

REDES NEURAIS ARTIFICIAIS E UMA APLICAÇÃO NA ÁREA DE CRÉDITO: UMA PROPOSTA DE METODOLOGIA

*Francisco Carlos B. dos Santos**

Resumo

Este trabalho tem por objetivo mostrar uma nova ferramenta estatística, não-linear e não-paramétrica, utilizando redes neurais artificiais como um método estatístico “dinâmico”, na avaliação de risco de crédito, em específico na otimização do modelo de credit scoring.

Variações de métodos estatísticos e conceitos totalmente novos estão passando a ser largamente utilizado na concepção de análise de crédito, colocando hoje um grande leque de opções tecnológicas à disposição de quem vai implementá-los.

Baseado nisto, será feito também uma análise de uma rede neural e sua utilização na otimização de crédito.

Palavras-chave

Redes neurais, risco, risco de crédito.

* Francisco Carlos B. dos Santos é economista formado pela FEA/USP, com especialização em Economia do Setor Financeiro (MBA FIPE/USP) e professor da Faculdade São Luís. E-mail: francisco.carlos@myrealbox.com

Introdução

Emprestar dinheiro é uma das profissões mais antigas do mundo. O Código de Hamurabi, surgido em 1800 a.C., apresentava diversas seções relativas à regulamentação do crédito na Babilônia.

A palavra “crédito”, dependendo da situação, pode ter vários significados. Para a atividade bancária, crédito é definido como a entrega de um valor presente mediante uma promessa de pagamento (Silva, 1997:63). Portanto, a concessão de crédito, em última instância, significa confiar que outrem, seja uma empresa, seja uma pessoa, irá honrar seus compromissos nas datas pactuadas anteriormente.

Porém, o mercado financeiro é afetado por um grande número de fatores econômicos, políticos e psicológicos, altamente relacionados e que interagem entre si de uma maneira bastante complexa. A maior parte destas relações é probabilística e não-linear. Portanto, tais relações dificilmente podem ser expressas através de regras determinísticas.

No setor de serviços financeiros, sistemas de suporte às decisões vêm sendo amplamente utilizadas para projeção de autorização de crédito.

Existem vários métodos tradicionais de se avaliar a concessão ou não do crédito, podendo ser citado os métodos de *Crédit Scoring*, que podem ser classificados como *estatísticos* ou *determinísticos*.

Os métodos estatísticos utilizam técnicas que incluem tecnologias como regressão linear e redes neurais. Os modelos determinísticos incorporam o conhecimento a respeito da análise do crédito. Experiências adquiridas por especialistas no tema são traduzidas em bases de regras de negócios, processados por um software, denominado *motor de inferência*, que encadeia as regras de acordo com o comportamento do usuário.

Estes processos de análise de crédito e monitoração de risco tendem a se tornar centralizados, utilizando as facilidades de integração e distribuição de processamento proporcionado pela tecnologia da informação.

Sistemas que utilizam redes neurais artificiais estão se tornando uma técnica muito atrativa para estas aplicações, pois oferecem uma alternativa mais eficiente para tratar informações desta natureza (probabilísticas e não-lineares).

1. A avaliação de crédito

O grande desafio para o mercado financeiro na área de crédito é a avaliação do risco dos clientes para determinar o volume de empréstimos

e a taxa a ser cobrada pelos bancos. Com a estabilidade econômica e a tendência de queda nas taxas de juros, as instituições financeiras estão revendo seus processos de sistemas de crédito, incorporando modelos complexos de avaliação. O gerenciamento de risco de crédito vem ganhando importância fundamental, devido principalmente aos seguintes fatores:

- Aumento progressivo das falências e concordatas em uma escala superior aos níveis de recessão;
- Aumento da desintermediação, com a redução dos créditos bancários;
- Redução das margens, pressionada pela concorrência entre as instituições financeiras e não-financeiras outorgadoras de crédito;
- Queda dos valores dos ativos reais e, conseqüentemente, das garantias utilizadas como lastro de crédito;
- Aumento da securitização de crédito, com o advento de novos instrumentos, como futuros, forwards, opções e swaps de crédito;
- Evolução dos sistemas de medida e concentração de risco de crédito;
- Surgimento dos “Early Warning Systems”, sistemas que monitoram a utilização de linhas de crédito, pontualidade no pagamento de parcelas e desempenho de mercados, detectando anomalias com antecipação;
- Sofisticação dos sistemas de rating, incluindo mais níveis de detalhamento;
- Integração entre as diversas etapas de análise de crédito, por meio da utilização de técnicas de workflow e gerenciamento eletrônico de documentos.

Pode-se observar que o crédito será uma grande fonte de recursos para os bancos nos próximos anos, levando o setor creditício a gerir de uma maneira cada vez mais eficaz suas carteiras de empréstimos, reduzindo, assim, seus custos e aumentando a sua lucratividade.

1.1. Risco e Retorno

Segundo Gitman (1997:202), risco pode ser definido como possibilidade de perda, ou como variabilidade de retornos esperados relativos a um ativo. A incerteza pode ser outro termo utilizado com a mesma conotação de risco. Para Silva (1997:75), risco é a probabilidade baseada em dados históricos de se estimar diferentes resultados, cabendo ao tomador de decisões julgar a aceitabilidade da estimativa. Logo, para esse, incerteza seria a decisão baseada em uma sensibilidade pessoal.

Portanto, risco e incerteza não podem ser considerados sinônimos, pois o risco é mensurado com base em dados históricos ou experiências passadas, enquanto que a incerteza refere-se ao *feeling* do tomador de decisões.

A relação risco-retorno está em qualquer relação de crédito e a sua administração é a “alma” da teoria bancária que visa proteger o capital do banco.

1.2. Risco de Crédito

O risco de crédito pode ser definido como a probabilidade de perda em que se incorre quando há inadimplência (*default*) de uma das contrapartes em operação de crédito.

Em grande parte, o risco de crédito do tomador está relacionado a fatores internos e externos (conjuntura econômica) que podem prejudicar o pagamento do empréstimo tomado junto à instituição financeira.

A avaliação do risco de crédito é reflexo da multiplicitude, qualidade e origem das informações disponíveis ao analista de crédito, logo, a necessidade da existência de um banco de dados sobre as operações de crédito já liquidadas é fundamental.

Novas técnicas de mensuração de risco de crédito, aliadas aos avanços dos recursos tecnológicos (TI), são de suma importância para auxiliar o analista de crédito na tomada de decisões.

Alguns dos novos métodos de análise de risco que vêm sendo utilizados são:

- Metodologia de Análises de Migração;
- Modelos Baseados em Opções;
- Método Risk Neutral Valuation;
- Método do Crédito Risk +;
- Método Crédito Métrics;
- Modelo KPMG;
- Sistemas Especialistas;
- Redes Neurais.

Os modelos de avaliação de risco de crédito facilitam a compreensão de um fenômeno e, eventualmente, sua exploração (Couett et al., 1999:118):

(...) por meio de métodos estatísticos procuramos determinar, direta ou indiretamente as respostas para as seguintes perguntas: dada nossa experiência anterior e nossas premissas quanto ao futuro, qual o valor de um empréstimo ou título de renda fixa? ou, qual o risco (quantificável) de que os fluxos de caixa prometidos não venham a ocorrer?

Pode-se observar que o objetivo dos modelos de mensuração do risco de crédito está em enunciar estimativas precisas das probabilidades dos créditos a serem pagos, permitindo, por meio do controle das variáveis utilizadas, a definição de um critério que vise à maximização das receitas ou minimização das perdas, fornecendo uma base estatística satisfatória para a comprovação das decisões.

Uma avaliação de risco satisfatória deve ter cinco características-chave:

1. Sensibilidade para detectar alterações reais na qualidade do crédito;
2. Definir horizontes de tempo para reconhecer mudanças reais de qualidade;
3. Estabilidade de risco onde não ocorrem mudanças fundamentais;
4. Escalonamento graduado de avaliação de risco que facilite racionalmente a precificação do crédito e o estabelecimento dos termos do empréstimo;
5. Consistência de risco entre setores, portes e localizações.

Uma das técnicas estatísticas de análise de dados mais utilizadas no desenvolvimento de avaliação de risco de crédito é a análise discriminante.

Segundo Couett et. al. (1999:581), os modelos estatísticos atribuem pesos pré-determinados para algumas variáveis do cliente, gerando um *scoring* de crédito, ou seja, trabalham com uma medida que possa separar, em dois grupos distintos, os clientes considerados bons e os maus pagadores.

Este método de avaliação, apesar de ser muito utilizado, apresenta problemas como: qual é a ponderação dos pesos que se deve dar para algumas variáveis?

Os sistemas baseados em redes neurais oferecem uma abordagem mais adequada para o tratamento de problemas deste tipo. Uma rede neural pode ser treinada, utilizando-se como *input* uma grande quantidade de exemplos significativos, o que possibilita obter uma performan-

ce elevada. Desta forma, a modelagem por meio de exemplos, de aplicações bem ou mal sucedidas que são obtidas do histórico dos clientes e não por meio de modelos objetivos ou quantificáveis da operação de avaliação é que irá avaliar o cliente.

Assim, uma rede neural seria capaz de encontrar padrões entre as diversas informações e incorporar os processos subjetivos do processo de avaliação. Com isto, obtém-se uma maior flexibilidade, pois quando mudanças se tornam necessárias, devido a fatores de conjuntura econômica, pode-se alterar apenas o conjunto de exemplos de treinamento, o que não requer mudanças estruturais do sistema.

Estes sistemas também podem ser considerados consistentes e eficientes, pois um mesmo sistema pode ser utilizado em diferentes locais, podendo ser operados por pessoal sem experiência gerencial e realizar avaliações mais rapidamente.

2. Sistemas neurais e sua utilização na área de crédito

A construção de um sistema neural para a avaliação de crédito é dividida em várias etapas, sendo:

1. Coleta de Dados: primeiramente necessita-se de um banco de dados que contenha todas as informações pertinentes ao cliente como valor de compras, datas, locais etc. Este conjunto de dados será utilizado para treinamento e validação da rede neural;
2. Pré-processamento dos Dados: os dados simbólicos e numéricos devem ser transformados em dados puramente numéricos, mais adequados para o aprendizado da rede;
3. Separação dos conjuntos de dados: os dados devem ser divididos em vários conjuntos para treinamento, validação e teste ou cada fase do desenvolvimento da rede;
4. Seleção da Arquitetura, Treinamento e Teste da Rede: o desempenho do sistema depende em grande parte da configuração da rede. O treinamento é feito através do algoritmo *backpropagation* e em seguida são realizados os testes. Estes se constituem de simulações com os dados de aprendizado e treinamento, procurando sempre minimizar os erros quadráticos médio, que é uma medida de desempenho da rede;
5. Integração da Rede Neural ao Sistema: uma vez treinada a rede, ela já pode ser integrada ao sistema, tendo sempre em mente a

necessidade de um sistema de banco de dados, para futuros treinamentos da rede, adequando o sistema ao ambiente operacional da empresa.

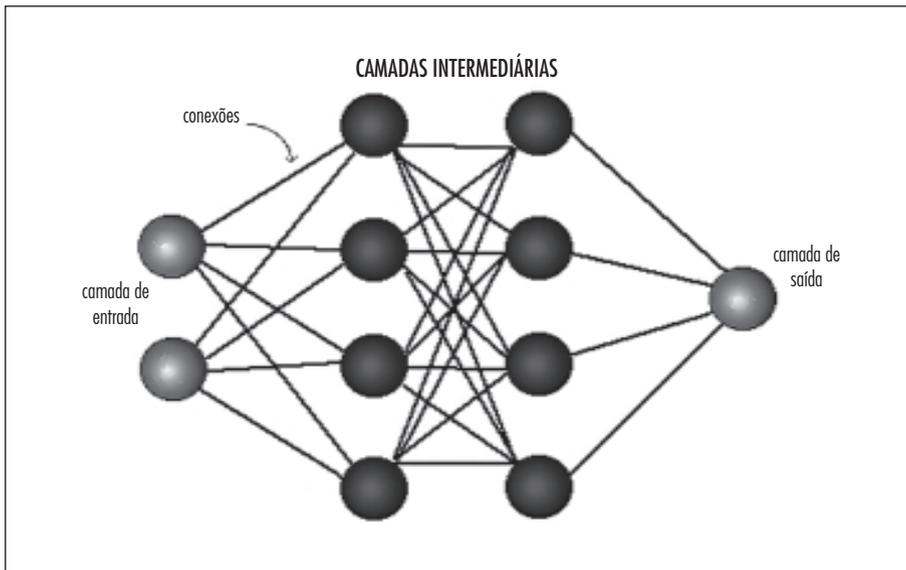
3. Modelo de rede neural: uma implementação

Redes Neurais Artificiais são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento por meio da experiência. Uma grande rede neural artificial pode ter centenas ou milhares de unidades de processamento, já o cérebro humano possui $10e^{11}$ neurônios.

Os neurônios se comunicam por meio de sinapses. Sinapse é a região onde dois neurônios entram em contato e por meio da qual os impulsos nervosos são transmitidos entre eles. Em média, cada neurônio forma entre mil e dez mil sinapses, sendo que o número de sinapses em um cérebro humano é da ordem de $10e^{14}$, formando redes complexas.

Já os sistemas de computação permitem a simulação de algumas centenas de neurônios.

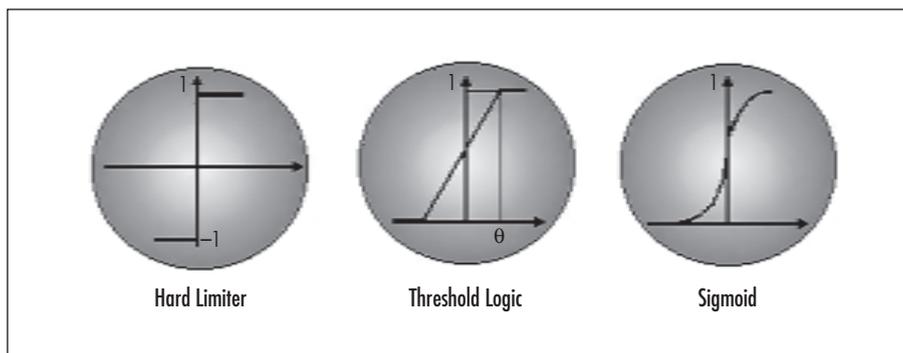
Figura 1. Rede Neural Artificial com seus respectivos nodos, conexões ou sinapses



Uma rede neural artificial consiste de:

- Elementos de processamento (“neurodos”), também referidos como nodos (Fig. 1); interconectados entre si na forma de uma topologia ou rede. Um nodo recebe sinais binários ou contínuos provindos do meio exterior ou de outros nodos da rede, por meio de conexões diretas (*sinapses*), processa estes sinais de alguma forma e envia o produto do processamento para outros nodos ou para o exterior. Os sinais de entrada formam um vetor $A = a_1, a_2, \dots, a_i, \dots, a_n$, onde a_i é o sinal ou nível de atividade da i -ésima conexão de entrada. Tipicamente existem dois tipos de conexões: (1) conexões excitatórias, ou positivas, e (2) conexões inibitórias, ou negativas, que tendem a aumentar ou diminuir, respectivamente, a ativação de um nodo;
- Associado a cada conexão a_i entre dois nodos, existe um valor ajustável denominado peso, formando, portanto, outro vetor $W_j = w_{1j}, w_{2j}, \dots, w_{ij}, \dots, w_{nj}$, onde w_{ij} corresponde ao peso da conexão entre os nodos a_i e b_j . No modelo generalizado pode existir também um parâmetro extra Θ_j , modulado pelo peso w_{ij} , que corresponde a um limiar a ser excedido para haver ativação do nodo. Para calcular o valor de saída b_j do nodo, realiza-se a soma ponderada de todos os $a_i w_{ij}$, e o resultado é aplicado a uma função de ativação, que usualmente é não linear (sigmóide, por exemplo, veja figura 2), e que equivale ao limiar de disparo do neurônio;

Figura 2. Funções de Ativação mais usadas



- Uma arquitetura de interconexão entre nodos da rede. Existem diversos tipos de arquitetura, desde as que interconectam total-

mente cada nodo constituinte da rede (cada nodo se conecta a todos os demais), como nas redes de Hopfield; até as redes estratificadas, formadas por 2 ou mais camadas (uma camada de entrada, uma camada intermediária, ou “oculta”, e uma camada de saída), tais como os perceptrons (Fig. 2). Podem existir conexões de diversos tipos, como as intracamadas (entre os nodos de uma mesma camada), intercamadas (entre os nodos de duas camadas), recorrentes (conexão de um nodo sobre ele mesmo). As conexões intercamadas podem ser anterógradas (no sentido da entrada para a saída) e retrógradas (sentido inverso);

- Finalmente, um importante parâmetro de operação de uma rede neural é a forma como os padrões aprendidos, ou incorporados à rede são recuperados. Se a arquitetura da rede é de tipo ativação progressiva (feedforward), ao se apresentar o padrão a ser classificado à camada de entrada I, as camadas sucessivas são ativadas por meio da propagação do sinal através das conexões ponderadas, até determinar um padrão de ativação final na camada de saída O, que corresponderá à classificação;
- Os perceptrons são bons exemplos de redes deste tipo. De outro lado, as arquiteturas de rede com ativação regressiva (feedback) ao se apresentar um padrão I de entrada, a própria camada devolve o padrão de saída O. A rede denominada Memória Associativa Bidirecional (BAM)¹ é um exemplo bem conhecido desse tipo de rede.

3.1. *Aprendizado Supervisionado*

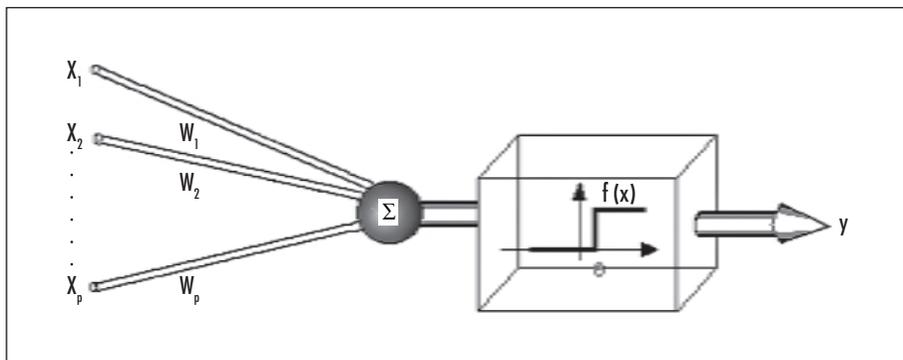
Como exemplo de uma rede neural artificial capaz de aprendizado supervisionado, apresentaremos um dos algoritmos mais utilizados atualmente, que é o da retropropagação (*backpropagation*).

O algoritmo básico de aprendizado de retropropagação é empregado em uma rede neural artificial com uma topologia do tipo perceptron

1. Pode-se dizer que memória associativa é uma variante de um classificador de padrões, em que as categorias de classificação são representadas pelos vetores armazenados. Um padrão arbitrário será classificado pela memória numa destas categorias p dependendo apenas do critério de proximidade que se definiu para a memória. Para maiores detalhes, veja: Kovacs (1996).

de três camadas, heteroassociativo, com conexões de alimentação direta (feedforward) da camada de entrada F_a para a camada intermediária F_b e desta para a camada de saída F_c (Fig. 3);

Figura 3. Esquematização simples de uma rede, com as camadas de entrada, uma função de transferência e uma saída

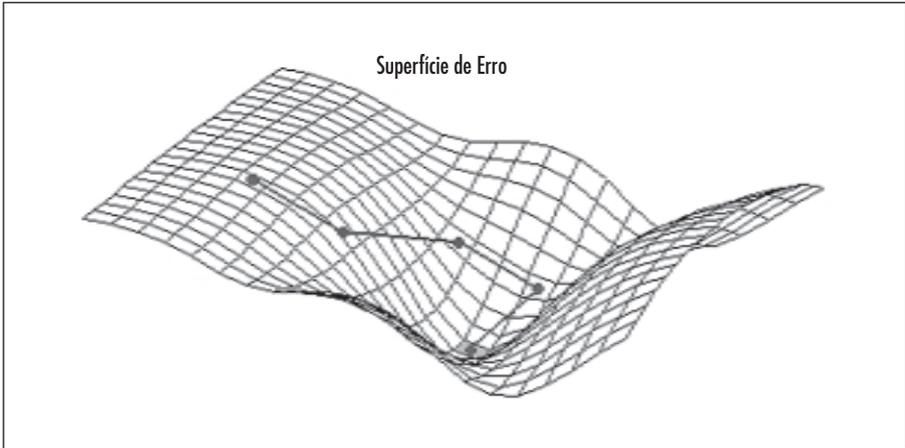


O perceptron de três camadas é capaz de armazenar pares de padrões arbitrários (A_k, C_k) , $k=1, 2, \dots, m$, onde o par de padrão k é representado por dois vetores, $A_k = (a_{1k} \dots a_{nk})$ de entrada e $C_k = (c_{1k} \dots c_{qk})$ de saída. A retropropagação aprende de forma supervisionada (por meio de exemplos), fora de linha (of line), em tempo discreto, e utiliza um método de descida de gradiente por correção de erro, ou seja, o algoritmo de codificação executa um mapeamento entrada-saída por meio da minimização de uma função de custo qualquer (Fig. 4).

A função de custo é minimizada realizando-se interativamente ajustes nos pesos sinápticos de acordo com o erro, ou diferença, entre os valores desejados e obtidos na camada de saída F_c ao se aplicar um par (A_k, C_k) na rede. A função de custo a ser minimizada normalmente é o erro quadrático (o quadrado da diferença entre os dois valores) acumulado para todos os padrões do conjunto de treinamento. Outras funções de custo podem ser utilizadas, mas independentemente disto, o procedimento de ajuste de pesos é realizado por meio do cálculo da mudança da função de custo com respeito à mudança em cada peso (método do delta). O processo de redução gradativa de erro que acompanha a minimização se denomina convergência. À medida que a rede aprende, o valor do erro converge para um valor estável, normalmente irreduzível.

O processo de aprendizagem prossegue até que algum critério seja estabelecido, como um valor mínimo de erro global, ou uma diferença sucessiva mínima entre erros calculados para cada interação (Fig. 4).

Figura 4. Superfície de Erro com o mínimo global



A rememoração é feita também pelo método de alimentação direta.

3.2. Aprendizado

Antes de utilizar o conjunto de treinamento, os valores de A_k e C_k devem ser todos normalizados em uma faixa adequada, por exemplo, convertidos para valores que fiquem entre 0 e 1, inclusive. Uma forma de realizar esta normalização é aplicar a seguinte fórmula para cada elemento A_k do vetor:

$$A_k = (A_k - A_{MIN}) / (A_{MAX} - A_{MIN})$$

Onde A_{MIN} é o valor mínimo encontrado ou determinado em A_k e A_{MAX} é o valor máximo encontrado ou determinado em A_k . A mesma fórmula se aplicaria para o conjunto de saída C . Outras funções de normalização e transformação (por exemplo, logaritmo) podem ser utilizadas.

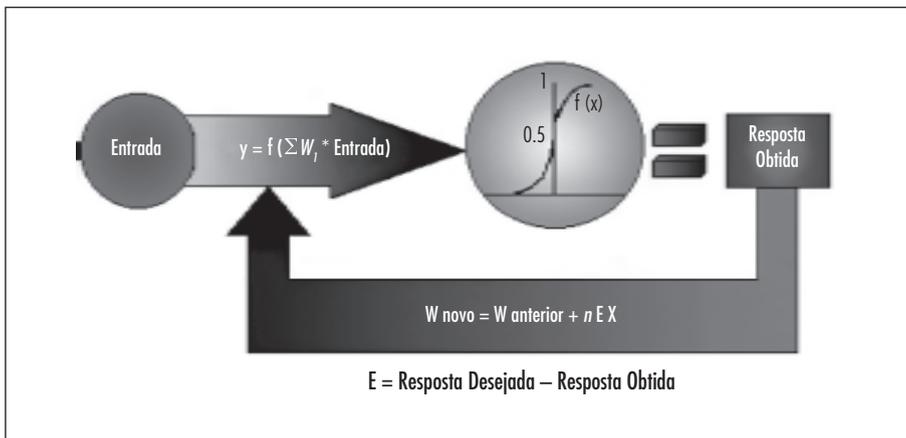
A normalização é importante, porque se um ou mais neurodos tiverem valores de ativação muito maiores ou menores do que os demais,

isto poderá produzir valores de erro desproporcionais, e levar a um desbalanço da rede, conseqüentemente prejudicando a sua convergência.

Os exemplos devem ser escolhidos de forma a representar todos os padrões existentes na possível população de padrões que se deseja classificar. De preferência, colocam-se vários exemplares de cada classificação de saída (por exemplo, vários conjuntos de sintomas associados a um determinado diagnóstico). Os exemplos podem ser sintéticos ou reais, sendo preferível utilizarem-se os segundos, caso seja possível. Por exemplo, um conjunto de exemplos pode ser derivado aleatoriamente de um conjunto maior de pares de padrões entrada-saída, e utilizado para treinar a rede até a convergência. Em seguida, os pares de padrões restantes podem ser utilizados para testar se a rede é capaz de classificar corretamente casos conhecidos.

O algoritmo básico de treinamento da rede consiste em aplicar um padrão A_k na camada de entrada e, em seguida, propagar os valores de ativação da camada de entrada até a camada de saída (Fig. 5). Os valores de ativação assim obtidos na camada de saída são comparados com os valores contidos em C_k (desejados), e uma função de erro é computada para cada neurônio de saída.

Figura 5. Funcionamento de uma Rede



Uma parte deste erro (chamado delta) é utilizada, então, para modificar os valores dos pesos sinápticos entre a camada intermediária

e a camada de saída, de tal forma que, quanto maior o erro, maior será a modificação do peso, sempre no sentido de diminuir o erro para cada par de neurodos. Este método de correção por erros é aplicado em seguida aos pesos sinápticos entre a camada de entrada e a camada intermediária, levando, deste modo, a uma propagação dos deltas no sentido inverso ao da ativação da rede (daí o nome de retropropagação).

Este procedimento é utilizado para cada um dos pares de padrões utilizados para treinar a rede (chamado de conjunto de treinamento), voltando-se a repeti-lo para todo o conjunto diversas vezes, até que o erro global da rede convirja para algum valor mínimo. Esta passagem do algoritmo por todos os exemplos do conjunto de treinamento é chamado de ciclo de treinamento. Dependendo da complexidade e do tamanho dos padrões, o método de aprendizado por retropropagação pode exigir entre milhares e centenas de milhares de ciclos de treinamento.

Na representação do algoritmo abaixo, utilizamos a notação convencionalizada por Simpson (1990).

- A) Atribua valores aleatórios entre -1 e $+1$ a todos os pesos sinápticos V_{hi} , das conexões entre cada limiar Θ_i da camada F_b e a cada limiar T_j da camada F_c .
- B) Para cada par de padrões (A_k, C_k) , $k = 1, 2, \dots, m$ existente no conjunto de treinamento (exemplos), realize o seguinte:
 - I) Transfira os valores do vetor correspondente ao padrão de entrada A_k , $k = 1, 2, \dots, m$ aos neurodos da camada de entrada F_a (um valor do vetor para cada um dos m neurodos);
 - II) Calcule os novos valores de ativação b_i dos neurodos da camada intermediária F_{bi} através do seguinte método:

$$b_i = f\left(\sum_{h=1}^n a_h\right)$$

Para todos os neurodos de F_{bi} , $i = 1, 2, \dots, p$, onde b_i é o valor de ativação do neurodo i da camada intermediária F_b , Θ_i , é o valor do limiar i do neurodo da camada F_b , e $f(x)$ é a função de limiar sigmóide (Fig. 2):

$$f(x) = \left(1 + e^{-x}\right)^{-1}$$

- I) Calcule os novos valores de ativação c_j dos neurodos da camada de saída F_c , através da seguinte equação:

$$c_j = f\left(\sum_{i=1}^p b_i w_{ij}\right) + \Gamma_j$$

Para todos os neurodos de F_{c_j} , $j = 1, 2, \dots, q$, onde c_j é o valor de ativação do neurodo j da camada de saída F_c , Γ_j é o valor do limiar j do neurodo da camada F_c e $f(x)$ é a função de limiar sigmóide.

- II) Calcule os valores de discrepância (erro) d_j entre os valores obtidos e desejados dos valores dos neurodos da camada de saída F_c , usando a seguinte equação:

$$d_j = c_j(1 - c_j)(c_j^k - c_j)$$

para todos os neurodos de F_{c_j} , $j = 1, 2, \dots, q$, e onde c_{jk} é o valor do neurodo correspondente ao elemento c_k do vetor de saída do par de treinamento k .

- III) Calcule os erros para cada neurodo de camada intermediária F_b correspondente a cada d_j , usando a equação:

$$e_i = b_i(1 - b_i) \sum_{j=1}^q w_{ij} d_j$$

para todos os neurodos de F_{b_i} , $i = 1, 2, \dots, p$, onde e_i é o erro computado para o neurodo i .

- IV) Ajuste os pesos das conexões entre as camadas de saída F_c e intermediária F_b , utilizando:

$$\Delta w_{ij} = \alpha b_i d_j,$$

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

para todos os neurodos da camada intermediária F_{b_i} , $i = 1, 2, \dots, p$ e todos os neurodos da camada de saída F_{c_j} , $j = 1, 2, \dots, q$ e onde Δw_{ij} é o valor da mudança a ser realizada da conexão entre o neurodo i da camada intermediária F_b e o neurodo j da camada de saída F_c (chamado delta); e α é uma constante positiva que controla a taxa de aprendizado (quanto maior α , maior

será a mudança delta, portanto mais acentuado será o gradiente e mais rápido o aprendizado, daí o significado deste parâmetro). Note, portanto, que o valor do delta para cada sinapse equivale ao produto do erro do neurodo de saída pelo valor de ativação do neurodo intermediário, pela taxa de aprendizado.

V) Ajuste também os limiares dos neurodos F_c usando delta:

$$\Delta_i = \alpha d_i$$

para todos os neurodos da camada F_{c_j} , $j = 1, 2 \dots q$.

VI) Ajuste agora os pesos sinápticos das conexões entre os neurodos da camada de entrada F_a e da camada intermediária F_b , utilizando o delta para todos os neurodos da camada de entrada F_{ah} , $h = 1, 2 \dots n$ e todos os neurodos da camada intermediária F_{bi} , $i = 1, 2 \dots p$; e onde A_{vij} é o valor da mudança a ser realizada da conexão entre o neurodo h da camada intermediária F_{ah} e o neurodo i da camada intermediária F_{bi} ; e b é uma constante positiva semelhante à a , que controla a taxa de aprendizado desta camada. Normalmente, “ b ” é feito igual “ a ”, mas o algoritmo permite que sejam diferentes, o que pode ser interessante em certos casos.

VII) Ajuste também os limiares dos neurodos F_b usando o delta para todos os neurodos da camada F_{bi} , $i = 1, 2 \dots n$.

VIII) Pegue um novo par de padrões (A_k , C_k) do conjunto de treinamento, e execute novamente o passo 2, repetindo-o até que os valores de correção de erro d_j para cada neurodo da camada de saída F_{c_j} , $j = 1, 2 \dots p$, para cada exemplo $K = 1, 2, \dots, m$ seja suficientemente baixo.

Uma função global de erro para cada ciclo de treinamento pode ser computada com a seguinte equação:

$$W(k+1) = w(k) + \alpha \sum_l^L (y_l^d - y_l)$$

onde:

$$\alpha \sum_l^L (y_l^d - y_l)$$

é o erro associado ao n -ésimo exemplo, ponderado pela taxa de aprendizado a , que pode ser utilizada para fins de critério de parada de treinamento. Esta parada pode ser efetuada por intervenção do operador, por meio do acompanhamento visual dos valores globais (D) a cada 100 ou 1000 ciclos de treinamento (um gráfico de evolução do D em função do número de ciclos de treinamento); ou dos valores locais (d_j) de erro, por meio da exibição de um histograma indicando a distribuição dos valores de (d_j) a cada número fixo de ciclos. Alternativamente, a parada pode ser efetuada automaticamente, quando um dos critérios discutidos acima for atingido.

3.3. Ativação

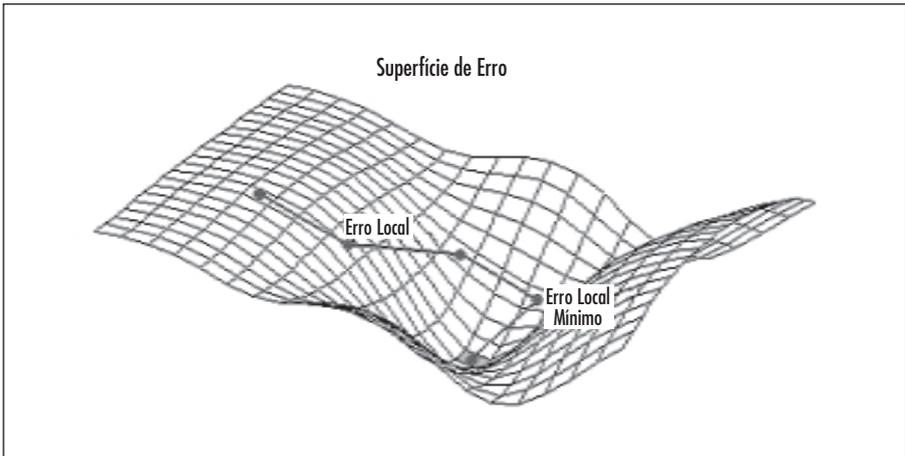
A chamada ativação, ou rememoração de um padrão armazenado, ocorre propagando-se um padrão de entrada “A” à camada de entrada F_a do perceptron assim treinado, até a camada de saída F_c , utilizando-se os conjuntos de pesos V_{hi} e W_{ij} estabilizados após convergência:

- A) Da camada de entrada F_a para a camada intermediária F_b , para todos os neurodos de F_{bi} , $i = 1, 2 \dots p$, onde b_i é o valor de ativação do neurodo i da camada intermediária F_b .
- B) E da camada intermediária F_b para a camada de saída F_c , para todos os neurodos de F_{cj} , $j = 1, 2 \dots q$, onde c_j é o valor de ativação do neurodo j da camada de saída F_c .

3.4. O Algoritmo de Retropropagação (BACKPROPAGATION)

O algoritmo de retropropagação não garante que o erro global mínimo seja achado durante o processo de treinamento, mas apenas algum erro local mínimo (Fig. 7). Este fato pode levar às oscilações do erro global da rede durante o treinamento, que fica aumentando e diminuindo, sem convergir. Pode acontecer, também, que o erro global para o qual a rede converge é inaceitavelmente alto (as classificações resultantes após treinamento serão erradas em um número muito grande de vezes). Em ambos os casos, diz-se que a rede é incapaz de aprender.

Não existe uma solução pré-determinada para resolver estes problemas. Portanto, quando uma rede for incapaz de aprender, devem-se realizar modificações em sua topologia ou parâmetros de aprendizado, na forma de tentativa e experimento.

Figura 6. Superfície de Erro com Mínimo Local

- A) Normalização: verificar se todas as variáveis de entrada são normalizadas, ou se existe algum erro de digitação que torne o conjunto incoerente com a classificação que deve ser aprendida.
- B) Seleção de variáveis: verificar se existem variáveis irrelevantes para o problema de classificação. Isto pode ser feito por meio de testes estatísticos de associação e correlação, como o χ^2 (qui-quadrado), entre cada variável de entrada e a classificação de saída. Assim, podem ser incluídas apenas as variáveis que apresentam um alto grau de associação positiva ou negativa, e retiradas àquelas que não apresentam associação ou correlação significativas.
- C) Número e variabilidade dos exemplos: o aprendizado e/ou efetividade de classificação pode ser prejudicado se o número de exemplos for muito pequeno (não cobrindo todas as possibilidades de classificação), ou se os exemplos de uma mesma classificação forem muito variáveis entre si (confusão), ou se existirem exemplares muito similares entre diferentes classes (ambigüidade). Pode-se, nestes casos, utilizar critérios de eliminação de “outliers”, ou seja, exemplos que estejam provocando confusão ou ambigüidade.
- D) Camada intermediária: verificar se há um número ótimo de neurodos na camada intermediária. Não existe uma fórmula para determinar

esse número. O *Teorema da Existência* de Kolmogorov (1957)² diz que a convergência é garantida se o número de neurodos intermediários for igual ao número de neurodos da camada de entrada (tamanho do padrão), mais um. Entretanto, essa convergência nem sempre acontece ou fica extremamente lenta. O número necessário de neurodos na camada intermediária também depende de formação não previsível de outros fatores, como a variabilidade dos exemplos, a ocorrência de separações não-lineares complexas entre padrões, a relação entre o número de neurodos de entrada e de saída, entre outros. Algumas pesquisas mostraram também que a efetividade de uma rede pode ser aumentada, utilizando-se um procedimento de duplo treinamento: primeiro treina-se a rede a aprender até um valor (não ótimo), em seguida analisa-se a matriz de pesos e retiram-se os neurodos intermediários que contribuem pouco, ou seja, há valores de pesos próximos de zero. Em seguida, retreina-se a rede usando este conjunto diminuído de neurodos na camada intermediária. Outra forma mais complexa seria “desconectar” algumas sinapses que contribuem pouco, ao invés de todo o neurodo intermediário.

- E) Taxa de aprendizado: os fatores A e B de aprendizado, usados para transferir parte do delta para a correção dos pesos, são determinantes em duas coisas: a velocidade do aprendizado (tanto maior quanto maiores forem estas taxas), e a ocorrência de mínimos locais. O uso de uma taxa elevada leva a um aprendizado rápido, mas se houver problemas na topologia da rede ou na estrutura dos padrões, poderão ocorrer oscilações do erro global. Nestes casos, é preferível utilizar uma taxa menor, levando a modificações menores dos pesos, com maior estabilidade na evolução da convergência. Alguns algoritmos modificados de retropropagação têm mudanças adaptativas da taxa de aprendizado, ou seja, utilizam taxas de aprendizado maiores no início, levando a quedas grandes de erro, e que são diminuídas de forma proporcional ao número de ciclos de treina-

2. O Teorema de Kolmogorov-Nielsen diz que: *Dada uma função contínua arbitrária $f: [0,1]^n \rightarrow \mathbb{R}^m$, $f(\mathbf{x})=\mathbf{y}$, existe sempre para f uma implementação exata com uma rede neural de três camadas, sendo a camada de entrada de vetor de dimensão n , a camada oculta composta por $(2n+1)$ neurônios, e a camada de saída com m neurônios representando as m componentes do vetor \mathbf{y} .* (Kolmogorov, 1957:114). Para maiores detalhes veja: Kolmogorov (1957) e Kovacs (1996).

mento, proporcionando maior estabilidade em fases avançadas do treinamento.

- F) Limiares e função de transferência: estes são parâmetros que também podem ter influência na capacidade e velocidade de aprendizado. Os valores limiares α e T dos neurodos das camadas de saída e intermediária, respectivamente, são normalmente constantes que regulam a sensibilidade de um neurodo a variações em suas funções de ativação. Embora sejam inicializados aleatoriamente no início do aprendizado, eles podem percorrer gamas mais baixas ou mais elevadas, conforme as características da rede neural empregada. Da mesma forma, é possível utilizar outras funções de transferência além da sigmóide, tais como logística, sigmóide segmentada, senóides ou cossenóides, cossenóides truncadas etc. O importante é que seja uma função não-linear. A escolha da função também pode ter repercussões sobre a velocidade de aprendizado, principalmente pelo número de operações em ponto flutuante que exigem.

Outro problema de retropropagação é a lentidão do algoritmo, que normalmente exige muitos ciclos de treinamento para convergir. Observando-se a velocidade de convergência nos primeiros ciclos, e o início da tendência à estabilização, pode-se ter uma idéia se a rede está convergindo efetivamente ou não, e se os parâmetros escolhidos são adequados.

4. Estimativas de pesos em redes neurais

Um dos problemas centrais em aproximações funcionais usando redes neurais é a estimativa do erro de previsão ou generalização da rede. Para o caso de regressões lineares há fórmulas analíticas para o erro esperado e a variância. No caso de regressões não-lineares, estas fórmulas não existem. Alguns métodos estatísticos podem ser aplicados para o cálculo do erro de generalização esperado da rede.

Cabe lembrar que o erro esperado não é o erro *root mean square* no conjunto de treinamento, pois os parâmetros da rede foram ajustados justamente para reduzir estes dados. Caso muitos exemplos estejam disponíveis para o conjunto de teste pode-se computar o erro de generalização diretamente, como o erro *root mean square* neste conjunto de teste. No entanto, na maior parte das situações, os dados são escassos e seria desejável usar todos os dados para treinar a rede e também prever o erro de generalização.

O método do *bootstrap*, proposto por Efron em 1979, trata deste tipo de problema estatístico. A observação básica é que os dados existentes podem ser usados para o ajuste dos parâmetros livres, mas ele também contém informação sobre a distribuição dos dados que serão utilizados para a previsão. Este método consiste em a partir de um conjunto de dados X com m pontos, gerar N conjunto X^* a partir de X com amostragem aleatória com substituição. A hipótese básica é que a probabilidade de distribuição dos dados F pode ser aproximada pela amostragem aleatória de X com substituição.

O algoritmo do método bootstrap é o seguinte:

Selecionar N amostras independentes $x_1^*, x_2^*, \dots, x_n^*$, cada uma com n dados selecionados aleatoriamente com substituição de X .

Computar a estatística desejada S em cada amostra através da seguinte equação:

$$\hat{\Theta}(b) = S(x_b^*), \text{ com } b = 1, 2, \dots, N$$

Estimar o erro padrão das N réplicas:

$$\hat{S}_N = \sqrt{\frac{\sum_{b=1}^N \left(\hat{\Theta}(b) - \hat{\Theta} \right)^2}{N-1}}, \quad \hat{\Theta} = \frac{1}{N} \sum_{b=1}^N \hat{\Theta}(b)$$

No caso de redes neurais, calcula-se o erro quadrático médio Q_i^* para o i -ésimo conjunto de treinamento gerado pelo bootstrap. O desvio padrão dos valores de Q_i^* é uma aproximação para o verdadeiro desvio padrão do ajuste.

Há uma complicação que ocorre com as redes neurais; como o treinamento não é determinístico devido à existência de vários mínimos locais na função de erro que podem prender os parâmetros durante o treinamento, o re-treinamento da rede pode gerar soluções completamente diferentes dos pesos envolvidos, apesar do erro ser semelhante.

Outro método estatístico que pode ser usado é conhecido como *jackknife*, que é mais antigo que o método do bootstrap. Este método consiste em obter novos conjuntos de treinamento a partir do conjunto original de dados, mas de uma maneira mais simples. Caso n exemplos existam, a rede é treinada com $(n-1)$ pontos e o resultado final é dado

pela média dos n diferentes conjuntos de treinamento. Também pode-se estimar os erros de generalização dos n diferentes exemplos deixados fora dos conjuntos de treinamento.

Também é comum o uso da técnica de *cross-validation*, em que um conjunto de validação não usado na fase de treinamento é usado para medir o erro. O erro estimado é uma boa aproximação do erro verdadeiro caso tanto o conjunto de treinamento quanto o conjunto reflitam a distribuição de probabilidade dos dados de entrada. Uma maneira de melhorar este método é com o chamado *k-fold cross validation*, que consiste em dividir todos os dados aleatoriamente em k subconjuntos. A rede é treinada k vezes, cada vez deixando um dos subconjuntos à parte, para estimar o erro.

Os métodos de bootstrap, jackknife e cross-validation permitem determinar os intervalos de confiança para as previsões das redes neurais. Também é possível treinar uma outra rede neural nos erros da rede original, usando-a para previsão de erros das previsões.

No capítulo abaixo, será apresentado um método simples de estimar um intervalo de confiança para a saída da rede neural.

4.1. Intervalo de Confiança na Saída da Rede

Um dos problemas de uma rede neural é que sua predição não é avaliada por nenhum método estatístico convencional, sendo assim, o modo como a rede neural trabalha se torna uma verdadeira “caixa preta”.

Uma estatística que dê um intervalo de confiança com uma grande acurácia, pode se tornar de grande valia para determinar o quão boa uma rede pode ser. A rede neural trabalha com regressões não-lineares e é sabido que há uma teoria para estimar intervalos de confiança para tal tipo de regressão.

A derivação matemática abaixo é válida para toda rede neural onde o tipo de ativação é progressiva (*feedforward*) e onde procura minimizar a soma dos quadrados dos erros, como é o caso da rede desenvolvida neste trabalho.

A equação geral para se estimar todo intervalo de confiança é dado por:

$$\hat{y}_0 \pm Z_{\alpha} \frac{\sqrt{\left(S^2 + Var(\hat{y}_0) \right)}}{2} \quad (\text{equação I})$$

onde:

\hat{y}_0 = saída (predição) da rede neural

$\frac{Z_\alpha}{2}$ = valor crítico

s^2 = variância da estimativa das variáveis endógenas que foram treinadas

$\widehat{Var}(y_0)$ = variância da predição

Na prática, a variância do termo da equação (I) é computada usando:

$$\widehat{Var}(y_0) = \sum_{i=1}^n \left(\frac{\partial f_0}{\partial w_i} \frac{\partial f_0}{\partial w_i} \text{Côv}(w_i w_j) \right) + 2 \left(\sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{\partial f_0}{\partial w_i} \frac{\partial f_0}{\partial w_j} \text{Côv}(w_i w_j) \right)$$

(equação II)

onde:

$\frac{\partial f_0}{\partial w_i}$ = derivada da rede neural em relação aos seus pesos

n = número de pesos da rede neural

$\text{Côv}(w_i w_j)$ = covariância estimada com relação aos pesos da camada i e j

A equação (II), que é a derivada parcial com relação aos pesos que são computados pela rede, não apresenta dificuldades de cálculo. De fato, a maneira computacional é semelhante àquela para calcular os pesos durante o treinamento da rede, sem associar à somatória dos mínimos quadrados dos erros.

Sempre que se treina a rede é necessário salvar os pesos das conexões (matriz de pesos) em cada ciclo de treinamento e estimar a covariância. Com isto, teoricamente teríamos uma covariância computada para cada ciclo de treinamento, totalizando N pesos, cada qual com sua covariância. Porém, na prática se usa apenas os pesos do último treinamento. Se o resultado da rede for estável, significa que os N pesos da matriz de peso, do último ciclo de treinamento, será o “ideal”. Se, pelo contrário, os resultados forem instáveis, isto indica uma inadequação do modelo obtido pela rede.

Após o treinamento, aplica-se a equação abaixo para cada par (i,j) dos pesos que foram gravados pela rede.

$$C\hat{o}v(w_i w_j) = \left(\frac{1}{N} \sum_{k=1}^N (w_{ik} w_{jk}) \right) - \overline{w_i w_j}$$

onde:

w_{ik} = k-ésimo elemento da matriz de peso da i-ésima matriz de peso salva

$\overline{w_i}$ = valor médio do i-ésimo peso da rede neural

w_{jk} = k-ésimo elemento da matriz de peso da j-ésima matriz de peso salva

$\overline{w_j}$ = valor médio do j-ésimo peso da rede neural

N = número de pesos salvos

Resultados empíricos afirmam que uma boa média a ser salva é em torno de cinqüenta matrizes de pesos (Rubino e Donald, 1996:46-49).

Substituindo a equação (II) na equação (III), e depois substituindo na equação (I), temos a equação geral do intervalo de confiança fornecido pela rede neural:

onde:

$$\hat{y}_0 \pm Z_{\alpha/2} \sqrt{\left(S^2 + \sum_{i=1}^n \left(\frac{\partial f_0}{\partial w_i} \frac{\partial f_0}{\partial w_i} \left(\frac{1}{N} \sum_{k=1}^N (w_{ik} w_{jk}) \right) - \overline{w_i w_j} \right) + 2 \left(\sum_{i=1}^{n-1} \sum_{j=1+i}^n \left(\frac{\partial f_0}{\partial w_i} \frac{\partial f_0}{\partial w_j} \left(\left(\frac{1}{N} \sum_{k=1}^N (w_{ik} w_{jk}) \right) - \overline{w_i w_j} \right) \right) \right) \right)}$$

$$Var(\hat{Y}_0) = \sum_{i=1}^n \left(\frac{\partial f_0}{\partial w_i} \frac{\partial f_0}{\partial w_i} \left(\frac{1}{N} \sum_{k=1}^N (w_{ik} w_{jk}) \right) - \overline{w_i w_j} \right) + 2 \left(\sum_{i=1}^{n-1} \sum_{j=1+i}^n \left(\frac{\partial f_0}{\partial w_i} \frac{\partial f_0}{\partial w_j} \left(\left(\frac{1}{N} \sum_{k=1}^N (w_{ik} w_{jk}) \right) - \overline{w_i w_j} \right) \right) \right)$$

Considerações finais

Os conceitos verificados na literatura, as proposições efetuadas no artigo e os comentários quanto aos principais aspectos abordados, permitem chegar a algumas conclusões.

Os modelos de risco de crédito são de suma importância, pois proporcionam ao analista de crédito o *insight* ou conhecimento que não

estaria, de outra maneira, prontamente disponível para auxiliar a tomada de decisão sobre a concessão do crédito.

Em um mercado em que as margens de lucro estão diminuindo rapidamente e a pressão para a redução de custos é incessante, a análise de risco é uma vantagem competitiva para se atingir tais objetivos, proporcionando à empresa creditícia uma maior abrangência de mercado.

Referências Bibliográficas

- BERNI, M. T. (1999). *Operação e concessão de crédito: os parâmetros para decisão de crédito*. São Paulo: Atlas.
- BLATT, A. (1999). *Avaliação de risco e decisão de crédito: um enfoque prático*. São Paulo: Nobel.
- EFRON, B. (1979). *Bootstrap Methods: another look at the Jackknife*, Annals of Statistics, 7, 1-26.
- (1987). *Better Bootstrap Confidence Intervals*, Journal of the American Statistical Association, 82, 171-185.
- FAUSETT, L. (1995). *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*. Nova York: Prentice Hall.
- GUITMAN, Lawrence J. (1978). *Princípios de Administração Financeira*. São Paulo: Editora Harper & Row do Brasil Ltda.
- HAYKIN, S. (2001). *Redes Neurais Artificiais: Princípios e práticas*. Porto Alegre: Bookman.
- HOPFIELD, J. J. (1982). *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, Proceedings of the National Academy of Sciences, vol 79.
- KOLMOGOROV, A. N. (1957). *On the representation of Continuous Functions*. Dokl. Acad. Nauk., USSR, 114.
- KOVASC, Z. L. (1996). *Redes Neurais Artificiais: Fundamentos e aplicações*. São Paulo: Edição Acadêmica.
- LeBARON, B. e WEIGEND, A. (1993). *A Bootstrap Evaluation of the Effect of Data: Splitting on Financial Time Series*. Department of economics, Universit of Wisconsin. <http://www.econ.wisc.edu/~blake>.
- RUBINO JR., T. B. e DONALD, NIMEY P. (1996). *Option strategies and neural networks: technical analysis of stocks & commodities*, September.
- SILVA, J. P. da. (1993). *Análise e decisão de crédito*. São Paulo: Atlas.
- SIMPSON, P. (1990). *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations*, Pergamon Press Inc.