

USO DE REDES GSN EM TAREFAS DE ORDENAÇÃO (RANKING) APLICADAS À MEDIDA DE ADIMPLÊNCIA DE CLIENTES

Júnio César da Silva^{1,2}, Weber Martins^{1,3}

¹Universidade Federal de Goiás. Escola de Engenharia Elétrica. Grupo PIRENEUS

²Universidade Salgado de Oliveira. Coordenação de Análise de Sistemas.

³Universidade Católica de Goiás. Departamento de Psicologia.

Goiânia – GO

E-mails: jcesar, weber@pireneus.eee.ufg.br

Abstract

Boolean neural networks are known for their low cost and ease of hardware implementation [1]. Networks of Goal Seeking Neurons (GSN) [2] encompasses an interesting class and were designed to solve problems from networks of Probabilistic Logic Neurons (PLN) [3], successors of the very first Boolean model.

The use of neural networks in financial applications is being consolidated through the last decade. This paper proposes a GSN based system to cope with ranking problems. We are concerned with the ranking of clients with respect to financial solvency.

On the contrary of more popular models, GSN networks have a one-shot learning algorithm (where each example is shown only once) combined with the non-destruction of acquired knowledge. These features lead to inability to learn non-representative patterns and show it is possible to define the similarity degree of a client with respect to the whole class. The proposed system aims, therefore, to define how solvent and insolvent a client is in relation to the universe of classified clients presented to the network. The proposed system also implements earlier research [4] on the treatment of false examples by GSN control of presentation order.

1. Introdução

Este artigo propõe um modelo para ordenação (“ranking”) que faz uso da característica das redes GSN de rejeitar padrões que estão em desacordo com os padrões aprendidos anteriormente pela rede. Demonstramos que é possível criar um mapeamento das características de um determinado cliente na forma de um número que expressa a medida de similaridade do padrão apresentado em relação ao universo do conjunto de todos os padrões.

O primeiro modelo booleano foi desenvolvido por Aleksander [5, 6], sendo basicamente uma memória RAM que utiliza o princípio das n-tuplas [7]. Seu

sucessor, PLN, buscou a inserção de caráter não determinístico para as respostas neuronais, possuindo posições internas de valor indefinido (valor “u”), onde probabilisticamente a saída é definida. O neurônio GSN difere do PLN pela possibilidade de produzir como saída o valor indefinido “u”, que pode ser interpretado como ‘0’ ou ‘1’. O neurônio pode, portanto, endereçar mais de uma posição interna. Este endereçamento múltiplo agrupa padrões similares, economiza memória e impede que o neurônio fique saturado rapidamente. Além disso, o GSN possui um algoritmo rápido de treinamento, denominado de “único passo” (one-shot learning), que rejeita pares de entrada-saída que rompem (entram em contradição) com a informação previamente armazenada no neurônio. Ao contrário do PLN, o algoritmo de execução do GSN disponibiliza uma forma determinística de definição das saídas [2, 10].

O neurônio GSN pode estar em um de três estados: validação, aprendizagem e execução. No estado de validação, o neurônio decide se é possível fazer o mapeamento entrada-saída corrente. Durante a fase de aprendizado, a prioridade é dada à reutilização de posições de memória do neurônio, o que aumenta a generalização. Se não existe uma posição de memória que faça o mapeamento entrada-saída desejado, o neurônio mapeia utilizando uma posição de memória não utilizada (com valor “u”). Uma vez modificada do valor “u”, tal posição não é colocada a disposição para novas alterações. No estado de execução, são utilizadas regras simples para produzir a saída do neurônio (por exemplo, o valor de saída do neurônio será o valor de maior ocorrência nas suas posições internas).

O aprendizado de redes GSN feedforward com arquitetura piramidal tem semelhanças com redes backpropagation [8]. No estado de validação, um padrão de entrada é apresentado à rede que é estimulada a responder “u”, caso exista uma célula do conteúdo endereçado com o valor “u”, ou existam ‘0’ e ‘1’ no conjunto endereçado. O neurônio irá responder ‘1’ ou ‘0’, somente se todos os valores do conjunto endereçado forem ‘1’ ou ‘0’ respectivamente (ver Figura 1.1).

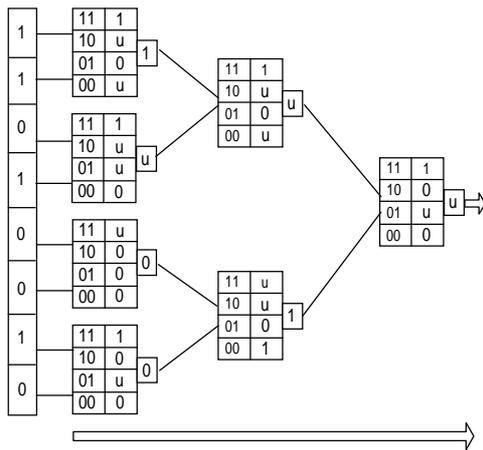


Figura 1.1: Rede no estado de validação de um padrão

Se a saída da rede corresponder ao valor de saída desejado pelo padrão apresentado, ou for igual a “u”, isto significa que a rede pode fazer o mapeamento entre a entrada-saída corrente. Então, cada neurônio entra no estado de aprendizado e o valor de saída desejado é propagado pela rede no sentido do nível de saída para o de entrada. Quando o neurônio do nível de saída define qual endereço de memória será utilizado para armazenar o valor de saída, este endereço é propagado para os neurônios dos níveis anteriores como a saída desejada destes neurônios (ver Figura 1.2). Este ciclo irá se repetir até o nível de entrada ser alcançado.

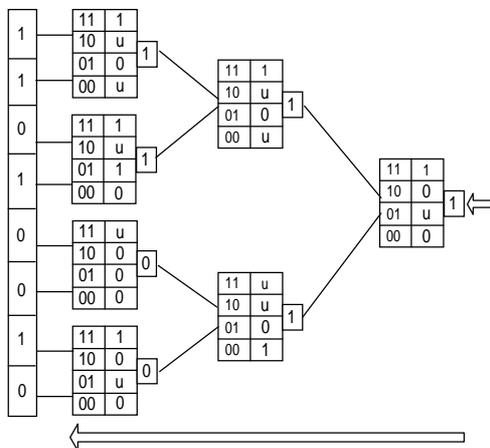


Figura 1.2: Rede no estado de armazenamento de um padrão

2. Sistema Proposto

O classificador GSN originalmente proposto [2] é constituído por várias redes. O processo de classificação consiste na submissão de um conjunto de padrões de treinamento às redes numa ordem fixa de apresentação. Nesta fase, a rede fará o mapeamento dos padrões com as saídas desejadas. Após o ciclo de treinamento de cada rede, sua capacidade de reconhecimento é avaliada

mediante a apresentação de padrões que não fizeram parte do conjunto de treinamento e na avaliação da capacidade das redes treinadas em reconhecer os padrões apresentados. Normalmente, para cada bit do código de classe dos padrões, são treinadas várias redes.

O sistema proposto aproveita-se apenas das características de treinamento seqüencial de redes GSN, exercendo controle sobre a ordem de apresentação dos padrões. O que nos interessa é a posição típica onde cada padrão é aprendido, que será obtida através de várias abordagens de treinamento.

Um outro diferencial que propomos é a forma do tratamento de falsos exemplos. Redes GSN são bastante sensíveis quanto à ordem de apresentação dos padrões de treinamento. Falsos exemplos apresentados à rede prematuramente podem inviabilizar totalmente a performance de um classificador [9]. O sistema de “ranking” aqui proposto trata este assunto inspirado em idéias conhecidas [4] para lidar com classificadores submetidos a falsos exemplos.

Alguns mecanismos de reforço/punição foram implementados. No primeiro deles (ver Figura 2.1a), denominado “saltos constantes” (constant shifts), os padrões que são aprendidos saltam na ordem de apresentação (a ser seguida pela próxima rede) para posições mais distantes do início. Em contrapartida, padrões que não são aprendidos saltam para posições de apresentação mais próximas do início. O salto para os padrões aprendidos é maior que o salto para os padrões não aprendidos sendo que estes valores de salto são constantes. Tal estratégia resulta numa alta velocidade de migração dos exemplos aprendidos para o final da fila e numa baixa velocidade de migração dos exemplos não aprendidos para o início da fila. Esta técnica faz com que padrões que diferem muito do conjunto, que denominamos de “exigentes” (talvez, falsos exemplos), somente sejam aprendidos se apresentados no início da fila de apresentação no treinamento da rede.

Na segunda abordagem (ver Figura 2.1b), denominada “saltos variáveis” (variable shifts), os padrões que são aprendidos saltam para o início da fila levando-se em consideração sua posição atual. Para padrões que conseguiram ser aprendidos e estão no final da fila de apresentação os saltos são maiores. À medida que estes padrões vão se aproximando do início da fila o tamanho dos saltos vai diminuindo. O inverso acontece aos padrões que não conseguem ser aprendidos. Neste caso, estes padrões vão migrando para o final da fila de apresentação com saltos maiores quando estão próximos ao início da fila de apresentação. Se estiverem se aproximando do final da fila, os saltos serão menores. Além disso, esta abordagem adota uma barreira que impede que os padrões que estão com dificuldades em serem aprendidos alcancem uma posição muito próxima do início da fila de apresentação.

Na terceira abordagem, foi adotado um tratamento aleatório na apresentação dos exemplos às redes, não

levando em consideração se o exemplo foi aprendido ou não. Em outras palavras, o treinamento de uma rede não interfere na de outra, abrindo, inclusive, espaço para treinamentos concorrentes (em paralelo). Neste caso, interessa-nos a quantidade de vezes em que cada exemplo foi aprendido.

As políticas de tratamento de falsos exemplos são exibidas na figura 2.1:

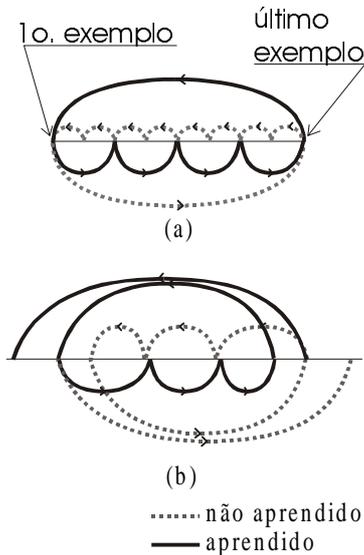


Figura 2.1: Políticas de tratamento de exemplos.

2.1 Dinâmica do Treinamento no Contexto

Os padrões foram apresentados a um sistema composto por 1000 redes. Inicialmente, todos os neurônios da rede têm suas células de memória inicializadas com o valor “u”. Após esta inicialização, a rede obtém de forma aleatória 16 bits do padrão de entrada, sendo 2 bits para cada neurônio, pois estamos usando neurônios com duas entradas somente. A rede entra no estado de validação e cada neurônio, com base em suas entradas, produz sua saída que é propagada para os níveis posteriores até que o neurônio do último nível produza a saída da rede. Caso a saída seja igual a “u” ou igual ao valor desejado pelo padrão, este padrão será armazenado na rede. O endereço da célula do neurônio de saída, onde o padrão foi armazenado, será propagado para os neurônios dos níveis anteriores até se alcançar o nível de entrada. Uma matriz de resultados (Figura 2.2) é alimentada sempre que um padrão é aprendido, de forma a tornar possível o cálculo da posição média de aprendizagem dos padrões.

Id. Padrão	Quantidade de vezes aprendido	Σ posições onde o padrão foi aprendido
1	12	234
2

Figura 2.2: Matriz de resultados

Após todos os padrões serem apresentados à rede, a fila de apresentação dos padrões para a próxima rede será construída segundo um dos algoritmos apresentados na seção 2.

Finda a apresentação dos padrões a todo o conjunto de redes, obtém-se uma matriz da posição média onde cada padrão foi aprendido (Tabela 2.1).

Tabela 2.1: Matriz das médias

Id.	Posição Média	Nr. Aprendidos
1	180,0265	603
2	148,8661	605
3	151,8905	603
4	180,9934	604
5	189,836	622
6	133,773	577
7	187,4541	621
8	140,7334	574
9	190,2612	605
.	.	.
.	.	.

A matriz representada na Tabela 2.1, quando classificada pela posição média, produz o “ranking” dos padrões. Os padrões com a posição média baixa são padrões que tiveram dificuldade de serem aprendidos pelas redes, indicando serem padrões que diferem do conjunto apresentado (da classificação previamente definida).

3. Experimentos

A topologia de rede utilizada no sistema foi a de uma pirâmide GSN com 3 níveis. Estas redes possuem 8 neurônios na camada de entrada e acessam 16 bits do padrão de entrada. (ver Figura 3.1).

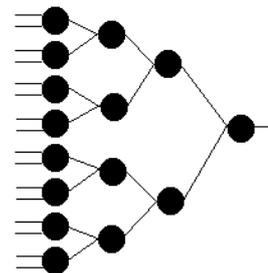


Figura 3.1: Topologia da rede utilizada nos experimentos.

3.1 Dados Fictícios

Foram utilizados dois conjuntos de dados fictícios que estão dispostos em forma de um mapa de bits de 16 por 11. As formas dos padrões de entrada são exibidas na Figuras 3.2 e 3.3 (“a” são protótipos de clientes

adimplentes e “b” são protótipos de clientes inadimplentes). O uso de dados fictícios é importante para validar o sistema proposto, tendo em vista que conhecemos a resposta desejada do mesmo.

Um algoritmo de inserção de ruído foi aplicado aos protótipos, gerando uma massa de dados de 200 padrões sendo 100 da classe adimplente e 100 da classe inadimplente com a seguinte configuração para cada classe: 35 padrões com 10% de ruído, 35 padrões com 20% de ruído, 20 padrões com 30% de ruído e 10 padrões com 40% de ruído. Tal distribuição baseia-se na suposição de que os bons exemplos são mais numerosos.

Nos experimentos com dados fictícios, a identificação dos padrões seguiu a política exibida nas tabelas 3.1 e 3.2. O motivo de estabelecermos dois conjuntos de padrões é para explicitarmos a independência com relação ao rótulo inicial.

Tabela 3.1: Primeiro conjunto de padrões de treinamento

Identificação	Nível de ruído	Classe
001-035	10%	Adimplente
036-070	20%	Adimplente
071-090	30%	Adimplente
091-100	40%	Adimplente
101-135	10%	Inadimplente
136-170	20%	Inadimplente
171-190	30%	Inadimplente
191-200	40%	Inadimplente

Tabela 3.2: Segundo conjunto de padrões de treinamento

Identificação	Nível de ruído	Classe
001-010	40%	Adimplente
011-045	10%	Adimplente
046-080	20%	Adimplente
081-100	30%	Adimplente
101-110	40%	Inadimplente
111-145	10%	Inadimplente
146-180	20%	Inadimplente
181-200	30%	Inadimplente

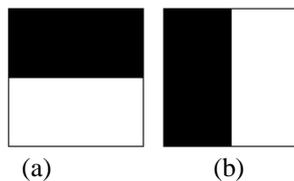


Figura 3.2: Protótipos de clientes adimplentes e inadimplentes. (a) adimplente, (b) inadimplente.

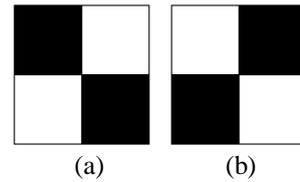


Figura 3.3: Protótipos de clientes: (a) adimplente, (b) inadimplente.

O gráfico das posições médias onde os padrões foram aprendidos, utilizando-se o algoritmo de tratamento de falsos exemplos denominado saltos constantes, está exibido na Figura 3.4 e 3.5. Nas figuras 3.6 e 3.7 estão os gráficos das médias das médias e desvio padrão referentes aos gráficos 3.4 e 3.5 respectivamente. Verificou-se que exemplares que diferem muito do conjunto de padrões (possuem maior ruído) exigem ser apresentados sempre no início do treinamento da rede, ou seja, sua posição média de aprendizado pela rede será normalmente inferior à dos padrões com menos ruído. Isto significa que estes padrões possuem características que os diferenciam significativamente dos padrões da classe utilizada para rotulá-los.

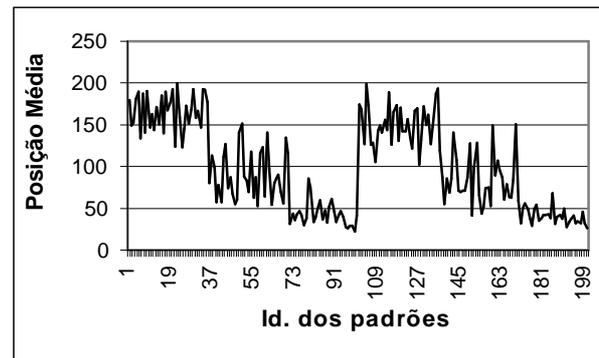


Figura 3.4: Posição média de apresentação dos padrões. Padrões com maior ruído por último.

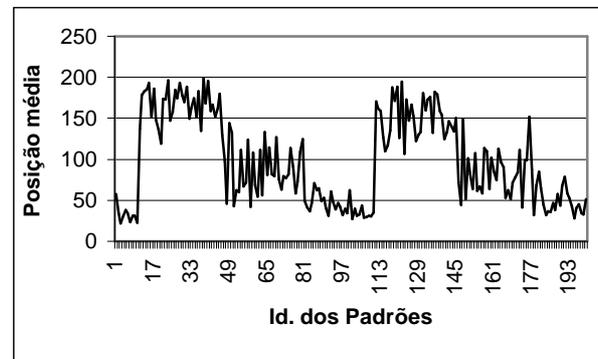


Figura 3.5: Posição média de apresentação dos padrões. Padrões com maior ruído primeiro

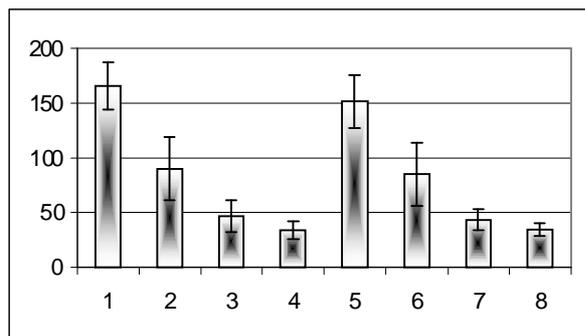


Figura 3.6: Média das médias (barras) e desvio padrão (intervalo). Padrões com maior ruído por último.

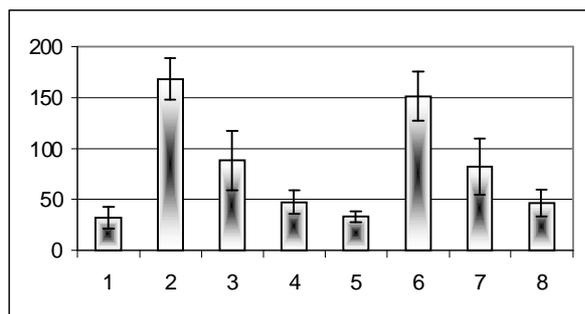


Figura 3.7: Média das médias (barras) e desvio padrão (intervalo). Padrões com maior ruído primeiro.

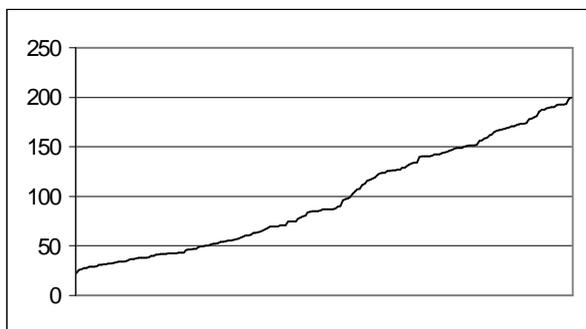


Figura 3.8: Posição média (de forma ordenada) de apresentação dos padrões. Dados Fictícios

3.2 Resultados com Dados Fictícios

Os resultados dos experimentos utilizando dados fictícios demonstram que o sistema é capaz de criar corretamente um ranking dos padrões. Neste ranking, padrões que diferem do conjunto apresentado às redes (possuem maior ruído) apresentam posição média de aprendizado inferior à posição média dos padrões que possuem características semelhantes. Isto indica que estes padrões necessitam ser apresentados às redes nas primeiras posições para serem aprendidos. Este mecanismo, portanto, é apto para efetivamente ordenar padrões segundo suas características específicas.

3.3 Dados Reais

Demonstrado que o mecanismo proposto é capaz de fazer a detecção de padrões que diferem do conjunto de padrões apresentado, vamos utilizá-lo sobre uma massa de dados de clientes de uma instituição financeira. Estes dados referem-se a informações cadastrais de contas garantidas (cheque especial) e registram algumas características dos clientes tais como: classe (adimplente/inadimplente), tempo de conta corrente, quantidade de cheques devolvidos, utilização média do cheque especial, limite de crédito, quantidade de garantias, quantidade de excessos ao limite durante o contrato, quantidade média de talões usados pelo cliente, etc.

A massa de dados utilizada é composta por um conjunto de 30000 clientes adimplentes e 287 clientes inadimplentes. Perceba, portanto, que se trata de duas classes representadas de forma bastante desequilibrada. As informações dos clientes foram transformadas em uma matriz de bits de 16 por 11.

Uma vez que existe uma disparidade muito grande na quantidade de amostras de clientes adimplentes e inadimplentes, o sistema escolhe aleatoriamente 100 clientes de cada classe para treinamento das redes.

O mesmo processo de treinamento do sistema, descrito no tópico 2.1, foi aplicado aos dados reais. Os resultados dos experimentos estão apresentados nas Figuras 3.9, 3.10 e 3.11.

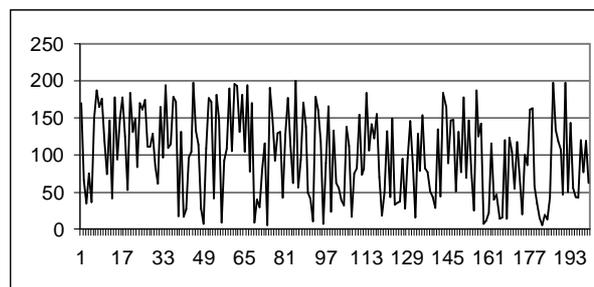


Figura 3.9: Posição média de apresentação dos padrões. Dados Reais

A idéia da aplicação do sistema em dados financeiros se prende à necessidade das instituições analisarem o risco das operações de crédito. O mecanismo proposto é útil uma vez que permite, dado um conjunto de dados de informações de clientes inadimplentes e adimplentes da instituição, verificar o grau de “adimplência” ou “inadimplência” de um determinado cliente em relação ao conjunto de clientes. Um cliente mesmo com status de adimplente pode, segundo seu ranking, denotar que possui características que o distinguem dos clientes adimplentes típicos, o que indica a necessidade por parte da instituição financeira de obter maiores garantias do cliente para lhe conceder crédito.

Estes clientes são grandes causadores de prejuízo às instituições, uma vez que pertencem à classe dos inadimplentes e, portanto, conseguem crédito.

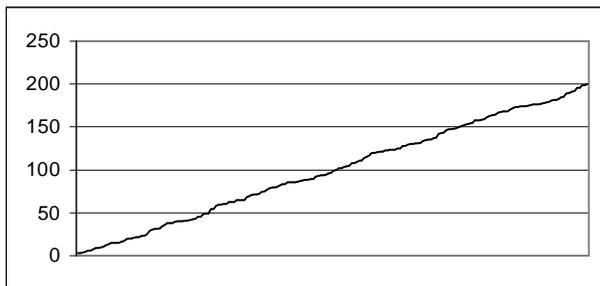


Figura 3.10: Posição média (de forma ordenada) de apresentação dos padrões. Dados Reais

Se o cliente é adimplente, mas possui características que diferem muito do cliente adimplente típico, é provável que ele não venha a honrar seu compromisso. O sistema proposto neste artigo, dado um conjunto de clientes, consegue fazer o mapeamento de suas características para um número que expressa o grau de similaridade do mesmo em relação ao conjunto apresentado.

4. Conclusões

Neste trabalho, propomos um sistema capaz de discriminar, de um conjunto, componentes que possuem características que os diferenciem notadamente da classe à qual eles foram rotulados. Esta habilidade é útil quando existe a necessidade de uma análise mais detalhada da medida de "qualidade" (no sentido de similaridade) dos componentes de uma classe.

Redes GSN são bastante sensíveis quanto à qualidade dos dados utilizados em seu treinamento. Padrões que diferem da classe à qual pertencem apresentam problemas ao serem aprendidos pela rede. A idéia básica do sistema proposto neste artigo parte deste fato e, evoluindo esta idéia, consegue fazer o mapeamento dos padrões de forma que os mesmos são "ranqueados" segundo suas similaridades.

A incapacidade dos analistas de trabalharem com o grande número de dimensões do problema torna a tarefa de análise bastante complexa. Portanto, a capacidade de determinar o grau de adimplência de um cliente, proposto neste trabalho, é um mecanismo que pode ser útil na diminuição de concessão de crédito a clientes que provavelmente não irão honrar seus compromissos.

As redes GSN possuem algumas características interessantes que justificam sua aplicação em problemas financeiros. A alta velocidade com estas redes podem ser treinadas é um fator determinante em situações onde as análises necessitem de respostas rápidas. Além disso, a simplicidade destas redes e a possibilidade de sua implementação em hardware torna a solução interessante para a criação de sistemas de

hardware/software [11] portáteis que possam ser facilmente transportados pelos analistas de crédito.

A aplicação do sistema proposto, porém, não se prende somente ao objeto de estudo em questão (análise de risco de crédito), mas pode ser estendido para várias outras áreas onde é necessário fazer a distinção de padrões de uma mesma classe em forma gradativa, o que chamamos de "ranqueamento". Um de nossos interesses futuros é a aplicação do sistema na área de sistemas tutores inteligentes.

Quanto aos resultados, podemos ver que o sistema apresentou bons resultados diante dos dados sintéticos (para os quais a solução é conhecida). Perceba, da Figura 3.10, que a posição média realmente ranqueia o conjunto de dados reais, ocasionando poucos empates. Quanto menor a posição média, mais exigente, menos representativo é o padrão. Quando alteramos a classe de tais padrões, percebemos que o sistema não rejeita tantos padrões comparado à situação onde alteramos a classe dos padrões mais representativos (os últimos, de posição média mais elevada). Tal fato corrobora com a qualidade do ranking produzido

Referências

- [1] Aleksander, I & Stonham, T J, Guide to pattern recognition using random-access memories, Computers and digital techniques, vol 2, nr. 1, 1979.
- [2] Filho, E, "Investigation of boolean neural network based on a novel Goal-Seeking-Neuron", Tese de Doutorado, Electronic Engineering Laboratories, University of Kent, UK, 1990.
- [3] Kan, W K and Aleksander I, "A probabilistic logic neuron network for associative learning", in Proc. IEEE Intl Conference on Neural Networks, volume II, 541-548, San Diego, USA, 1987.
- [4] Martins, W, "Avoiding False Examples By Use of Cooperation", World Congress on Neural Networks, 795-798, San Diego, Califórnia, USA, 1996.
- [5] Aleksander, I. Adaptive systems of logic networks and binary memories. Proc. Spring Joint Computer Conference, 1967.
- [6] Aleksander, I e Albrow, R C. Adaptive logic circuits. Computer J., 11:65, Maio 1968.
- [7] Bledsoe, W W and Browning, I. Pattern recognition na reading by machine. In Proceedings of The Eastern Joint Computer Conference, 225-232, 1959.
- [8] Rumelhart, D E et alli, Parallel Distributed Processing, Vol 1: Foundations, chapter Learning internal representations by error propagation, The MIT Press, 318-362, 1986.
- [9] Martins, W, "The Improvement of Classifiers Based On Weightless Neural Networks", Tese de Doutorado, Department of Electronics, University of York, UK, 1994.
- [10] Filho, E et alli, A goal-seeking neuron for Boolean neural networks. Proceedings of the IEEE International Neural Networks Conference, Paris, França, 894-897, 1990.
- [11] Aleksander, I et alli, Wisard: a radical step forward in image recognition, Sensor Review, vol 4, nr. 3, 120-124, 1984.