extrair regras mais compreensíveis. Foi possível, com essas modificações, reduzir a complexidade do reticulado conceitual com a associação entre *itemsets* frequentes e conceitos formais frequentes, propondo-se, assim, a representação do conhecimento extraído das redes neurais por meio do reticulado conceitual *Iceberg*. Utilizando um suporte mínimo, foi possível selecionar os conceitos formais frequentes, obtendo um compromisso entre o conhecimento extraído, a representatividade das regras e o suporte mínimo.

Ao utilizar o reticulado conceitual *Iceberg*, apenas conceitos formais mais relevantes são representados, o que possibilita, em muitos casos, a visualização do reticulado. A visualização do reticulado é adequada para processos de aprendizagem no qual o usuário procura compreender algum processo através do método FCANN. Outra vantagem da visualização do reticulado é permitir facilmente a inferência de novas regras.

Utilizou-se, também, a extração de regras de implicação diretamente do reticulado conceitual. Essa extração, quando comparada com a extração do conjunto mínimo de regras, apresenta algumas vantagens. Extraí-se uma quantidade menor de regras, apenas as mais relevantes são selecionadas e existe uma relação direta entre as regras e o reticulado conceitual, facilitando a aprendizagem. Entretanto, o conjunto de implicações é redundante, podendo essas redundâncias serem eliminadas em tempo polinomial.

Por fim, é importante destacar que as mudanças propostas para o método FCANN podem ser adotadas por outros métodos para reduzir a complexidade do reticulado conceitual. Além disso, é possível adotar outros paradigmas de construção de reticulados.

Referências

- Altincay, H. and Demirekler, M. (2002). Why does output normalization create problems in multiple classifier systems? In *In Proc. ICPR*, pages 775–778.
- Benítez, J. M., Castro, J. L., and Requena, I. (1997). Are artificial neural networks black boxes? *IEEE Transactions on Neural Networks*, Vol. 8:1156–1164.
- Carpinetto, C. and Romano, G. (2004). *Concept Data Analysis: Theory and Applications*. John Wiley & Sons, England.
- Craven, M. W. (1996). *Extracting Comprehensible Models from Trained Neural Networks*. PhD thesis, University of Wisconsin-Madison.
- Ganter, B. and Stumme, G. (2003). Formal concept analysis: Methods and applications in computer science. Technical report, Otto - von - Guericke - Universitat Magdeburg.
- Ganter, B. and Wille, R. (1999). *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, Germany.
- Godin, R., Saunders, E., and Gecsei, J. (1986). Lattice model of browsable data spaces. *Inf. Sci.*, 40(2):89–116.
- Kovács, Z. L. (1996). *Redes Neurais Artificiais*. Edição Académica São Paulo, Brasil.
- Maier, D. (1983). *The Theory of Relational Databases*. Computer Science Press.
- Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., and Lakhal, L. (2002). Computing iceberg concept lattices with titanic. *Data and Knowledge Engineering*, 42:189–222(34).
- Tickle, A. B., Andrews, R., Golea, M., and Diederich, J. (1998). The truth is in there: Directions and challenges in extracting rules from trained artificial neural networks. *IEEE Trans Neural Networks*, pages 1057–1068.
- Vimieiro, R., Zárate, L. E., Pereira, E. M. D., and Vieira, N. J. (2005). Using the nextclosure algorithm to extract rules from trained neural networks application in solar energy systems. In *IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications SMCia*, pages 184– 189, New York. IEEE express.
- Zárate, L. E. and Dias, S. M. (2009). Qualitative behavior rules for the cold rolling process extracted from trained ann. *Engineering Applications of Artificial Intelligence*, In Press, Corrected Proof(-):–.
- Zárate, L. E., Dias, S. M., and Song, M. A. J. (2008). Fcann: A new approach for extraction and representation of knowledge from ann trained via formal concept analysis. *Neurocomputing*, 71:2670–2684.
- Zárate, L. E., Pereira, E. M. D., Oliveira, L. A. R., Gil, V. P., Santos, T. R. A., and Nogueira, B. M. (2006a). Techniques for training sets selection in the representation of a thermosiphon system via ann. pages 2736–2741. in Proc IJCNN.
- Zárate, L. E., Song, M. A. J., de Castro B. S. Alvarenga, B., Nogueira, B. M., Alvarez, A. A., Dias, S. M., Santos, T. R. A., and Vimieiro, R. (2006b). An approach to knowledge extraction from ann through formal concept analysis - computational tool proposal: Sophiann. In *Proceedings of the (ISIE) 15th International Symposium on Industrial Electronics*, pages 43–48. IEEE express.